TU WIEN Informatics

Development and Visual Assessment of a Cross-Modal Brain Tumor Segmentation Pipeline

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Visual Computing

eingereicht von

Caroline Magg, MSc.
Matrikelnummer 01225388

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assistant Prof. Dr. Renata Raidou

Wien, 7. Dezember 2021

Caroline Magg	Renata Raidou

Erklärung zur Verfassung der Arbeit

Caroline Magg, MSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 7. Dezember 2021

Caroline Magg

Danksagung

Zuallererst möchte ich Renata Raidou meinen tiefsten Dank aussprechen. Ich finde die Symmetrie mehr als passend, dass sie meinen ersten Kurs im Master Visual Computing unterrichtet und mich nun durch die letzte Phase meines Studiums begleitet hat. Als ich sie vor 6 Monaten bat, meine Diplomarbeit zu betreuen, wusste ich, wenn jemand mir bei meinem ehrgeizigen Plan helfen könnte, dann sie. Sie stand mir mit Rat und Tat zur Seite und hatte immer ein offenes Ohr. Sie war wirklich die beste Betreuerin, die ich mir hätte wünschen können.

Ein großes Dankeschön geht auch an meine Studienkollegen und Freunde, die mich während meines Studiums begleitet haben. Besonders in Zeiten, in denen die Lehrveranstaltungen aufgrund von Lockdownmaßnahmen online gehalten wurden, waren mir die Verbindungen zu ihnen sehr viel wert.

Nicht zuletzt möchte ich meinen Eltern danken. Sie haben mich in all meinen Entscheidungen unterstützt und es mir ermöglicht, meinen Weg zu finden. Ich bin wirklich dankbar für jedes lange spätabendliche Telefonat mit meiner Mutter, wo sie sich meine Zweifel anhörte und mich daran erinnerte, an mich selbst zu glauben.

Dankeschön!

Acknowledgements

First of all, I would like to express my deepest gratitude to Renata Raidou. I find the symmetry more than fitting that she taught my first course in the Master Visual Computing and has now accompanied me through the last phase of my studies. I asked her to be my thesis supervisor 6 months ago because I knew if anyone could help me with my ambitious plan, it would be her. She stood by me with advice and support and always had an open ear. She was genuinely the best supervisor I could have asked for.

A big thank you also goes to my student colleagues and friends, who accompanied me during my studies. Especially during times when courses were held online due to lockdown measures, the connections with them were very valuable to me.

Last but not least, I would like to thank my parents. They supported me in all my decisions and allowed me to find my way. I am truly grateful for every long late night phone call with my mom where she listened to my doubts and reminded me to believe in myself.

Thank you!

Kurzfassung

Automatisierte Segmentierung ist ein wichtiger Schritt in der Therapieplanung für Hirntumore, wie das Vestibularis-Schwannom. Behandlungsprotokolle umfassen kontrastverstärkte T1-gewichtete (ceT1) und hochauflösende T2-gewichtete (hrT2) MR-Scans. CeT1 Scans bietet einen höheren Kontrast, verwenden aber Kontrastmittel, die kumulative Nebenwirkungen verursachen können. Daher gibt es Bemühungen, vollständig auf hrT2 umzusteigen. Da die Verfügbarkeit großer, vollständig annotierter Datensätze begrenzt ist, sind Strategien zur Nutzung modalitätsübergreifender Daten erforderlich. Nach der Entwicklung eines Segmentierungs-Algorithmus müssen Künstliche Intelligenz (KI) Entwickler die Ergebnisse ihrer Modelle mit Ground-Truth Labels und anderen Algorithmen vergleichen. Eine visuelle Analyse verbessert das Verständnis für solche automatisierten Lösungen. Aktuelle Visual Analytics (VA) Anwendungen bieten jedoch keine flexiblen Vergleichsmöglichkeiten, mit denen sich große Patientenkohorten bis hin zu einzelnen Bildschichten aufschlüsseln lassen. Außerdem sind sie nicht in der Lage, Korrelationen zu anderen aus Datensätzen und Bildern abgeleiteten Merkmalen, zu erkennen.

Diese Arbeit hat zwei Schwerpunkte. Erstens entwickeln wir **zwei Methoden, die Information von ceT1- auf hrT2-Scans übertragen**. Das Ziel ist die automatische Tumorsegmentierung auf hrT2-Bildern. Es werden Kohortdaten von 242 Patienten verwendet, die jeweils aus annotierten ceT1- und nicht annotierten hrT2-Aufnahmen bestehen. Die Methoden werden durch ein klassifikationsgesteuertes Modul erweitert, das falsch-positive Vorhersagen von Scans vermeidet. Zweitens entwerfen und implementieren wir eine **interaktive webbasierte VA-Anwendung** für die Bewertung der Algorithmusergebnisse. Wir führen eine quantitative Evaluierung durch und demonstrieren vier Anwendungsszenarien. Das vorgestellte Tool ermöglicht es den Benutzern, mehrere Modelle auf verschiedenen Detailebenen zu vergleichen und Korrelationen zwischen Fehlermetriken und Radiomics-Merkmalen zu finden. Unsere besten Methoden erreichen 61.14% und 92.62% Dice Score auf Tumorschichten bzw. dem gesamten Datensatz. Unser VA-Ansatz liefert zusätzliche Erkenntnisse, die für die Bewertung der entwickelten Algorithmen nützlich sind.

Abstract

Automatic segmentation is an important step in therapy planning for brain tumors, such as Vestibular Schwannoma. Treatment protocols include contrast-enhanced T1-weighted (ceT1) and high-resolution T2-weighted (hrT2) MR scans. Although ceT1 scans provide higher contrast, they use contrast agents which can cause cumulative side-effects. Therefore, efforts are underway to move to hrT2 completely. Because the availability of large, fully annotated data sets is limited, strategies for using cross-modality data are needed. After developing an automated algorithm, artificial intelligence (AI) engineers must evaluate the results of their models against ground truth labels and compare them to other algorithms. Visual assessment through Visual Analytics (VA) improves an in-depth understanding of such automated approaches. However, current VA applications are limited and do not provide flexible comparison capabilities that are able to drill down from large cohorts of patients into individual image slices. Also, they are not able to provide a view on correlations to other dataset- and image-derived features, such as from radiomics.

This thesis has two main contributions. First, we develop **two domain adaptation methods** that transfer knowledge from ceT1 to hrT2 scans. The goal is to generate automatic tumor segmentation on hrT2 images. Cross-modal data of a cohort of 242 patients, each consisting of annotated ceT1 and non-annotated hrT2 scans, are used. The methods are enhanced with a classification-guided module which avoids false positive predictions of slices. Second, we design and implement an **interactive web-based VA application** for the assessment of algorithm performance and results. We perform a quantitative evaluation and demonstrate four use case scenarios. The proposed tool allows the users to compare multiple models and subjects on different levels of detail and find correlations between performance values and radiomics features. Our best methods achieve 61.14% and 92.62% Dice Score on only tumor slices and the entire dataset, respectively. Our VA approach provides additional insight, useful for the assessment of the developed algorithms.

Contents

CHAPTER 1

Introduction

This chapter provides a brief overview of the main concepts presented in this thesis. In particular, we focus on the motivation of our thesis and our goals. At the end we outline the structure of this work.

1.1 Motivation

Radiation therapy with ionizing radiation is the most common treatment for tumors. Manual delineation of the clinical target volume (CTV) and the organs-at-risk (OAR) is an essential part of the clinical workflow in radiation therapy [116]. The CTV is the target of irradiation, whereas the OARs are organs and structures that should be spared from radiation to avoid damage to healthy surrounding tissue, such as the cochlea or the facial nerve. When treating brain tumors, identifying different brain structures in a consistent manner is critical for a successful treatment planning and for avoiding treatment failures [68]. This task, often performed manually by radiologists, is time-consuming and can result in intra- and inter-observer variability [116, 28]. Accurate automatic segmentation would improve the clinical workflow and its efficiency, but automatic approaches are not yet robust and general enough to account for all cases. Results of recent challenges in the medical imaging community suggest that neural networks are a successful approach for automatic brain structure segmentation [116].

Treatment protocols make use of multi-modality images, such as computed tomography (CT), T1-weighted and T2-weighted magnetic resonance imaging (MRI) [116, 132], since each modality provides supplementary information for treatment planning. However, using multiple modalities introduces additional challenges to neural network approaches. Common supervised deep learning techniques need a **fully labeled dataset**, which increases the workload as the number of modalities increases. Thus, there are only a few large, open source datasets available for deep learning techniques [46]. In addition,

neural networks learn the distribution of the training data. Therefore, it is not a feasible solution to use just one modality for training and **predict on another modality** at inference time without any adaptations during the training process. This creates a challenge in compiling an appropriate training and evaluation dataset. Finally, there are visually preferred modalities for brain structures based on contrast and resolution [151]. This may result in **annotations** that are not present, are biased, or are derived from other modalities and therefore not as accurate [151]. Thus, multi-modality frameworks, domain adaptation (DA) and unsupervised approaches are recently of strong interest in the medical imaging community.

In this work, we investigate how domain adaptation could be employed in the context of brain tumor segmentation across two image modalities, i.e., cross-modal data. In addition, we discuss how the results of such an approach could be visually assessed in order to support performance explainability and fine-tuning of the segmentation method.

1.2 Aim of the Work

The aim of the work is to answer the following two research questions:

(RQ1) *How can we generate brain tumor segmentations automatically for cross-modal data under the assumption that no labeled data for the target domain are available for training?*

(RQ2) *How can we visualize the outcomes of automatic segmentation methods to support software developers and artificial intelligence (AI) engineers in evaluating their developed models?*

In order to answer the research questions stated, this work is split into two parts. We translate the questions to tasks and identify limitations and requirements for each task.

Research Question 1 (RQ1): Automatic brain tumor segmentation on cross-modal MRI data

The first task is to develop an algorithm that automatically generates delineations of a brain tumor on unseen cross-modal data. The clinical use case and the resulting dataset impose some constraints on the development of the algorithm. The dataset available for development contains two image modalities: source (T1) and target (T2) modality. Paired images and ground truth labels for the source domain are available during development. However, the algorithm should predict delineations on the target modality where only unannotated raw image are available for development. For test purposes, the target modality images must have associated ground truth labels. Another assumption for the algorithm design is that the data samples of the two modalities are not paired. Finally, the algorithms should not have extreme requirements in hardware. A Nvidia GPU with 8 GB of memory is specified as reasonable equipment.

Research Question 22 (RQ2): Visual assessment of algorithm performance and results

The second task is to design and implement a visual analysis tool that supports the model evaluation process by providing an interactive interface to investigate the performance in a visual manner. Usually, the performance of the method is evaluated based on numerical measures that reduce a complex behaviour to an individual numerical value. The target users are deep learning experts and AI engineers who design solutions and train neural networks. The tool should help the user to inspect and compare the results of different segmentation methods (high level), as well as to identify data samples where errors or inaccuracies occur (low level). In addition to solving the task, the tool should ensure scalability, generalizability, and usability. The final tool should go beyond comparing just two results and be applicable to a larger number of segmentation algorithms. Although the main focus of the tool is the investigation of brain tumor segmentation results, the underlying principles should be applicable to segmentation algorithms for other body parts. The functionality provided should be understandable and usable by the users who may have varying levels of experience with interactive visualizations.

1.3 Methodological Approach

The first task is accomplished by using algorithms that employ deep learning. After researching previous work on domain adaptation, we developed two segmentation pipelines based on image alignment. Several supervised segmentation models and image synthesis networks are trained and combined into frameworks predicting the tumor segmentation masks on the target modality. In order to provide baseline methods for the evaluation, we train a fully supervised segmentation network on target modality data as upper boundary and a supervised segmentation network on source modality data, without considering the different modalities, as lower boundary. Different training settings for the approaches are tested and compared. We have chosen individual concepts that are frequently used, and therefore well-known and evaluated.

An interactive visual analysis tool fulfills the second task by providing functionality to navigate through the dataset at different levels of detail. The implementation is web-based to ensure accessibility despite different hardware settings. The results of multiple segmentation methods are presented with various Visual Analytics (VA) techniques on different levels of detail. Following Shneidermann's Mantra *"Overview first, zoom and filter, then details on demand"* [115], we provide an overview of the entire test set with all 3D volumetric data samples, followed by single patient data sample with a stack of image slices, and finally, single image slices within the 3D data sample. We present the distribution of a pre-defined performance metric with heatmaps. The relationship between the performance, the selected data and the tumor features are shown with parallel set diagrams. The visual assessment of multiple segmentation predictions for a single image slice is shown with heatmaps using explicit encoding.

This thesis contains **two main contributions**. First, we develop an automatic seg-

mentation framework to predict on the target modality based on the source modality. Second, we design and implement a VA application for artificial intelligence (AI) and machine learning (ML) experts to support the evaluation and comparison of multiple segmentation algorithms. The combination of the two contributions is novel within the context of cross-modal domain adaptation.

1.4 Outline of the Thesis

The following chapters of this thesis are structured as follows: Chapter 2 provides clinical background about Vestibular Schwannoma tumor, the brain tumor targeted, and the dataset used. In Chapter 3, we focus on related work in the field of image segmentation and comparative visualization of medical data. This includes automatic segmentation with deep learning for medical images in general and brain data in detail, domain adaptation in medical image analysis, comparative visualization and visual assessment of segmentation outcomes. In Chapter 4, we explain our data pre-processing, and provide the technical background of model architectures, activation functions, and error metrics utilized, followed by the design of our segmentation methods and their implementation. In Chapter 5, we proceed with the design and implementation details of our interactive visual analysis tool. Chapter 6 collects the results of our segmentation algorithms and insights about the model performances gained with our VA tool. Finally, Chapter 7 concludes the thesis where we reiterate about the work performed with regards to the research questions stated, reflect upon limitations of our implementations and provide potential approaches for future work.

CHAPTER 2

Clinical Background

The goal of this chapter is to provide clinical background information about Vestibular Schwannoma tumor, to position our work within the therapy workflow, and to introduce the dataset used in this work.

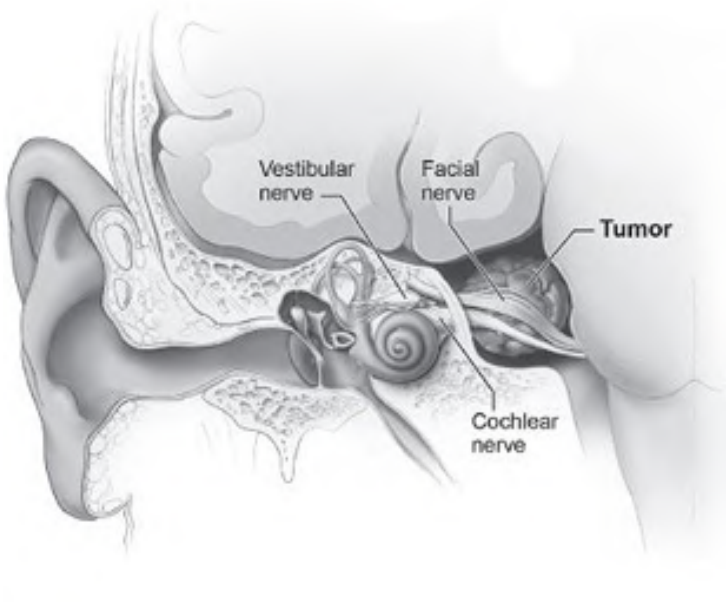


Figure 2.1: Illustration of Vestibular Schwannoma location and surrounding nerves [5].

2.1 Vestibular Schwannoma

Vestibular Schwannoma (VS), also known as acoustic neuroma, is a benign tumor affecting the balance and hearing nerves connecting the brain and the inner ear (see Figure 2.1). It accounts for approximately 8% of intracranial brain tumors and for the majority of tumors of the cerebellopontine angle (CPA) (85 – 90%). The incidence has increased over the years and was reported with 1.2 per 100,000 people per year in 2016. More than 90% of the patients have a unilateral and sporadic tumor. [21]

2.2 Therapy Workflow

Figure 2.2 shows an illustration of the steps involved in therapy planning. The series of steps is compressed to diagnosis, imaging, treatment planning, and treatment, which are described in the following subsections. Our work falls within the scope of treatment planning, where data segmentation, data exploration and analysis are subtasks.

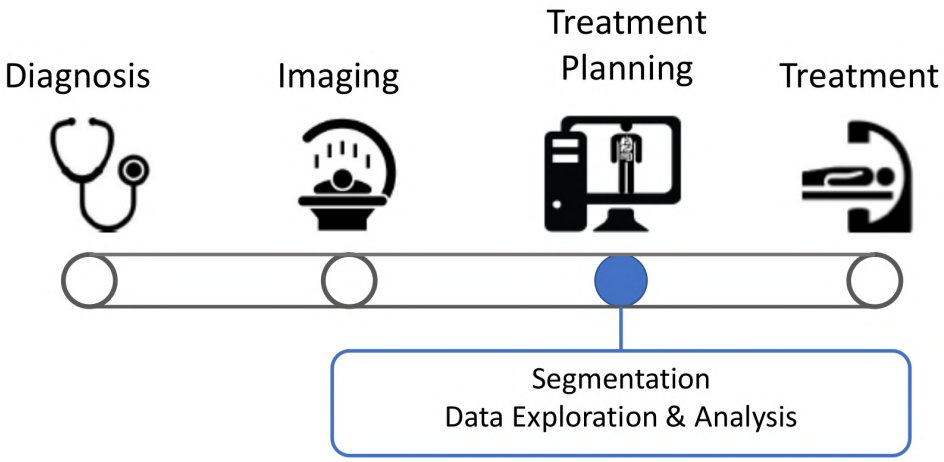


Figure 2.2: Compressed workflow of therapy planning as described by Schlachter et al. [111] for radiation therapy planning containing the steps: diagnosis, imaging, treatment planning, and treatment. Our primary focus is on treatment planning and the associated subtasks of segmentation and data exploration and analysis.

Diagnosis

The symptoms of VS include hearing loss and tinnitus, resulting from the involvement of the cochlea nerve, as well as imbalance and cranial nerve deficits [21]. Although, VS tumors are rarely fatal, they reduce the patient's quality of life. The diagnosis consists of a multidisciplinary evaluation including neurological examination, audiometric testing (i.e. audiogram) and contrast-enhanced MRI [21]. The Koos staging system with scores I - IV is a reliable tumor classification with scores rating the tumor size [37].

Imaging

The current MRI protocol include contrast-enhanced T1-weighted (ceT1) and high-resolution T2-weighted (hrT2) scans. The tumor is more visible on the former, while surrounding structures, such as the cochlea, are more visible on the latter [133]. However, recently there has been increasing concern about the use of gadolinium as contrast agent in contrast-enhanced MRIs [125, 36], supporting the shift to using other MRI modalities. Wang et al. [133] motivated their segmentation approach on T2-weighted MRI with reducing the "potentially harmful cumulative side-effects of gadolinium contrast agents". The MICCAI challenge *Cross-Modality Domain Adaptation for Medical Image Segmentation* description also claims that high-resolution T2 scans "could be a reliable, safer, and lower-cost alternative to ceT1" [3].

Treatment Planning

Since radiotherapy can cause side-effects harming the patients, the treatment needs to be carefully planned. Surrounding Organs at Risk (OARs), such as surrounding nerves and the cochlea, should be spared from damaging radiation to effectively treat the patient. Thus, treatment planning involves a lot of different essential steps. The tasks include data registration, data fusion, data segmentation, and data exploration and analysis [111]. Our work contributes to the last two steps: automatic data segmentation and segmentation outcome exploration and analysis. In clinical practice, the tumor and the relevant surrounding structures are delineated manually, which results in inter-observer variability [100, 95]. Automatic deep-learning approaches can help to reduce the time and the human effort required for delineation and make clinicians results more consistent [33]. Another aspect of achieving more stable delineations is the use of multiple image modalities [147]. Different image modalities contain different information aspects that can complement each other. However, this leads to multi-parametric, multi-modality medical imaging data that must be explored and analyzed to derive conclusions for the treatment planning and outcome. Several experts are involved in treatment planning: radiation oncologists, medical physics, radiologists, radiotherapist, and dosimetrists. The introduction of automated segmentation approaches expands this group of experts to include machine learning (ML) and deep learning (DL) engineers/experts.

Treatment Strategies

After VS is diagnosed, there are different treatment strategies [21]:

- **Microsurgery**
  Surgical resection is considered for large tumors that pose a threat of neurological impairment. The surgeon has different approaches to chose from, each with its own associated advantages and drawbacks.
- **Stereotactic Radiosurgery (SRS)**
  SRS is considered for patients with a tumor size less than or equal to $3-4$ cm. Tools

for modern SRS include the Gamma Knife [21]. It has become a preferred initial treatment due to the minimal invasive nature and the excellent clinical outcomes.

- **Fractionated Stereotactic Radiation Therapy (FSRT)**
  FSRT is applied to patients with a tumor size greater than or equal to $3-4$ cm and when functional preservation has the highest priority. Since the tumor grows slowly, this approach causes late toxicity to the surrounding normal structures, such as brainstem and cranial nerves.

- **Observation**
  Surveillance without any initial treatment is used especially for elderly patients and patients with small, asymptomatic lesions. In the surveillance period, the patient receives brain MRI scans every 1 to 2 years and additional tests, such as audiograms, are performed.

- **Combined therapy**
  A combination of multiple treatment strategies is quite rare, but may be surgery followed by radiation therapy or vice versa.

The main treatment strategies targeted by the present thesis are SRS and FSRT.

2.3 Dataset

The dataset used in this work is publicly available in The Cancer Imaging Archive (TCIA) [114] and is a collection of labeled MRI scans from 242 patients diagnosed with VS undergoing Gamma Knife stereotactic radiosurgery (GK SRS). The provided image modalities are **ceT1 and hrT2 MR images**. Facial features were obscured before the dataset was made available for anonymization reasons. An illustrative data sample before and after facial obscurification is shown in Figure 2.3. In addition, the patient's radiation therapy (RT) plan is included, denoted as RTDose, RTStructure and RTPlan. The structural information of VS tumor and **cochlea** is provided as segmentation contour lines (JSON format). The manual delineations were generated by treating neurosurgeon and physicist considering both image modalities. Due to visual preference, the tumor was segmented mostly on T1 images and cochlea on T2 images. The annotation process was performed on axial slices in the Gamma Knife planning software (Leksell GammaPlan, Elekta, Sweden) using an in-plane semi-automated segmentation method.

After downloading the dataset from TCIA, the recommended preprocessing steps are applied [73]. This includes restructuring the patient folders and converting DICOM images and JSON files into NIFTI format. The images are co-registered. Since the tumor is mainly annotated in T1 images, and the training is conducted using the pair T1 and VS tumor segmentation for the majority of our segmentation methods, we register T2 images to T1 images with the registration matrices (TFM format) to prioritize the T1 images. The affine transformation matrices are calculated based on reference points of the Leksell Stereotactic System MR Indicator box which fixates the patient's head

during image acquisition (see white dots in Figure 2.3). The calculation is performed automatic by LeksellGammaPlan software, and the resulting registration matrices are provided along with the dataset. After the co-registration, empty (i.e., all-zero) slices are removed from the volume. T1 and T2 volumes are aligned so that the correspondence of non-empty slices in the dataset for both and the mapping between the layers is preserved.

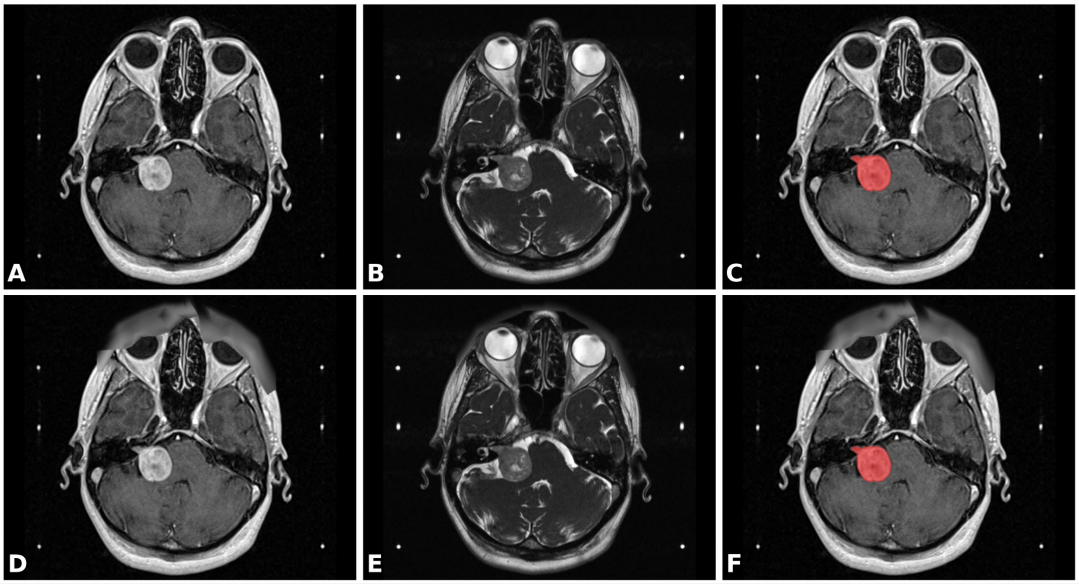


Figure 2.3: Dataset example of patient with right sided VS tumor. The six white dots in each image are reference points used for image registration. A) ceT1 MRI; B) hrT2 MRI; C) ceT1 MRI scan with VS tumor segmentation; D)-F) Corresponding images after facial obscurification [114].

The image scans were taken with a 32-channel Siemens Avanto 1.5T scanner with single-channel head coil. The contrast-enhanced T1-weighted scans have an in-plane resolution of 0.4×0.4 mm and an in-plane matrix of 512×512. The in-plane resolution of the high-resolution T2-weighted scans is 0.5×0.5 mm, the in-plane matrix is either 384×384 or 448×448. The slice thickness for both was between 1.0 and 1.5 mm.

CHAPTER 3

Related Work

The topic covered in this chapter is two-fold, i.e., image segmentation and comparative visualization of medical data. The first two sections provide an overview of semantic segmentation for biomedical applications in general (Section 3.1) and brain image data (Section 3.2). Next, we cover techniques for domain adaptation in medical image analysis with special focus on unsupervised methods, to deal with domain shift caused by different image modalities (Section 3.3). Apart from the deep learning related topics, we also take a look at medical visualization. In Section 3.4, the techniques for comparative visualization are discussed, followed by visual assessment of segmentation results in Section 3.5.

3.1 Semantic Segmentation in (Bio)Medical Images

In the last decade, deep neural networks achieved great results in computer vision tasks such as classification, object detection and segmentation. Especially Convolutional Neural Network (CNN) architectures such as *AlexNet* [78], *VGGNet* [118], *GoogleLeNet* [119], *ResNet* [50], *MobileNet* [52], and *DenseNet* [53] are very popular for image classification. After their success in classification challenges, CNNs have also been adopted for semantic segmentation in a variety of applications.

One of the first deep learning networks for semantic image segmentation was introduced by Long et al. [86]. A *Fully Convolutional Network (FCN)* with skip connections between downsampling and upsampling path is trained end-to-end for supervised pixelwise predictions. Since only convolutional layers are used, the network can take an image of arbitrary size. However, FCNs are slow in real-time inference and they are not using the global context information in an efficient way. One of the most common encoder-decoder network is *UNet* by Ronneberger et al. [108]. It was originally developed for electron microscopic (EM) images and won the International Symposium on Biomedical Imaging (ISBI) cell tracking challenge 2015. Although developed for biomedical data, it has been

one of the major breakthroughs for medical image segmentation and is now also used outside the medical domain [92, 14].

Numerous extensions and adaptations of the *UNet* have been developed to overcome limitations of the original architecture and to adopt the idea to different kinds of images. This paragraph only covers a selected collection. Drozdzal et al. [35] investigate the importance of skip connections for biomedical image segmentation by comparing results of FCN with long and short skip connections. Another network for multiple medical images segmentation tasks, which deals with further development of skip connections, is *UNet++* by Zhou et al. [148]. Aside from deep supervision, they use nested and dense skip connections to fuse semantically more similar feature maps. *UNet3+* by Huang et al. [54] uses full-scale skip connections and deep supervision for liver and spleen CT segmentation. In addition, they introduce a classification-guided module to reduce false-positives in a non-organ image. A 3D version of *UNet*, called *3D UNet*, for volumetric data of the Xenopus kidney is introduced by Çiçek et al. [29]. Another well-known 3D variant using the encoder-decoder scheme is *V-Net* by Milletari et al. [93], which was first used for volumetric prostate MRI data. Li et al. [84] combine a 2D *DenseUNet* and a 3D counterpart in their *H-Dense UNet* for liver and tumor segmentation in CT scans. *Attention-UNet* by Oktay et al. [102] inserts attention gates to increase the focus on the target structures in multi-class abdominal CT segmentation. Li et al. [83] combine the idea of attention gates, dense skip connection and deep supervision in the *Attention-UNet++* for liver CT segmentation. Zhang et al. [145] use residual units in the UNet architecture and apply their deep *ResUNet* to road extraction from aerial images. *X-Net* by Bullock et al. [19] combines two *UNets* in series for bone and soft tissue segmentation in X-Ray images. Isensee et al. [60] introduce a robust and self-adapting framework called "no-new-Net" (*nnU-Net*). The idea behind the framework is that solving segmentation on a novel dataset is influenced by a lot of inter-connected choices regarding architecture, pre-processing, training and inference (i.e., post-processing). Designing a new architecture is not always the best way, since a lot of components influence the performance in the end. The framework tries to find the best combination based on 2D and 3D vanilla *UNet* and a *Cascaded UNet*.

Another class of segmentation networks are R-CNN based models. A well-known candidate of this family is the *Mask-RCNN* by He et al. [48], which can be used for instance segmentation. Mask-RCNN consists of two stages. First, the Region Proposal Network (RPN) predicts bounding box candidates for objects. Then, extracted features are used to perform classification, bounding box regression and pixel-wise segmentation in parallel. Zhou et al. [148] used their resdesigned skip connections to improve segmentation backbone and boost the medical image segmentation performance. *SegAN* for medical image segmentation was published by Xue et al. [138]. Their adversarial network consists of a FCN as segmentor and a critic network trying to differentiate between predicted and ground truth segmentation maps. The two networks are trained end-to-end in a min-max manner. We refer the interested readers to the survey about image segmentation using deep learning by Minaee et al. [94].

3.2 Semantic Segmentation in Brain Images

In recent years, there have been several segmentation challenges on medical data focused on advancing research in different fields. At the Medical Image Computing and Computer Assisted Intervention (MICCAI) 2020 alone, nearly all challenges included a segmentation task for various (clinical) use cases. We only focus on the most relevant challenges dealing with multi-modal brain data.

The most related MICCAI challenge is the MICCAI challenge 2021 *Cross-Modality Domain Adaptation for Medical Image Segmentation* (*crossMoDA*) [3]. The task is to predict VS and cochlear segmentation on hrT2 images. Compared to the following introduced challenges, the limitation for network training is, that only ceT1 with corresponding segmentation masks and unpaired hrT2 can be used. For validation the mean Dice score and Average Symmetric Surface Distance per structure and overall are used. In the first validation phase, 5 challenge teams (10%) reach a performance above 80% mean Dice score and 45 teams (85%) achieved results below 60% mean Dice score. If we look at the top 10 teams, there is a clear trend in strategies [3]. The top three methods are combinations of image alignment and *nnU-Net*. What sets the first two apart, is the use of self-supervision. The challenge winner generates pseudo-labels on non-annotated hrT2 images with a trained pipeline consisting of *CycleGAN* and segmentation network. The *nnU-Net* segmentation network is then re-trained with the labeled synthetic T2 and the pseudo-labeled T2 scans. The team on the second place uses *NiceGAN*, an extension of *CycleGAN*, for pixel alignment. For the segmentation, they trained several *nnU-Nets* tailored to the two different hrT2 modalities and the different label classes. The goal of the third team is to use publicly available frameworks. Therefore, their image alignment is performed with *CUT* (Contrast unpaired translation) that uses patchwise contrastive learning and adversarial learning and *nnU-Net*. Due to the usage of large networks, all strategies are trained on Nvidia GPUs with more than 8 GB. An extended dataset of the challenge published via TCIA is used for this work (see 2.3). Wang et al. [133] proposed the first automatic VS tumor segmentation. They combine 2D convolutions in earlier layers and 3D convolutions in deeper layers to form a 2.5D UNet. Additionally, they extend the attention module by Oktay et al. [102] to supervise the attention learning on different scales explicitly. The network was trained with and tested on hrT2 images.

The MICCAI 2020 challenge *Anatomical Brain Barriers to Cancer Spread: Segmentation from CT and MR images* (*ABCs*) [1] aimed at developing fully automatic segmentation algorithm for brain structures that will make radiotherapy planning more efficient and consistent. Two segmentation tasks are performed on paired and registered CT and MRI scans. The first task is focused on the clinical target volume (CTV) and limiting brain structures, the second task deals with supporting structures used to optimize the radiotherapy treatment plan. The results of the top 10 challenge participants are summarized by Shusharina et al. [116]. An analysis regarding submitted architecture shows that eight out of ten used a *3D UNet* [29] and four out of ten used *nnU-Net* [60], including the first two. Ning et al. [99] won the challenge with their residual *UNet* architecture. For optimization, they use a hybrid loss between Tversky loss and Dice

loss. Further improvement is achieved through an ensemble strategy. The algorithm by Chen et al. [27] uses a coarse-to-fine approach in three stages. The ROI for the targets extracted in Step 1 is used in Step 2 to perform fine segmentation on zoomed in image. The final decision is made in Step 3 based on a fusion of all previous feature maps and predictions. Zou and Dou [151] leverage domain knowledge for model training and the symmetric of structures for label merging. At test time, the best two models are used in a model ensemble strategy. An uni-modal approach, only leveraging one modality, is described by Langhans et al. [81]. Gay et al. [41] used a bi-directional two-stage framework. Large structures were segmented with an *Attention U-Net* on MRI images only. For small structures, the ROI is determined by training a *Inception-ResNet-v2* and using either an *Attention U-Net* or a *V-Net* depending on the target structure for automatic segmentation on smaller image volumes.

A challenge that already exists since 2012 is the MICCAI challenge *Brain Tumor Segmentation Challenge* (*BraTS*) [2]. Over the years, the complexity of the challenge tasks has widened from segmentation to overall survival prediction and classification uncertainty prediction. The challenge dataset is often used as benchmark resource for brain tumor segmentation, due to the publicly accessible multi-modal dataset offered to the medical research community [10, 91]. The dataset consists of native and post-contrast T1-weighted, T2-weighted and T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) MRI scans. In their survey, Agravat and Raval [10] summarize the change of trends in automated glioma brain tumor segmentation approaches from methods using handcrafted features to the usage of deep neural networks. They cover the BraTS challenges 2012-2019. Another survey by Magadza and Viriri [91] focuses on different strategies and building blocks for brain tumor segmentation. Isensee et al. [59] won the first place in the BraTS 2020 challenge with an *nnU-Net* adapted to the brain tumor segmentation task. Jia et al. [63] extended their previous 3D High-resolution and Non-local Feature Network (*HNF-Net*) [64] to use a two-stage cascaded HNF-Net. Together with Wang et al. [135] they rank second. Wang et al. [135] develop a modality-pairing network with two parallel branches connected via skip connections.

The *HEad and neCK TumOR segmentation challenge* (*HECKTOR*) [4] was held in conjunction with MICCAI 2020 and MICCAI 2021 to motivate researchers to leverage bi-modal information for head and neck primary tumor segmentation from PET and CT images. While the focus of the first year was only on segmentation, in the second year the challenge was expanded to include patient outcome prediction with and without the tumor delineation. The challenge organizers provide a baseline with 3D V-Net and a 2D version [13]. They compare bi-modal networks with early and late fusion and single modality dedicated networks. All top 10 ranked teams with one exception submitted a 3D-UNet like architecture with different strategies regarding pre- and post-processing, loss function and usage of data augmentation and ensemble strategies. The challenge winner is a residual UNet with integrated 'Squeeze and Excitation' (SE) normalization by Iantsen et al. [58]. The final prediction is generated by an ensemble of 8 networks. The SE layer was developed by Iantsen et al. [57] for a submission at the BraTS 2020

challenge. Ma and Yang [89] train a *3D UNet* and perform at test time an uncertainty estimation with model ensembles. For high uncertainty cases, active contours is used as post-processing. A *3D nnU-Net* model supplemented with 'spatial and channel Squeeze and Excitation' (scSE) blocks is proposed by Xie and Peng [137]. The dynamic Scale Attention Network (*SA-Net*) by Yuan [143] uses full-scale skip connections to combine feature maps at different scales. The same strategy was also submitted at BraTs 2020 [142]. Chen et al. [26] use a three-step framework to refine the segmentation results in an iterative manner by using additional information from upstream trained models. Ghimire et al. [43] introduce a patch-based *3D UNet* with both conventional and dilated convolution to tackle the 3D image memory issue and use small and large receptive fields. Yousefirizi and Rahmim [141] modified a GAN for medical image segmentation (*SegAN*) with an improved polyphase *V-Net* as generator and an encoder-similar discriminator. The only 2D approach is used by Zhu et al. [150] in a two-stage framework. A ResNet-based classifier predicts the axial slices displaying a tumor which are then segmented via a *2D UNet*.

Our work is inspired by the MICCAI challenge *crossMoDA* dataset and its clinical use case. The task of unsupervised domain adaptation with underlying cross-modality is very different from the automatic segmentation algorithms studied. While they address the multi-modality aspect, they lack the challenge of missing target labels, which pose an additional difficulty in segmenting brain structures. This is the research direction that the present thesis aims to contribute to.

3.3 Domain Adaptation in Medical Image Analysis

Transfer learning (TL) aims to transfer the knowledge which has been learned from task T_A on domain A to the task T_B on domain B. Either the domain (feature space) or the task (label space) change in transfer learning. *Domain adaptation* is a special case of TL, where only the domain changes. As a toy example to explain TL and DA, we employ the common deep learning example of cat and dog image classification. Let task T_A be the classification of cat images within the domain A of natural images. Applying the knowledge learned by this task to the problem of classifying dog images (different task $T_B \neq T_A$) on natural images (same domain $B = A$) is TL with constant domain. One strategy for TL with same domain is to re-use the weights of a model trained on task T_A as a starting point for the task T_B training. Transferring the knowledge to the classification of cats (same task $T_A = T_B$) in cartoon images (different domain $A \neq B$) is still considered TL, but more precisely it is DA. The change in data distribution between the training data (source domain) and the test data (target domain) is referred to as data shift problem [104]. The data shift is common for medical image analysis due to different scanners and scanning parameters, subject cohorts and multi centers studies, and changing image modalities. Thus, DA is of great interest for the medical image analysis community to overcome the domain shift and heterogeneity among datasets [46, 77].

There are different taxonomies for DA. Guan and Liu [46] provide five categories to describe a DA method. The descriptor classes are not mutually excluding to each other and are based on model types, label availability, modality difference, number of sources and adaptation steps. Figure 3.1 shows each category and where our work is located. Based on the strategy of knowledge transfer, we can distinguish four major alignment techniques for unsupervised DA methods: feature alignment, image alignment, feature+image alignment, and disentangled representation. Another way of subdividing unsupervised DA is given by Kouw and Loog [77]. They distinguish between sample-based, feature-based and inference-based. We use the alignment strategy for categorization. Table 3.1 provides an overview of different domain adaption methods in medical image analysis that is discussed in this section. For a more comprehensive literature review, we refer the reader to existing review papers [46, 77, 120].

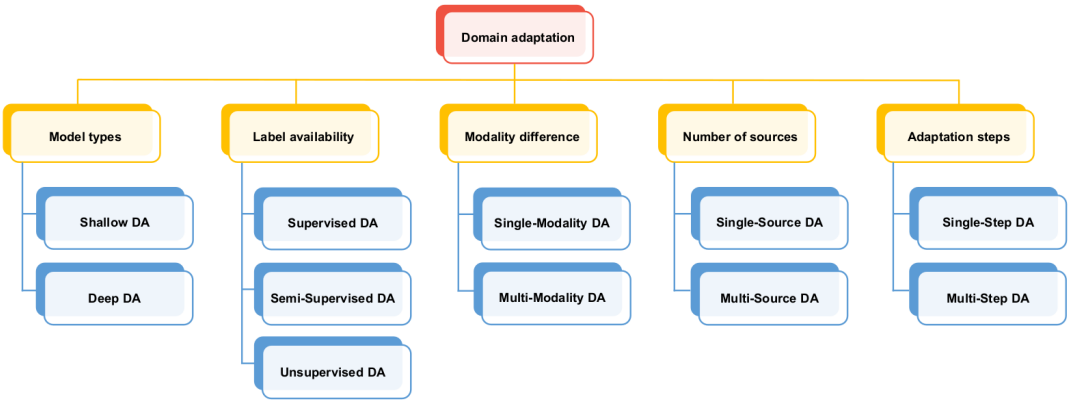


Figure 3.1: Overview of domain adaptation categories after Guan and Liu [46].

A well known deep generative model is the Generative Adversarial Network (*GAN*) [45]. It consists of a generator and a discriminator which are trained simultaneously in a min-max-game manner. The generator takes random noise as input and generates synthetic images. The discriminator tries to distinguish between fake and real images. For realistically looking synthetic images, a good balance between the generator and discriminator is necessary. Based on the idea of GANs, Zhu et al. [149] introduced *CycleGAN*, an unpaired image-to-image translation with cycle-consistent adversarial networks. For unsupervised domain adaptation, CycleGAN-based frameworks are used to perform *image alignment* either by transforming source domain images into target domain images or vice versa [51, 17, 109].

In the context of medical image segmentation, *image alignment* is the most commonly used approach (see Table 3.1). The goal of the work by Chartsias et al. [22] is to perform cardiac synthesis on unpaired CT and MRI data. A direct application of *CycleGAN* leads to an anatomical shift that causes the segmentation masks between the original and synthetic images to be misaligned. Thus, they concatenate images and corresponding

Publication	target GT	modality	segm domain	alignment
Charsias et al. [22]	✓	CT & MRI	both	image
Jiang et al. [65]	✓	CT & MRI	target	image
Valindria et al. [126]	✓	CT & MRI	both	feature
Zhang et al. [146]	✓	CT & MRI	both	image+feature
Chen et al.[23]	✗	X-Ray	source	image
Zhang et al. [144]	✗	X-Ray & DRR	source	image
Huo et al. [55, 56]	✗	CT & MRI	target	image
Kamnitsas et al. [67]	✗	MRI	target	feature
Joyce et al. [66]	✗	CT & MRI	source	feature
Dou et al. [34]	✗	CT & MRI	target	feature
Chen et al. [24, 25]	✗	CT & MRI	target	image+feature
Yan et al. [139]	✗	MRI	source	image+feature
Yang et al. [140]	✗	CT & MRI	source	disentangled

Table 3.1: Overview of domain adaptation methods in medical image analysis. The first part of the table includes methods where ground truth labels in the target domain are available for training neural networks, the second part is lacking this ground truth data.

segmentation masks to a two-channel input for the *CycleGAN*. This mitigates the problem of dislocated segmentation label for synthetic images. Jiang et al. [65] introduce a two-step tumor-aware approach for semi-supervised lung cancer segmentation in MRI data. The first step is a *CycleGAN* supplemented with tumor-aware loss to preserve tumor structure. After the MRI synthesis, a semi-supervised segmentation network is trained with synthesized and a limited number of real MRIs. The "Synthetic Segmentation Network" (*SynSeg-Net*) by Huo et al. [56, 55] is a similar combination of *CycleGAN* and segmentor. In contrast to the previously introduced approaches, the segmentation is only performed on synthetically generated target images since no target ground truth is available. *CycleGANs* enhanced with semantic consistency are used to transform X-Ray images (target domain) towards source images and perform X-Ray segmentation by Zhang et al. [144] and Chen et al. [23].

Next, we discuss methods using *feature alignment*. Ganin et al. [40] introduce the "Domain Adversarial Neural Networks" (*DANN*) framework. The *feature alignment* is implemented by a shared CNN encoder between a task-specific classifier and domain classifier. The purpose of the domain classifier is to support a domain-invariant representation. This idea is extended by Kamnitsas et al. [67] for MRI segmentation of traumatic brain injuries. The task-related classifier is exchanged for a segmenter invariant to domain-specific representations. "Adversarial Discriminative Domain Adaptation" (*ADDA*) by Tzeng et al. [123] is focused on a discriminative representation. After pre-training a source encoder with supervised classification task, the weights are used as initialization for adversarial adaptation. The source and target mapping are independently trained in an adversarial manner. At test time, target images are transformed with the target

encoder and classified with the source classifier. Joyce et al. [66] use feature alignment to stabilize the training of an adversarial segmenter. Instead of images, they directly generate synthetic segmentation masks that have no point-to-point correspondence with the original image. A reconstruction loss, a loss to regularize the size and the intensity of segmentation masks are introduced as additional unsupervised costs. Dou et al. [34] introduce *PnP-AdaNet*, an unsupervised cross-modality DA framework for cardiac multi-class image segmentation. MR (source) and CT (target) images are unpaired. Their feature alignment strategy is based on the convention, that the first layers learn low-level features (domain-specific) and the higher level task-specific features (domain-invariant). Two independent encoders, one for each domain, share one segmentation decoder. The adversarial learning is realized by two discriminators, one for multi-level features and another for predicted segmentation masks. Another feature-alignment technique is to share latent representation between two domains at different locations in the network [126]. However, for this, source and target ground truth labels need to be available.

The next category covers approaches using *image and feature alignment* techniques. Zhang et al. [146] combine a *CycleGAN* with two *UNets* in an end-to-end supervised framework that works on unpaired but labeled data. The synthetic data from the generators is boosting the segmentor training and the generator loss includes a segmentor supervised shape-consistency loss. Unlike this work, the following methods do not rely on labeled target data. "Cycle-Consistent Adversarial Domain Adaptation" (*CyCADA*) by Hoffmann et al. [51] employs a combination of *image and feature alignment* where cycle-consistency is used along with a semantic loss that constrains the mapping between domains. This approach was tested on a number of natural image recognition and prediction tasks. Chen et al. [24, 25] are solving the task of cardiac multi-class image segmentation with their "Synergistic Image and Feature Adaptation" framework (*SIFA*). Image transformation with a CycleGAN data flow ensures the image alignment. The segmentation model trained on real and synthesized target images shares a feature encoder with the decoder stream that generates source-like images. Discriminators in semantic prediction and generated image space complete the feature alignment. A similar unsupervised framework is proposed by Yan et al. [139]. Their *UNet-GAN* consists of two stages. First, a UNet is trained on labeled source data in a supervised manner. Then, a *CycleGAN* is trained with a loss function that combines image- and feature-level loss. To facilitate feature-level alignment, the MSE of features extracted from the *UNet* encoder between the original and generated images is calculated. At test time, the target data is transformed to source-like data that is fed to the *UNet*.

Disentangled representations of CT and MRI scans are used by Yang et al. [140] in a two-step domain adaptation framework. In the first step, images from both domains are transformed into a domain-specific style space and a domain-invariant content space. The representation in the content space contains anatomical structural information that is independent of the modality. Content-only CT images are used to train a segmentation model that is applied to content-only MRI data at test time.

Our task classifies as deep, multi-modality, single source DA according to the taxonomy by Guan and Liu [46]. Existing works use only data samples with corresponding segmentation labels for the segmentation dedicated framework part. In order to extend the dataset that can be actively used during training, we add a classification-guided module to two unsupervised approaches that resemble an *UNet-GAN* with image-feature only alignment. This algorithmic extension is the main technical contribution of the present thesis. Compared to methods that were submitted to the *crossMoDA* competition and rely on high GPU memory (> 8 GB), our approaches work on a GPU with 8 GB of memory.

3.4 Comparative Visualization

The visual assessment of at least two data samples with respect to each other is defined as comparative visualization. Kim et al. [71] classify comparative visualization approaches into four fundamental techniques, as displayed in Figure 3.2. These are Juxtaposition, Superimposition, Interchangeable, and Explicit Encoding. Juxtaposition is a side-by-side view of multiple data samples in different coordinate spaces. Superimposition is an overlay of data samples integrating them to the same coordinate system. An approach is interchangeable when data samples are viewed sequentially in the same coordinate system, e.g. by means of animation. Explicit encoding refers to calculating and displaying a composite of multiple data samples, such as the difference or intersection. Thereof, the authors propose hybrid methods, i.e., to combine multiple of the traditional approaches or switch between them, to overcome drawbacks of single methods. An additional dimension in comparative visualization is the amount of instances to be compared. There are 1-by-1 comparisons (one patient vs another), 1-by-n (one patient vs a cohort, i.e., group of patients), and n-by-n (a cohort vs another).

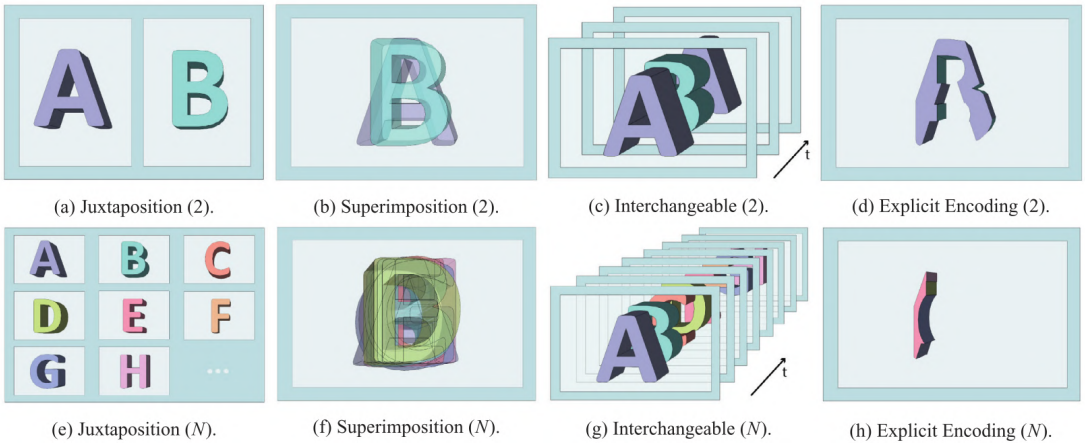


Figure 3.2: Four fundamental visualization techniques for comparative visualization for 2 (top row) and N (bottom row) data samples: Juxtaposition (a,e), Superimposition (b,f), Interchangeable (c,g), Explicit Encoding (d,h) [71].

For the comparison of polygonal meshes, the mesh processing system MeshLab [30] and PolyMeCo [117] for mesh analysis and comparison are proposed. The drawback of both tools is that they only support pairwise comparison. Schmidt et al. [112] introduce YMCA (Your Mesh Comparison Application) for meshes produced from point clouds. The surface reconstruction of multiple algorithms can be compared against each other in an interactive way. In addition to a 2D overview, local variations in the spatial information are visualized using focus-and-context techniques. The main target of the application is the visual quality of the mesh reconstruction.

Displaying multiple data samples at once to provide important insight is especially useful for multi-modal medical data visualization. In this context, the different data samples originate from multiple image modalities. Lawonn et al. [82] name basic visualization techniques to deal with occlusion, poor depth perception and rendering relevant data components from multiple sources in their survey. Besides smart visibility techniques to avoid visual clutter or occlusion while maximizing the visual information, they discuss focus-and-context as an approach to highlight a focus object in the context of surrounding structures in their survey.

The topics of ensemble visualization and cohort analysis are related to comparative visualization and are not only relevant for the medical field. The interested reader is referred to other publications [134, 75, 16].

3.5 Visual Assessment of Segmentation Outcomes

The visual assessment of segmentation results can have different types of emphases and applications. As a result, the research field is very broad and contains (not exclusively) parameter space exploration, uncertainty or sensitivity visualization, and Visual Analytics (VA) approaches . We focus on a selection of related works dealing with medical image segmentation. For VA applications in the area of public health in general, the interested read is referred to the survey by Preim and Lawonn [103].

Tuner by Torsney-Weir et al. [122] and *GEMSe* Fröhler et al. [38] are interactive tools to explore the parameter space. *Tuner* is intended for developers of segmentation algorithms to help them find a "good" parameter setting. *GEMSe* allows an interactive exploration of parameter combinations in multi-channel segmentation algorithms by using a hierarchically clustered image trees.

In general, uncertainty and sensitivity information is important to visualize. As uncertainty is present in each stage of the radiotherapy planning workflow, it influences the individual steps [105]. The task of uncertainty-aware visualization in medical imaging is the communication of uncertainties in the therapy workflow that might have an influence on the decision-making process and the treatment planning [44]. *ProbExplorer* by Saad et al. [110] is an interactive tool for analyzing and editing probabilistic segmentation results. The uncertainty is quantified based on Bayesian decision theory and is color-encoded for suspicious region highlighting. Al-Taie et al. [11] introduce an uni-modal approach

for uncertainty estimation and visualization. Using the Kullback–Leibler divergence, also called the total variation divergence, an uncertainty map is calculated where each pixel that can not be assigned to a class with a certain probability is color-encoded. The authors extend their work to multi-modal imaging data [12]. The segmentation results for the multi-modal application are produced by a combination of probabilistic uni-modal algorithms, i.e., an ensemble of classifiers. The use case is the segmentation of multiple sclerosis lesions involving multiple MR modalities. In addition to comparing different ensemble classifier, they estimate and visualize the brain tumor growth over time as the areas of segmentation mask growth or shrinkage result in high uncertainty values. Nair et al. [97] utilize Monte-Carlo dropout to calculate multiple uncertainty measures that are visualized with the segmentation results in the original image. Another tool that is used to generate uncertainty measures for medical image segmentation is Bayesian neural networks [79, 62]. Apart from the segmentation mask, they also express the uncertainty of the segmentation results. For additional literature, we refer the reader to surveys by Gillmann et al. [44] and Raidou [105].

Visual Analytics (VA) is a combination of automated data analysis and interactive visualization to enable an effective analysis of data [70]. VA has also dealt with the topic of analyzing segmentation outcome results. Landesberger et al. [130, 131, 80] provide several VA tools that support the visualization of the segmentation process and outcomes of medical image segmentation with statistical shape models. The first approach [130] visualizes the convergence behavior for global (i.e., full organ) and local data (i.e., organ regions and landmarks). The user can select only one subject for analysis. In a follow-up work by Landesberger et al. [131], the visualization applications is extended to the entire workflow, i.e., data pre-processing, model selection, model-based segmentation, and model evaluation, including linked-views of the histogram of quality values for the dataset with detailed views for selected objects [131]. In a later work, Landesberger et al. [80] deal with the task of detecting segmentation error that occur systematically. First, an overview of the similarity in segmentation quality for the whole dataset is provided. Then, outliers and instances of special interest are shown in a detailed view for the inspection of samples selected by the user. In order to find regions with common segmentation quality profiles across a dataset, point correspondence of landmarks is required. The approach is limited to one dataset, i.e., a ground truth and automatic segmentation prediction from a single algorithm. Geurts et al. [42] published a method to compare and evaluate several statistical shape models for 3D medical image segmentation. Scatterplots of global quality measures, such as Hausdorff Distance and Average Surface Distance, provide a pairwise algorithm comparison. Regions showing systematic quality properties are clustered with a quality-based clustering method and the best algorithm per region is determined. The results of this analysis are shown in a regional quality comparison view. Raidou et al. [106] propose a VA tool that facilitates exploration and visual assessment of segmentation errors. Local quality measures and response profiles for features are supported. The results from a single shape model segmentation algorithm are considered on the level of a cohort and a individual subject. Reiter et al. [107] have extended the web-based VA tool to allow the analysis of shape and size variability and

their influence on the segmentation result. There are also VA applications that are not specialized in medical segmentation, but have applications in other areas of medical image processing [39, 85].

Table 3.2 compares the VA tools for medical image segmentation with respect to the criteria of multiple model support, multiple (i.e., cohort) and individual subject visualizations. All methods provide some kind of "overview-and-focus" approach and have underlying segmentation meshes from (statistical) shape models. Our work supports the comparison of multiple deep-learning based segmentation algorithms for a whole dataset (i.e., a cohort) and individual data samples (i.e., individual subjects). Based on our experience with deep learning model development, we have defined five tasks. This are overall performance comparison (**T1**), per patient performance comparison (**T2**), per slice performance comparison (**T3**), relationship to imaging-derived features (**T4**), and anatomy-based predictions (**T5**). The approaches support **T1**, **T2**, and mostly a mesh-based variation of **T3** (denoted with **T3***). By exploring and analyzing data at different levels of detail, our approach is more flexible than the state-of-the-art and covers all five tasks.

Publication	multiple models	multiple subjects	individual subject	comparison	tasks
Landesberger et al. [130]	✗	✗	✓	-	**T1**,**T2**
Landesberger et al. [131]	✗	✓	✓	1-by-1	**T1-T3**
Landesberger et al. [80]	✗	✓	✓	1-by-n	**T1-T3***
Geurts et al. [42]	(✓)	✓	✓	1-by-1	**T1-T3***
Raidou et al. [106]	✗	✓	✓	n-by-n	**T1**,**T2**
Reiter et al. [107]	✗	✓	✓	n-by-n	**T1**,**T2**
Ours	✓	✓	✓	flexible	**T1-T5**

Table 3.2: Overview of VA methods in medical image segmentation with (statistical) shape models. For multiple models, (✓) stands for pairwise comparison only, whereas ✓is for more than only two models. **T3*** is the mesh-based equivalent of per slice performance comparison.

CHAPTER 4

Automatic Segmentation Methods for Cross-Modal Data

This chapter is dedicated to describe the automatic segmentation approaches that predict VS delineations on hrT2 scans. After covering the data processing pipeline and theoretical background, we introduce our methods and explain their implementation.

4.1 Data Pre-Processing

The dataset consists of 242 folders containing ceT1 and hrT2 volumetric MRI scans with corresponding tumor segmentation masks. The source and content are described in Section 2.3. In this section, we discuss the processing steps applied to the data before it is fed into a neural network. The pipeline is depicted in Figure 4.1. First, we split the dataset into subsets for training, validation, and testing. We analyze the data distribution of all three subsets. Then, we calculate for each data volume statistical values, i.e., 1st and 99th percentile, mean, standard deviation, overall minimal and maximal value. They are used at loading time to pre-process the raw T1 and T2 slices. In addition, the data augmentation used during training for selected neural networks is explained.

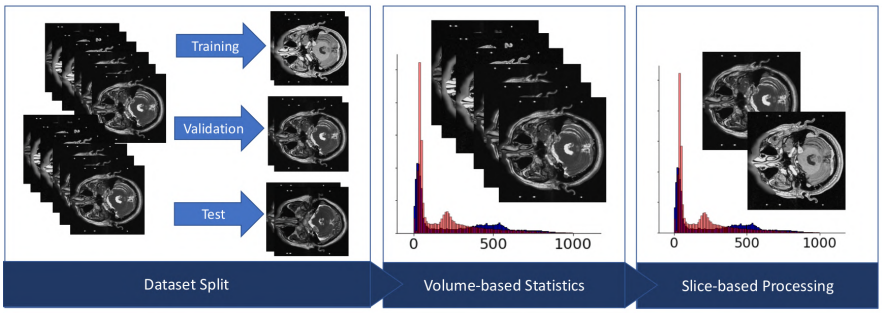


Figure 4.1: Pipeline for data pre-processing steps. First, the data is split into three subsets. Then, volumetric statistics that are used to process each slice are calculated.

4.1.1 Dataset Split and Distribution

It is common practice to divide a dataset for neural network training into three subsets. The training dataset is used for the actual training, the validation dataset is used for validation or optimization during the training phase and the test dataset is locked away for a final evaluation. Our split is 64 : 16 : 20 for training, validation, testing, respectively, which means 80% of the data is used during the algorithm development and 20% is used for the final evaluation, which is discussed in Section 6.1. The split is folder-based and each folder contains volumetric data with a varying number of slices. Some subject IDs are removed due to being incomplete or because they were missing in the first place. Details about the dataset split are summarized in Table 4.1. Depending on the task, there are different filtering options for the subsets based on the VS segmentation information:

D1 Full dataset with all non-zero slices of all subjects. The dataset is highly imbalanced with respect to tumor presence in the individual image slices since only a subset of slices per volume display the tumor (see Figure 4.2a).

D2 Balanced dataset with respect to tumor presence, i.e., the number of image slices with and without tumor presence is equal. The slices not displaying a tumor are chosen randomly.

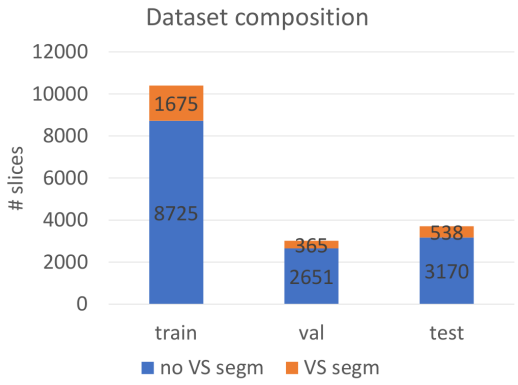
D3 Slices displaying the tumor. The slices are filtered by checking the segmentation mask size after resizing to the image size specified (default: $H = 256$, $W = 256$).

D4 Slices displaying the tumor with a certain size, i.e., number of pixels in the segmentation mask. Figure 4.2b shows that many slices have a small (i.e. < 50 pixel counts) delineations associated. This information is useful when the data sample is viewed as 3D volume. Then, the tumor volume is more accurate since the tumor boundary is not cut off at the top and bottom. For training 2D methods, the small masks are very challenging to detect which is discussed in Section 6.1.

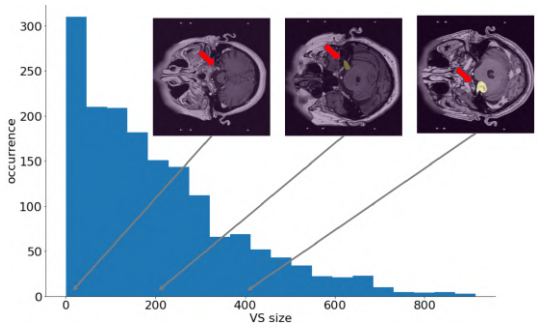
	# folders	IDs	removed IDs	%	# slices	# VS slices
train	155	1-158	[39, 97, 130]	64%	10400	1675
validation	39	159-199	[160, 168]	16%	3016	365
test	48	200-250	[208, 219, 227]	20 %	3708	538
$\sum$	242	1 - 250	total of 8 IDs	100 %	17277	2578

Table 4.1: Overview of training/validation/test split.

Figure 4.3 shows the histogram of pixel values for slices with VS segmentation for the training, validation and testing dataset of 1675, 365 and 538 images, respectively. It demonstrates the data shift between source (ceT1) and target (hrT2) domain which means that the distributions are different.

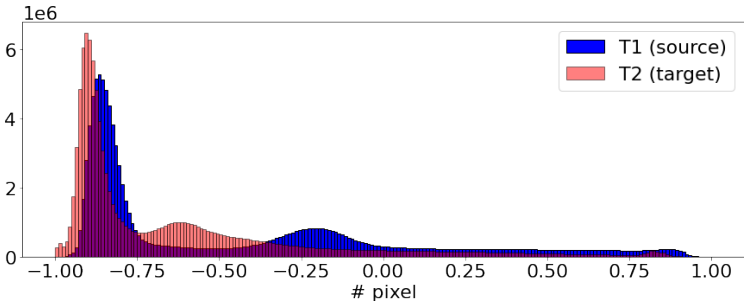


(a) Full dataset.

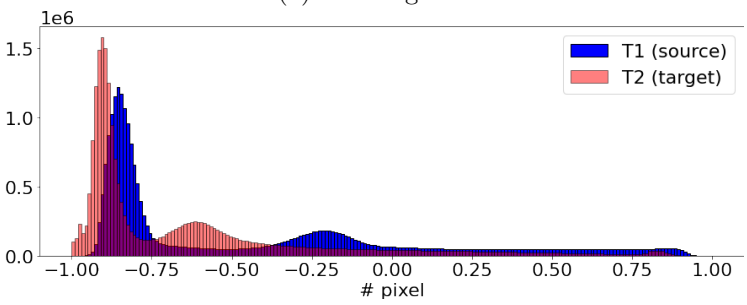


(b) Tumor size distribution in training set.

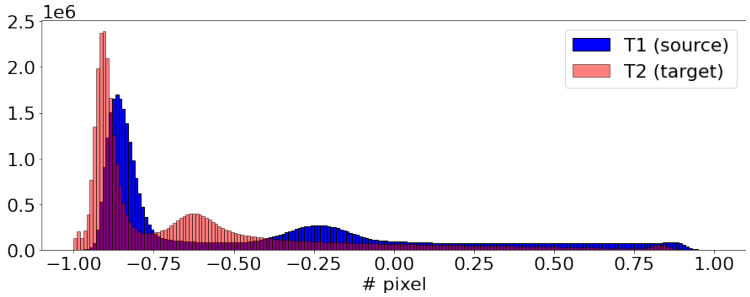
Figure 4.2: Illustration of dataset composition in training, validation, testing subset: (a) Full dataset with/without tumor presence; (b) Tumor size distribution in training subset.



(a) Training set.



(b) Validation set.



(c) Test set.

Figure 4.3: Data shift for source and target domain visualized by histogram of pixel values for training (a), validation (b) and testing (c) subset.

4.1.2 Volume-based Statistics

Statistical values of the volumetric data sample X are pre-calculated to be used in the slice-based processing. The following values are calculated for ceT1 and hrT2 separately:

s1 1st and 99th percentile of the pixel values in a sample X to reduce outliers.

s2 Mean μ_X and standard deviation σ_X for z-score normalization.

s3 Overall minimal $min(X)$ and maximal $max(X)$ intensity values for normalization.

4.1.3 Slice-based Processing

At loading time, we deal with 2D image slices since we use 2D models. However, the underlying data is volumetric, i.e. a stack of slices. In order to preserve the volume statistics and process the slices belonging to a data volume the same way, the volume-based pre-calculated statistical values are used to apply processing steps to each slice $x \in X$. The pre-calculation is beneficial for computational efficiency since not the entire data volume needs to be accessed to calculate the parameters for every slice at run time. Figure 4.4 shows the histogram of a exemplary data volume for T1 and T2 and how the steps **S1-S4** transform the image value range.

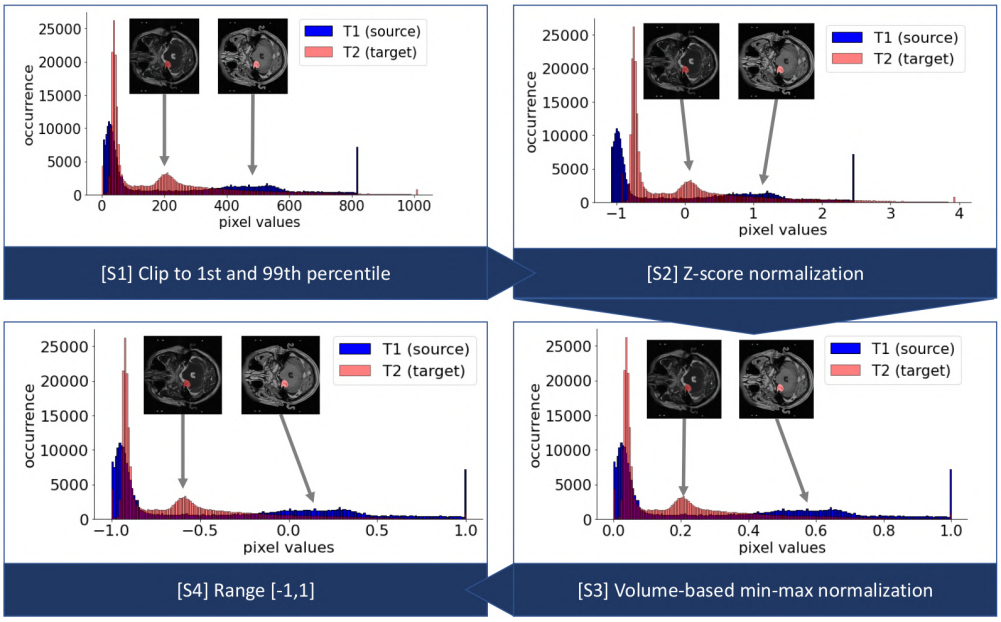


Figure 4.4: Histograms of image values to illustrate the slice-based processing steps **S1-S4**: clip intensity values to 1st and 99th percentile, z-score normalization, volume-based min-max normalization, and slice-based min-max normalization.

S1 Clip the image values of a data slice x below the pre-calculated 1st and above the 99th percentile to remove extreme outliers [81].

S2 Perform z-score normalization with pre-calculated mean μ_X and standard deviation σ_X: $\frac{x-\mu_X}{\sigma_X}$.

S3 Apply volume-based min-max normalization $\frac{x-min(X)}{max(X)-min(X)}$ to normalize the value range to $[0, 1]$ according to the volume data.

S4 Apply slice-based min-max normalization to shift the pixel values to a common scale $[\alpha, \beta]$ within the slice x: $\left[\frac{x-min(x)}{max(x)-min(x)}\right] \cdot (\beta - \alpha) + \alpha$. The intervals $[\alpha, \beta]$ are neural network dependent and for this work, we use the common intervals $[0, 1]$ and $[-1, 1]$. This is the last operations in the processing pipeline. Before that, other modifications such as data augmentation (Section 4.1.4) can be applied.

4.1.4 Data Augmentation at Training Time

For the training of selected neural networks, the data is randomly augmented during training time to avoid overfitting [19]. A random selection of the following methods is chosen at run time: affine transformation for translation, scaling and rotation, flip the image vertically around the x-axis, blur the image with a Gaussian, median, motion filter with random kernel size. All methods increase the data variability to better generalize the model to unseen data at test time. Augmented images are shown in Figure 4.5.

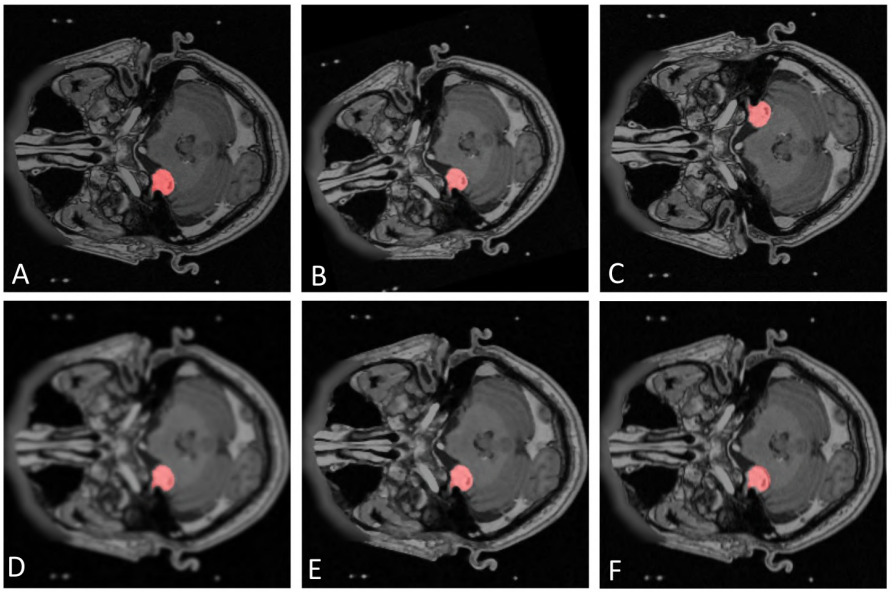


Figure 4.5: Examples for data augmentation during training time: (A) Original; (B) Affine transformation; (C) Vertical flipping; (D) Gaussian blur; (E) Median blur; (F) Motion blur. Best seen in high-resolution.

4.2 Background

In this section, we cover the error metric choice, the theory of selected network architectures, corresponding loss functions, and activation functions.

4.2.1 Error Metrics

A standard for image segmentation and object detection is the **Jaccard Index**, also known as **Intersection over Union (IoU)**, (Figure 4.6a)

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{4.1}$$

The region-based measurement compares the intersection to the union of two binary regions A and B and is easy to understand and implement. The notation $|.|$ is used to indicate the set size. A perfect segmentation has an IoU value of 1, while totally disjoint regions have an IoU value of 0.
Related to the IoU is the **Dice Similarity Coefficient (DSC)** (Figure 4.6b)

$$DSC = \frac{2|A \cap B|}{|A| \cup |B|}. \tag{4.2}$$

The ratio between IoU and DSC is

$$\frac{IoU}{DSC} = \frac{1}{2} + \frac{IoU}{2}. \tag{4.3}$$

which means they are always within a factor of 2 to each other:

$$DSC/2 \leq IoU \leq DSC. \tag{4.4}$$

While the formulation of IoU penalizes a mismatch between two regions even if there is a substantial strong overlap, DSC is less sensitive and more likely to estimate an averaged behavior.

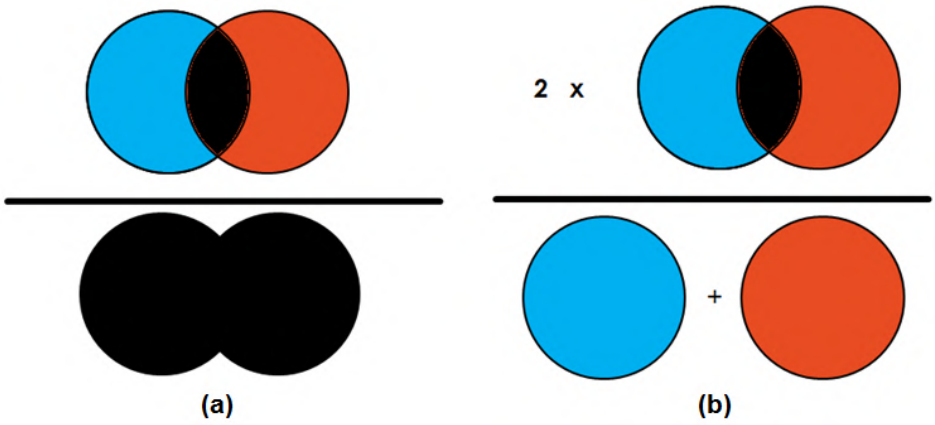


Figure 4.6: Illustration of IoU (a) and DSC (b) [121].

Another error measurement for segmentation performance is the **Average Symmetric Surface Distance (ASSD)**. ASSD calculates the average distance between two surfaces which in our case are the segmentation contours of ground truth and prediction. For each pixel belonging to the segmentation contour, the distance to the closest pixel of the other contour is calculated and all distances are averaged. If two segmentation masks are a perfect match, the distance is 0 mm. The worst case is that the ASSD is the maximal distance of an image.

In the context of image synthesis, we use different performance measurements because we are not comparing binary masks, but image objects X and Y with total pixel number n. **Mean Absolute Error (MAE)** calculates the arithmetic average of the absolute errors and **Mean Squared Error (MSE)** penalizes larger differences more by calculating the average of the errors squared:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|X_i - Y_i|, \tag{4.5}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2. \tag{4.6}$$

The detection of the tumor presence is a binary classification problem with the two possible outcomes present and absent. The predictions fall into one of the four categories: True positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions (see Figure 4.7). The **Accuracy (ACC)** is given based on the different outcome categories:

$$ACC = \frac{TP + TN}{P + N}. \tag{4.7}$$

The **True Positive Rate (TPR)**, also called sensitivity and recall, and the **True Negative Rate (TNR)**, also called specificity and selectivity, are analyzing the correct present and absent classification in more detail:

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{4.8}$$

$$TNR = \frac{TN}{TN + FP} = \frac{TN}{N}. \tag{4.9}$$

DSC and ASSD are easy to understand, widely known and often used in similar situations [3]. Thus, their ability to asses the prediction accuracy is used in our work for semantic segmentation evaluation. We report MSE for image comparison, to weight larger differences between two images stronger. An indication about how good the algorithm can detect tumor presence in image slices is given by ACC, TPR, and TNR.

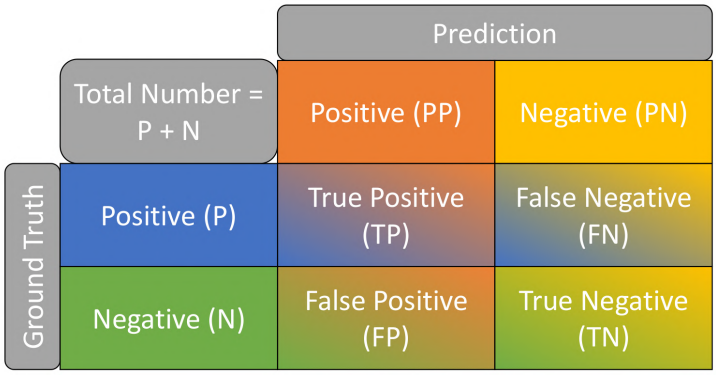


Figure 4.7: Confusion matrix for binary classification.

4.2.2 X-Net

X-Net has been developed by Bullock et al. [19]. It is designed to provide a deep neural network that can extract high-level features and fine-grained details without overfitting to a small dataset. Figure 4.8 shows the model architecture, which consists of two encoder-decoder modules. It can be interpreted as two UNets connected in series with a long-range skip connection. As inspired by *UNet* [108] or *SegNet* [15], the encoder is a series of convolutional layers and pooling for feature extraction and downsamling, respectively. The decoder uses convolutional layers and upsampling over multiple stages to produce a segmentation mask of same resolution as the input image. The original paper used the activation function ReLU, which is explained in Subsection 4.2.6. The downsampling is performed by maximum pooling and upsampling with nearest-neighbor interpolation. Skip connections are transferring information from downsampling to upsampling path, providing fine-grained details at later layers. The connections are realized by concatenating feature maps with the same dimensions.

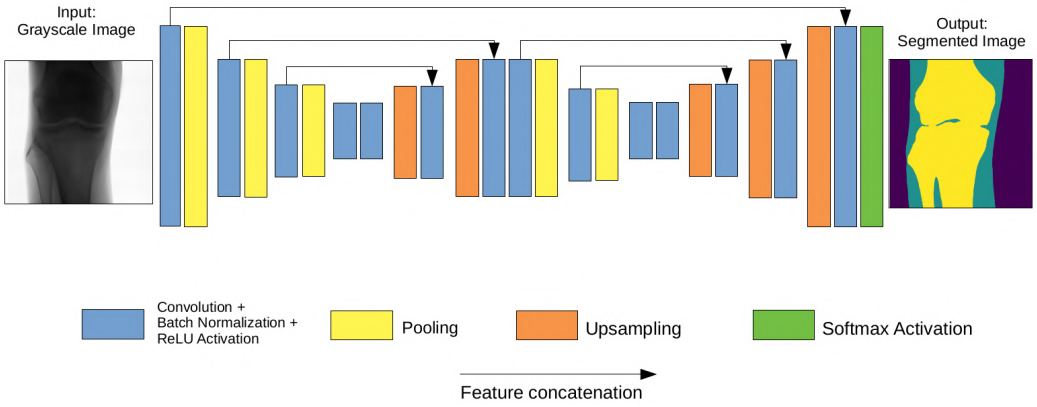


Figure 4.8: X-Net Architecture [19].

For semantic tumor segmentation the label pair consists of the ground truth A and predicted B segmentation masks. A common loss function, based on the DSC, is the **Dice loss**:

$$\mathcal{L}_{dice} = \frac{2|A \cap B| + \epsilon}{(|A| \cup |B|) + \epsilon}. \tag{4.10}$$

A factor ϵ is added to the fraction to stabilize the training by avoiding a vanishing denominator. Dice-related losses are a common choice for medical image segmentation due to their training robustness and rank stability [88]. $\mathcal{L}_{dice}$ should be low.

4.2.3 Classification-guided Module

The idea of a classification-guided module (CG module) is introduced by Huang et al. [54] for *UNet 3+* to avoid over-segmentation in medical image segmentation. *UNet 3+* is an extension of *UNet* and has an encoder-decoder structure. Figure 4.9 depicts the original classification-guided module used on different levels in the decoder part of the network. For the classification-guided module, a block of operations including dropout, convolution, pooling, and sigmoid activation is attached to different stages of the decoder. The resulting 2D tensor is processed to a single value $\{0, 1\}$ encoding the probability for presence (i.e. 1) and absence (i.e. 0) of an object (i.e. organ in the original work). The value is used as multiplier for the predicted segmentation mask. If the object is found absence by the classification branch, the segmentation mask is silenced, i.e., set to 0, and if the object is found present, the segmentation mask is kept as is.

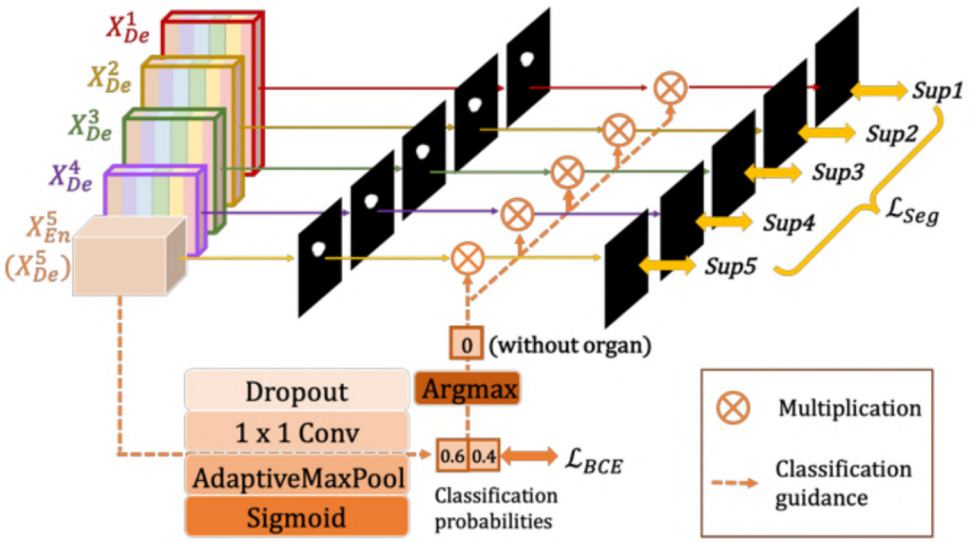


Figure 4.9: Classification-guided module in *UNet 3+* to avoid over-segmentation [148].

For training the classification-guided module, the **Binary Cross Entropy (BCE)** loss function [32] is used:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i}^{N} y_i \cdot log(p(y_i)) \tag{4.11}$$

with the targets y_i and the probability function $p(.)$. A high loss value means that the prediction deviates from the target. The total loss function is a combination of the segmentation (4.2) and classification (4.11) loss. The lower the loss, the better.

4.2.4 CycleGAN

The *CycleGAN* by Zhu et al. [149] uses the image-to-image translation idea of *GANs* [45] combined with a cycle-consistency loss that allows image synthesis for unpaired data. The unpaired data comes from two different domains/distributions S (source) and T (target). The framework consists of two generators G_{S2T}, G_{T2S} and two discriminators D_S, D_T (see Figure 4.10a). The goal is to learn a mapping $G_{S2T} : S \rightarrow T$ such that the distribution of $G_{S2T}(S)$ is similar to the distribution of T. To facilitate the training, another generator $G_{T2S} : T \rightarrow S$ is optimized to make the distribution of $G_{T2S}(T)$ similar to that of X. To make the problem mathematically bijective, i.e. $G_{S2T} = G_{T2S}^{-1}$, the constraint of cycle consistency is introduced: $G_{T2S}(G_{S2T}(S)) \approx S$ (Figure 4.10b) and $G_{S2T}(G_{T2S}(T)) \approx T$ (Figure 4.10c). This means, if we translate one domain to another and then run the reverse mapping, we should arrive back at our starting point. The cycle-consistent networks $G_{T2S} \circ G_{S2T} : S \rightarrow S$ and $G_{S2T} \circ G_{T2S} : T \rightarrow T$ can be interpreted as two autoencoders with an intermediate state in another domain.

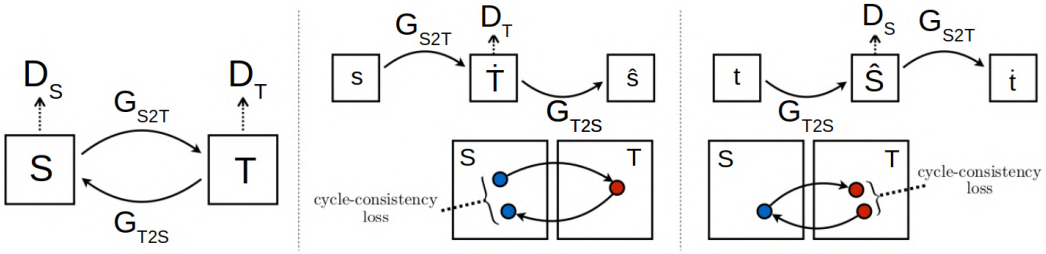


Figure 4.10: Illustration of *CycleGAN* architecture: (a) The model consists of two generator functions and two associated discriminators; (b) Forward cycle-consistency for generator G_{S2T}; (c) Backwards cycle-consistency for generator G_{T2S} [149].

The generator architecture consists of three components. First, an encoder part with convolutional blocks is extracting high level features and compressing the input image. Then, the feature vector is processed in the transform part by residual blocks [50]. Figure 4.11 illustrates a residual block, which ensures that the block input is available for later layers by adding a residue to the output node. This preserves the characteristics of the input and makes the changes for the output less abrupt. Finally, the decoder path creates the output with original input size from the feature vector. The patch-based

discriminator is a PatchGAN network [61]. Overlapping image patches of size 70×70 are classified as real or fake. The advantage of a patch-level discriminator over a full-image discriminator is the reduced number of parameters.

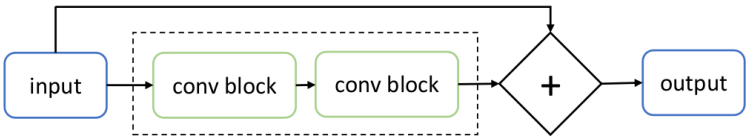


Figure 4.11: Illustration of a Residual block, which adds the input to the output node to preserve characteristics of the input.

For image synthesis with *CycleGAN*, several loss functions are used to guide the training for discriminator and generator. The loss functions are described for the generator $G_{S2T} : S \to T$ and discriminator $D_S : S \to \{0, 1\}$. The losses for generator $G_{T2S} : T \to S$ and discriminator $D_T : T \to \{0, 1\}$ work analogue.

- The discriminator D_S wants to distinguish between real $s \in S$ and fake $\hat{s} = G_{T2S}(t) \in S$ samples. The **discriminator loss** is the weighted MSE of real and fake sample:
$$\mathcal{L}_{discr} = ((D_S(s) - 1)^2 + (D_S(\hat{s}))^2 \cdot \frac{1}{2}. \tag{4.12}$$
- The generator G_{S2T} wants to fool the discriminator D_T, which means that the fake sample $\hat{t} = G_{S2T}(s) \in T$ produced by the generator should pass as real sample, i.e. discriminator should predict 1. Thus, the **adversarial loss** [45] is the MSE of discriminator prediction of the fake sample and the target 1:
$$\mathcal{L}_{adv} = (D_T(G_{S2T}(s)) - 1)^2. \tag{4.13}$$
- The **cycle consistency loss** for *CycleGANs* is introduced to ensure the bijectivity of the generators, i.e., after translating one domain to another and performing the reverse mapping, the starting point should be reached again. $s \in S$ and $t \in T$ are real samples.
$$\mathcal{L}_{cyc} = |s - G_{T2S}(G_{S2T}(s))| + |t - G_{S2T}(G_{T2S}(t))|. \tag{4.14}$$
- The **identity loss** measures how similar a real sample $s \in S$ and an identity sample $G_{T2S}(s) \in S$, generated by applying the "wrong" generator to the real sample, are:
$$\mathcal{L}_{id} = |s - G_{T2S}(s)|. \tag{4.15}$$
- The total **generator loss** is a weighted combination:
$$\mathcal{L}_{gen} = \lambda_1 \cdot \mathcal{L}_{cyc} + \lambda_2 \cdot \mathcal{L}_{id} + \mathcal{L}_{adv}, \tag{4.16}$$
with $\lambda_1 = 10$ and $\lambda_2 = 1$ to emphasize the importance of the cycle consistency loss. The identity loss is considered optional, i.e., $\lambda_2 = 0$ is possible.

$\mathcal{L}_{discr} = 0$ means that the discriminator can distinguish 100% correctly between real and fake samples. $\mathcal{L}_{gen} = 0$ means that the synthetic images of the generator always trick the discriminator ($\mathcal{L}_{adv} = 0$), the generator reproduces the same real sample ($\mathcal{L}_{cyc}$), and inserting a real target image results in the same real image ($\mathcal{L}_{id} = 0$). Since $\mathcal{L}_{discr}$ and $\mathcal{L}_{gen}$ are in competition with each other, we would like to see a balance between the two.

4.2.5 SIFA

Synergistic Image and Feature Adaptation (SIFA) by Chen et al. [24, 25] is a unified framework designed for unsupervised domain adaptation employing both image and feature alignment. Figure 4.12 illustrates the components of the SIFA framework and the two data flows for image and feature alignment. Both alignment strategies can benefit from each other. Image synthesis is exploited to narrow the domain shift by aligning the image appearance between source and target images. This is realized by the generator G_t transforming images from source to target domain and the combination of shared encoder E and the decoder U performing the reverse transformation. The encoder E is shared with the classifier C producing segmentation masks. Hence, the encoder is trained on both pixel-to-pixel transformation and semantic segmentation at the same time, permitting multi-task learning. The imbalance in medical image segmentation is addressed by using a hybrid loss of cross-entropy loss and Dice loss as segmentation loss for the supervised training of the segmentation branch. Since the data is unpaired, adversarial learning and the cycle consistency loss [149] are injected (see Section 4.3.2). There are three discriminators D_t, D_s, and D_p distinguishing their inputs between source and target based content. Two of the discriminators D_s and D_p enhance the feature adaptation. In contrast to a common strategy of inserting the discriminator directly in a high-dimension feature space, they operate in lower-dimensional spaces, i.e. semantic prediction and generated image space. The goal is to support domain-invariant feature extraction in the shared encoder, as there are two discriminators attached from different feature spaces. All components are trained every training step in an end-to-end manner with a sequential update: $G_t \rightarrow D_t \rightarrow E \rightarrow C \rightarrow U \rightarrow D_s \rightarrow D_p$.

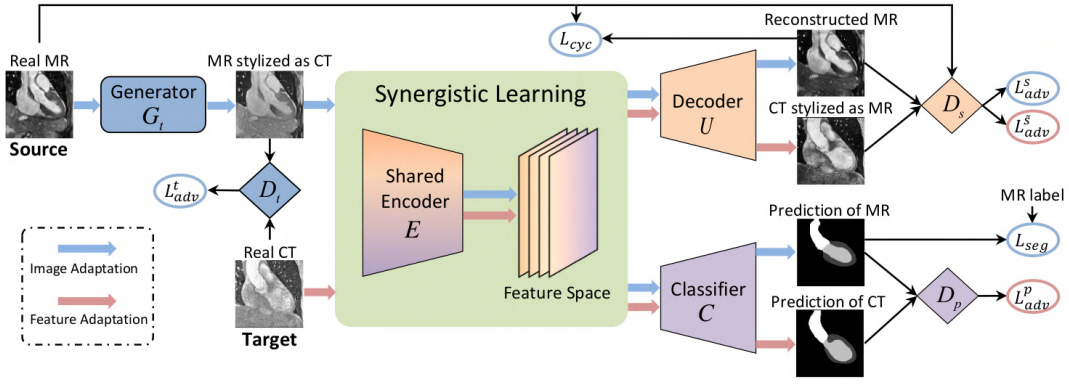


Figure 4.12: Illustration of SIFA [25].

4.2.6 Activation Functions

Activation functions are used after convolutional blocks to control their output. They introduce non-linearity to networks and ensure that the gradient values are in a certain range during back-propagation [101]. Table 4.2 and Figure 4.13 summarize the activation functions relevant for this work. Rectified Linear Unit **ReLU** [98], **Leaky ReLu** [90] and Scaled Exponential Linear Unit **SeLU** [74] are used in convolution blocks within neural networks. A small comparison is performed, to evaluate, if one is superior to others for DA. ReLU is one of the most used activation functions [31]. However, since negative values are set to zero, the effect of the dying ReLU problem can occur, i.e. large parts of the network have no contribution to the training process. Leaky ReLU is relaxing this definition and is a well-known alternative [90]. SeLU claims self-normalizing properties [74], which might be beneficial for the data shift problem. In the last layer of semantic segmentation networks, **logistic sigmoid** activation is normalizing the output to the interval $[0, 1]$.The last layers in *GAN* and *Cycle-GAN* have **hyperbolic tangent (tanh)** activation, which normalizes the output to the interval $[-1, 1]$.

Activation function	Formula
Rectified Linear Unit (ReLU)	$f(x) = max(x, 0) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$
Leaky ReLU	$f(x) = \begin{cases} \alpha x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$
Scaled exponential Linear Unit (SeLU)	$f(x) = \tau \begin{cases} \alpha e^x - \alpha & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
Hyperbolic tangent (tanh)	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Table 4.2: Overview of activation functions, with the function value x and parameters α and τ.

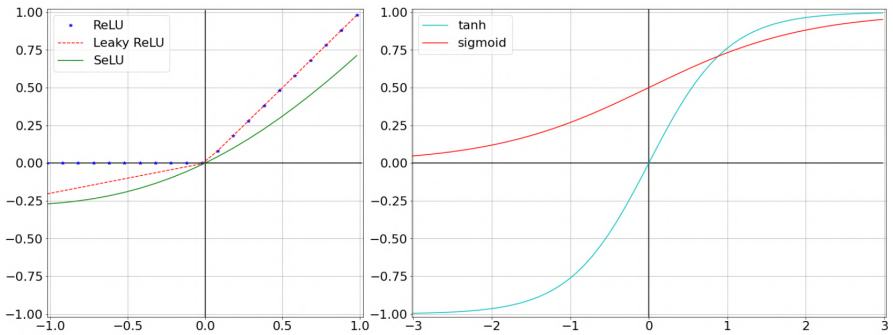


Figure 4.13: Activation functions (left) within a neural network for a range of $x \in [-1, 1]$ and (right) before the network output for a range of $x \in [-3, 3]$.

4.3 Automatic Segmentation Methods

This section is dedicated to the methods implemented for automatic VS segmentation on hrT2 images. As shown in Figure 4.14, the assumption is that annotated ceT1 images and raw hrT2 images are available for training. At inference time, the brain tumor mask should be predicted on hrT2 images. First, supervised methods are described to establish a baseline. They make use of the data pair of image and segmentation mask for both image modalities. Then, domain adaptation methods are introduced. Two methods are employing a *CycleGAN* for image alignment. Finally, we introduce a new method that performs domain adaptation with image and feature alignment exploiting the entire dataset. Theoretically, all methods can be applied to reversed source S and target T domains. However, since the goal is to predict on hrT2 scans, T2 is considered as target domain and is focused in the following section.

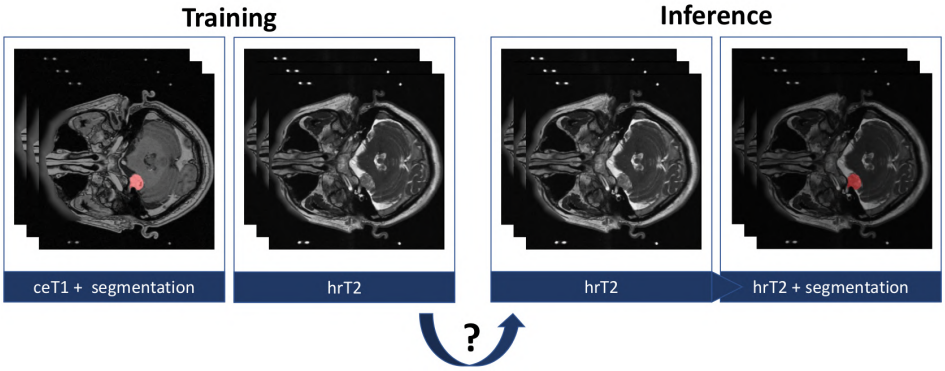


Figure 4.14: Data availability for DA: annotated ceT1 and not annotated hrT2 images for training. At inference, masks on hrT2 images are predicted.

4.3.1 Supervised Segmentation

Supervised segmentation networks are trained as baseline to evaluate the performance of later introduced unsupervised techniques and to investigate the performance differences between different activation functions. In the supervised training setting, the input images are available together with corresponding segmentation masks.

First, we considered the *UNet* architecture [108]. However, initial empirical tests have shown an unstable training. Advancement of *UNet*, such as *UNet++* or *UNet3+*, have a higher number of model parameters resulting in a higher GPU memory consumption. Thus, our baseline segmentation network is *X-Net* [19] (for theoretical background see Section 4.2.2). The network has been developed especially for small medical datasets and has a similar parameter number as a comparable *UNet*. After removing slices without tumor presence, only a couple of hundred remain. Thus, *X-Net* seemed particularly suited for our purposes. There are three different versions of the simple setting without further adaptation:

- **X-Net T1**: *X-Net* trained on ceT1 slices with brain tumor present (**D3**) in a supervised manner. Applying the pre-trained network to hrT2 images gives a lower bound on performance if the difference between the data modalities is not considered.
- **X-Net T2**: *X-Net* trained on hrT2 slices with brain tumor present (**D3**) in a supervised manner. This network is the upper performance bound when performing supervised training with available image and segmentation data pairs.
- **X-Net T1+T2**: *X-Net* trained on ceT1 and hrT2 slices with brain tumor present (**D3**) in a supervised manner. Instead of taking one input channel, this version has two input channels that are concatenated. As a result, both modalities are included in the training in a paired and supervised manner.

Another supervised method is the classification-guided segmentation, short **CG X-Net**. The original classification-guided module was added to a *UNet* extension to avoid over-segmentation (see Section 4.2.3). For our work, the classification-guided module complements the *X-Net* architecture. The training and validation dataset are extended with non-tumor slices to build a balanced dataset (**D2**). As shown in Figure 4.15, a classification branch is added at the second bridge. The operations are Dropout with rate 0.5, 1×1 convolutional layer, global max pooling and sigmoid activation. Then, values above 0.5 are mapped to 1, values below 0.5 to 0. The resulting value encodes the probability of tumor presence or absence and is used for multiplication with the predicted segmentation mask. The loss function is a combination of the Dice and BCE loss: $\mathcal{L}_{dice} + \mathcal{L}_{BCE}$ (see Figure 4.15). Empirical testing showed that a weighted loss results in unstable training. Thus, the sum of loss functions is used for optimization.

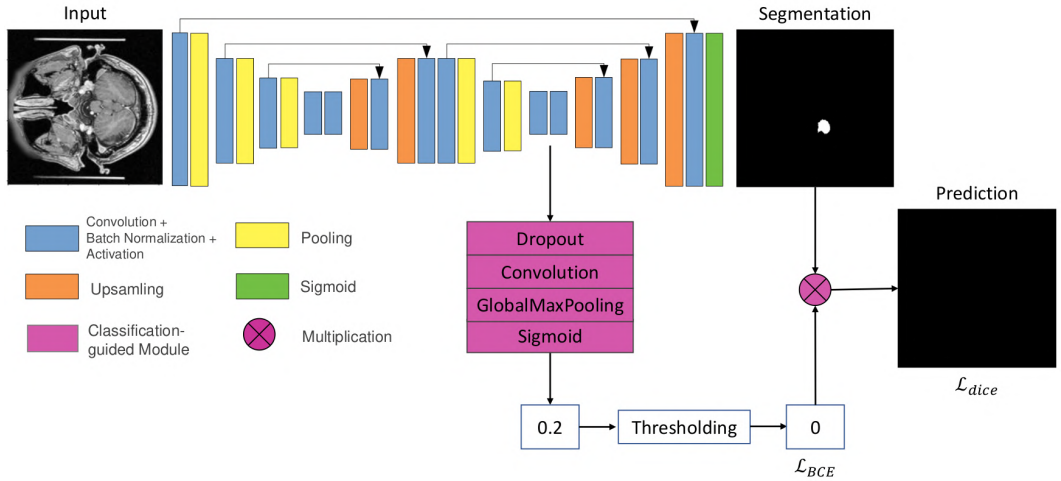


Figure 4.15: **CG X-Net** architecture with classification-guided module which predicts a probability of tumor presence to silence "empty" masks.

4.3.2 CycleGAN

The *CycleGAN* implementation follows the original paper [149]. The generator is a ResNet-based network and the discriminator follows the PatchGAN architecture. Only slices with brain tumor are used for training (**D3**), because they are images with the most relevant information for tumor segmentation while the training time is reduced by not using the entire dataset. To stabilize the training of the discriminator, an image pool of fake patterns is collected during training, from which an example is then drawn for error computation. The size of this fake image pool is set to 50 samples. They are randomly replaced by new generated fake samples throughout the training process. In one training epoch, each training data sample, i.e. unpaired T1 and T2 image slices, is drawn and processed exactly once like shown in Figure 4.16. For each data sample the following steps are performed:

- Train the generators $G_{S2T} : S \to T$, $G_{T2S} : T \to S$ where S (source) is the domain of ceT1 images and T (target) of hrT2 images. The generator loss in (4.16) is used.
- Either fill the image pool with the newly generated fake samples or replace random samples in the already filled pool. Then, draw fake samples from the updated fake image pool for the discriminator training.
- Train the discriminators $D_S : S \to \{0, 1\}$ and $D_T : T \to \{0, 1\}$ with the discriminator loss in (4.12). Depending on the specific training settings, this update step for the discriminators is not executed for every training sample.

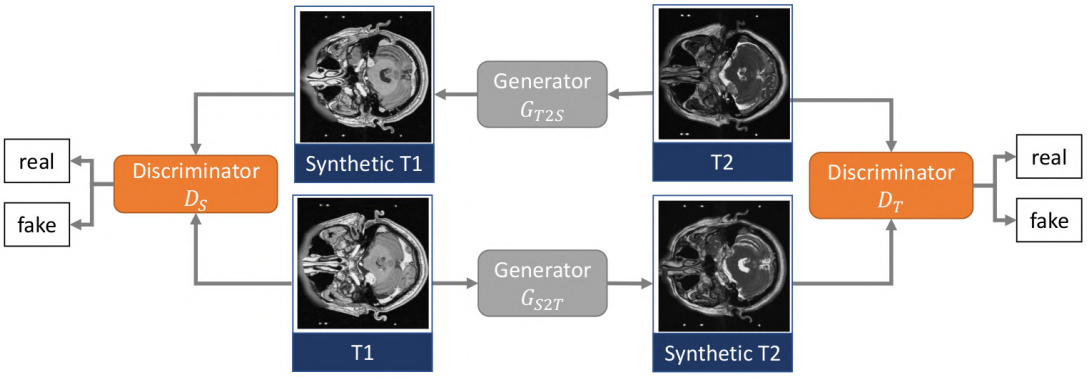
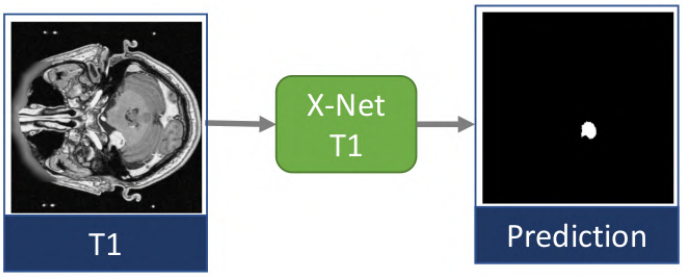


Figure 4.16: CycleGAN training with ceT1 (S) and hrT2 (T) images. The main components are two generators, G_{S2T} and G_{T2S}, and two discriminators, D_S and D_T, which are trained depending on each other.

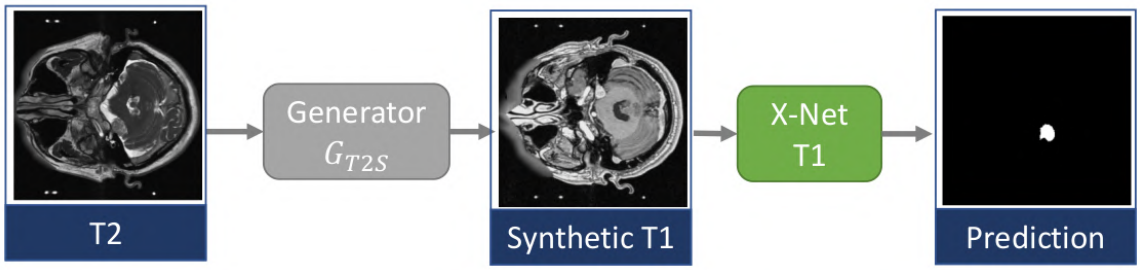
T1 images show better defined tumor boundaries and have a better contrast and resolution compared to T2 images. Thus, we expect that the transformation from T1 to T2 works better than the opposite case.

4.3.3 G_{T2S}+SegmT1

The first approach that employs the generator G_{T2S} from a pre-trained CycleGAN is **G_{T2S}+SegmT1**. Both networks, a *CycleGAN* and a segmentation network, are trained separately. The workflow at inference time is illustrated in Figure 4.17b. The generator translates the target T2 images into synthetic T1 images. Then, they are fed into a **SegmT1**, which was trained on real T1 image and segmentation pairs in a supervised manner (see Figure 4.17a). Finally, the segmentation mask predicted on synthetic T1 images is used as target segmentation mask. This pipeline has not only two stages at training time, but also at inference which makes the framework more complex.
Since the training of both networks does not depend on each other, they can be trained in parallel if enough hardware resources are available. However, there is no end-to-end training possible. We expect that the framework works well on images that preserve the tumor boundary from T2 to T1 transformation. However, due to the expectation that generating synthetic T1 images from T2 images is challenging, synthetic T1 images may not resemble the real T1 images in the quality that the segmentation network is used to. The real T1 images have a more distinct tumor boundary, which are used to predict the tumor delineation. For images where this clear boundary cannot be reconstructed, the segmentation network will most likely fail.



(a) Training of segmentation.

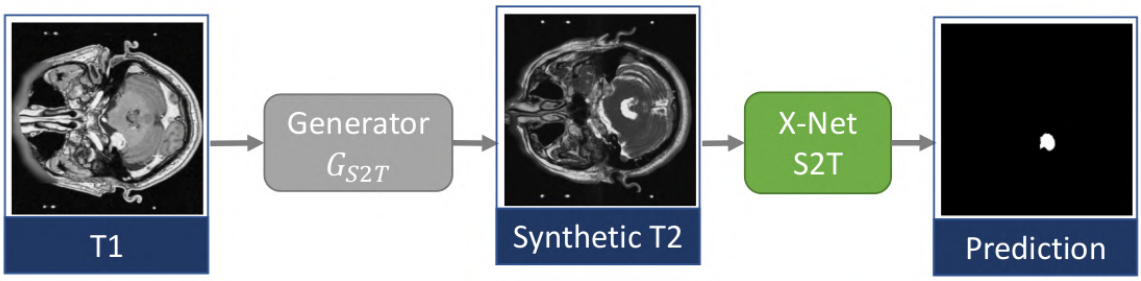


(b) Inference.

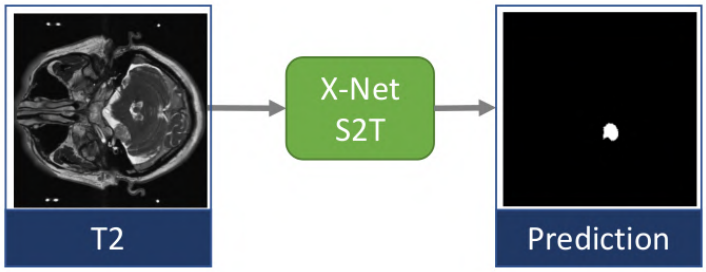
Figure 4.17: **G_{T2S}+SegmT1**: (a) Segmentation network is supervised SegmT1; (b) Inference consists of two steps: 1) Generate synthetic T1 images by applying the generator G_{T2S} to the T2 image. 2) Predict the segmentation mask with a specialized SegmT1. *X-Net T1* is employed as an example for SegmT1.

4.3.4 SegmS2T

The pipeline **SegmS2T** employs the trained generator G_{S2T} for image synthesis during semantic segmentation training. Figure 4.18 shows the data flow at training and inference time. T1 slices with corresponding masks are taken as input (**D3**) which means that neither real T2 images nor related masks are necessary. Then, the T1 images are translated to T2 images before the network predictions are optimized (see Figure 4.18a). Assuming that the transformation preserves the brain tumor, the segmentation masks can be transferred and used as ground truth. Therefore, the training data pair consists of synthetic T2 images with transferred segmentation masks from the original T1 images. Compared to $\mathbf{G_{T2S}}$**+SegmT1**, predictions can be performed directly on real T2 images at inference time (see Figure 4.18b). This reduces the inference to a single step.
We expect that the synthetic T2 images preserve the tumor structures better than synthetic T1 images. The potentially lower quality of the tumor boundary compared to real T2 images is supposed to aid the training. It can be interpreted as integrated data augmentation. If the network is capable of extracting the tumor information for low quality T2 images, it should also be able to handle real T2 images. Thus, we expect that the approach produces better results.



(a) Training of segmentation.



(b) Inference.

Figure 4.18: **SegmS2T**: (a) Segmentation network training by using synthetic T2 and transferred segmentation masks. (b) For inference, the trained **SegmS2T** predicts on real T2 images. *X-Net T2* is employed as an example for the segmentation model.

4.3.5 CG SIFA

The original *SIFA* framework [25] addresses the class imbalance in medical imaging with a hybrid loss of cross-entropy and Dice loss. In our approach, we integrate the classification-guided module from *UNet3+* [54] into the segmentation branch of *SIFA*. Figure 4.19 shows the resulting **CG SIFA** framework, which is short for "Classification-Guided Synergistic Image and Feature Adaptation" framework. The main components are taken from the original *SIFA* framework (see Section 4.2.5): the generator G_{S2T}, the discriminator D_T, the shared encoder E, the decoder U and the two discriminators D_S and D_{Segm}. The segmentation classifier is exchanged for a classification-guided segmentation decoder called *CG Segm*. The classification-guided module allows a dataset of non-tumor samples to be used in a more guided approach. Multiple deconvolutional layers and a classification branch are combined in the last layer by multiplication of segmentation mask and classification prediction. This leads to a silenced segmentation mask when tumor absence is classified. For the cases where the silencing is correct, i.e. the ground truth is empty, the Dice Loss $\mathcal{L}_{dice}$ is zero and is not contributing to the total loss. The total loss is: $\mathcal{L}_{dice} + \mathcal{L}_{bce}$. However, the segmentation decoder is trained together with the shared encoder E. The loss for the generator G_{S2T} and $U \circ E$ follows the CycleGAN loss principle. Same applies to the discriminator losses. At inference time, real T2 images are inserted into the encoder to generate the representation in feature space. Then, the representation is taken by the segmentation network to predict the presence classification and the segmentation mask.
SIFA is a hybrid approach using not only image, but also feature alignment. According to the literature [25], it should be more successful than only image alignment. Improved by the CG module, **CG SIFA** should perform even better on the entire dataset.

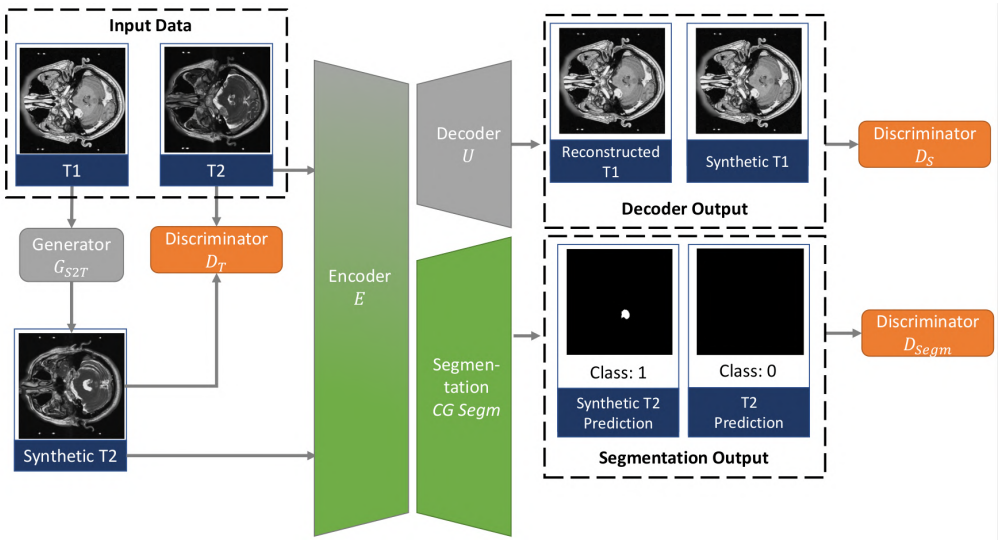


Figure 4.19: **CG SIFA** framework with classification-guided segmentation decoder in addition to the image synthesis workflow.

4.4 Implementation

In this section, we report the implementation and training settings of the methods. Detailed illustrations of the network architectures are given in Appendix A.1. All methods are implemented with Tensorflow and Keras [9]. Each model has been trained with the settings described below on an Nvidia RTX 3080 Laptop GPU with 8 GB memory. The input image size for all networks is $256 \times 256 \times 1$ (HxWxC).

4.4.1 X-Net & CG X-Net

In the original *X-Net* implementation, ReLU activation is used. Thus, our basic **X-Net** version uses ReLU activation and He Normal [49] initialization for the convolutional weights. Additionally, we also test the performance of Leaky ReLU with He Normal initialization and SeLU with LeCun Normal initialization [74]. In comparison to the original filter depth of $[64, 128, 256, 512]$, the filter depth are reduced to $[32, 64, 128, 256]$ to minimize the required GPU memory. A batch size of 4 is chosen after first test runs. Adam optimizer [72] with learning rate 0.001 and Dice loss function are used for training. The learning rate is reduced by a factor of 0.5 to a minimum of 0.0001 if the loss on the validation dataset is not improving for 10 epochs. In total, the training is scheduled for 150 epochs with augmentation switched on at epoch 100. The first epochs can be interpreted as a pre-training of the network, whereas in the last third, additional variation in the dataset is introduced to reduce overfitting. In addition, early stopping is applied, which means that the training is stopped if the validation loss is not improving for 20 epochs which reduces the epoch number and training time if the network is already converging. The training schedule was fixed after testing different settings and observing the loss behavior to verify the convergence of the model. For **CG X-Net**, the same implementation parameters and training settings are applied, except a bigger batch size of 8 is chosen to stabilize the training.

4.4.2 CycleGAN

Our *CycleGAN* implementation follows the original [149]. The ResNet-based generator consists of two down-sampling blocks with filter depth $[64, 128, 256]$, followed by nine residual blocks that keep the filter size the same. The up-sampling is done by a transposed convolutional layer with the reverse filter sizes. After a final convolutional layer, the activation function tanh is applied bringing the image pixel range into $[-1, 1]$. Like the original *CycleGAN* implementation, we use instance normalization [124]. The PatchGAN discriminator is a convolutional network looking at 70×70 image patches to define if they are real or fake. After an initial convolutional layer, Leaky ReLU and instance normalization are integrated in three down-sampling convolutional blocks. The final convolution layer outputs a 32×32 tensors. The generators and the discriminators use one Adam optimizer [72] each, with initial learning rate 0.002. The total number of epochs is 100. The learning rate scheduler starts a linear decay at epoch 50.

To produce natural appearing synthetic images, the generator and the discriminator need to be balanced. When the discriminator gets too strong, the generator training suffers and vice versa. We observed that the discriminator improves very quickly, since the generator produces unrealistic images at the beginning. Thus, skipping the discriminator update, allows the generator "to catch up". This behavior is controlled with the training parameter `dstep`. The options `dstep=1,5,10` have been tested. This means that the discriminators are only trained at each step, at every 5th and at every 10th step in one training epoch. The generated fake samples are stored for all steps in the fake image pool and are therefore not lost for training.

4.4.3 G_{T2S}+SegmT1 & SegmS2T

Both approaches can use either the standard *X-Net* or the classification-guided enhanced version. Depending on the segmentation model, they are referred to as **G_{T2S}+X-Net T1** and **X-Net S2T** or **G_{T2S}+CG X-Net T1** and **CG X-Net S2T**. The model and training settings for **G_{T2S}+SegmT1** and **SegmS2T** are the same as described in Section 4.4.1 for the *X-Net* applications and in Section 4.4.2 for the *CycleGAN* implementation. The only difference is that due to a less stable validation loss, early stopping is not applied for the segmentation network in **SegmS2T**.

4.4.4 CG SIFA

The implementation of the **CG SIFA** framework is a combination of *CycleGAN*, the original *SIFA* and the CG module within the segmentation branch. The individual components were trimmed down to fit the framework into 8 GB of GPU memory. A lot of different combinations were tested for a limited numbers of epochs (< 20 epochs). The description in this paragraph is from the latest implementation with which more epochs have been trained. As generator G_{S2T}, a ResNet-based generator is used. In the end, the constellation of three instead of two down-sampling blocks with filter depths $[32, 64, 128, 256]$, followed by four instead of nine residual blocks with filter depth 256 showed satisfying reconstruction results. With the modifications, we managed to cut the number of trainable parameters into half (from over $11,000$ to roughly $5,500$ trainable parameters). The discriminators in the framework follow our CycleGAN implementation. As for the *CycleGAN* implementation, we introduce the parameter `dstep` to regulate how often the discriminators are trained in a training epoch. The shared encoder is the largest component. By removing the large layers with filter depths 256 and 512 that occur deep in the network, the encoder size is reduced so that the memory is sufficient. Several residual convolution blocks with filter depths $[16, 32, 64, 128, 256]$ remain in the resulting network. For a more detailed description, see the network architecture visualization in Appendix A.1. The number of trainable parameters is reduced from over $27,000$ to under $7,000$. Three residual convolutional blocks with filter depth 128 and three deconvolutional layers with filter depths $[64, 64, 32]$ build the decoder branch. The segmentation branch is not only extended by the CG module as described in 4.3.1, but also with additional deconvolutional layers with the same filter depths as for the decoder. The original *SIFA*

segmentation branch implementation consists of only a single convolution layer and the result is then resized. First empirical testing without the CG module showed that the results are not satisfactory enough and, therefore, we changed the architecture. In this way, we aim to mimic the *X-Net* strategy of down-scaling with convolutions and up-sampling with deconvolutions. If the CG module is skipped in the segmentation network, we refer to the framework as our **SIFA** method from now on, although it is not exactly the original implementation. For the optimizer and learning rate definition, we have orientated ourselves on the *SIFA* implementation. An Adam optimizer [72] with initial learning rate of 0.002 is used. Due to the training strategy discussed later on, there are four optimizer in the framework, one for generator and segmentation each and two for discriminators. The discriminators for the reconstruction share an optimizer while the segmentation discriminator has its own. Due to the limited resources, the batch size is restricted to 1. A larger batch size would be preferable to stabilize the training but also requires more GPU memory. In some training runs, linear decay was tested after 1/2 or 3/4 of the total number of epochs, but showed no advantages. According to Chen et al. [24], an optimizer with constant learning rate is more stable for training. For this reasons, linear learning rate decay was not used later on.
Compared to the original *SIFA*, the training strategy is modified. First, the framework is trained like a *CycleGAN*, i.e. no segmentation branch update, for an initial period to teach the encoder an initial feature extraction. After $5, 10, 15$ or 20 epochs, the segmentation updates are turned on. Then, the encoder is only updated in combination with the segmentation branch. Although the loss function includes the generator loss $\mathcal{L}_{gen}$, the λ parameters are set to 1 and the segmentation loss multiplier is 10. This means that although the reconstruction is still included in the loss function, the influence is overshadowed by the segmentation training. The total epoch number varies between 50 and 100. The training settings and strategy were developed iteratively. A setup was first tested for a few epochs (< 20) for the **SIFA** implementation. When training progress was visible, improvements were tried out. Otherwise, larger modifications such as architectural changes were made. Only then was the CG module added and the same settings used for training **CG SIFA**.

CHAPTER 5

Visual Assessment

To analyze the segmentation algorithms from the previous section, usually quantitative error metrics, such as Dice Similarity Coefficient and Jaccard Distance (see Section 4.2.1), are employed. This means that the complex model behavior is reduced to a single value, which gives a good indication about the overall performance, especially if a big number of models is comparatively evaluated. In order to understand the model behavior better, obtaining additional information through detailed visual inspection is necessary. In this chapter, we describe an interactive, web-based tool to support the visual assessment as part of the model evaluation and to visualize segmentation outcomes.

5.1 Data Preparation

The underlying dataset for the visualization is the test set, which is a subset of the entire dataset (see Section 4.1.1). Only the target domain with T2 images is used. There are three different levels of detail in which the dataset can be viewed. First, the **entire dataset** is considered and one value per feature is derived. Second, the dataset is divided into **patient samples** and one value per patient ID is calculated. Third, for each patient sample, the volumetric data scan is split into the **individual 2D scan slices**. Following the mantra by Keim *"Analyze first, Show the Important, Zoom, filter and analyze further, Detail on demand"* [70] which is an extension of Shneidermann's mantra for Visual Analytics applications, all three scales are covered with our visualization tool. In this section, the generation of the model prediction, the feature types and extraction are described.

5.1.1 Segmentation Predictions

The segmentation algorithms described in Chapter 4 are all applied to T2 images of the test set (**D1**) — independent of tumor presence or absence. The inference pipeline depends

on the segmentation algorithm and is described in more detail in Section 4.3. After retrieving the predicted segmentation mask, each segmentation mask is post-processed by thresholding with the threshold value 0.5. Pixel values below 0.5 are mapped to 0, values above are mapped to 1. This results in a binary segmentation mask with 0 for the background and 1 for the tumor delineation, as shown in Figure 5.1.

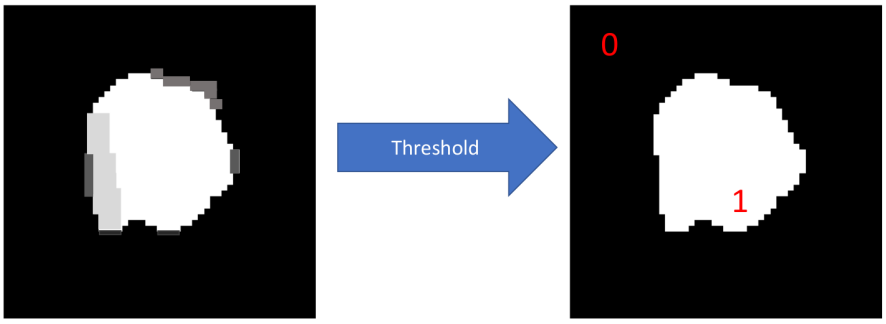


Figure 5.1: Binary segmentation mask with 0 for background and 1 for tumor delineation generated by thresholding with value 0.5.

5.1.2 Feature Extraction

We hypothesize that the outcomes of the models relate to features in our data. Aside from the predicted segmentation masks, we extract different 2D and 3D features from the dataset. This includes performance measures, e.g., DSC, ASSD, and ACC, radiomic features [152] and task-specific dataset characteristics, e.g., number of tumor slices. The following three feature categories are considered.

Performance Measures

DSC (4.2) and ASSD, which are very common error metrics for segmentation predictions, are supplemented by ACC (4.7), TPR (4.8), and TNR (4.9) for the tumor presence detection. The quantitative results are calculated for the entire dataset (`all`) and the subset only containing slices with tumor presence (`only tumor`). The scores are calculated for different levels of detail. First, the performance is determined per slice. Then, the results for each stack of slices is averaged per patient. Finally, the values are averaged over the entire dataset to retrieve a single value per segmentation method. In addition to the numerical values, the performance can also be divided into three classes: good, medium, bad. The scores for all selected segmentation algorithms are averaged per patient ID before translating them to categories.

Radiomic features

The Image Biomarker Standardization Initiative (IBSI) provides a standardized definition for features that quantify characteristics in medical imaging [152]. These hand-crafted

feature are called radiomic features and are calculated for a region of interest (ROI). The ROI is limited to the image region within a given segmentation mask, i.e., defining a tumor or other structures of interest. The radiomic features include first-order (distribution of gray values), second-order (relationship between voxels/pixels), and higher-order texture features (relationships between more than three voxels/pixels), and morphological features (size and shape). The features can be both 2D and 3D. Based on the significance for the models and our dataset, we have selected a subset of first-order texture and shape features, which is described below. Some feature values show only a small variance for different patient samples or coincide into one category when creating the signature. The total list of features also contains related and dependent features. Thus, not all features from one category have been selected.

The definitions for all features used in our work follows Zwanenburg et al. [152]. The following notations are used to define the features:

- X set of N voxels/pixels in ROI
- $\overline{X}$ mean value of X
- $P(k)$ first-order histogram with N_b non-zero bins
- $p(k)$ normalized first-order histogram, i.e., $P(k)/N_b$
- μ_3 is the 3rd central moment with $\mu_3 = \frac{1}{N}\sum_{k=1}^{N}(X(k) - \overline{X})^3$
- μ_4 is the 4th central moment with $\mu_4 = \frac{1}{N}\sum_{k=1}^{N}(X(k) - \overline{X})^4$
- X set of N voxels/pixels in ROI
- N_f number of faces/triangles (3D) or lines (2D) in the mesh
- V mesh volume in mm^3
- A mesh surface area in mm^2
- a_i, b_i, c_i vertices in the mesh

First-order texture features describe statistics of gray value intensities within the image region defined by the segmentation mask.

- **Energy** $= \sum_{k=1}^{N} X(k)$ measures the magnitude of the gray values of the ROI.
- **Variance** $= \frac{1}{N}\sum_{k=1}^{N}(X(k) - \bar{X})^2$ is the mean of the squared distance of gray values from the mean value. It is an indication for how the distribution is spread around the mean.

- **Skewness** = $\frac{\mu_3}{\sigma^3}$ measures how asymmetric the distribution of gray values is around the mean value. The values can be positive (right tail is longer) or negative (left tail is longer) and are divided into two classes with 0 as partition value.
- **Kurtosis** = $\frac{\mu_4}{\sigma^4}$ measures the "tailedness" of gray value distribution. A univariate normal distribution has a Kurtosis value of 3. A value less than 3 is platykurtic (less extreme outliers than normal distribution), a value more than 3 is leptokurtic (more extreme outliers than normal distribution). Thus, the values are split into two classes with 3 as limit.
- **Range** = $max(X) - min(X)$ defines the range of gray values in the ROI.
- **MAD** (Mean Absolute Deviation) = $\frac{1}{N}\sum_{k=1}^{N}|X(k) - \bar{X}|$ is the mean of the absolute differences between gray values and mean value.

Morphological features describing the shape of the tumor are based on the mask and do not consider the gray value intensity in the ROI. The mask is transformed to a mesh-based representation with triangles for volume and surface measurements [87]. We refer to the IBSI manuscript [152] for more details about the mesh generation and mesh volume calculations. Depending on the underlying data structure, the features are calculated for volumetric 3D data and/or slice-wise 2D data. The 2D features are:

- **Mesh Area** = $\sum_{k=1}^{N_f} A_k$, with $A_k = \frac{|a_k \times b_k|}{2}$, is the sum over the surface areas of the triangle edges.
- **Perimeter** = $\sum_{k=1}^{N_f} P_i$ with $P_i = \sqrt{(a_i - b_i)^2}$ is the sum of all sub-area perimeters.
- **Sphericity** = $\frac{\sqrt{4\pi A}}{P}$ describes how circle-like the slice-based ROI is. The values are in the range of $[0, 1]$ with 1 for a perfect circle. With the partition value 0.5, the slice-based sphericity values are assigned to two classes.
- **Max2DDiam** is the largest euclidean distance between pairwise vertices on the ROI surface mesh for the 2D ROI.
- **Elongation** = $\sqrt{\frac{\lambda_{minor}}{\lambda_{major}}}$ is the relationship between the two most prominent axis lengths of the ROI-enclosing ellipsoid, i.e., the major axis length λ_{major} and minor axis length λ_{minor}. The values are in the range of $[0, 1]$ with 1 representing non-elongated (sphere/circle-like) and 0 maximal elongated. Two classes are divided by the threshold value 0.5.

The 3D features are:

- **Mesh Volume** $= \sum_{k=1}^{N_f} V_k$, with $V_k = \frac{a_k \cdot (b_k \times c_k)}{6}$, is the sum of all volumes of a tetrahedron defined by the vertex points a_k, b_k, c_k of a face k.

- **Surface Area** $= \sum_{k=1}^{N_f} A_k$, with $A_k = \frac{|a_k b_k \times a_k c_k|}{2}$, is the sum over the surface areas of the triangle faces.

- **Sphericity** $= \frac{(36\pi V^2)^{(1/3)}}{A}$ describes how sphere-like the volumetric ROI is. The values are in the range of $[0, 1]$ with 1 for a perfect sphere. With the partition value 0.5, the volumetric sphericity values are assigned to two classes.

- **Max3DDiam** is the largest euclidean distance between pairwise vertices on the ROI surface mesh for the 3D ROI.

- **Elongation** $= \sqrt{\frac{\lambda_{minor}}{\lambda_{major}}}$ is the relationship between the two most prominent axis lengths of the ROI-enclosing ellipsoid, analogous to the 2D version of the feature.

- **Flatness** $= \sqrt{\frac{\lambda_{least}}{\lambda_{major}}}$ is the relationship between the maximal and minimal axis lengths of the ROI-enclosing ellipsoid, i.e., the major axis length λ_{major} and least axis length λ_{least}. The values are in the range of $[0, 1]$ with 1 representing non-flat (sphere-like) and 0 totally flat. Two classes are divided by the threshold value 0.5.

Task-Specific Characteristics

Aside from image data features, we also extract task-specific dataset characteristics. The total number of slices and the number of slices with tumor presence are reported for each patient sample in the dataset. Although, we do not expect a big impact, we want to rule out possible hidden connections and gain an overview of our dataset. In addition, the tumor presence and size derived from the ground truth delineation are extracted per slice. These two features give insight into the composition of a 3D dataset and we expect a direct link to the model performance.

5.2 Visualization Techniques

This section is intended to provide an overview of the chosen visualization techniques and their underlying reasoning. We present the tasks in which our target users are interested. Then, we will walk through our strategy and describe each component.

5.2.1 Overview

The main target users of our application are AI/ML engineers who develop segmentation algorithms. To show how close the predictions of their models come to the quality of human delineations, they need to validate their models against ground truth (GT). In addition to performance assurance, the evaluation results are also used to compare different models. A detailed assessment drives an iterative development process, where the weak points are revealed and can be eliminated step by step.

To this end, we define five tasks based on the experience we gained from the algorithm development in the previous chapter:

T1 Overall performance comparison, i.e., for all models and all patients

T2 Per-patient performance comparison, i.e., all models for one patient

T3 Per-slice performance comparison, i.e., all models for a specific slice

T4 Relationship to features, i.e., correlation of performance with the dataset- and image-derived features discussed in Section 5.1

T5 Anatomy-based predictions, i.e., link to anatomical space

Our design strategy for a visualization tool to cover all five tasks follows Keim's mantra *"Analyze first, Show the Important, Zoom, filter and analyze further, Detail on demand"* [70]. This is an adaptation of Shneidermann's mantra *"Overview first, zoom and filter, then details on demand"* [115] for VA. We need a visual representation that can show the performance on three different levels of detail, i.e., averaged over the entire dataset (**T1**), per patient sample (**T2**) and per 2D scan (**T3**), the correlation of performance to features (**T4**), and the prediction on image scans for multiple algorithms (**T5**) in a comparative manner. To fulfill the VA mantra, we need to integrate interaction. Starting from the overall performance comparison, the user needs the possibility to select specific patient or slice IDs based on the next higher data scale to browse the different levels of the dataset. Brushing specific subgroups and applying this selection as filter to the performance comparison can create a link between performance comparison and feature relationships. In addition, the visual representations must be configurable. This means that the user must be able to control some general settings, such as model and feature selection. Figure 5.2 summarizes the required tasks **T1**-**T5** and indicates the interaction directions.

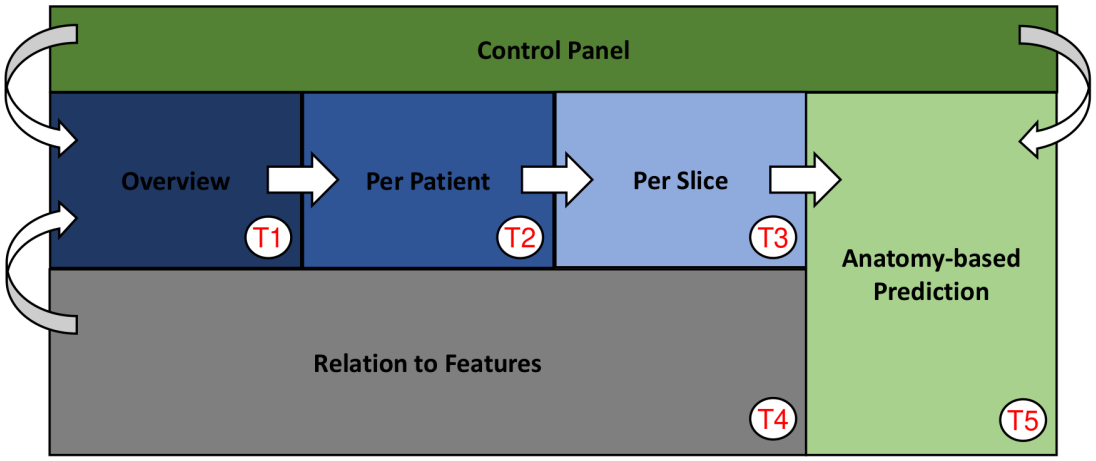


Figure 5.2: Illustration of the tasks **T1-T5** and interaction directions.

5.2.2 Performance Heatmaps (T1-T3)

The performance measures are calculated for the results of multiple segmentation algorithms applied to several data samples, i.e., either patient data samples (**T2**) or image slices (**T3**). In addition, a performance summary of the entire data set (**T1**) provides an overview. To satisfy all three levels of detail, the comparison type needs to be flexible, depending on what the user is focusing on.

The data can be interpreted as matrix with two dimensions. The human processing of numbers is slower than of colors. Therefore, the matrix is visualized by color coding the numerical values, as illustrated in Figure 5.3. The data samples are the matrix rows, the performance results the columns. This visualization technique is called heatmap visualization. Thus, we call this component **performance heatmap**. The color-encoding is based on a sequential color map. Sequential color maps are used for data ranging from low to high values and they imply order [47]. Frequently used sequential color maps are Plasma and Viridis. We have chosen the Plasma color map since this aligns better with the other components of the application. In addition to the actual matrix visualization, the average performance per column is displayed in the first row from the top. The colored matrix cells are annotated with the numerical values in the annotated heatmap.

Depending on the focus of the user, different comparisons are possible. The performance heatmaps allow a **1-by-1** or **1-by-n comparison**. A single patient or slice data (matrix row) or a single model (matrix column) is compared with others. Using Figure 5.3 as an example, different comparison options are discussed. The user can single out the blue cell in the middle of the yellow cells (slice ID 24, model CG_SegmS2T_GAN1_relu) and compare this cell with other elements of the row or column in which it is in, i.e. perform a 1-by-1 comparison. Another option is to look at the entire column of the CG_SegmS2T_GAN1_relu model and compare the results to a subset or all models. The same can be done for the row with slice ID 24. This is a 1-by-n comparison.

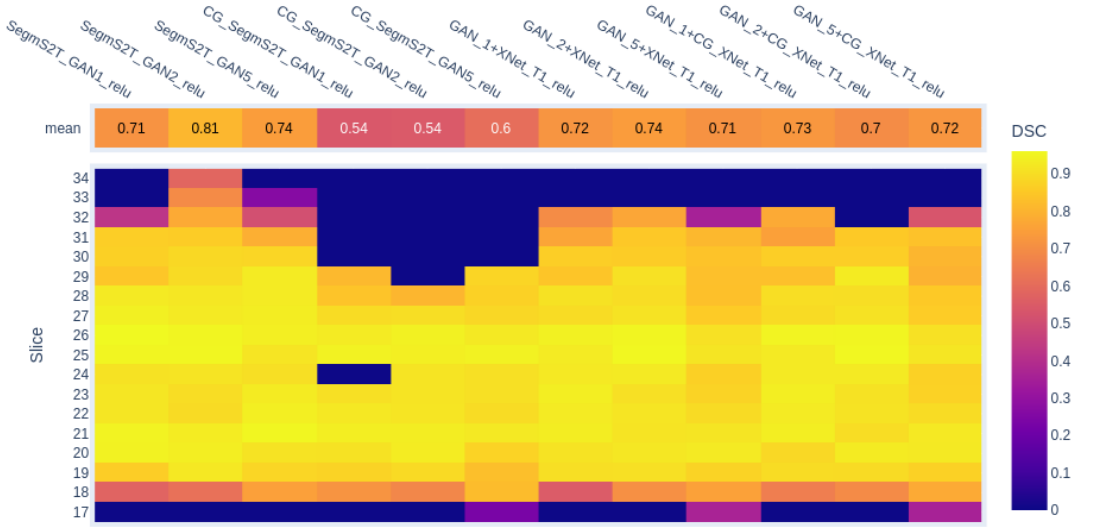


Figure 5.3: Performance heatmap: The DSC performance results of multiple DA segmentation algorithms on multiple data samples are color-encoded with the Plasma color map. The data is taken from patient ID 233.

5.2.3 Parallel Set Diagrams (T4)

Combining all features from Section 5.1.2, i.e., x features from performance measures, y from radiomics, and z from task-specific characteristics, results in a vector per data sample with dimensionality $x + y + z = n$. The exact number of features depends on the feature selection and the data sample. The data sample is either the volumetric scan per patient ID or the 2D slice. Consequently, the space of all features is n-dimensional.

The n-dimensional spaces with categorical data can be reduced to two or three dimensions by using high-dimensional clustering, such as t-SNE (t-distributed stochastic neighbor embedding) [127]. The low-dimensional representation shows the clusters per signature, but not the composition of the signature itself. Since we are also interested in the individual categories that make up the signature, t-SNE is not suitable.

Given the actual numerical values, we assign the numbers of each feature to categories. The mapping provides a categorical meaning, e.g., high DSC values correspond to a good performance, whereas low DSC values to a bad performance. The aim is to use all features to create a signature for each data sample. Samples with the same signature show similarities, and we hope to find patterns in these relationships to draw conclusions about the developed models. Therefore, the signatures are visualized in order to represent the correlation to features (**T4**).

The numerical values of the features are grouped to classes. The combination of all feature classes forms a signature per data sample. We are only interested in dividing the value range, not in finding clusters in the numerical data. Therefore, the bin edges are

defined by the minimal and the maximal feature value. If not stated otherwise, the values are assigned to three classes that are determined by evenly spaced bins over the value range. For some features, e.g. skewness and kurtosis, fixed limit values with statistical significance are used to divide the range of values. A simple example with two classes A, B with limit value 0.5 and a feature value range of $[0, 1]$ is illustrated in Figure 5.4.

	Feature 1	Feature 2	Feature 3	Feature 4		Signature
Sample 1	0.1	0.2	0.1	0.6	$\longrightarrow$	AAAB
Sample 2	0.3	0.4	0.4	0.7		AAAB
Sample 3	0.6	0.1	0.2	0.8		BAAB
Sample 4	0.9	0.8	0.4	0.7		BBAB

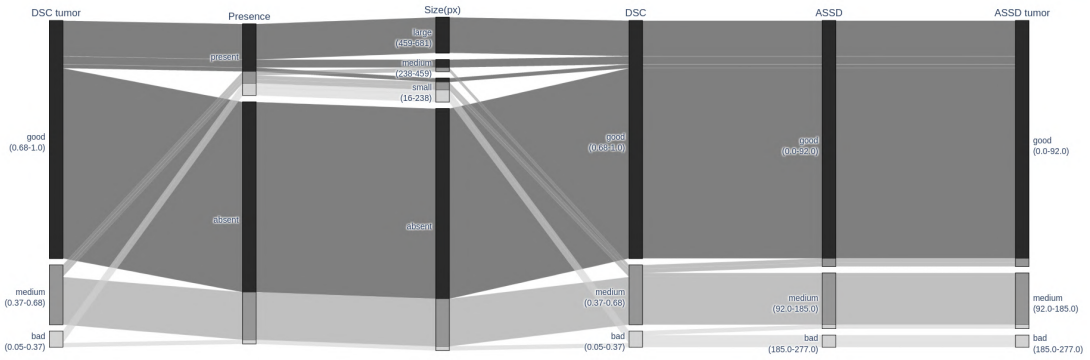
Figure 5.4: Illustration of signature creation for the features extracted for a data sample, i.e., either the volumetric scan per patient ID or the 2D slice.

We visualize the multivariate, categorical data in a **parallel set diagram** (PSD), also called parallel category plot [76]. A PSD has parallel vertical axes, one per category, consisting of blocks which represent the classes. The height of a block is proportional to the count. Each point/signature is represented by a polyline with vertices on the parallel axes going from left to right. The key features of the most left axes is defining the color groups. For data samples with the same signature or at least same signature parts, the polylines are merged to form a band. The PSD with all performance metrics and the task-specific features for the patient ID 233 is shown in Figure 5.5. The same diagram can be generated for all possible feature combinations. We support two different color themes to adapt to the user's needs. The first theme uses grayscale levels, ranging from dark for good to light for bad (see Figure 5.5a). It avoids too many different colors in our application for a more subtle PSD representation. With many overlaps, the polylines and bands can sometimes be difficult to follow. Thus, the second option is a more distinguishable color palette. The color map contains yellow for good, purple for intermediate and blue for bad results as shown in Figure 5.5b. The three colors are taken from the Plasma color map to establish color consistency.

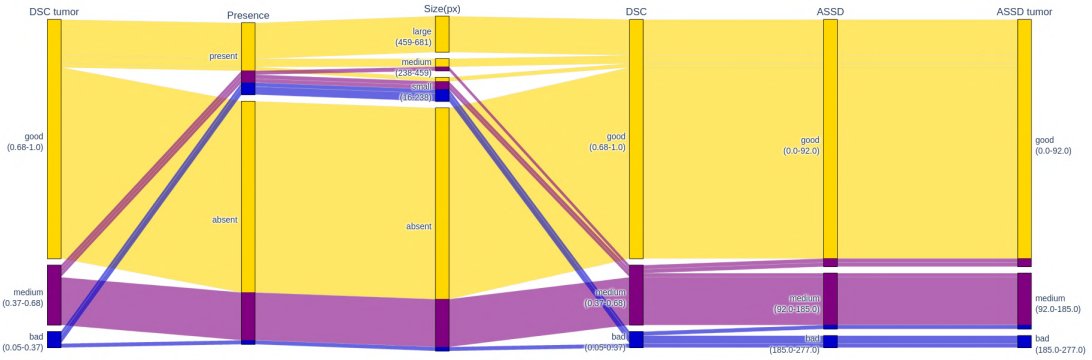
The PSD representation is a **n-by-n comparison** per definition. The individual data samples are assigned to subsets and compared with other subsets in terms of their characteristics.

5.2.4 Prediction Heatmap (T5)

The prediction comparison per slice should happen on the one hand among the algorithms results and on the other hand in comparison to the ground truth (GT) label (**T5**). Comparison of multiple predictions shows the agreement or disagreement between algorithms and reveals consistent errors made by a group of models. Since the user is not



(a) Grayscale color map



(b) Plasma color map

Figure 5.5: PSD with all performance metrics for DA models and the task-specific features for the slices with tumor presence in two different color themes. The data is taken from patient ID 233.

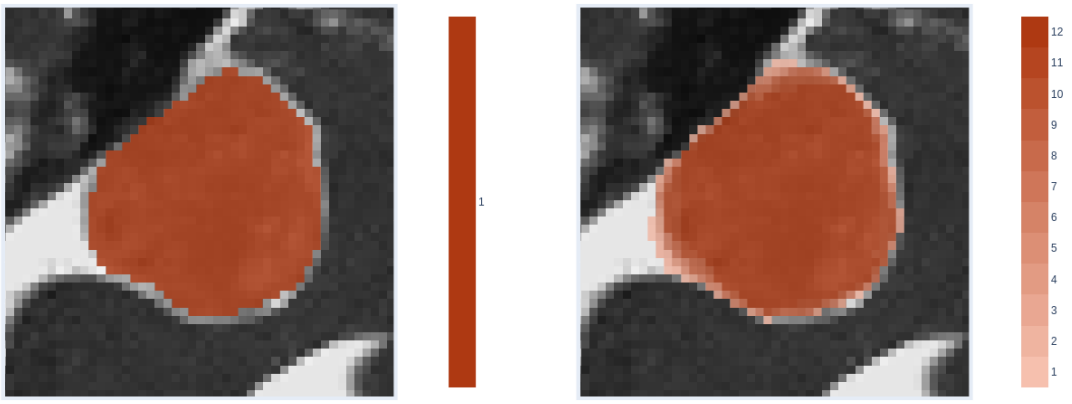
only interested in the performance of the algorithms compared to each other, but also in how the results compare spatially to the GT, this information must also be included.

In order to present several delineations comparatively, techniques of comparative visualization are useful [71]. Juxtaposition can be extended to multiple instances in one view, but each instance has its own coordinate system. This means that the spatial comparison between the delineations is lost. Superimposition combines multiple instances in the same coordinate system but suffers from occlusion issues. The method does not scale well with a large number of segmentation and we have more than 10 masks depending on the model selection. The scalability issue also occurs when using interchangeable. The instances are displayed in the same coordinate system but not at the same time which makes the direct comparison difficult for the user. This leaves the explicit encoding, which represents a processed version of all the delineations in the same coordinate system with no limit on the number of segmentations.

The generated mask is superimposed over the T2 scan with variable opacity. In theory,

the T2 scans can be exchanged for T1 scans, but since the goal of our approaches is the prediction on T2 images, we only support one modality. The image slice provides additional anatomical context. The user can examine the anatomical structure of the tumor, surrounding structures and regions with false positive or false negative predictions. The GT label is integrated as contour or mask overlay for reference. The prediction heatmap together with the GT superimposition is a **1-by-1** or **1-by-n comparison** for single or multiple algorithms, respectively.

The first explicit encoding is the **total sum** of all masks (see Figure 5.6). If only one segmentation algorithm is selected, the representation is just the mask itself, as shown in Figure 5.6a. For more than one mask, the number of times a pixel occurs in the masks is counted. Thus, a value of 0 in the final mask means that no segmentation mask contains this pixel. The maximal value is the number of selected segmentation algorithms, which means that all masks match in pixel labeling. Figure 5.6b shows the final mask for an example with 12 methods. The pixel values are color coded. Red corresponds to the maximum value, and as the count decreases, the color fades to white. This representation shows the user the agreement and consistency between algorithms. Nevertheless, the ensemble of models can be very confident in the merged mask, and still be incorrect in comparison to the GT.

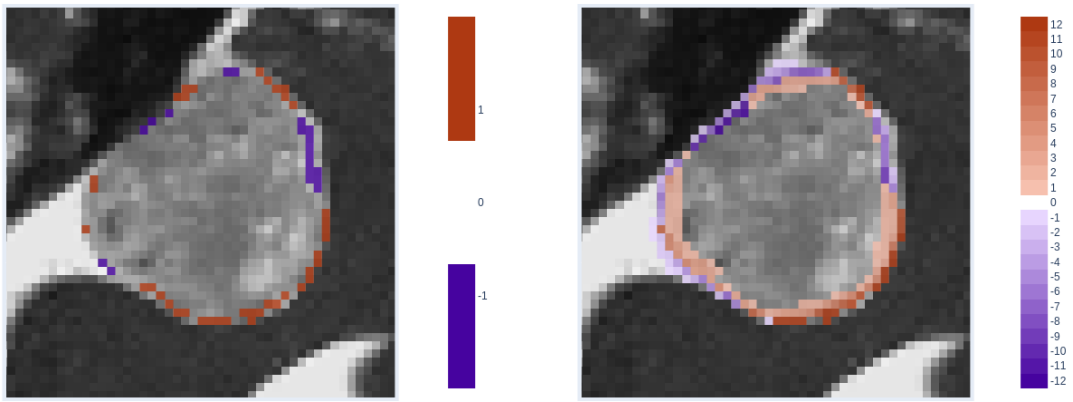


(a) Single segmentation mask (b) Sum of multiple segmentation masks

Figure 5.6: Prediction heatmap with summed segmentation masks for **X-Net S2T** `dstep=2` (a) and 12 (all) DA models (b). High values represent a high agreement between methods. The data is taken from patient ID 233, slice 25.

The second explicit encoding is **subtraction**. The sum described above is subtracted from the GT mask where the pixel value 1 is replaced by the number of models. Consequently, the resulting mask has positive and negative values. This concept works for both single and multiple masks, as shown in Figure 5.7. Positive values represent pixels that are not segmented by all methods. Negative values occur if at least one prediction includes false positive pixels, i.e., positive predictions for non-tumor pixels. Consequently, if the GT mask is empty, negative values mean that there is at least one false positive

tumor presence prediction. Depending on how many segmentation masks show the same behavior of local under- or oversegmentation, the representation color is more intense. The symmetric, diverging color map employs the concept of "cool" and "warm" for low and high values, which is natural for humans [96]. Hence, low values are mapped to blue, high values to red. The blue and red shades are taken from the Plasma color map to harmonize the color use in the visualization application. The brightest color is taken from the monochromatic color axis to interpolate the color from intense red and blue towards white in discrete steps. The best case scenario is an empty subtraction mask. This means that all selected methods correctly predicted each pixel.



(a) Single segmentation mask (b) Sum of multiple segmentation masks

Figure 5.7: Prediction heatmap with predicted masks subtracted from the GT mask for **X-Net S2T** `dstep=2` (a) and 12 (all) DA models (b). High values represent missing segmentation regions (under-segmentation), low values represent segmentation regions which do not belong to the tumor (over-segmentation). The data is taken from patient ID 233, slice 25.

5.2.5 Interaction

Several interaction techniques are known for VA applications and we also use them in the current work. Multiple (Coordinated) Views [136] show different perspectives of the data in multiple views. The design method is often used in combination with Brushing and Linking [69, 18], where one or multiple elements are selected in one view and highlighted in another. The link between individual views makes it easier to identify relationships. Focus+Context (F+C) [20] uses different levels of detail to decorate very important elements with more details, while less relevant elements remain in the view but without additional information.

5.3 User Interface and Interaction

The visualization techniques described in the previous section are combined and linked in a web-based visualization application. The components of the application and the interaction options are described in the following.

5.3.1 Application components

The layout of the web interface is shown in Figure 5.8 with the link to the tasks **T1-T5** defined in the previous section. The main components of the web interface, with which the user interacts, are described below.

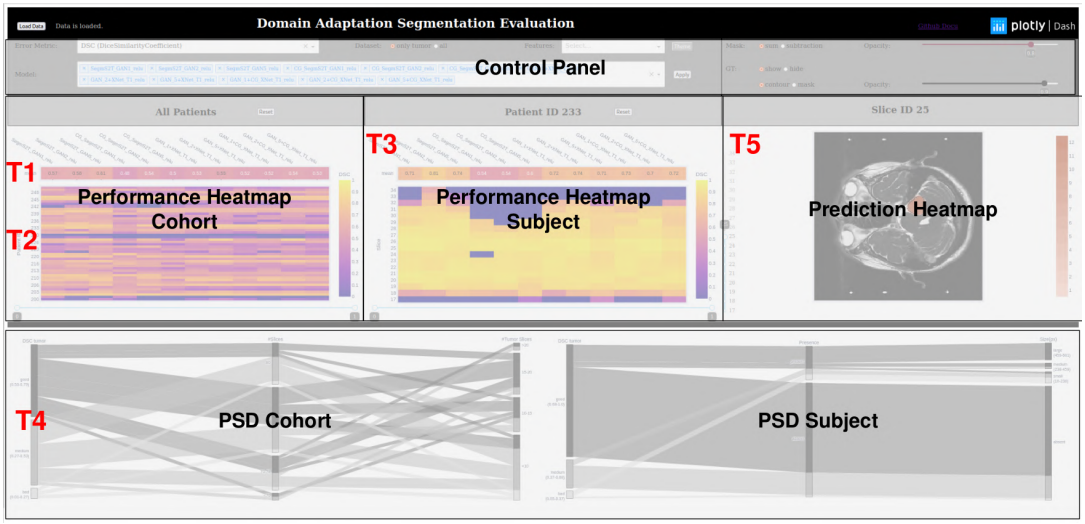


Figure 5.8: Illustration of user interface layout with the main components: Control Panel, Performance Heatmap Cohort (**T1**, **T2**), Performance Heatmap Subject (**T3**), Prediction Heatmap (**T4**), and PSD (**T5**).

Header

The header section contains a `Load Data` button on the left side. Data, that is not modified during run time and serves as a lookup for further processing, filtering, or selection, is loaded into memory during execution. On the right side, the link to the readme file of the github repository with the code is linked.

Control Panel

The control panel allows the user to regulate settings for the performance heatmap and the PSD on the left side and for the 2D prediction heatmap on the right side. The possible settings are displayed in Figure 5.9.

Figure 5.9: Control panel and website header.

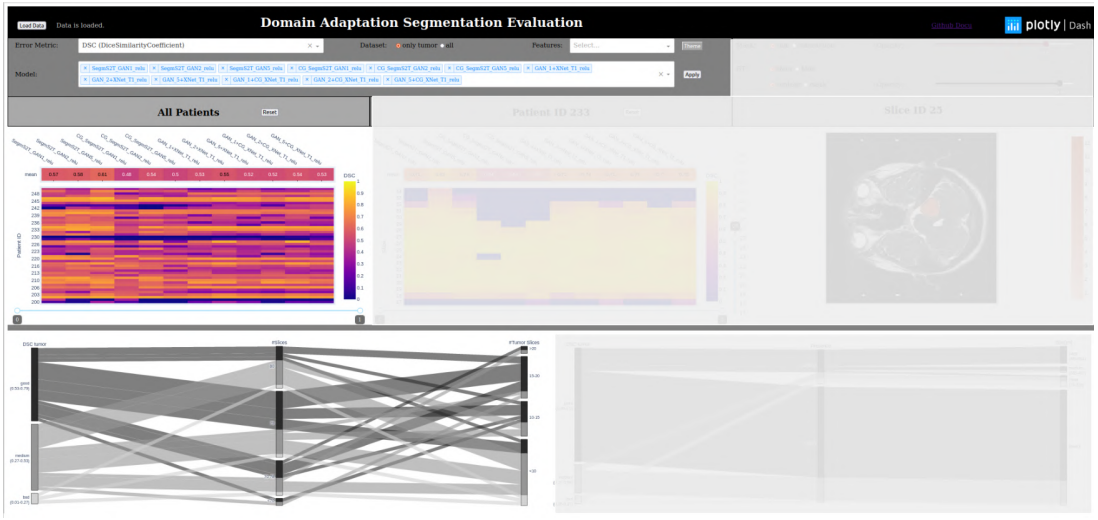
The user can select the performance metrics from DSC, ASSD, ACC, TPR, and TNR, as well as the dataset used, i.e., all slices or only slices with tumor presence. The combination of these two settings determine the performance feature for the performance heatmaps and the main feature of the PSD. The segmentation algorithms to be analyzed can be selected from a drop-down menu, listing the individual models and predefined groups. Models are grouped based on characteristics for easier selection. We have groups for all baseline (`Baseline`) and domain adaptation (`DA`) methods. The domain adaptation methods can further be divided into their approach, i.e. `SegmS2T` and `Gen+Segm`. Overall the methods can be split into classification-guided (`CG`) and standard (`NOT_CG`) methods. Finally, there is a class `Best` collecting the best methods as listed in Table 6.7 and a class `All` including all methods. For the PSD, a selection of the features is possible. The user can chose between the shape (`Shape`), first-order (`Firstorder`) or performance (`Performance`) features. The selected performance metric and the task-specific features are always the first left parallel axes and the most left parallel axis is the key feature determining the color coding. Next to the feature selection is a button to change the color theme of the PSD.

Cohort and Subject Views

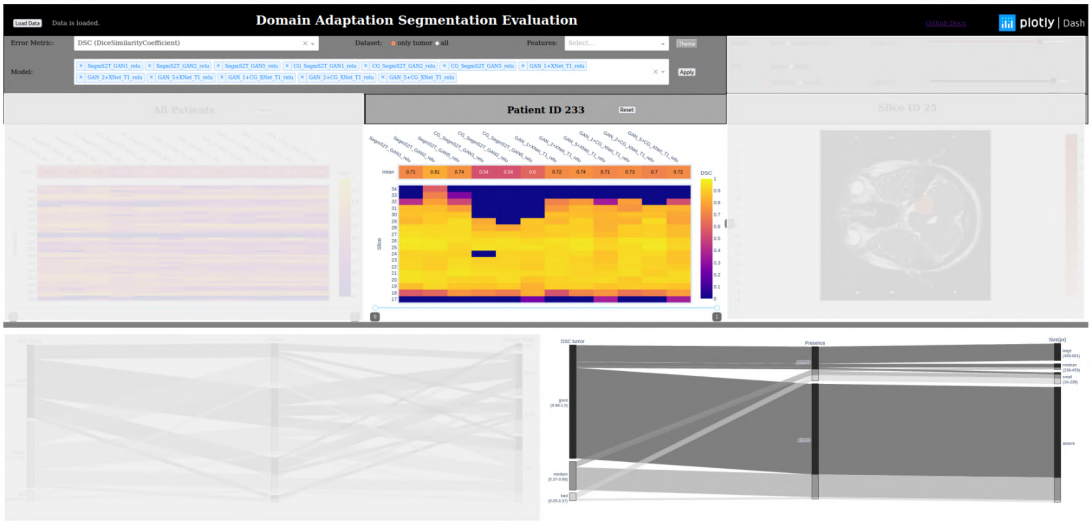
Subheadings designate the cohort and the subject elements of the application. Both include a performance heatmap for the error metrics and a PSD for the selected features, either per patient or per slice. In order to make the best use of the space, the PSD spans over 50% of the total width. The perception of the overarching PSD should also symbolize the connection between the different levels. The visualizations for the cohort and the subject are shown in Figure 5.10a.

Slice Views

The two views available for the slice-wise prediction heatmap, i.e., sum and subtraction encoding, are spatially aligned with the corresponding subheadings and the control panel options as the most left element in the application. The GT is either displayed as contour with thickness 1 (see Figure 5.11a) or as filled mask (see Figure 5.11b) with the color black. The opacity can be adjusted so that not all underlying layers of the visualization, such as the segmentation heatmap, are completely obscured.

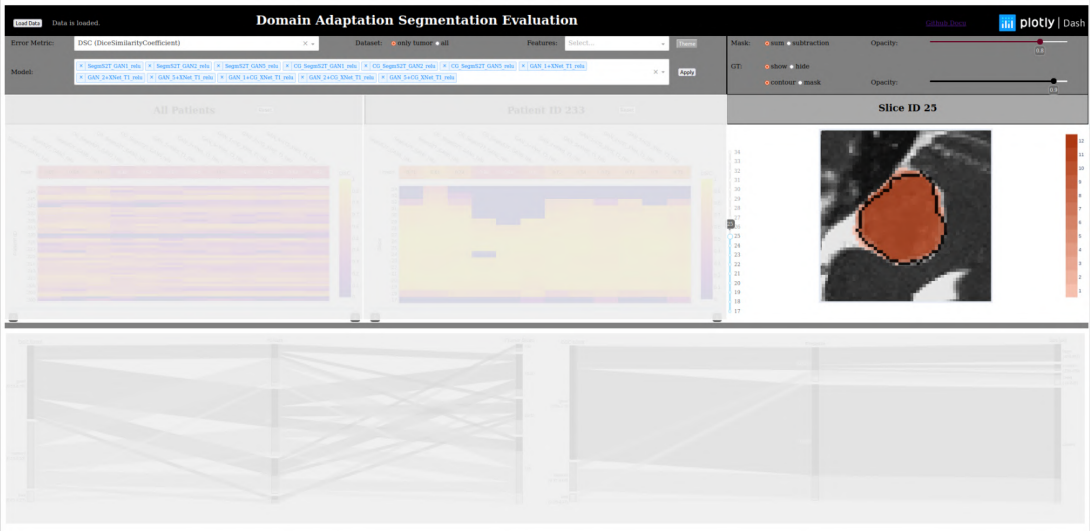


(a) Representation used for the exploration and analysis of a cohort.

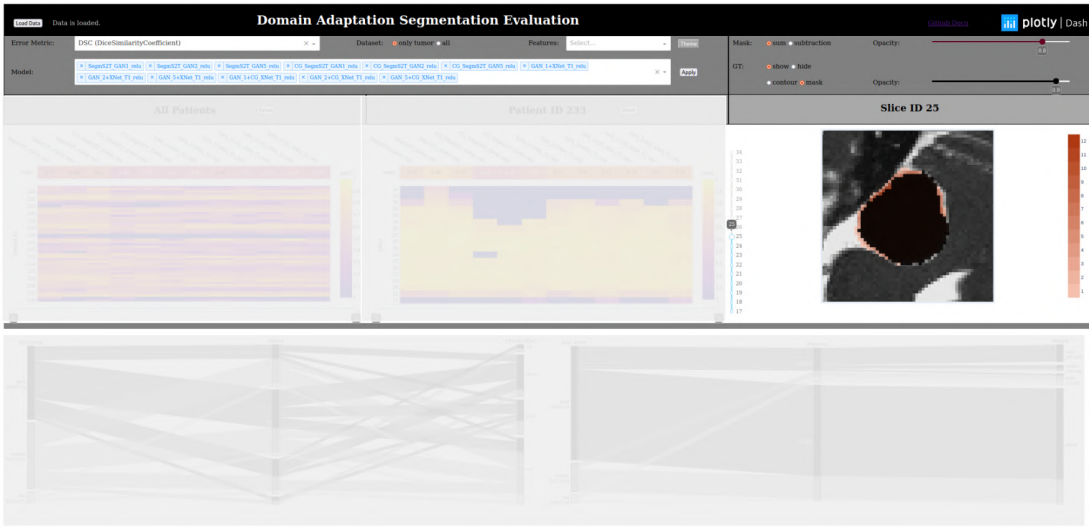


(b) Representation used for the exploration and analysis of an individual subject.

Figure 5.10: Cohort (a) and subject (b) visualization in form of performance heatmap and PSD. Other components of the application are grayed out.



(a) Sum encoding with GT contour.



(b) Sum encoding with filled GT mask.

Figure 5.11: Prediction heatmap with overlapping GT contour (a) and filled mask (b). Other components of the application are grayed out.

5.3.2 Interaction

We follow the concept *"Analyze first, Show the Important, Zoom, filter and analyze further, Detail on demand"* by Keim et al. [70]. The realization of the supported interaction techniques is described in the following subsection. An overview is given in Figure 5.12. The arrows indicate the direction of the connection.

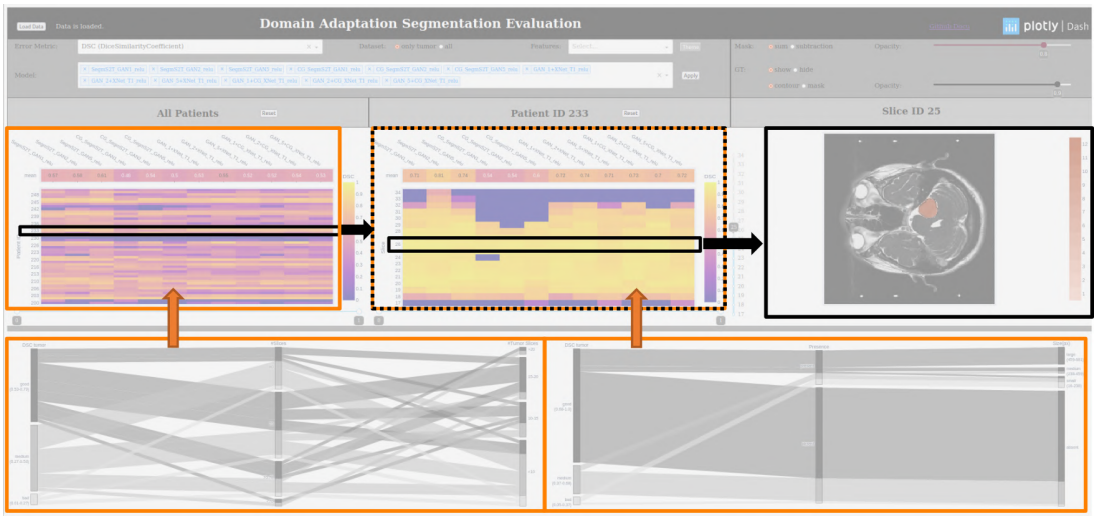


Figure 5.12: Illustration of some selected interaction possibilities in the application: Selection (black) and filtering (orange). The arrows indicate which component is affected.

Selection

If the data for the remaining elements is not yet available to the application, a placeholder with text annotation is used. The text provides information about the reason for the missing data, e.g., no data or models are selected. Before the user selects a slice for the first time, the slice in the middle of the volume (rounded down for odd slice numbers) is shown. The user can also slice through the volume of a patient data with the slider to the left of the segmentation figure. This should make the navigation in the volumetric data easier since the arrow keys can be used while hovering over the segmentation mask for pixel information. Selecting a row triggers an update of the subheadings to include the selected ID.

Filtering

The space of segmentation algorithms is filtered by the drop-down menu in the control panel. The same applies to the features for the PSD and the key performance metric.

The PSDs and performance heatmaps are linked, with PSDs always showing the entire dataset of the corresponding level and the heatmap showing only a filtered selection. When the mouse pointer is moved over individual elements of the PSD, the selected IDs are highlighted in the heatmap view. The not-selected IDs are grayed out, but are retained in the matrix to demonstrate where the selected data samples are located in relation to the whole set. Once the selection is confirmed by clicking on the element, it is transferred to the underlying matrix and only the chosen subset is shown. The subset is now fixed, and when the user hovers over other elements, the matrix returns to this subset. The selection of other subsets is possible by choosing other elements or

pressing the reset button. This restores the unfiltered matrix according to the specified parameters.

Filtering to control the focus of visual perception in the performance heatmap is supported by the slider under the heatmap diagrams. The range sliders regulate the color axis. Values below or above the limits on the range slider are mapped to the same color at the end of the Plasma color map. This increases the color spectrum for the values within the range and results in a more fine-graded color coding.

Hovering

Each component has its own hovering information. The performance heatmap reports the model name and the patient ID from the two dimensions of the matrix and the name and value of the performance metric that determines the color value. For the annotated heatmap row with mean values, the hovering is deactivated, since the value is already provided as cell label. In the PSD, hovering over a block in the parallel axes or a band of polylines is possible. It returns the number of samples in the group in absolute and relative numbers. The prediction heatmap limits the hovering to non-zero segmentation mask pixels to avoid information overload. The pixel values are reported.

5.4 Implementation

The visualization is realized with Dash [6] and plotly [7]. For the radiomics features, the python library pyradiomics is used with the default settings and implementations [128, 8]. For the generation of the segmentation prediction, we need a Nvidia GPU with at least 8 GB memory and Tensorflow and Keras [9].

Due to the high number of segmentation algorithms and slices in the test set, the inference takes some time and is done offline. With the hardware available, it takes up to 4 hours to generate all the predictions and process them. They are needed to calculate the performance measures. Therefore, they need to be available at application start and can not be predicted on demand during the run time. The calculation of the other features (radiomic and task-specific features) takes 1-3 minutes. In order to provide interaction in real-time, the data preparation as described in Section 5.1 is performed offline. The information is stored in JSON files and is accessible by the visualization application. The segmentation masks are reduced to contours and the list of coordinates are stored in JSON files as well. This reduces the memory storage need. Another advantage of offline calculation is that the hardware requirements for the machine on which the visualization application is running can be circumvented. To generate and store the data on a machine, allows to transfer the data to another machine. At reading time, the coordinates are processed to a contour which can be filled to gain a binary mask with a pre-defined image size of 256×256 pixels (H×W).

CHAPTER 6

Results and Discussion

In this chapter, results of the previously introduced segmentation approaches are presented. After looking at the quantitative values with the predefined error metrics, we use our visualization tool to comparatively analyze the predictions of the models and some characteristics of the dataset that might be related to the performance. Due to the high number of models and individual dataset samples in relation to the total number of slices, we showcase few interesting use case scenarios that relate to the main tasks defined in Chapter 1.

6.1 Quantitative Results for Segmentation

We will take a look at the performance of the baseline methods to establish an upper and lower boundary when the task is solved by supervised training operating with the ground truth data of both modalities. Then, the performance of DA methods is covered. If a method is referred to as the best, this is meant with respect to DSC. We put more focus on the DSC because it is highly used in related literature [60, 34, 25]. The DSC and the ASSD are calculated on binary masks, i.e. with values 0 for background and 1 for VS segmentation. They are generated by mapping pixel values above 0.5 to 1, pixel values below to 0. The ACC for tumor presence detection is inferred from the segmentation prediction. For architecture types including a CG module, the classification influences the segmentation mask by silencing the segmentation output if no tumor is detected. However, even if the tumor presence is detected correctly, the segmentation might still fail, i.e. the mask is all zero. This case is also considered a missing tumor segmentation, since our focus is the tumor delineation on T2 images. The classification extension only contributes to the training, the output is not considered for evaluation. In the tables presenting the results, we use the notation $\uparrow$ and $\downarrow$ to indicate that the desired values are either high or low. We report the error scores on two datasets: for image slices containing ground truth delineations for brain tumor (**D3**), and the entire

test dataset (**D1**). On one hand, we are interested in the performance on actual tumor samples. On the other hand, in clinical practice, the data sample obtained is volumetric containing more anatomical information than just the brain tumor. In order to apply an automatic segmentation algorithm to the entire 3D data sample without pre-filtering relevant slices or applying another detection algorithm beforehand, the algorithm should be able to deal with non-tumor slices. Thus, we analyze both datasets. In addition to analyzing each method individually, we also provide a summary and comparison of the best models per strategy and qualitative examples.

6.1.1 Baseline

The baseline results are produced with segmentation architectures trained in a supervised manner (see Section 4.3.1). These models are not taking the domain difference into account. Each modality has its own specialized network, which has the modality in the model name as suffix, such as **X-Net T2** or **CG X-Net T2**. We distinguish two different application possibilities: intended and "off-label" use. For intended use, the model is applied to data from the training domain. "Off-label" use is the prediction on images from a domain different than the training domain, such as apply **X-Net T1** to T2 images and **X-Net T2** to T1 images.

X-Net

Aside from **X-Net T1** and **X-Net T2** for T1 and T2, a simple multi-modality approach **X-Net T1+T2** is realized by concatenating the two input images and using the fused input for training. The results for each model are summarized in Table 6.1 for both modalities and three different activation functions. The best T2 model uses ReLU activation and reaches 0.71 DSC, 40.45 ASSD, and 89.22% ACC. This means that the overlap measured with DSC is on average 71% and the average deviation between contours is 41 pixels. The lower bound, with 0.071 DSC, 88.23 ASSD, and 84.57% ACC, is given by a T1 model using Leaky ReLU activation applied to T2 images. This method represents an approach using only available T1 ground truth data for training, but ignoring the domain shift. It proves to be very ineffective as the measured overlap drops to 1/10 of the previous value and the average deviation doubles. Table 6.2 summarizes the results for T2 segmentation of **X-Net T2** and **X-Net T1**. The ReLU activation is superior not only for intended but also for "off-label" use when looking at the entire test set. The highest DSC are 0.8425 and 0.5796 for **X-Net T2** and **X-Net T1** applied to T2 images, respectively.

The choice of activation function influences the performance with a margin of ± 0.04 DSC, which does not show a clear predominance of an activation function. However, ReLU proved to be stable considering the combination of DSC, ASSD and ACC for intended use and Leaky ReLU for unintended use. SeLU neither produces the best DSC scores nor predominant ASSD and ACC values. We could not observe a quantitative benefit of SeLU over ReLU or Leaky ReLU. Since the domain adaptation methods incorporate the

difference in the modalities and the application is not considered "off-label", we continue with ReLU as activation function if not stated otherwise.

Overall, we observe that the difference in performance is more than 0.10 DSC between T1 and T2, which means that T1 predictions are consistently more accurate than T2 predictions. Moreover, using T2 methods on T1 images works better than vice versa. We conclude that T2 prediction is more challenging than T1 prediction. This is explained by the fact that the quality of T1 images is better and T2 images have lower contrast, which makes the exact tumor boundary harder to detect [133].

X-Net T2 - D3				
target modality	activation	DSC ↑	ASSD ↓	ACC ↑
	relu	0.71 ± 0.32	40.5494 ± 111.8	89.22%
T2 (intended)	leaky	0.6912 ± 0.33	42.5344 ± 113.33	88.85%
	selu	0.6793 ± 0.33	40.2957 ± 109.83	89.59%
	relu	0.3304 ± 0.41	178.6781 ± 177.08	51.86%
T1 ("off-label")	leaky	0.3642 ± 0.41	71.5728 ± 127.39	84.20%
	selu	0.3476 ± 0.4	133.0548 ± 165.58	65.99%
X-Net T1 - D3				
target modality	activation	DSC ↑	ASSD ↓	ACC ↑
	relu	0.8238 ± 0.28	23.9276 ± 87.87	93.68%
T1 (intended)	leaky	0.8164 ± 0.27	22.7581 ± 84.1	94.24%
	selu	0.8152 ± 0.28	27.9048 ± 93.56	92.75%
	relu	0.0426 ± 0.14	226.0561 ± 163.77	41.08%
T2 ("off-label")	leaky	0.071 ± 0.17	88.2303 ± 120.01	84.57%
	selu	0.0608 ± 0.14	71.024 ± 113.82	86.99%
X-Net T1+T2 - D3				
target modality	activation	DSC ↑	ASSD ↓	ACC ↑
	relu	0.8266 ± 0.27	23.793 ± 87.87	93.68%
T1+T2	leaky	0.8343 ± 0.26	20.9521 ± 82.89	94.42%
	selu	0.8305 ± 0.26	21.3539 ± 82.88	94.42%

Table 6.1: Results for **X-Net T2**, **X-Net T1**, and **X-Net T1+T2** for test set **D3**. The models are trained in a supervised manner. DSC, ASSD, and ACC are reported for both modalities and three different activation functions. The notations ↑ and ↓ refer to the desired result, i.e. high and low values, respectively.

Results on T2 images - D1			
X-Net T2 (intended)			
activation	DSC ↑	ASSD ↓	ACC ↑
relu	0.8425 ± 0.34	47.6676 ± 122.09	86.89%
leaky	0.6464 ± 0.46	117.954 ± 169.35	67.50%
selu	0.5832 ± 0.48	139.8881 ± 175.9	61.46%
X-Net T1 ("off-label")			
relu	0.5795 ± 0.49	134.7212 ± 173.36	63.3%
leaky	0.2441 ± 0.42	237.6359 ± 167.75	35.65%
selu	0.1825 ± 0.38	256.9102 ± 160.87	29.99%

Table 6.2: Results for T2 application of **X-Net T2** (intended) and **X-Net T1** ("off-label") for test set **D1**. The notations ↑ and ↓ refer to the desired result, i.e. high and low values, respectively.

CG X-Net

We present **CG X-Net T1** and **CG X-Net T2** as the second baseline models. They are trained on a balanced dataset of sections with and without tumor (**D2**). The results on the test sets **D3** and **D1** are presented in Table 6.3. The performance measures for **CG X-Net T2** on **D3** are 0.696 DSC, 43.49 ASSD, and 88.48% ACC. For **CG X-Net T1** the scores are 0.81 DSC, 33.64 ASSD, and 90.89%. "Off-label" use of **CG X-Net T1** on **D3** T2 images results in 0.18 DSC, 160.38 ASSD, and 58.55% ACC. The behavior of *X-Net* versions previously described is maintained: The error metrics for T1 images are better compared to T2 images. The DSC for T2 images in **D3** is 0.014 points lower than for the standard *X-Net* version. This means that the difference is smaller than for different activation functions in the standard *X-Net* versions. In addition, **CG X-Net T1** applied to T2 images results in a DSC of 0.1803, which improves the lower baseline by more than 0.10 points. For T2 images of **D1**, the error scores improve to 0.94 DSC, 11.88 ASSD, and 96.74% for **CG X-Net T2** and 0.76 DSC, 6.54 ASSD, and 98.06% for **CG X-Net T1**. This is an improvement for both models on **D1**.

6.1.2 Domain Adaptation

The networks in this section are designed to account for the domain shift (see Section 4.3.3 and 4.3.4). Although the methods can be used to take either one of the image modalities as source and the other as target domain, the target domain for the results reported is set to T2 images. All methods are applied to T2 images, except stated otherwise.

CG X-Net T2			
D3			
target modality	DSC ↑	ASSD ↓	ACC ↑
T2 (intended)	0.6962 ± 0.33	43.4989 ± 115.0	88.48%
T1 ("off-label")	0.2456 ± 0.39	253.8364 ± 164.21	30.30%
D1			
target modality	DSC ↑	ASSD ↓	ACC ↑
T2 (intended)	0.9406 ± 0.2	11.8761 ± 63.77	96.74%
T1 ("off-label")	0.8358 ± 0.36	56.6478 ± 131.25	84.25%
CG X-Net T1			
D3			
target modality	DSC ↑	ASSD ↓	ACC ↑
T1 (intended)	0.8113 ± 0.29	33.637 ± 103.95	90.89%
T2 ("off-label")	0.1803 ± 0.28	160.3764 ± 170.06	58.55%
D1			
target modality	DSC ↑	ASSD ↓	ACC ↑
T1 (intended)	0.968 ± 0.15	6.5401 ± 47.85	98.06%
T2 ("off-label")	0.7616 ± 0.41	66.5179 ± 138.42	81.98%

Table 6.3: Results for **CG X-Net T2** and **CG X-Net T1** for test sets **D3** and **D1**. The models are trained in a supervised manner with a classification-guided module. DSC, ASSD and ACC are reported for both modalities for intended and "off-label" use. The notations ↑ and ↓ refer to the desired result, i.e. high and low values, respectively.

Generators G_{S2T} and G_{T2S}

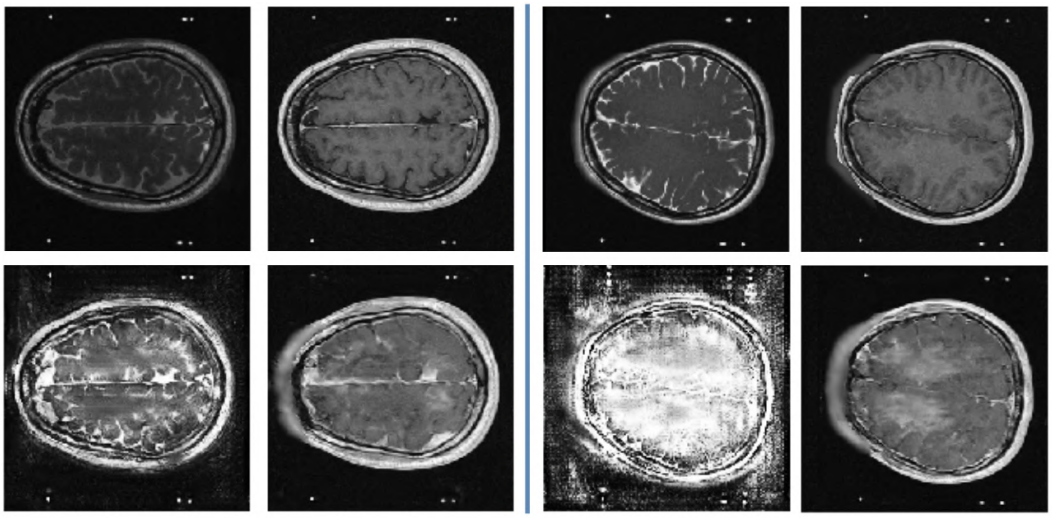
The generators G_{S2T} and G_{T2S} are used for training and inference, respectively. Their performance is crucial for the results of the DA methods. Generator G_{S2T} is applied to T1 images to generate synthetic T2 images ($\hat{T2}$) which are then used to train **X-Net S2T**. Generator G_{T2S} provides generated T1 images ($\hat{T1}$) for the inference of $\mathbf{G_{T2S}}$**+X-Net T1**. Depending on how often the corresponding discriminators were trained in an epoch, there are different generator versions. The parameter `dstep` describes at which steps the discriminators are trained, e.g. `dstep=5` means that the discriminator is trained every 5th step in an training epoch. MSE of real and synthetic images and the generator loss, i.e. how good the generator can fool the discriminator for `dstep=1`, are reported in Table 6.4. The results show that there is no training strategy that is best for all applications.

Generator G_{S2T} (training)				
D training	**D3**		**D1**	
	MSE(T2,$\hat{T2}$) ↓	MSE(1,$D_T(\hat{T2})$) ↓	MSE(T2,$\hat{T2}$) ↓	MSE(1,$D_T(\hat{T2})$) ↓
`dstep=1`	0.0563	2.390	0.0767	2.621
`dstep=2`	0.0533	2.389	0.0941	2.631
`dstep=5`	0.0520	2.654	0.069	2.662
Generator G_{T2S} (inference)				
D training	**D3**		**D1**	
	MSE(T2,$\hat{T1}$) ↓	MSE(1,$D_T(\hat{T1})$) ↓	MSE(T2,$\hat{T1}$) ↓	MSE(1,$D_T(\hat{T1})$) ↓
`dstep=1`	0.0487	2.201	0.0730	2.158
`dstep=2`	0.0446	2.086	0.0674	2.032
`dstep=5`	0.0520	2.394	0.0760	2.167

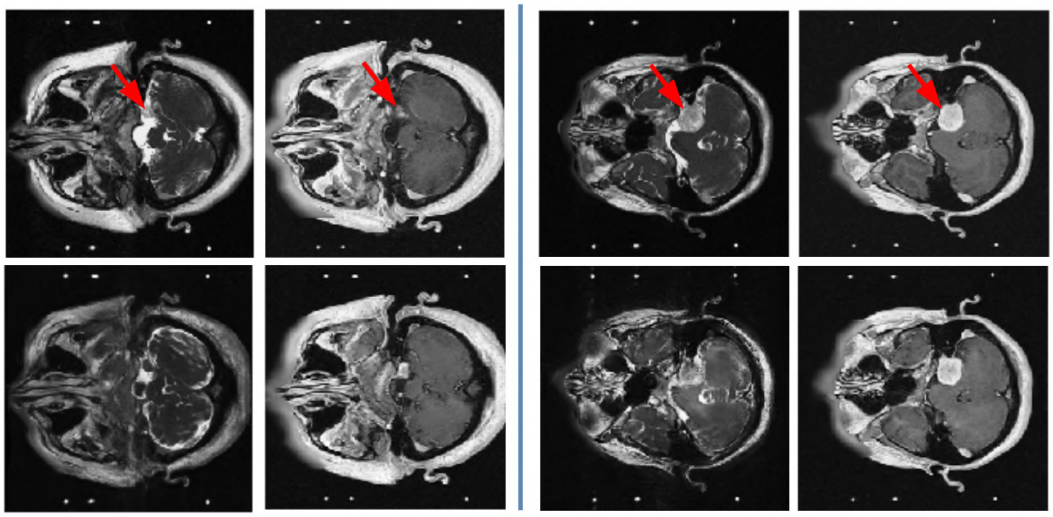
Table 6.4: Results for test set **D3** and **D1** for generators G_{S2T} and G_{T2S}. MSE of real and synthetic image pair and MSE between 1 and the discriminator results of synthetic images are reported. The notations ↑ and ↓ refer to the desired result, i.e. high and low values, respectively.

G_{S2T} produces the most resembling images with `dstep=5`, G_{T2S} with `dstep=2` for **D3**. Training the discriminator not every step benefits the generator training, since the discriminator is not too strong from the beginning and a better balance is found. The MSE increases for **D1** which can be explained by the fact that only **D3** has been used for training.

Reconstruction examples are shown in Figure 6.1. The tumors are marked with red arrows. The first row depicts the original images and the second row the synthetic images produced by the corresponding generator. In the test set D1, there are also slices from the top and bottom of the slice stack that have a different appearance. Even though the *CycleGan* has not seen images of this appearance, the results are still astounding. Nevertheless, in both applications, the ability of the generators to preserve and reconstruct the tumor structure correctly, influences the segmentation. Qualitative assessment shows a better tumor preservation on synthetic T1 images which corresponds to our assumptions.



(a) Samples from D1 test set without a tumor.



(b) Samples from D3 test set.

Figure 6.1: Four examples for generator $\mathbf{G_{S2T}}$ and $\mathbf{G_{T2S}}$ separated by blue line: Original images are in the first row - left: T2, right: T1; Synthetic images are in the second row - left: synthetic T2 image generated with G_{S2T}, right: synthetic T2 image generated with G_{T2S}. Best viewed in high resolution.

$\mathbf{G_{T2S}}$+X-Net T1 and $\mathbf{G_{T2S}}$+CG X-Net T1

For the **G_{T2S}+X-Net T1** framework, the **X-Net T1** models of the previous sections are combined with different *CycleGAN* generators G_{T2S} to predict on T2 images. By exchanging **X-Net T1** with a network employing the CG module, we get **G_{T2S}+CG X-Net T1**. The results of the model combinations are listed in Table 6.5 for both test

sets **D3** and **D1**. The best result on dataset **D3** is 0.553 DSC, 72.13 ASSD, and 81.41% ACC for the `dstep=2` version without CG module. If we consider the entire dataset **D1**, `dstep=1` with CG module works best with 0.869 DSC, 36.20 ASSD, and 90.13% ACC. There is no clear trend which discriminator training strategy is more beneficial.

G_{T2S}+X-Net T1			
D3			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.5318 ± 0.38	70.6398 ± 137.15	81.97%
`dstep=2`	0.5531 ± 0.38	72.1256 ± 138.98	81.41%
`dstep=5`	0.524 ± 0.36	60.3845 ± 127.47	84.94%
D1			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.6093 ± 0.47	127.1085 ± 172.05	65.10%
`dstep=2`	0.5794 ± 0.48	139.2345 ± 175.46	61.73%
`dstep=5`	0.5094 ± 0.48	161.3519 ± 179.14	55.66%
G_{T2S}+CG X-Net T1			
D3			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.5214 ± 0.38	89.0582 ± 151.31	76.58%
`dstep=2`	0.5443 ± 0.38	87.9963 ± 151.75	76.58%
`dstep=5`	0.5264 ± 0.36	75.2572 ± 140.81	80.67%
D1			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.8602 ± 0.32	38.4022 ± 110.55	89.56%
`dstep=2`	0.8692 ± 0.31	36.198 ± 107.87	90.13%
`dstep=5`	0.7406 ± 0.42	79.9413 ± 149.31	78.13%

Table 6.5: Results for **G_{T2S}+X-Net T1** and **G_{T2S}+CG X-Net T1** for test sets **D3** and **D1**. The *CycleGAN* generator from T2 to T1 modality and a supervised *X-Net* or *CG X-Net* with ReLU activation trained on T1 images are connected in series. The generators used have different discriminator training schedules. The notations ↑ and ↓ refer to the desired result, i.e. high and low values, respectively.

Both results represent a significant improvement of the unsupervised baseline method by more than 0.30 DSC for **D3** and 0.10 DSC for **D1**. The gap between **D1** results is lower due to the high number of non-tumor slices. If an algorithm produces a lot of all-zero slices independent of tumor presence or absence, the prediction on **D1** which contains more non-tumor slices than tumor slices (see Figure 4.2a) is correct for the majority of slices. Applying this insight to the interpretation of the "off-label" baseline results, it can be seen that the better results for **D1** compared to **D3** are due to the high number of empty predictions. However, the same behavior of the "off-label" baseline models leads to poor quality for **D3** slices. In comparison, DA methods especially without the CG module encourage global over-segmentation. Examples for non-tumor slices with segmentation predictions are shown in Figure 6.2. The positive predictions occur in areas with bright spots that resemble tumor structures, such as in the left image. At these location, the position is also consistent with the spatial information from the training data. However, examples such as the picture on the right show that false predictions are not only limited to this area. There should not be any confusion in these images because the tumor boundary was not reconstructed accurately enough.

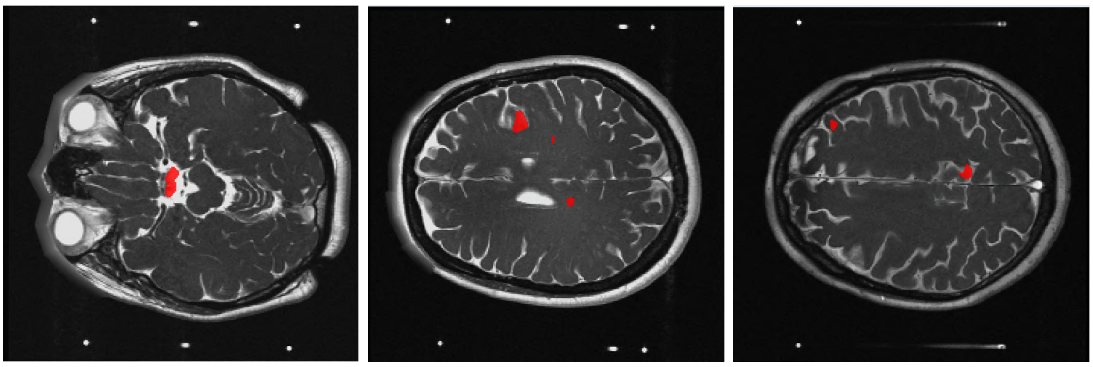


Figure 6.2: Examples for segmentation predictions (red) in slices without tumor presence, i.e. over-segmentation and a false positive prediction, produced with $\mathbf{G_{T2S}}$**+X-Net T1** `dstep=2`. Best viewed in high resolution and color.

X-Net S2T

For the **X-Net S2T** model, the generator produces synthetic T2 images which are used as training data together with the T1 label ground truth. At inference time, we apply the **X-Net S2T** model directly to real T2 scans. Table 6.6 provides an overview of the results. The best **D3** result is **X-Net S2T** with `dstep=5` reaching 0.611 DSC, 59.56 ASSD, and 84.57% ACC. For **D1**, it is the version using `dstep=1` and the CG module with 0.9262 DSC, 18.196 ASSD, and 94.90% ACC. As expected, this is another performance improvement compared to the lower boundary of the baseline methods and the other DA method. Similar to the first DA method, no preferable update parameter for the discriminator can be determined. The trend that classification-guided segmentation is

inferior to standard versions for **D3** but superiority for **D1** is preserved.

X-Net S2T			
D3			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.5695 ± 0.38	72.8934 ± 141.0	80.86%
`dstep=2`	0.5771 ± 0.37	74.5116 ± 142.64	80.30%
`dstep=5`	0.6114 ± 0.36	59.5639 ± 129.52	84.57%
D1			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.7932 ± 0.39	62.8065 ± 136.45	82.79%
`dstep=2`	0.7704 ± 0.4	71.7301 ± 143.75	80.31%
`dstep=5`	0.6502 ± 0.46	114.8601 ± 167.95	68.42%
CG X-Net S2T			
D3			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.4797 ± 0.41	134.0608 ± 172.66	63.57%
`dstep=2`	0.5398 ± 0.4	107.9157 ± 163.29	70.82%
`dstep=5`	0.4962 ± 0.4	114.3868 ± 164.96	69.33%
D1			
D training	DSC ↑	ASSD ↓	ACC ↑
`dstep=1`	0.9172 ± 0.25	22.087 ± 86.02	93.88%
`dstep=2`	0.9262 ± 0.24	18.196 ± 78.4	94.90%
`dstep=5`	0.9183 ± 0.25	19.7206 ± 81.17	94.53%

Table 6.6: Results for **X-Net S2T** and **CG X-Net S2T** with ReLU activation trained with different underlying generators G_{T2S} with `dstep=1,2,5` for test set **D3** and **D1**. The notations ↑ and ↓ refer to the desired result, i.e. high and low values, respectively.

CG SIFA

As already described in Section 4.4.4, we tried a lot of different architecture modifications and training strategy changes for both, **SIFA** and **CG SIFA**. Despite numerous experiments, we did not manage to train a stable version that showed satisfying results. We document in this section our results as guidelines for future implementations.

Figure 6.3 shows the TensorBoard tracking of the Dice Coefficient and Dice Loss of one of the few stable **SIFA** training in our experiments. Since no T2 GT labels are used during training, the progress on T2 images can not be documented. Although all loss values are reduced and the network seems to "learn", the results are not convincing at test time. The scores for **D3** are 0.27 DSC, 98.8696 ASSD, and 81.97% ACC and for **D1** 0.59 DSC, 123.00 ASSD, and 67.37% ACC. The framework was trained without segmentation for 25 epochs. Then, the segmentation branch including the segmentation discriminator are turned on and trained for another 75 epochs. To reduce the training time, `dstep` is set to 5. Another setting with 50 epochs pre-training and 50 epochs for segmentation training results in similar error scores at test time. The scores for **D3** are 0.25 DSC, 104.56 ASSD, and 77.32% ACC and for **D1** 0.61 DSC, 116.21 ASSD, and 68.8 ACC. These values are far below our expectations and are also lower than the other DA methods.

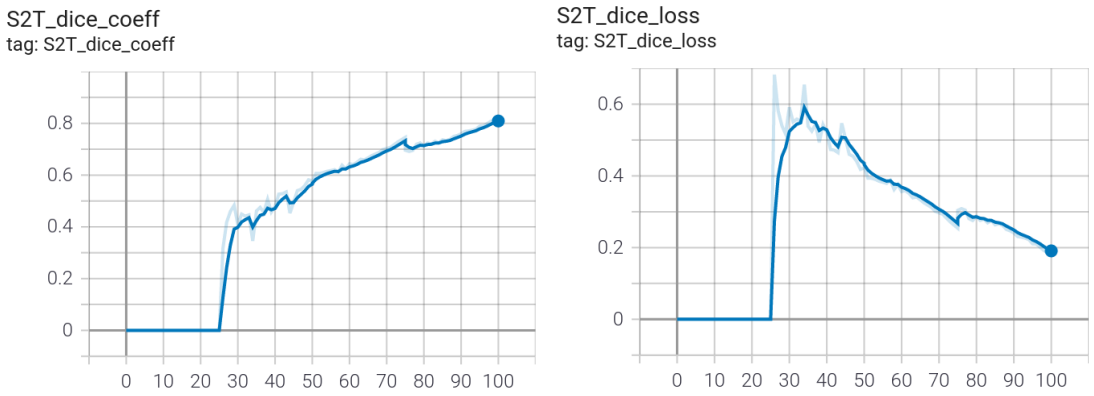


Figure 6.3: Dice Coefficient and Dice Loss of the synthetic T2 image predictions with the T1 GT labels tracked during a **SIFA** training where the model seems to "learn".

The **CG SIFA** training has proven to be very unstable. Especially if the segmentation with CG module is already optimized from the very first epoch. Thus, we experimented with training the framework without the CG module and the pretraining strategy described above for an initial adaptation period. Then, the pre-trained network weights can be used as starting point and to optimize the segmentation branch with the CG module. The TensorBoard tracking of the dice coefficient and loss for such a pretraining phase is shown in Figure 6.4. Training was done for 50 epochs, using 25 as pre-training. At epoch 45 the training collapsed, the loss functions jumped to higher values for no apparent reason. This behavior was observed several times with different training strategies, total number of epochs, and starting epoch of segmentation training. The performance measures are even worse than the lower baseline results. This approach was not pursued further, as it would have exceeded the scope of this work.

There are many potential reasons for the unexpected behavior. First, we would like to note that the original implementations were trained on GPUs with at least 12 GB and

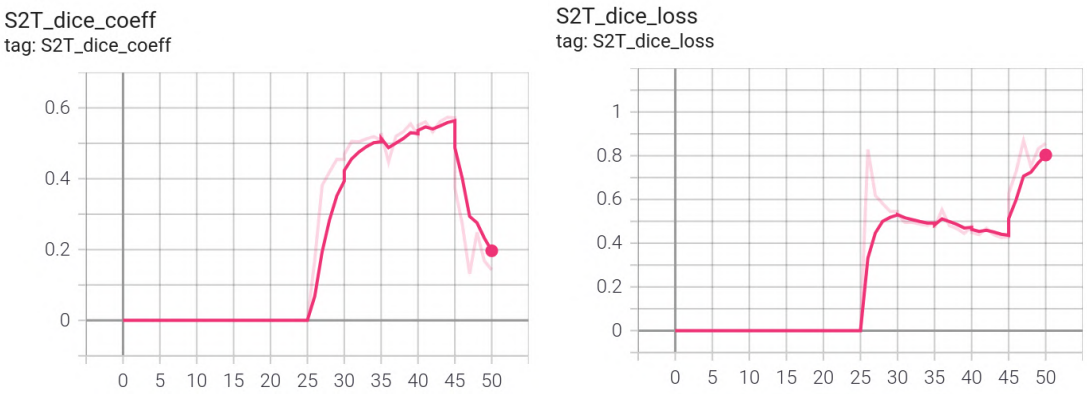


Figure 6.4: Dice Coefficient and Dice Loss of the synthetic T2 image predictions with the T1 GT labels tracked during a **CG SIFA** training where the model training collapses.

a batch size of 8 [24]. This allows for deeper networks with more capacity and a more stable training by using multiple samples instead of a single one. We hypothesize that the network modifications required to fit the framework into 8 GB limit the number of trainable parameters, which prevents the framework from extracting the relevant features. Another factor could be the training strategy and parameters used. The framework has many hyperparameters that need to be set, e.g. `dstep`, the learning rate, the start epoch for the segmentation optimization, parameters in the loss functions. They must be aligned to ensure successful training. Although most of the parameter values have been transferred from already existing implementations, they may not be ideal for our particular use case. In addition, human error during implementation could also be the cause of the unexpected training behavior. For the following evaluation, we excluded the **SIFA** and **CG SIFA** models. However, we think that a further exploration of such models is worthwhile for the future.

6.1.3 Summary

Table 6.7 holds DSC, ASSD, and ACC values for the test sets **D1** and **D3** for the best specialized baseline methods and the best DA methods applied to the target domain, i.e. T2 images. Compared on **D3**, models employing the CG module are defeated by standard segmentation approaches in all error metrics and for all methods. The only exception is the "off-label" use for models trained on labeled T1 images to determine a lower boundary. For this case, the CG module boosts the performance by more than 0.1 DSC, but ASSD and ACC deteriorate. Nevertheless, extending the dataset to **D1** shows the benefit of the CG module. Due to more TN and less FP predictions, all scores exceed the standard baseline methods and are significantly better. This means that the CG module can help to avoid global under- and oversegmentation, i.e. not segmenting slices with tumor and segmenting slices without tumor (see Figure 6.2), respectively. Hence, the overall segmentation performance does benefit from the additional data and

the multi-task learning introduced by *CG X-Net*. To rule out the possibility that only the additional training data is the reason for the better results, we tested training the baseline models with the balanced data sets **D2**. The training runs have shown an unstable training. The Dice Coefficient converges to a plateau below 0.1 at very early stages in the training. It seems that the network is not able to deal with empty ground truth labels. The optimization stagnate since empty masks for all slices might be a local minimum.

Figure 6.5 provides three visual zoomed-in examples for the best methods on **D3**. The first example is a large circular-like tumor which is covered by all methods with a few pixels deviation. The second example is a smaller tumor with a more coarse border. CG methods (bottom row) are more conservative resulting in smaller masks, whereas standard methods tend to over-segment. The third example is more ellipsoidal shaped and all methods having trouble production an accuracy delineation. The qualitatively best result is generated with $\mathbf{G_{T2S}}$**+X-Net T1**. **X-Net S2T** and **CG X-Net S2T** (last column) predict nothing, producing a FN result.

Overall, looking at the quantitative results, we can summarize the following observations:

- Segmentation algorithms with domain adaptation outperform models that do not encounter the domain shift. However, our DA methods cannot reach the scores of a supervised segmentation network trained on labeled T2 images.
- Our results show that it works better to train a segmentation network with synthetic images (**X-Net S2T**) than applying a network trained on real image to synthetic images at test time ($\mathbf{G_{T2S}}$**+X-Net T1**).
- We were not able to train a stable **SIFA** or **CG SIFA** version with the available hardware and number of images. Further work is required in this direction.
- The use of a CG module improves the performance on a dataset containing slices with and without tumor. At the same time, tumor segmentation DSCs remain in the same range as for networks without a CG module.

Best Results - D3			
Intended Use (upper boundary)			
Model	DSC ↑	ASSD ↓	ACC ↑
X-Net T2 ReLU	0.71 ± 0.32	40.5494 ± 111.8	0.8922%
CG X-Net T2	0.6992 ± 0.33	43.4989 ± 115.0	0.8848%
Domain Adaptation			
Model	DSC ↑	ASSD ↓	ACC ↑
G_{T2S}+X-Net T1 `dstep=2`	0.5531 ± 0.38	72.1256 ± 138.98	81.41%
G_{T2S}+CG X-Net T1 `dstep=2`	0.5443 ± 0.38	87.9963 ± 151.75	76.58%
X-Net S2T `dstep=5`	0.6114 ± 0.36	59.5639 ± 129.52	84.57%
CG X-Net S2T `dstep=2`	0.5398 ± 0.4	107.9157 ± 163.29	70.82%
"Off-label" Use (lower boundary)			
Model	DSC ↑	ASSD ↓	ACC ↑
X-Net T1 Leaky ReLU	0.071 ± 0.17	88.2303 ± 120.01	84.57%
CG X-Net T1	0.1803 ± 0.28	160.3764 ± 170.06	58.55%
Best Results - D1			
Intended Use (upper boundary)			
Model	DSC ↑	ASSD ↓	ACC ↑
X-Net T2 ReLU	0.8425 ± 0.34	47.6676 ± 122.09	0.8689%
CG X-Net T2	0.9406 ± 0.2	11.8761 ± 63.77	0.9674%
Domain Adaptation			
Model	DSC ↑	ASSD ↓	ACC ↑
G_{T2S}+X-Net T1 `dstep=1`	0.6093 ± 0.47	127.1085 ± 172.05	65.10%
G_{T2S}+CG X-Net T1 `dstep=2`	0.8692 ± 0.31	36.198 ± 107.87	90.13%
X-Net S2T `dstep=1`	0.7932 ± 0.39	62.8065 ± 136.45	82.79%
CG X-Net S2T `dstep=2`	0.9262 ± 0.24	18.196 ± 78.4	94.90%
"Off-label" Use (lower boundary)			
Model	DSC ↑	ASSD ↓	ACC ↑
X-Net T1 ReLU	0.5795 ± 0.49	134.7212 ± 173.36	63.30%
CG X-Net T1	0.7616 ± 0.41	66.5179 ± 138.42	81.98%

Table 6.7: Results for the best models of baseline and DA approaches for test set **D3** and **D1**. DSC, ASSD, and either predicted or inferred ACC are reported. The notations ↑ and ↓ refer to the desired result, i.e. high and low values, respectively.

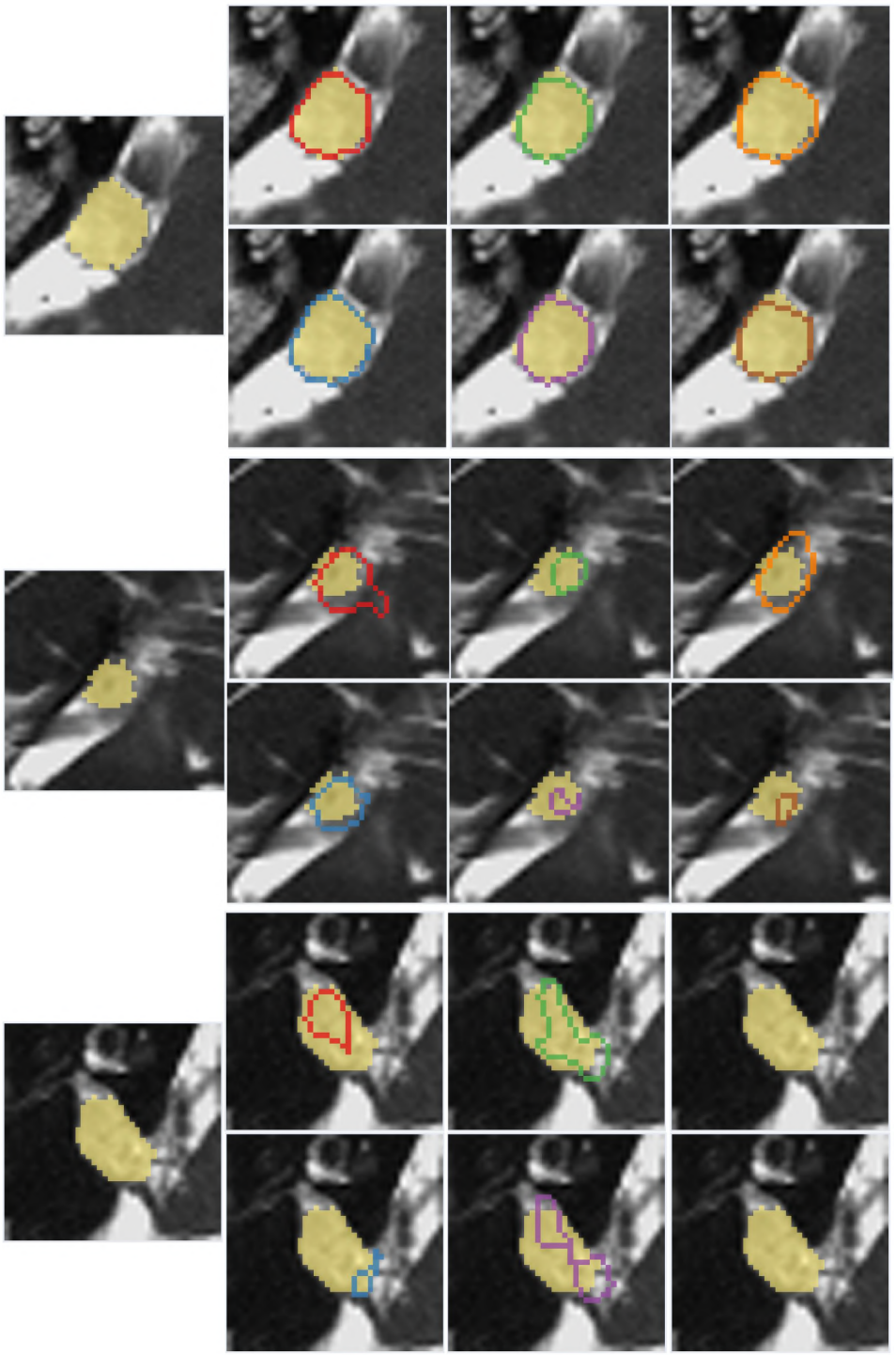


Figure 6.5: Examples of **X-Net T2** (red), **CG X-Net T2** (blue), $\mathbf{G}_{T2S}$**+X-Net T1** (green), $\mathbf{G}_{T2S}$**+CG X-Net T1** (purple), **X-Net S2T** (orange), and **CG X-Net S2T** (brown) with GT (yellow area). Best viewed in high resolution and color.

6.2 Visual Assessment

The high number of models and individual samples prevents a thorough investigation of all potential use cases in the scope of this thesis. Based on what AI engineers often look into, we have defined four use case scenarios in order to guide the visual assessment with our application. They demonstrate valuable insights about the behavior of our models and how this additional knowledge can drive further evaluation data analysis. The use cases are formulated as questions to be answered with the help of our tool in the following subsections. They have been defined empirically, based on our own experience with developing AI algorithms. The relation to tasks **T1**-**T5** defined in Section 5.2.1 is listed in parentheses per question.

- **Q1** How does the tumor size influence the segmentation performance? (related tasks: **T2**, **T3**, **T4**)
- **Q2** Are there other dataset characteristics related to the performance? (related tasks: **T2**, **T3**, **T4**, **T5**)
- **Q3** What is the difference between a standard model and its CG counterpart? (related tasks: **T1**, **T2**, **T3**, **T4**)
- **Q4** What are differences between all segmentation approaches? (related tasks: **T1**, **T2**, **T3**, **T5**)

A detailed documentation, by screenshots taken during the investigation, can be found in Appendix A.2. Here, we just document the most significant findings.

6.2.1 Tumor Size Analysis (Q1)

We limit the tumor size analysis to the T2 dedicated models. For an overview, the shape features are selected in the PSD (see Figure 6.6). The parallel axes are rearranged to group the performance measure and the tumor size related features, such as the mesh volume, the surface area and the maximal 3D diameter together. Inspecting the data by group affiliations reveals that bad results belong to small tumors. Medium results are mainly from small or medium-sized tumors, with some exceptions for large tumors. Tumor size in this context should be understood volumetrically. Next, we take a closer look at the 20 good results. Scanning through individual patient datasets shows a pattern in the performance heatmap for subjects as shown in Figure 6.7. The edge slices at the top and bottom, which are also the smallest in terms of pixel size, are consistently worse than the slices in the middle. Hovering over the subject PSD reinforces this discovery. Good results are always in the center of the volumetric tumor part, while intermediate results build two bands around the middle section.

Further data investigation shows two main observations that support the results of the visual assessment. First, the models have difficulties predicting especially small

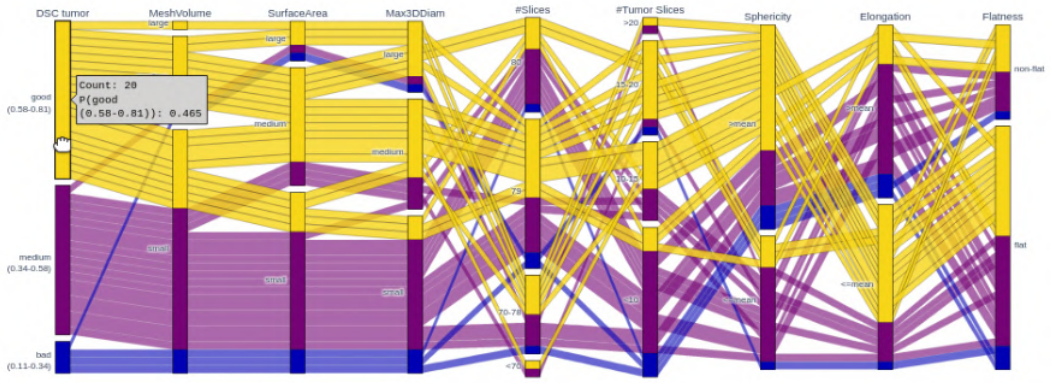


Figure 6.6: PSD with shape features for T2 dedicated methods and only tumor slices. The features are displayed per subject and the 20 subjects with good DSC values are highlighted.

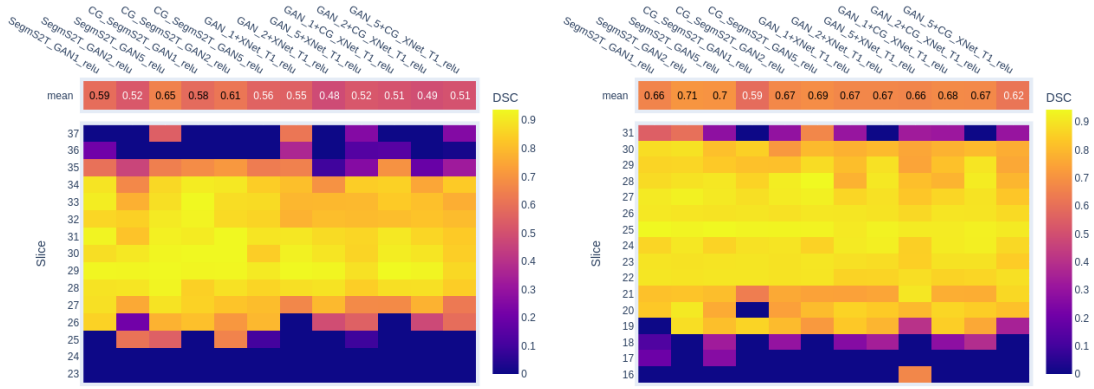


Figure 6.7: Performance heatmap of patient with ID 207 (left) and 241 (right). Both show a similar pattern of good results in the center and worsening results towards the edge slices at the top and bottom.

tumor occurrences. Second, removing segmentation masks with a size below a certain threshold from the evaluation increases the performances. For the test set **D4**, we remove segmentation masks with a tumor size below a certain threshold. As already mentioned in Section 4.1.1 and shown in Figure 4.2b, there are a lot of slices that depict only a small tumor with respect to pixel count. Figure 6.8 breaks down the distribution for the left range of the axis even further. The test set is determined by steps of ten between the tumor size of at least 0 to at least 100 pixels. The decomposition is used to illustrate how the performance depends on the tumor size. Figure 6.9 collects the histograms of DSC over tumor size for some selected models. It can be observed that for all methods, the performance, i.e., DSC score, increases when tested on filtered subsets. For the best specialized baseline method **X-Net T2 ReLU**, the DSC elevates from 0.71 to 0.85. For the domain adaptation approach performing the best, the DSC rises from 0.61 to 0.78

for **X-Net S2T** `dstep=5` and from 0.54 to 0.71 for **CG X-Net S2T** `dstep=2`. The same behavior can be observed for the other error metrics as well.

In conclusion, tumor size has a major impact on segmentation performance. **Slices with a small tumor, especially slices at the top and bottom of the volume, lead to poor results.** Sections in the center of the tumor volume are more likely to give good results because the number of pixels is higher.

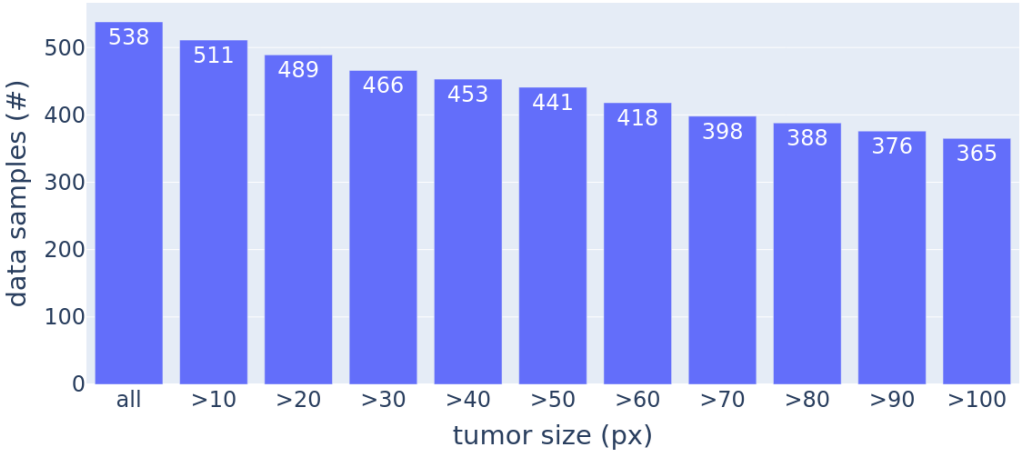
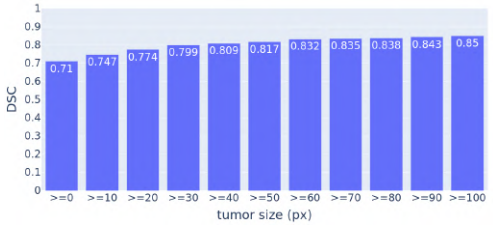
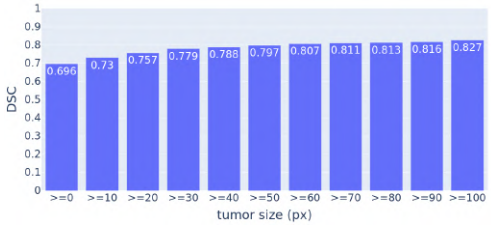


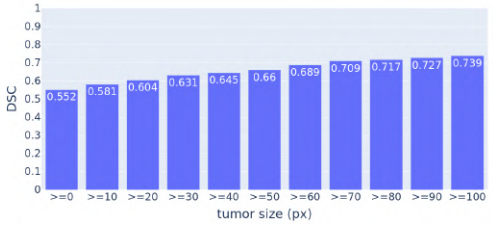
Figure 6.8: Tumor size distribution in test set.



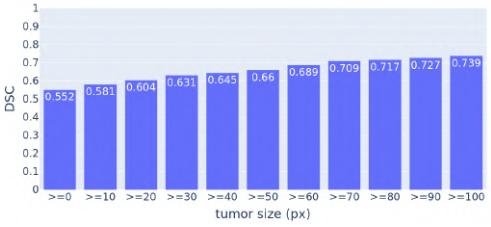
(a) **X-Net T2 ReLU**



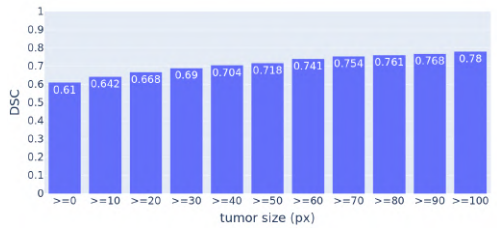
(b) **CG X-Net T2 ReLU**



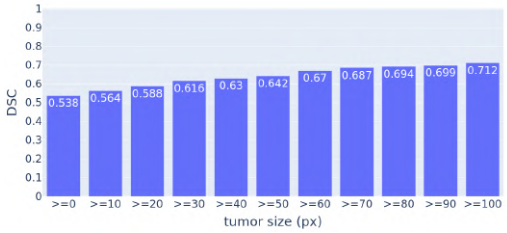
(c) $\mathbf{G_{T2S}}$**+X-Net T1** `dstep=2`



(d) $\mathbf{G_{T2S}}$**+CG X-Net T1** `dstep=2`



(e) **X-Net S2T** `dstep=5`

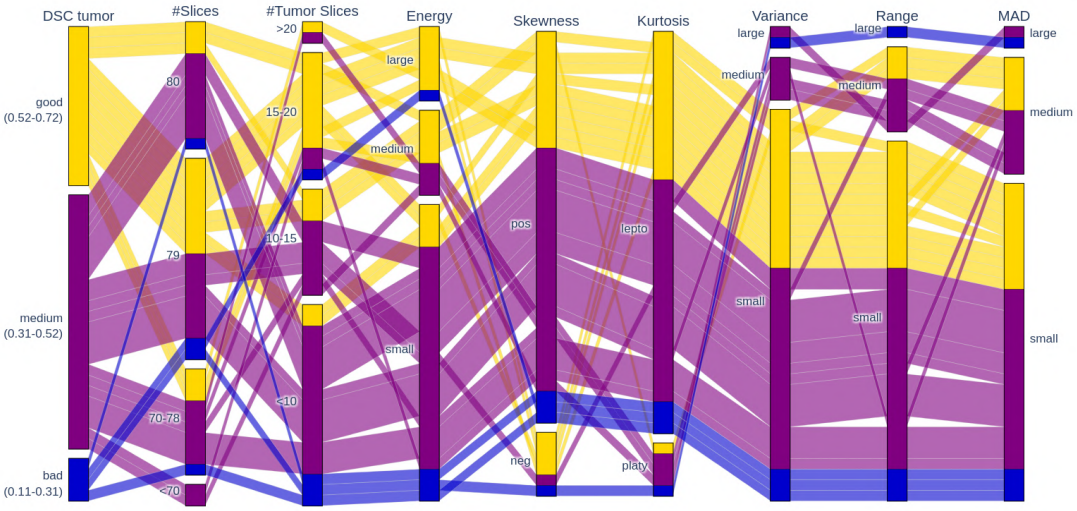


(f) **CG X-Net S2T** `dstep=2`

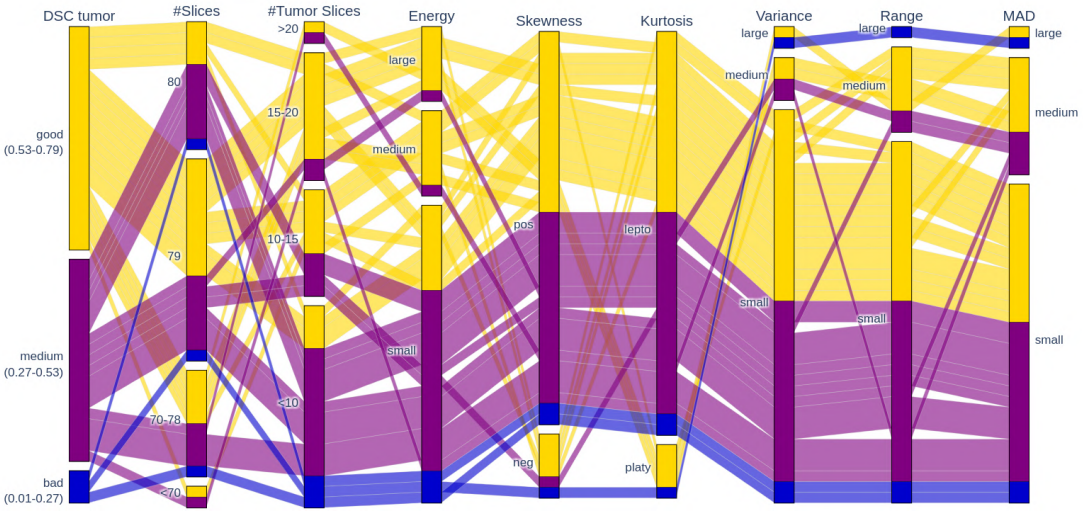
Figure 6.9: Tumor size analysis by plotting DSC over tumor size in pixels for selected models.

6.2.2 Subject Analysis (Q2)

In this subsection, we will take a look at the first-order features. The PSD and the association to feature groups changes depending on the model selection. Hence, the already prepared classes, like all, baseline, DA, and best, are used for the analysis. There are two general observations about the dataset characteristics. The PSD for all models and only for DA models is displayed in Figure 6.10. The intensity distribution in the dataset is mostly skewed to the left (positive) and leptokurtic (above 3). Whereas the variance, range, and MAD values are mostly small, just like the energy values. There is always one count for negative skewness and platykurtic kurtosis belonging to bad performance. Also one count for large range is consistently associated to bad performance. Other than that, we do not see any obvious pattern between the features and the patient IDs grouped by performance in the PSD. The blocks on the parallel axes are mixed performance groups.



(a) All models



(b) DA models

Figure 6.10: PSD with first-order features for (a) all models and (b) only DA methods.

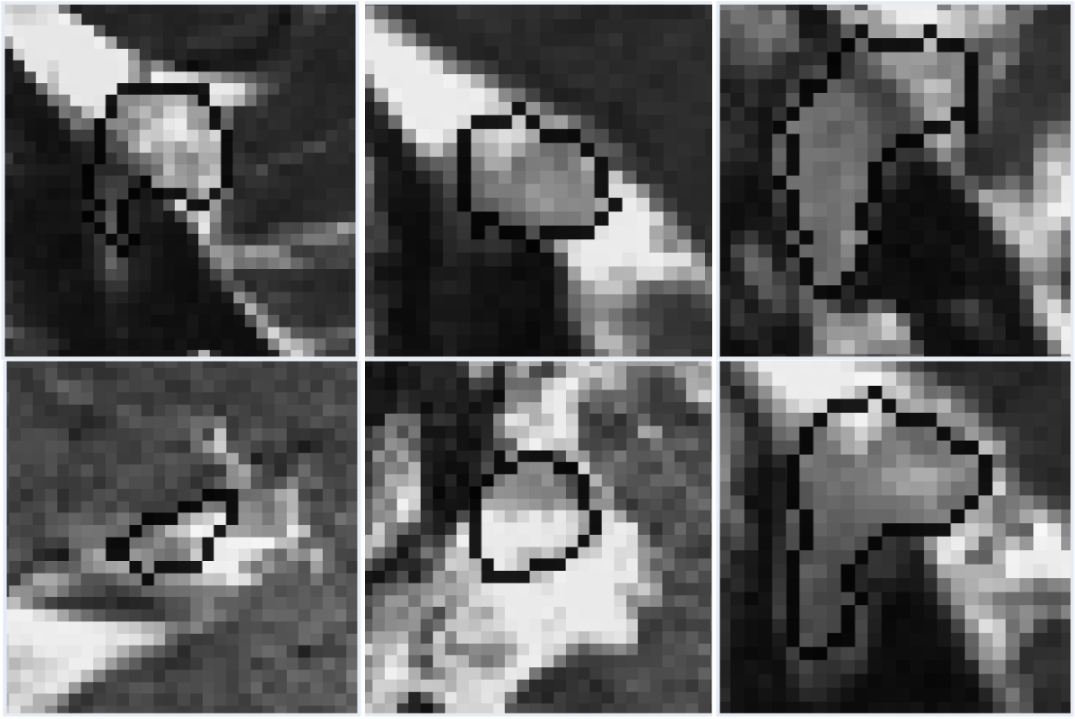
We visually inspected slices with large range or variance for two groups of patient IDs. The first group is patient IDs with bad performance measures averaged over all models. As shown in examples in Figure 6.11a, examples either do not have a clear tumor borders or are heterogeneous regions. The second group is a subject with patient IDs showing good overall performance measures averaged over all models. This group contains examples with a clear distinction from the surroundings and a homogeneous ROI (see Figure 6.11b).

The answer to **Q2** is that there are other correlations between dataset features and performance measures aside from the tumor size. However, they are not that obvious. Visual inspection shows that **homogeneous ROIs with distinct borders are easier to process by our algorithms.**

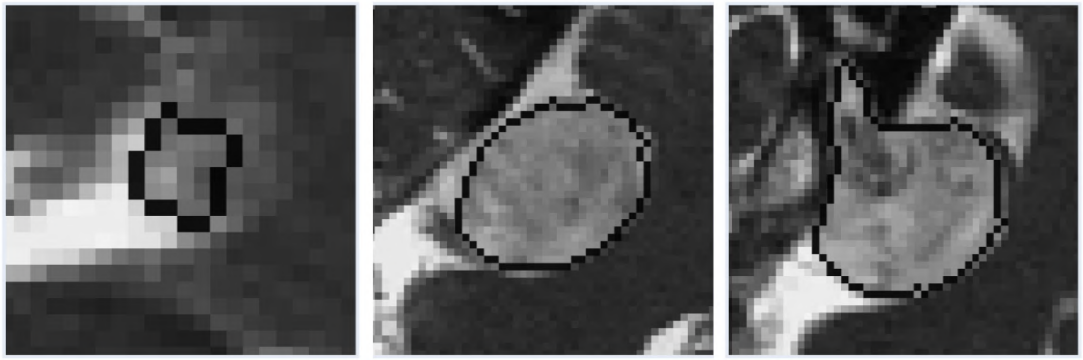
6.2.3 CG Module Analysis (Q3)

We will take a look at two different pairs of standard and CG module-enriched to find patterns in the result. First, we will look at the baseline models **X-Net T2 ReLU** and **CG X-Net T2 ReLU** as examples for fully supervised models. Then, the domain adaptation methods **X-Net S2T** `dstep=2` and **CG X-Net S2T** `dstep=2` (best model for **D1**) are analyzed. Both show the same patterns, which means that there is no difference whether the DA or the baseline are used for analysis.

We get an overview with the cohort visualization first only for layers containing a tumor, then for the whole dataset. Different performance heatmaps for **X-Net S2T** `dstep=2` and **CG X-Net S2T** `dstep=2` are collected in 6.12. For the representation of only tumor slices, the results are very mixed, without a clear pattern, but with a slight tendency for better results with the standard method (see Figure 6.12a). Including all



(a) Examples with low DSC values (bad).

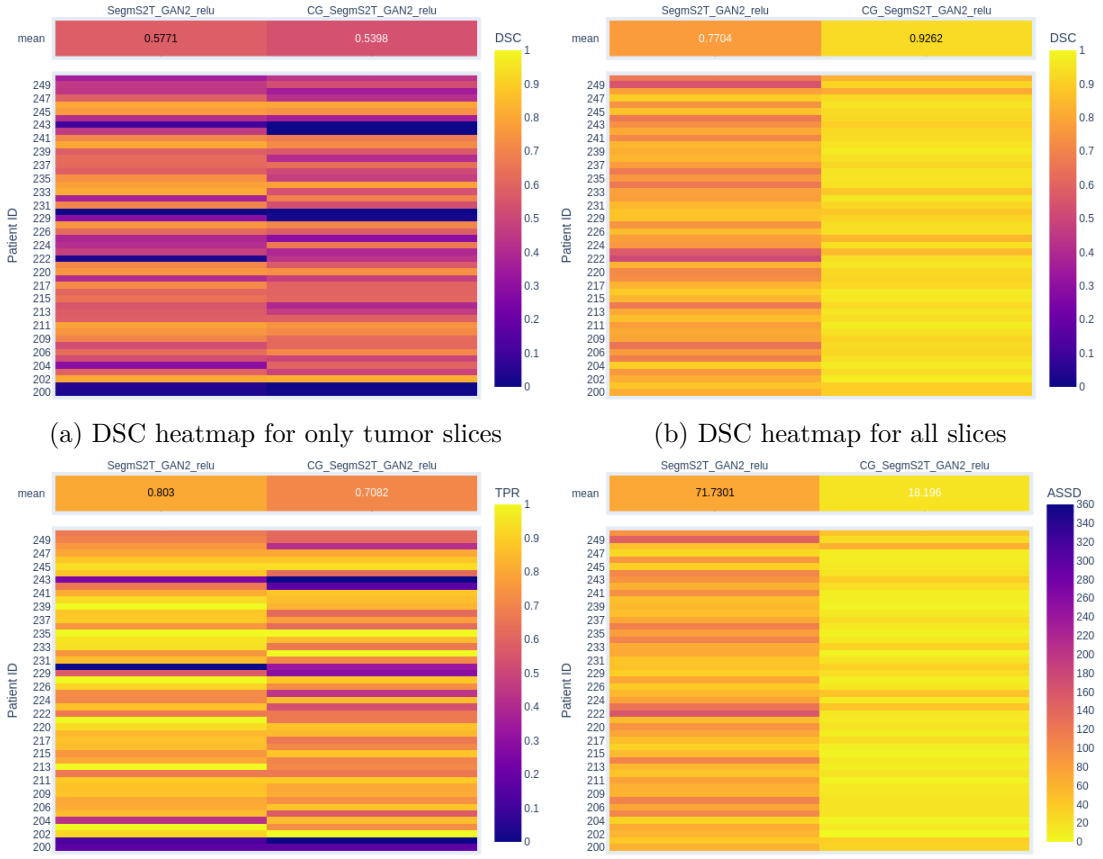


(b) Examples with high DSC values (good).

Figure 6.11: Examples of tumor ROIs: (a) bad performances for heterogeneous region without clear borders; (b) good performance for homogeneous ROI with clear borders.

slices shows better results for the CG version (see Figure 6.12b). Only the TPR does not show a major difference between standard and CG version since this performance measure is relevant for the tumor slices where the results are mixed (see Figure 6.12c). Extending the analysis from only tumor sections to all slices shows that the CG module is more beneficial overall. This conclusion is also supported when looking at the subject-based visualizations for random examples. The heatmaps per patient ID show more FP

predictions around the tumor volume for standard methods. Based on this findings, we formulate the hypothesis that the predictions of the CG models are more conservative, i.e., the regions are smaller.



(a) DSC heatmap for only tumor slices

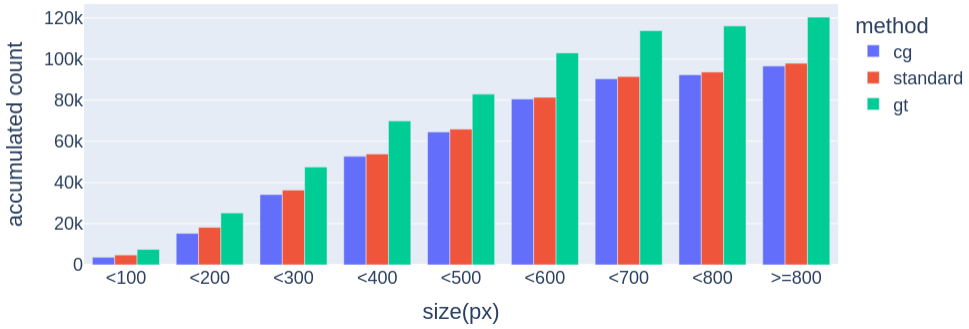
(b) DSC heatmap for all slices

(c) TPR heatmap for all slices

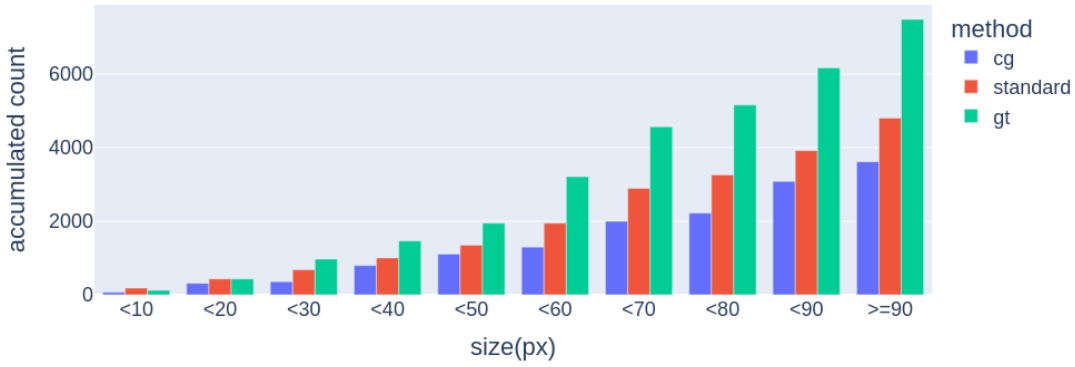
(d) ASSD heatmap for all slices

Figure 6.12: Performance heatmap for DSC, ASSD, and TPR values for **X-Net S2T** and **CG X-Net S2T**.

With these additional insights and assumptions, we return to a more detailed data analysis. The sum of the pixels in predictions with all tumor GT yields $97,897$ (**CG X-Net S2T** `dstep=2`) and $96,562$ (**X-Net S2T** `dstep=2`). The bar charts displaying the accumulated pixel count for different GT tumor sizes (px) in Figure 6.13 shows that both methods underestimate the mask size. However, the bar for CG method is consistently smaller than for the standard method, which especially shows if we zoom in on specific groups, such as all GT labels with size smaller than 100 (Figure 6.13b). This confirms our assumption that **CG X-Net S2T** `dstep=2` is overall more conservative than **X-Net S2T** `dstep=2`. For the non-tumor slices, this is the more favorable behavior. But for the tumor slices, this results in slightly higher error values.



(a) All GT labels.



(b) All GT labels with size smaller than or equal to 100.

Figure 6.13: Accumulated sum of pixels in segmentation for slices with different tumor sizes (px). The models are **CG X-Net S2T** `dstep=2` (cg) and **X-Net S2T** `dstep=2` (standard), together with the ground truth (gt).

In summary, the main difference between CG methods and their standard counterparts is **the lower number of false positive predictions in non-tumor slices**. This results in better overall performance with similar performance for tumor sections for the CG enhanced models.

6.2.4 Segmentation Approach Analysis (Q4)

The quantitative analysis we have already done in Section 6.1.3 is nicely underlined by the cohort heatmap. The **X-Net T1** models are excluded from the following analysis due to their bad performance.

First, the focus is on the three **X-Net T2** models with different activation functions. Making use of the heatmap slider and moving the maximal value towards 0 reveals

that for ReLU the columns get faster to yellow compared to the others. Similarly, the predominant color for Leaky ReLU and SeLU is blue for the minimal value going towards 1 (see Figure 6.14). This means that the activation function ReLU is slightly better, which agrees with the quantitative analysis. By visually selecting rows in a random manner and looking at the subject performance heatmap, we cannot recognize any particular patterns.

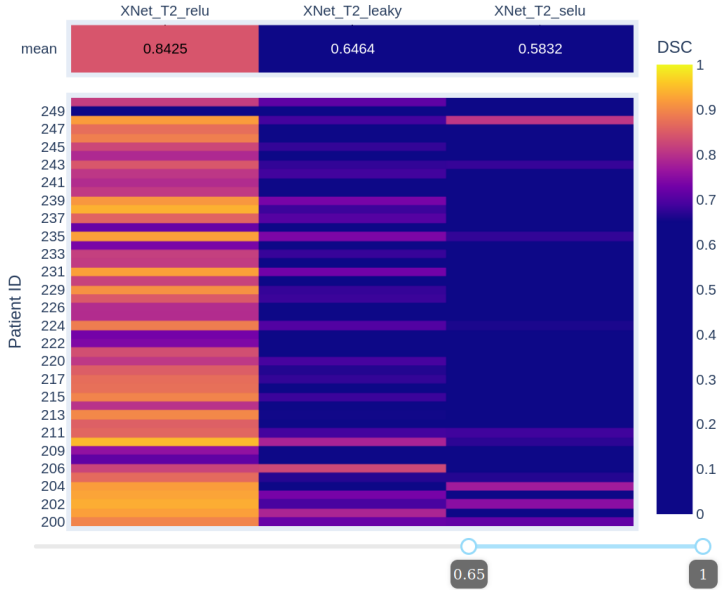


Figure 6.14: Performance heatmap of T2 baseline models with different activation function. The color map is changed by moving the minimal DSC value to 0.65. ReLU has higher DSC values than leaky ReLU and SeLU.

The individual DA methods are grouped into four approaches. This are **CG X-Net S2T**, **CG G_{T2S}+X-Net T1**, **X-Net S2T**, and **G_{T2S}+X-Net T1** with the same ordering as in Figure 6.15 from left to right. For the view with tumor slices, there are only weak visual clues in the heatmap, mainly in the row with the mean values. These changes for the full view where the ordering as described above is underlined with the cell colors, because it is the ranking of the approaches regarding DSC.

We made another observation for both cases during the visual assessment of the subject performance heatmap and the corresponding slice heatmaps. The segmentation predictions without overlap to GT are sometimes mirrored around the x-axis (see Figure 6.16), not in terms of shape but location. The cells are blue for the DSC representation, which means that the segmentation mask is not overlapping the GT mask. Changing the representation to ASSD shows a distance value between zero and the maximum value for some of these cells which is an indication for existing, but incorrect segmentation masks. There are examples of rows with such cells, that have at least one mask displayed on the opposite side in the prediction heatmap, as illustrated in Figure 6.16.

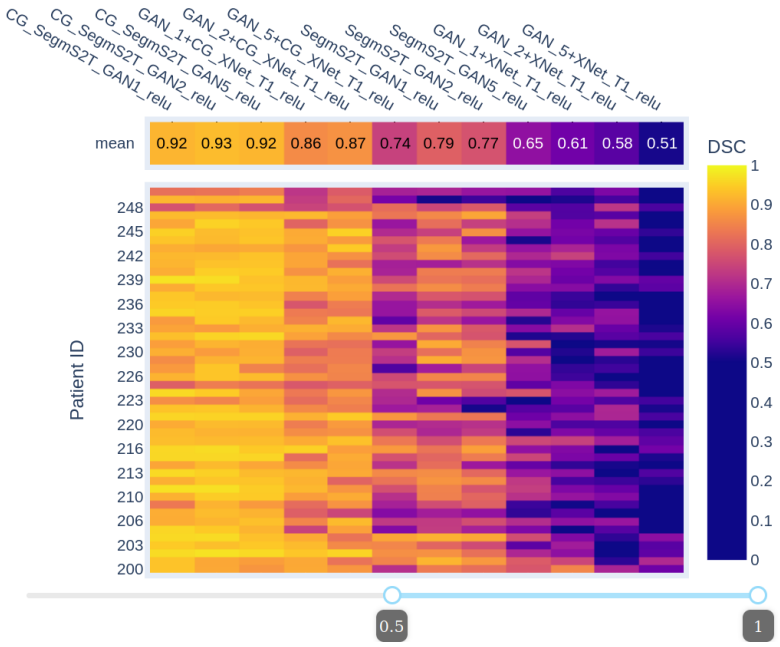


Figure 6.15: Performance heatmap of DA models sorted by the mean DSC values for all slices. The color map is changed by moving the minimal DSC value to 0.5.

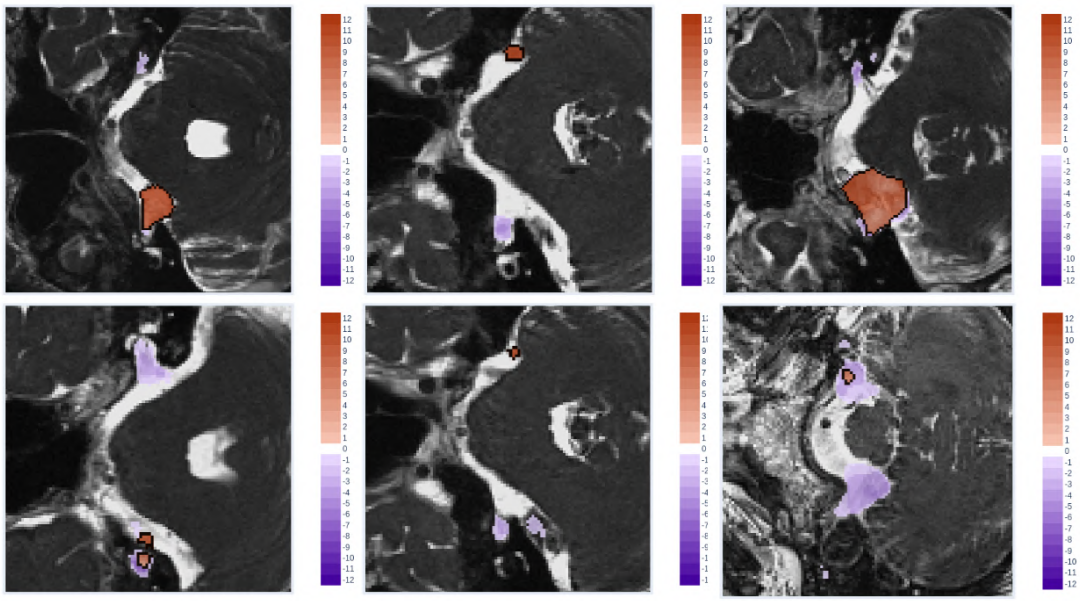


Figure 6.16: Examples for segmentation predictions mirrored around y-axis. The slice heatmap shows the subtraction encoding of all DA methods. Important are the blue pixels, i.e. there is no GT label, but at least one prediction. Best viewed in color and high resolution.

The comparison of the four different DA approaches results in an overall ranking of **CG X-Net S2T** and **CG G_{T2S}+X-Net T1** at the top, followed by **X-Net S2T** and **G_{T2S}+X-Net T1**. Another observation that cannot be assigned to any particular model is the occasional prediction to the "wrong" brain hemisphere.

6.2.5 Summary

As shown with the topics discussed above, we were able to gain additional insight into the performance of the models. Aside from knowing the overall performance metrics and comparing them as done in Section 6.1, we have a better understanding of the results and can make more targeted decisions for improvements.

For **Q1**, we make heavy use of the cohort visualization (**T2**) and the shape features related to size (**T4**) in order to filter the subjects. The main information is gained from the performance heatmaps (**T3**), i.e., especially slices with small segmentation masks and slices at the top and bottom of the tumor volume pose a problem to our methods. The consequence could be experimenting with strategies to teach the network to handle even small tumor incidences. Examples for this are oversampling small tumor masks, use a cut-out of the original image to increase the number of relevant pixels or applying a 3D approach to include additional volumetric information. The rows of the performance heatmaps (**T2**) and individual data samples (**T3**) are in focus of **Q2**. There seems to be a connection that is not visible with the PSD (**T4**), but with the heatmap plots per slice (**T5**). The ability of the tool to also view underlying data shows the visual difference between individual slices. These visual differences are apparently also picked up by the models when we cross-check this with the error values. The comparison of CG and standard methods in **Q3** is benefiting from the two dataset views. We already noticed the difference in performance when looking at the quantitative summary, but the visual assessment (**T1-T4**) showed that CG results are more conservative, i.e., the predicted segmentation masks are less in terms of pixel sum than GT. This could be mitigated by similar strategies as already mentioned for handling small tumor sizes. The focus of **Q4** is on the columns of the performance heatmaps (**T1-T3**). By checking out different performance metrics, we spot examples of segmentation predictions (**T5**) that do not have any overlap with the GT labels and are on the opposite brain hemisphere as the tumor. This could be prevented by taking advantage of the symmetry of the human body into account and use a training set that only contains original and flipped one-sided brain hemispheres. By answering the questions, weak points of algorithms are identified and can be targeted. The goal is not to provide an explanation why the algorithms work the way they do or to shed light on the inner workings of individual layers. The question is rather what does and does not work.

CHAPTER 7

Conclusion

In this chapter we conclude this thesis by summarizing our work. We address and answer our research questions by summarizing our methodological approaches and the results obtained. After a comparison with the current state-of-the-art methods, our methodological approach for the automatic tumor segmentation is compared with solutions submitted to the MICCAI challenge *crossMoDA* and our interactive visual analysis tool is compared with already existing VA applications. Finally, we discuss limitations of our implementation and potential future work.

7.1 Summary

This thesis deals with the task of automatic VS segmentation on hrT2 images under the constraint that only annotated ceT1 images and unannotated hrT2 images are available for the training phase. This introduces the problem of data shift, where the image modality between source (T1) and target (T2) data have different distributions. After developing automatic segmentation algorithms, deep learning engineers and domain experts often use single numerical values for the evaluation. In order to support the model evaluation process, visual assessment of the results is a beneficial tool to investigate and understand complex model behavior. In this section, we summarize our two-fold contribution by revisiting the research questions we set out to answer with our work.

Research Question 1: *How can we generate brain tumor segmentations automatically for cross-modal data under the assumption that no labeled data for the target domain are available for training?*
We developed **two domain adaptation frameworks** utilizing image alignment with *CycleGAN* followed by segmentation with a *UNet*-based network. On the one hand, we convert hrT2 images to ceT1 images and apply a T1-specific trained segmentation network to the synthetic T1 images. On the other hand, synthetic hrT2 images are generated from real ceT1 images and a segmentation network is trained with transferred

T1 labels in a supervised manner. Comparing the two approaches, the latter leads to better results. One reason is that image synthesis from T1 to T2 works better than vice versa because of the image contrasts. Another approach that utilizes not only image but also feature alignment is based on SIFA. It is documented as suggestion for further implementations. The main technical contribution is that the approaches are **enhanced with a classification-guided module**, which allows simultaneous training of semantic segmentation and classification of tumor presence. This extension shows its advantage when we evaluate the model performance on the entire test dataset containing slices with and without tumor presence. The effect of global over-segmentation, i.e., the prediction of a non-zero tumor segmentation mask on images without a tumor, is reduced by the inclusion of the CG module. In addition, we also trained two baseline models to compare our unsupervised domain adaptation approaches. First, a model using annotated T2 images is trained to set the upper boundary that is achieved by a fully supervised solution. Second, annotated T1 images are used to train a model as a lower bound, without considering the domain shift. Our DA models outperform the base method without DA, but do not achieve the performance of the T2 specialized baseline methods.

Research Question 2: *How can we visualize the outcomes of automatic segmentation methods to support software developers and artificial intelligence (AI) engineers in evaluating their developed models?*
We designed and implemented a **web-based visual analysis application that allows interactive visual assessment of the model performances and prediction**, and the exploration of the relationship to dataset characteristics. The interface contains several visualization techniques that display the information on three different levels of detail following the visualization mantra. First, a summary is provided, consisting of the performance measurements per algorithms averaged over the entire dataset. Then, a cohort visualization (entire dataset in overview) is presented by PSD and a heatmap of model performance values. Next, the PSD and performance heatmaps are used for a subject visualization (specific patient ID in detail on demand). Finally, a sum and subtraction encoding of the fused segmentation mask prediction is provided per slice. The main contribution of our VA application is the **flexible comparison of multiple segmentation algorithms** to solve the five tasks of overall, per-patient and per-slice performance comparison, as well as relationship of performance to features and anatomy-based predictions. We examined the model results and their relationship to the dataset using four empirically formulated use case scenarios. We found a strong correlation between tumor size and a weak correlation between the intensity values in tumor ROIs and the performance values. Slices with small tumor sizes, such as the upper and lower sections of the tumor volume, and tumors with non-homogeneous ROIs without clear boundaries tend to be poorly segmented. CG enhanced models produce less FP predictions, resulting in an overall better performance score. However, the quality on tumor slices is comparable to that of models without CG modules. Visual inspection of the prediction heatmaps revealed a shortcoming that could not be attributed to any particular method. Some predictions are in the wrong hemisphere of the brain, mirrored

around the x-axis on the opposite brain structure.

7.2 Comparison to State-of-the-Art Methods

With regard to our first goal, the same deep learning task that was solved in our work, i.e., VS segmentation on T2 images, was also part of the MICCAI challenge *crossMoDA*. The competition was not yet completed during the conduction of this thesis. The teams that submitted their solutions had a smaller subset of training data than what was uploaded to TCIA and used by us. A direct comparison is therefore not possible due to the different initial data. Nevertheless, we would like to compare the methodological ideas of the best methods with ours. The strategies are overall the same. Image alignment with *CycleGAN* or its extension *NiceGAN* is followed by segmentation with *UNet*-based networks. Where MICCAI solutions transfer T1 scans to look like T2 scans, we tried both, synthetic T1 for inference and synthetic T2 images for supervised segmentation training. The approaches using synthetic T2 scans for training lead to better results, which is also the option chosen by the competition teams. However, the top MICCAI submissions are a bit more advanced with the use of self-training, where pseudo-labels are generated and added to the training dataset. This is what distinguishes the first two methods from the other top 10, among others. To the best of our knowledge, no segmentation network in the context of domain adaptation has been trained with a classification-guided module facilitating multi-task learning of classification and segmentation so far.

With regard to our second goal, current VA approaches do not address the visual analysis of the results of multiple algorithms in the same, flexible manner as we do. Former approaches are mostly limited to single or pairwise result representation. Our method allows a comparison of several segmentation masks at the same time. Most state-of-the-art tools consider the comparison on several levels, namely overall, per patient, and per slice, or alternatively per triangular mesh. Since they are not designed for deep-learning methods, they do not include anatomical-based predictions. Moreover, they are specialized in the presentation of segmentation results and mask properties, so that no additional information, such as data characteristics or other image-derived features (e.g., from radiomics), is taken into account. Our application bridges this gap. We incorporate not only the segmentation results and their performance measures, but radiomic features in the analysis also.

7.3 Limitations & Future Work

In order to compare our approach to other existing state of the art methods, a more extensive benchmarking is necessary. In this context, the CG SIFA method can also be revisited. One approach would be to train the framework with higher capacity, where the original *SIFA* implementation with an increased network depth can be used. Looking at the results of the MICCAI *crossMoDA* challenge and analyzing the submitted approaches, we see a lot of potential to improve our DA segmentation models in future work. There

is the possibility to combine our classification-guided module with concepts used by the winning team. Guiding the image translation by *CycleGAN* with an additional segmentation input for the data where annotations are available during training, i.e., real T1 and fake T2 images, may improve the tumor reconstruction. Then, the segmentation loss can focus the training on this region when the loss weight is high. The second concept that can improve our approach is self-training. This means that we generate pseudo-labels on real T2 images using our current approaches and add them to the already existing synthetic T2 images with transferred T1 images. The segmentation component is trained with the merged dataset. This procedure of generating and improving pseudo-labels can be carried out in multiple iterations with the goal of stabilizing the weak labels further and further. In addition, dataset specific pre-processing, such as cropping the image around the center with a large enough margin to always include the tumor regions, increases the task-relevant pixel count which should help the training. However, this modification is very specific to the problem and not a generally applicable methodology.

A possible extension to the visualization application is a 3D view of the tumor segmentation heatmap. A three dimensional illustration can provide additional spatial context, but is also challenging in terms of occlusion and visibility. Here, smart visibility approaches should be used as discussed by Viola and Gröller [129]. Nevertheless, the value should be assessed prior to implementation. Another extension of the parallel set diagram is the interaction option of multi-selections. For our scenarios, we did not see an additional advantage or specific need for multiple selection support. However, other use cases or deep learning tasks might benefit from a more complex selection mechanism. The goal of the visualization could also be extended to explainable AI applications (XAI). Here, insights into the individual layers of the developed networks are provided to explain the reason for a prediction, such as Grad-CAM by Selvaraju et al. [113] that generates salience maps for classification networks. Further investigation of appropriate radiomics features and definitions could be beneficial, since first-order features and performance values do not seem closely related. One reason for this could be that the tumor center has less influence compared to the tumor borders. A different definition of ROI, such as a region around the tumor contour, could bring new insights and show stronger connections. For improvements in the visualization task, the target users need to be involved extensively. The application is currently empirically designed and based on our own experience in the development of deep learning algorithms. A user study for our visual analysis tool should be conducted to gain more feedback about potential issues and design flaws. It would also show what the main benefit for deep learning engineers is.

This thesis is an initial positive step towards using cross-modal domain adaptation for the segmentation of brain tumors and VA approaches for the flexible assessment of the segmentation outcomes.

APPENDIX A

Appendix

A.1 Network architecture plots

In this subsection of the Appendix, we provide additional information for Chapter 4. The architectures are visualized with blocks representing the tensors. The numbers corresponds to width, height and depth (channels, filter size) of the tensor shape. The arrows represent the neural network layers and processing methods. The notation used in the model architecture plots is summarizes in Table A.1.

Abbreviation	Description
Conv2D	Convolutional Layer 2D
BN	Batch Normalization
IN	Instance Normalization
ReLU	ReLU activation function
Leaky ReLU	Leaky ReLU activation function
tanh	tangens hyperbolicus activation function
sigmoid	sigmoid activation function
MaxPool	Maximum Pooling 2D
GlobalMaxPool	Global Maximum Pooling 2D
Conv2DTrans	Transposed Convolutional Layer 2D

Table A.1: Overview of notation used for model architecture plots.

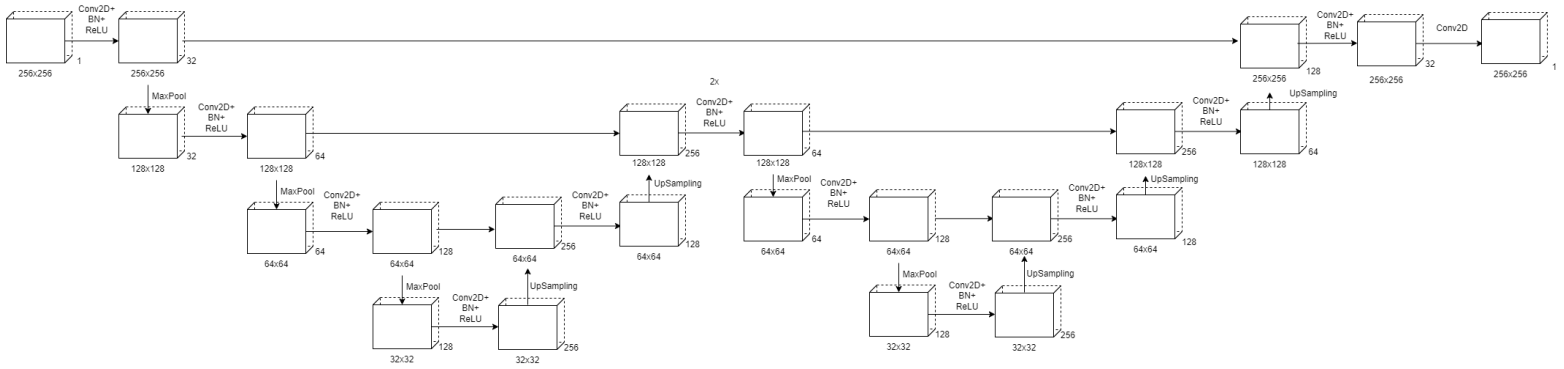


Figure A.1: *X-Net* architecture with two encoder-decoder parts.

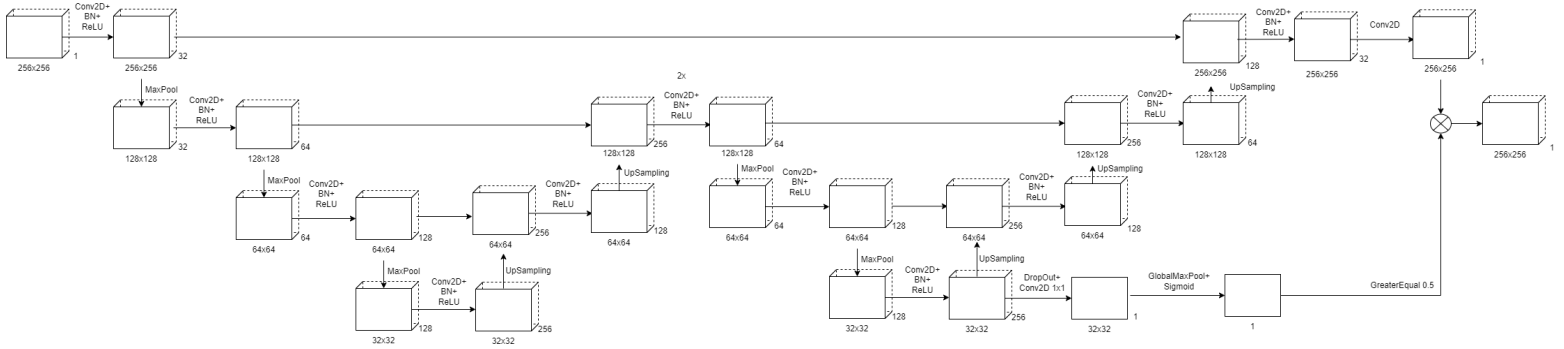


Figure A.2: *X-Net* architecture with integrated classification-guided module.

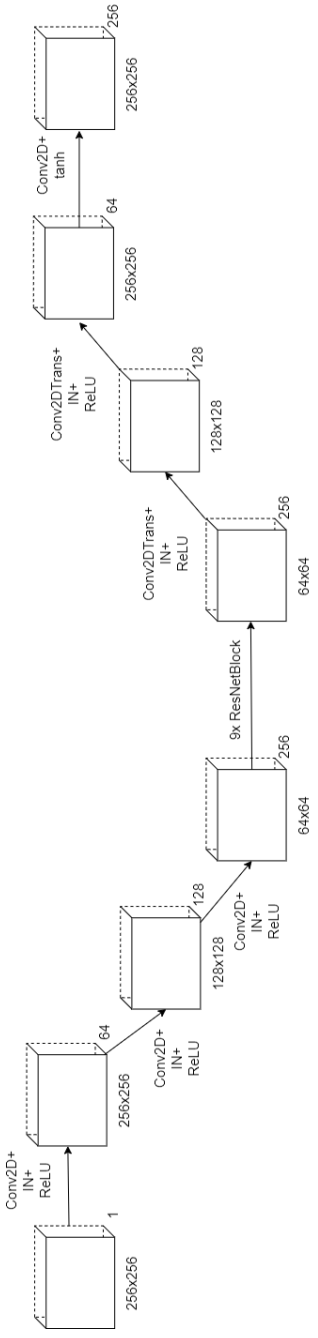


Figure A.3: ResNet generator for *CycleGAN* training.

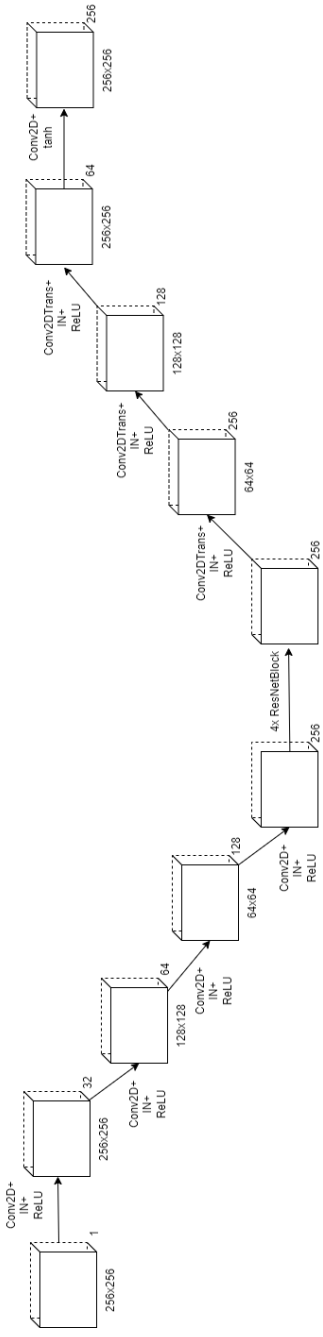


Figure A.4: ResNet generator for *SIFA* training. This network version is smaller than the *CycleGAN* version.

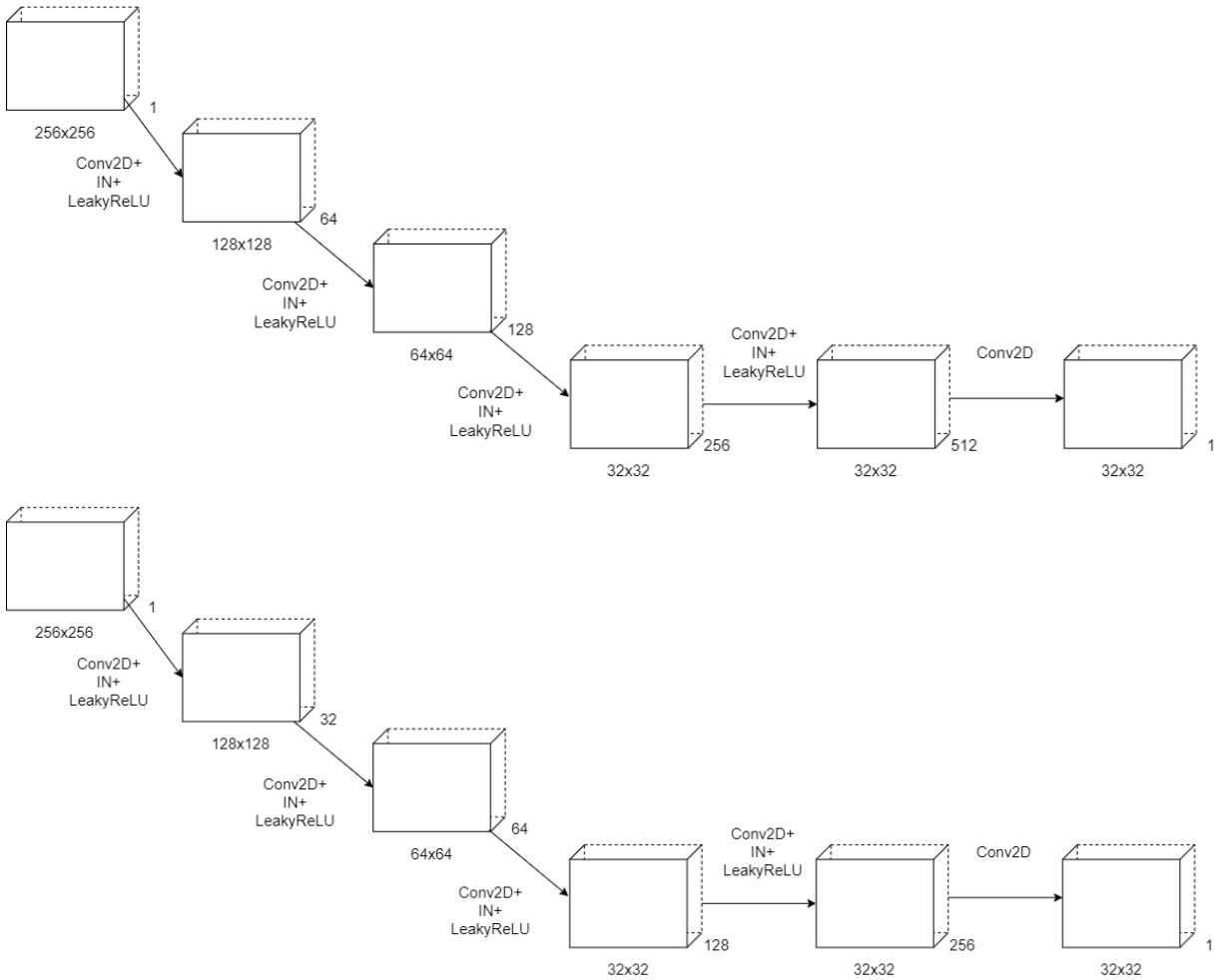


Figure A.5: Two versions of the discriminator used for *CycleGAN* and *SIFA* training with different filter depth size.

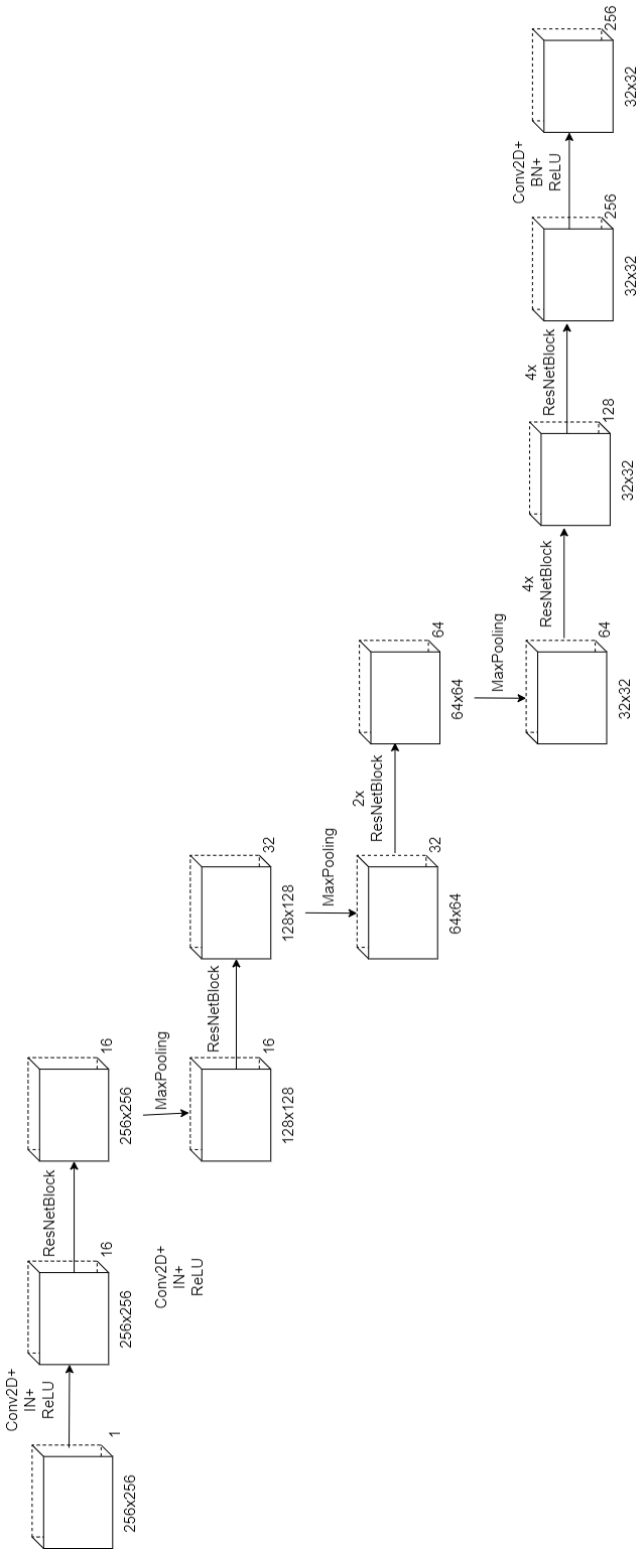


Figure A.6: Shared encoder for *SIFA* training.

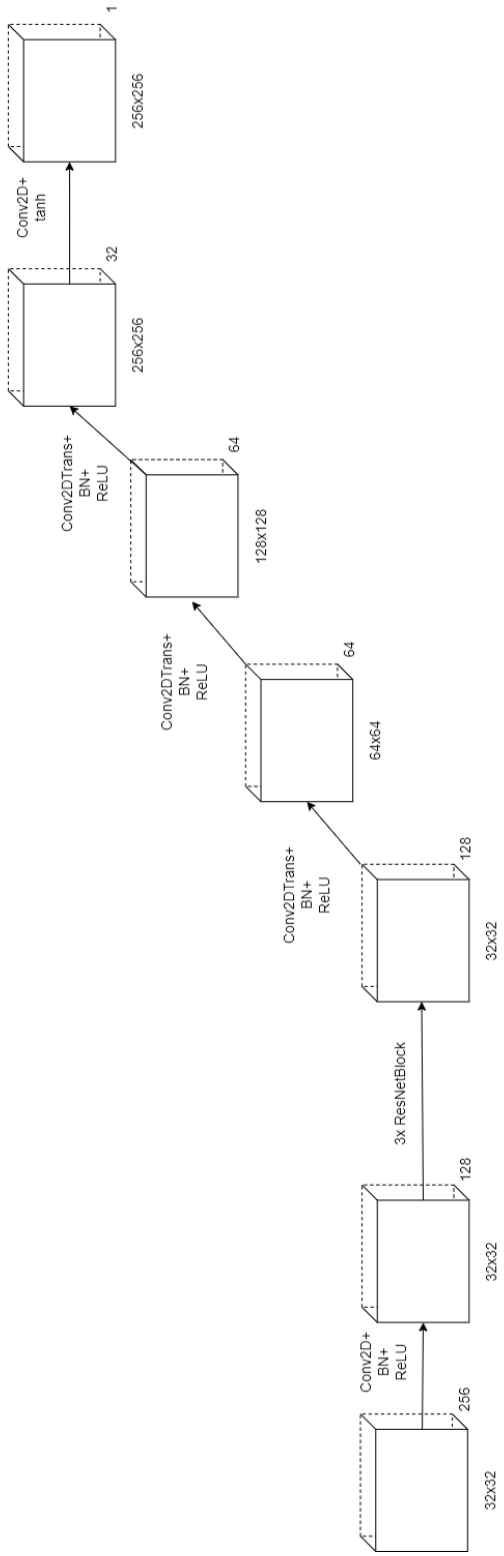


Figure A.7: Decoder for *SIFA* training.

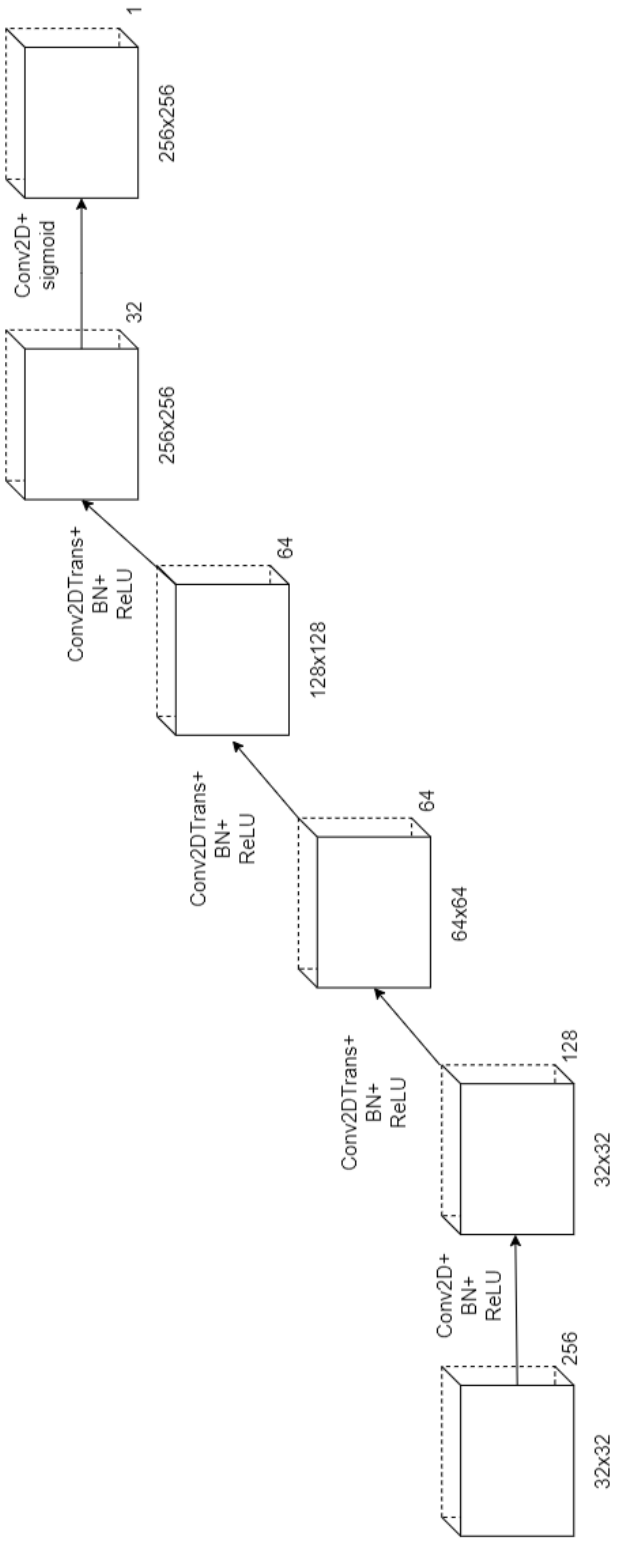


Figure A.8: Segmentation branch for *SIFA* training.

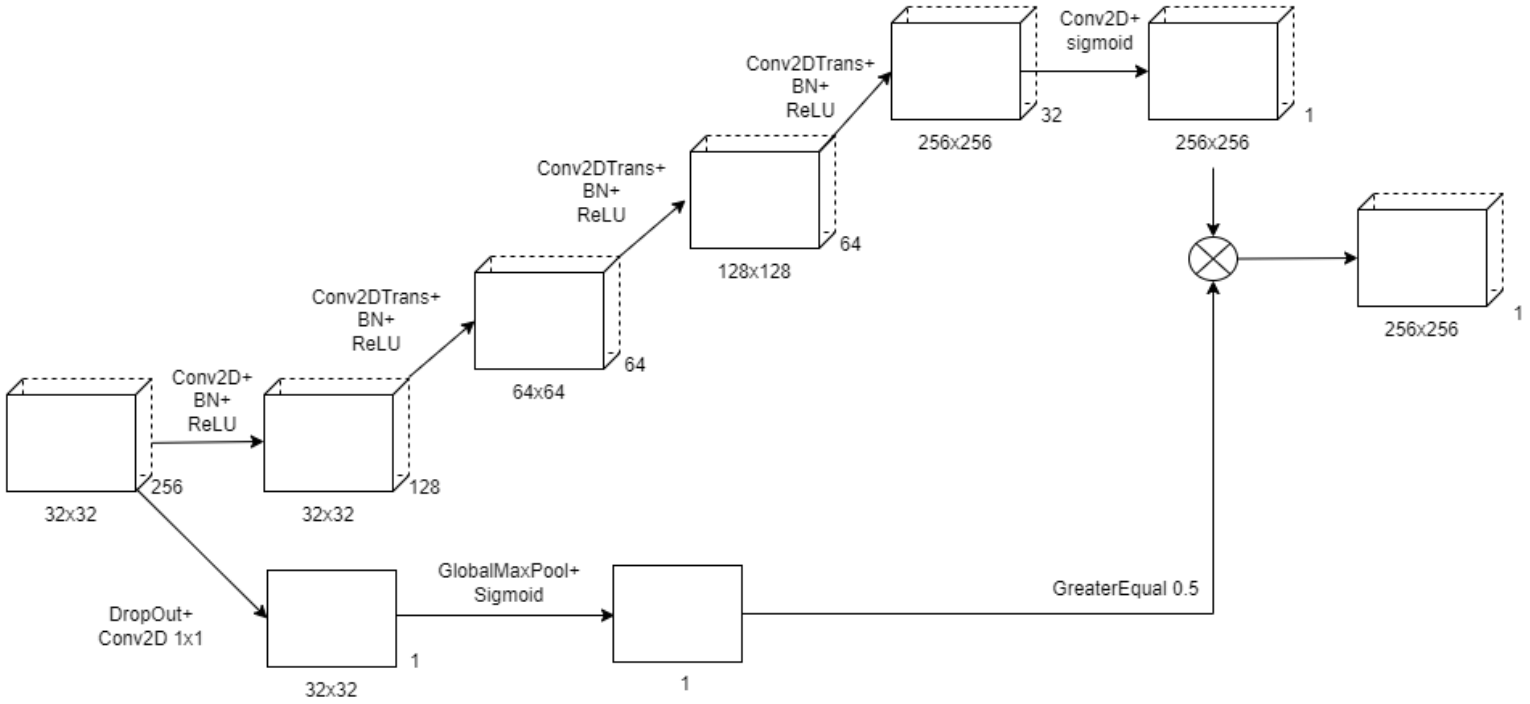
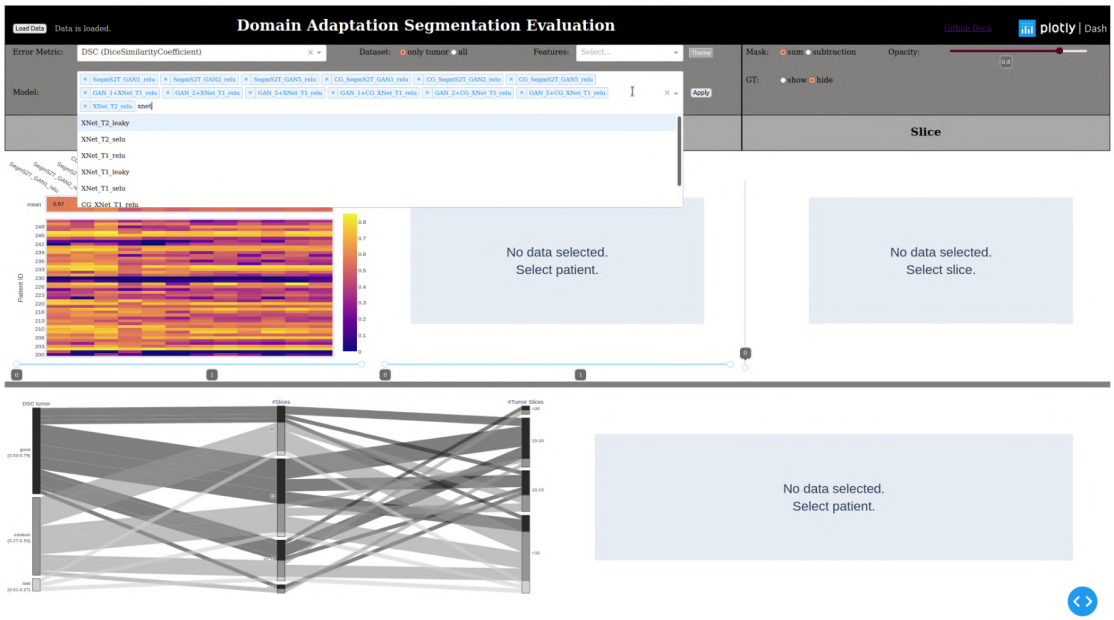


Figure A.9: Segmentation branch for *SIFA* training with classification-guided module.

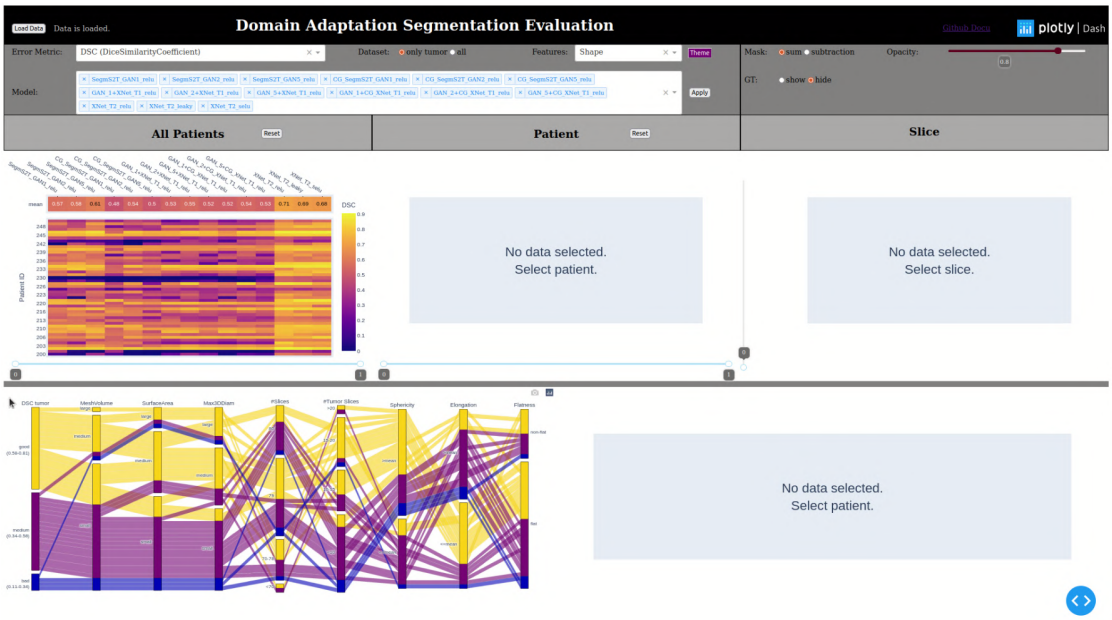
A.2 Step-by-Step Scenarios

In the second part of the Appendix, we showcase our application with screenshots to answer the questions from Section 6.2. The step-by-step explanation is reduced to a manageable number of figures. If several patient or slice IDs were analyzed, the figure caption contains an explanation.

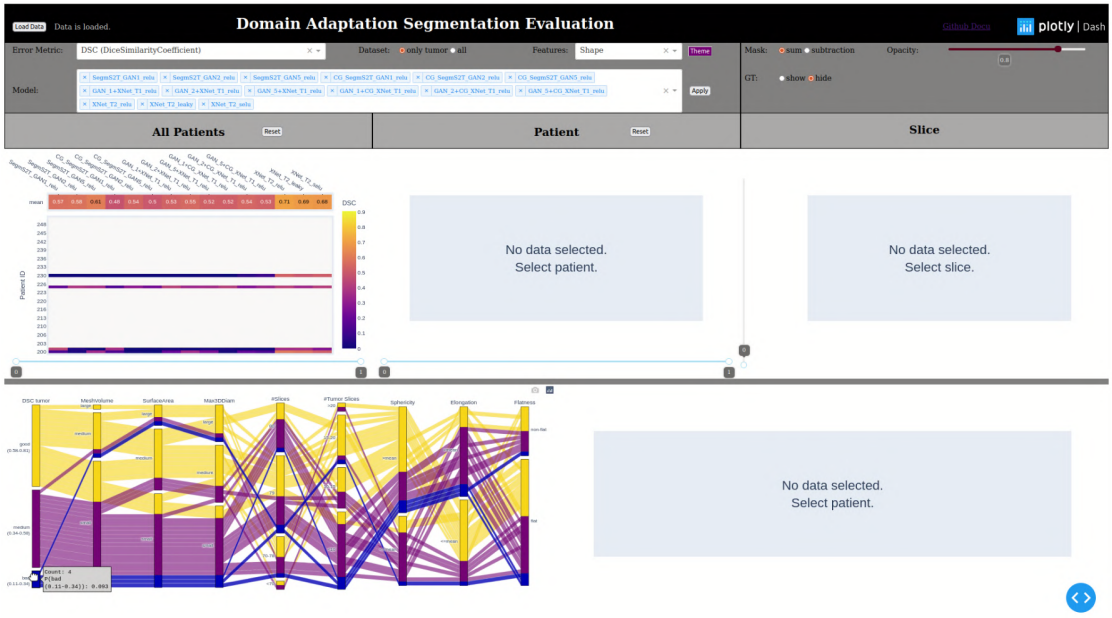
A.2.1 Tumor Size Analysis (Q1)



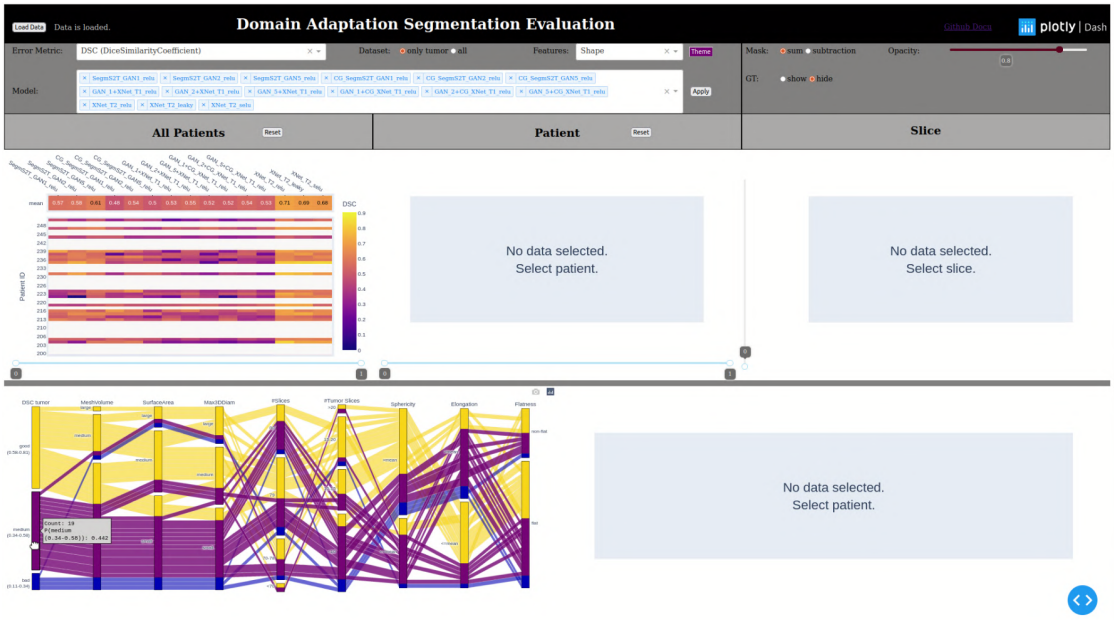
(a) First, all T2 methods are selected in the drop-down menu.



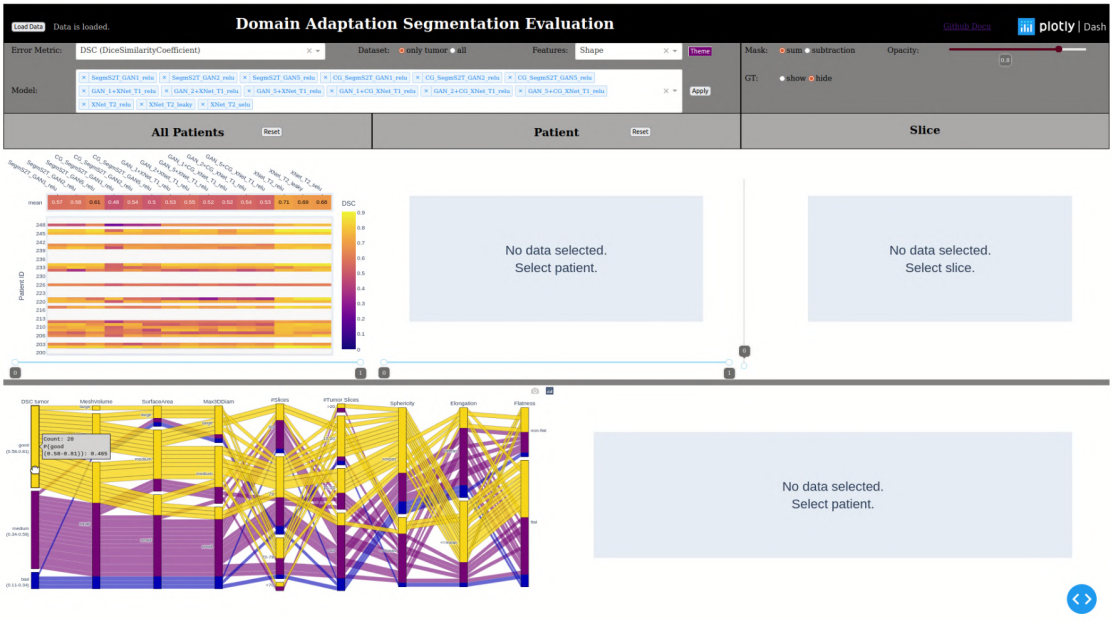
(b) Then, the theme is changed, shape features are selected and the parallel axes are rearranged.



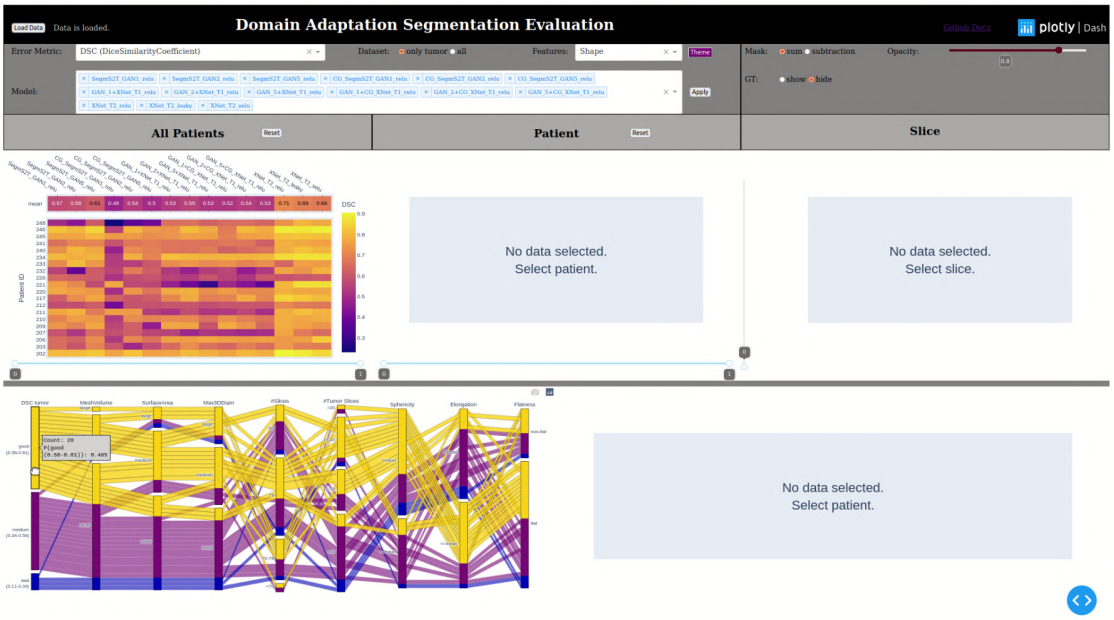
(c) We select the patient ID rows in the PSD which are associated with bad DSC values.



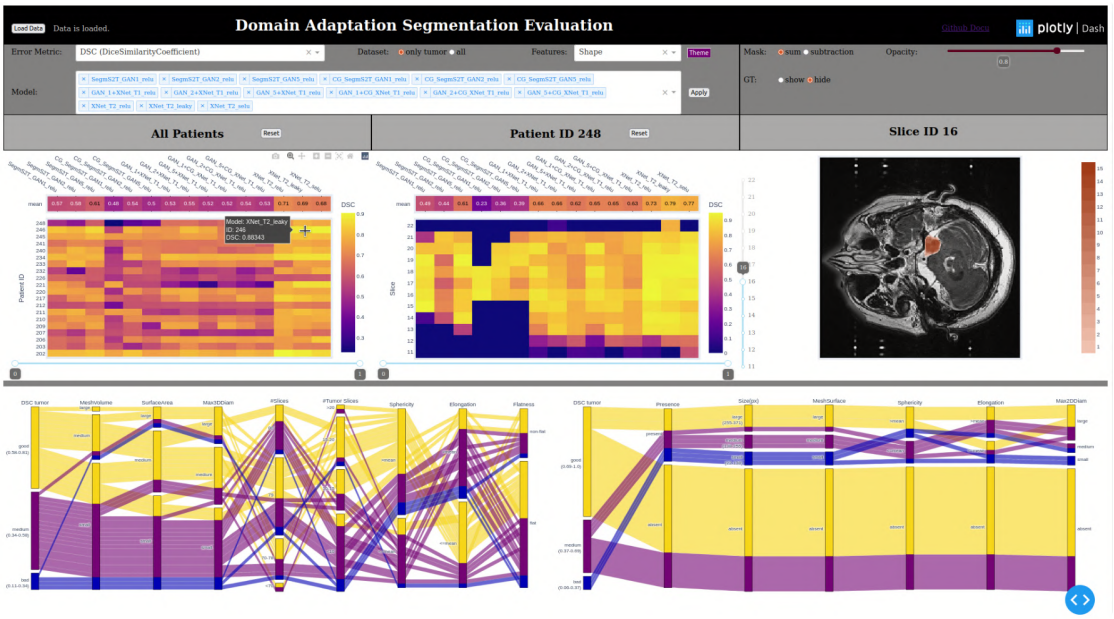
(d) We select the patient ID rows in the PSD which are associated with intermediate DSC values.



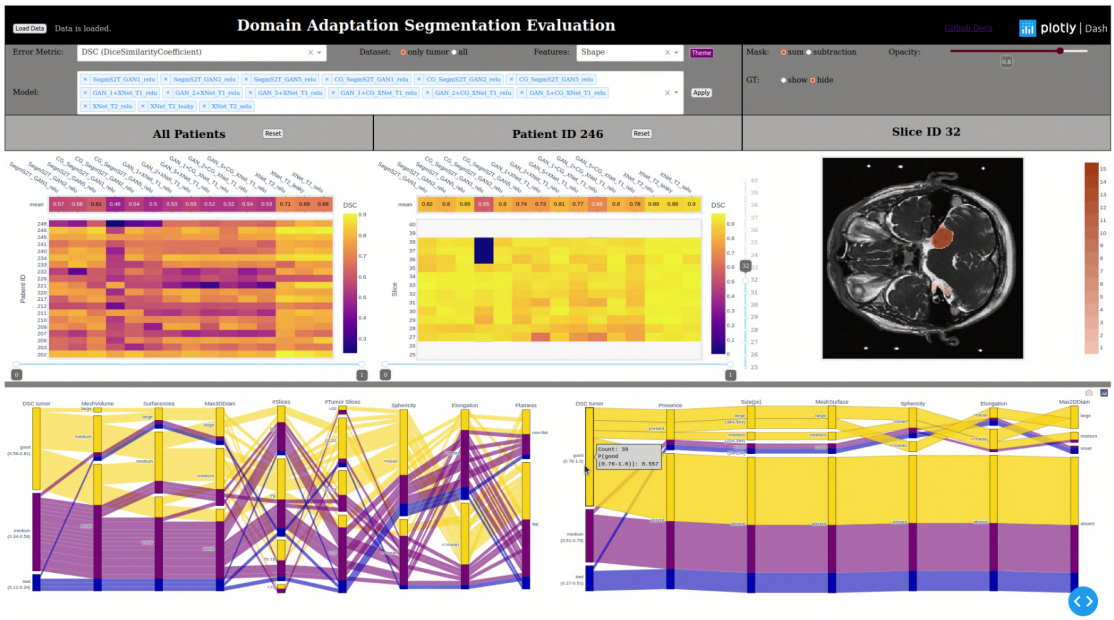
(e) We select the patient ID rows in the PSD which are associated with good DSC values.



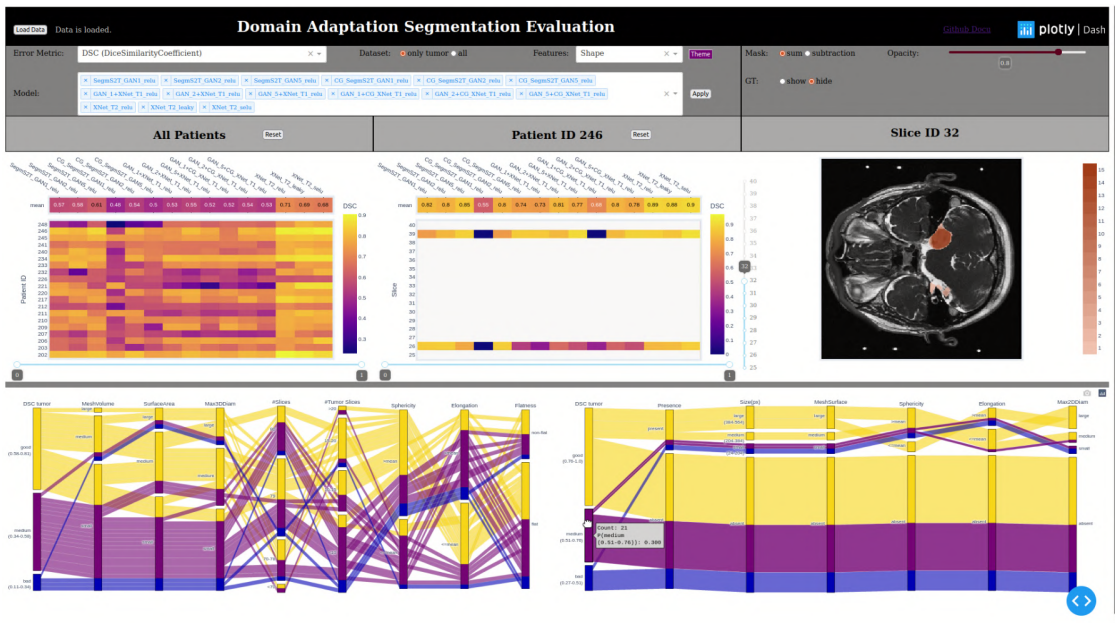
(f) The rows with the good patient IDs are selected and used as filter in the cohort performance heatmap.



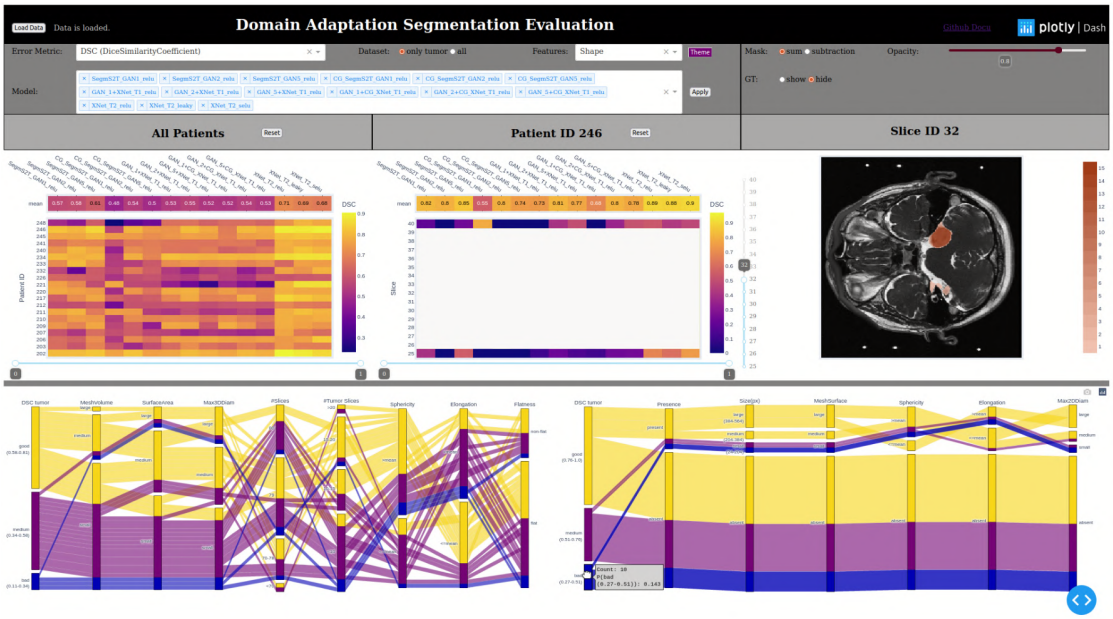
(g) The patient ID 246 is selected as an example. The characteristic pattern in the subject performance heatmap is visible.



(h) The slice rows are inspected in the PSD by hovering over good DSC values per slice.



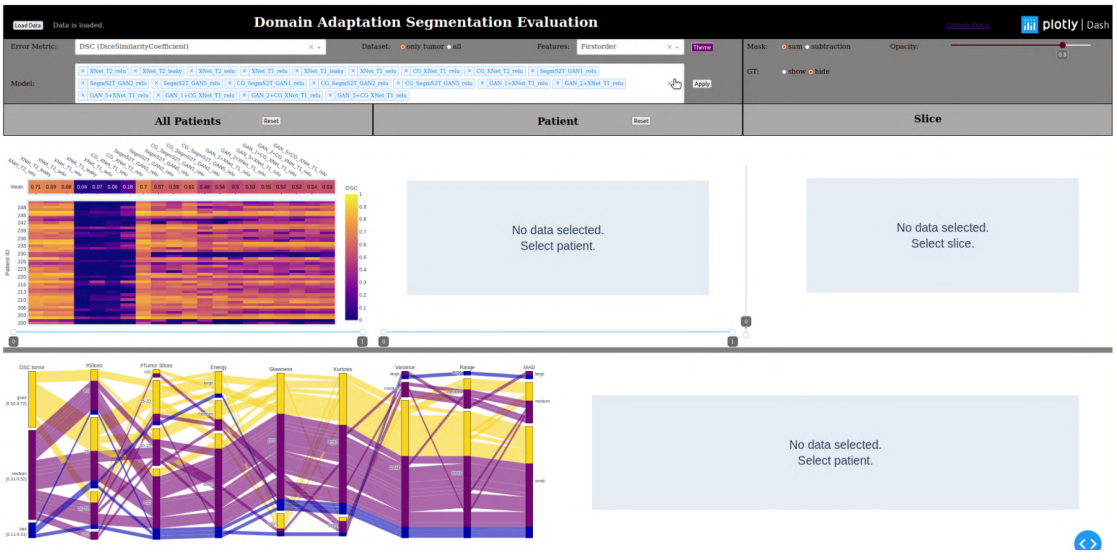
(i) The slice rows are inspected in the PSD by hovering over intermediate DSC values per slice.



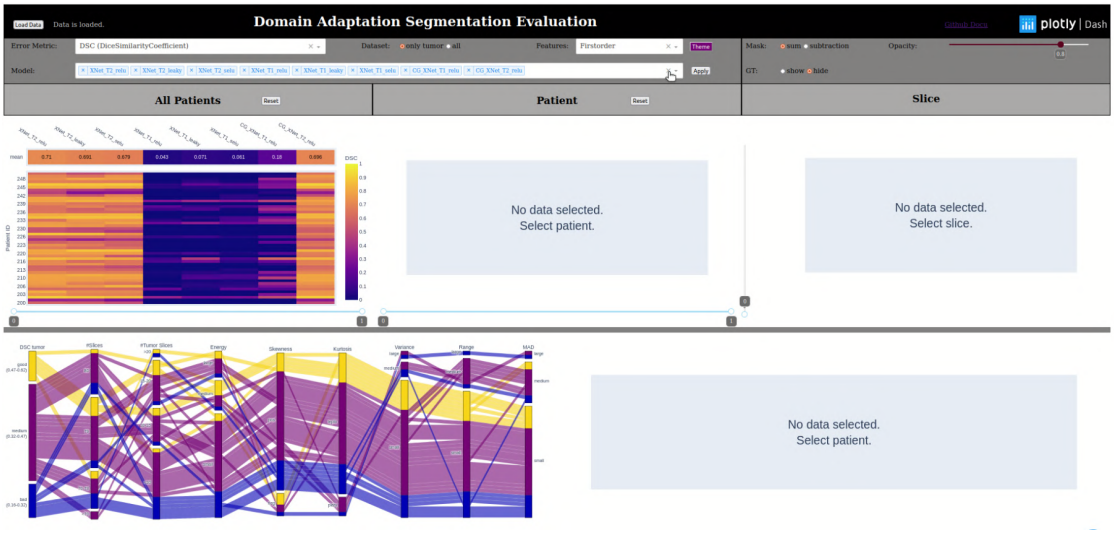
(j) The slice rows are inspected in the PSD by hovering over bad DSC values per slice.

Figure A.10: Workflow for **Q1** to investigate the tumor size relationship to performance. Steps (g)-(j) are repeated for multiple examples (patient and slice IDs).

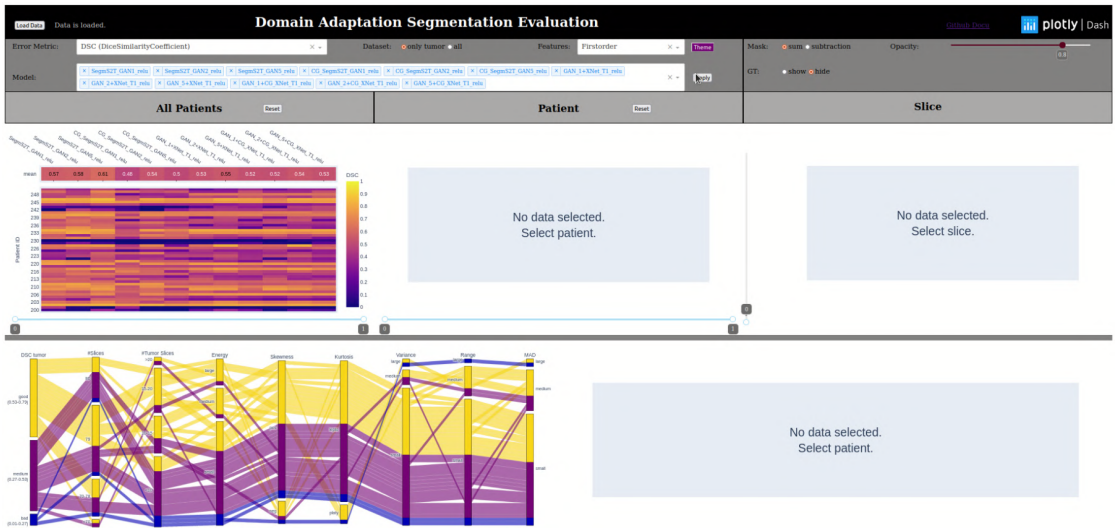
A.2.2 Subject Analysis (Q2)



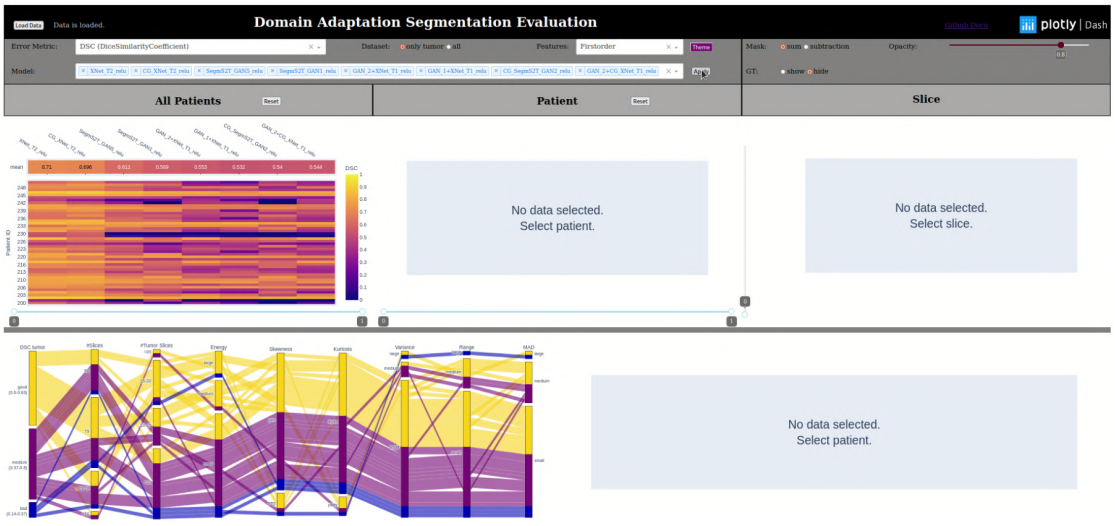
(a) We take a look at the first-order features for different model group selections. First selection is the group All.



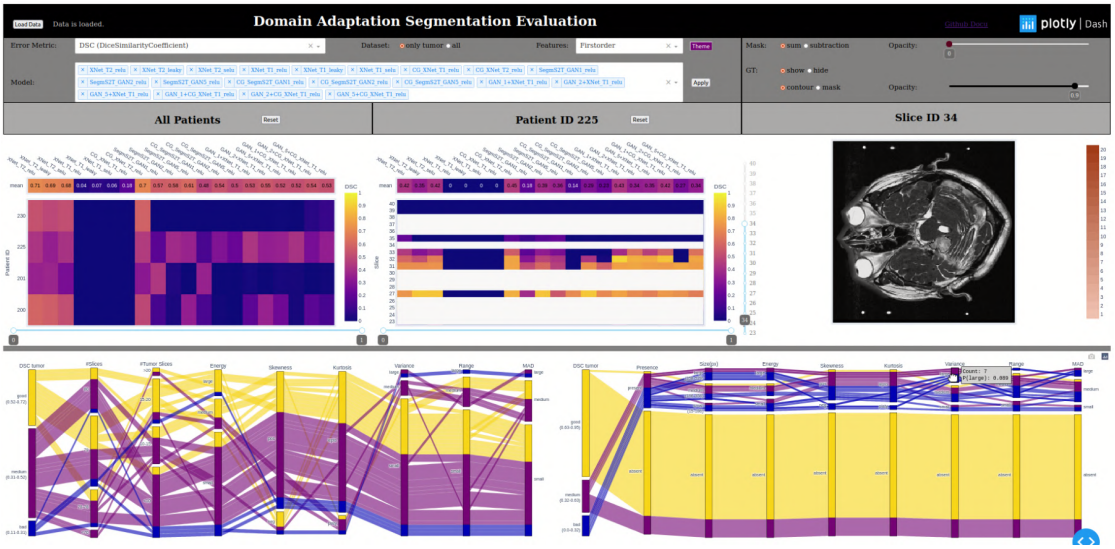
(b) The next group contains all baseline methods.



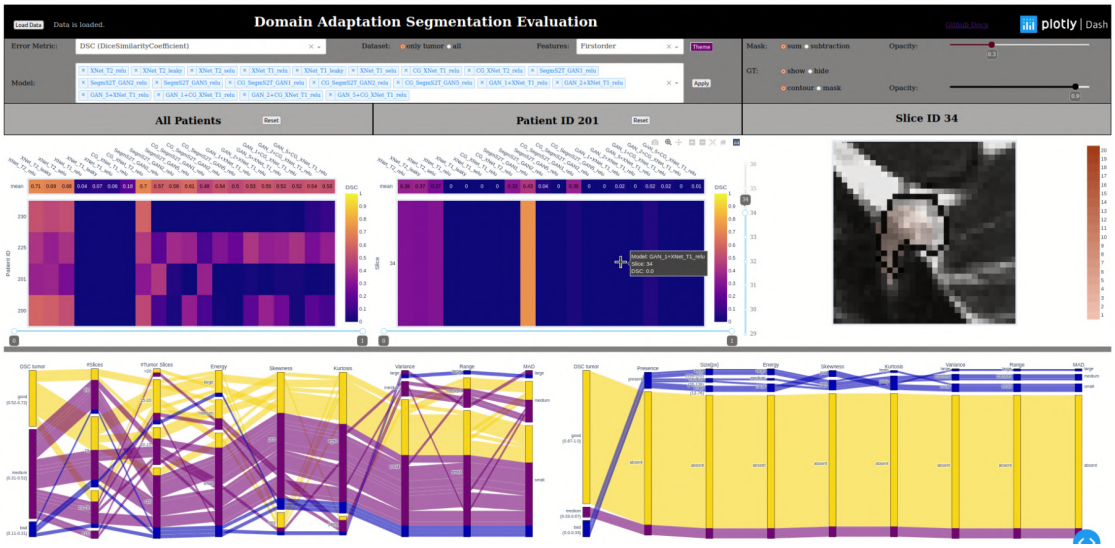
(c) Then, we look at all DA models.



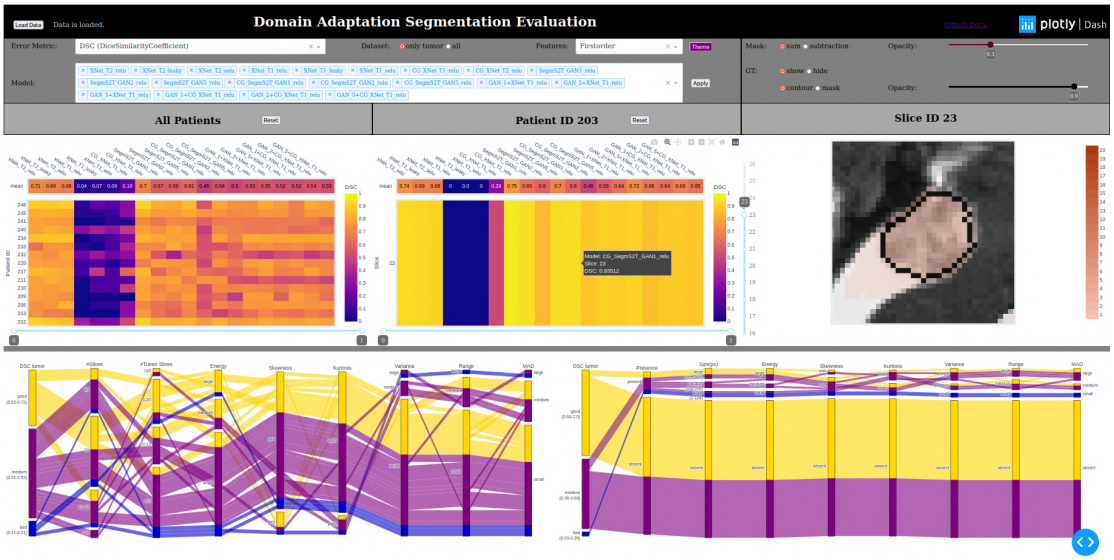
(d) We also select the best methods.



(e) After getting an overview, we select data samples with bad overall performance values. Patient ID 255 is an example where we inspect the subject PSD.



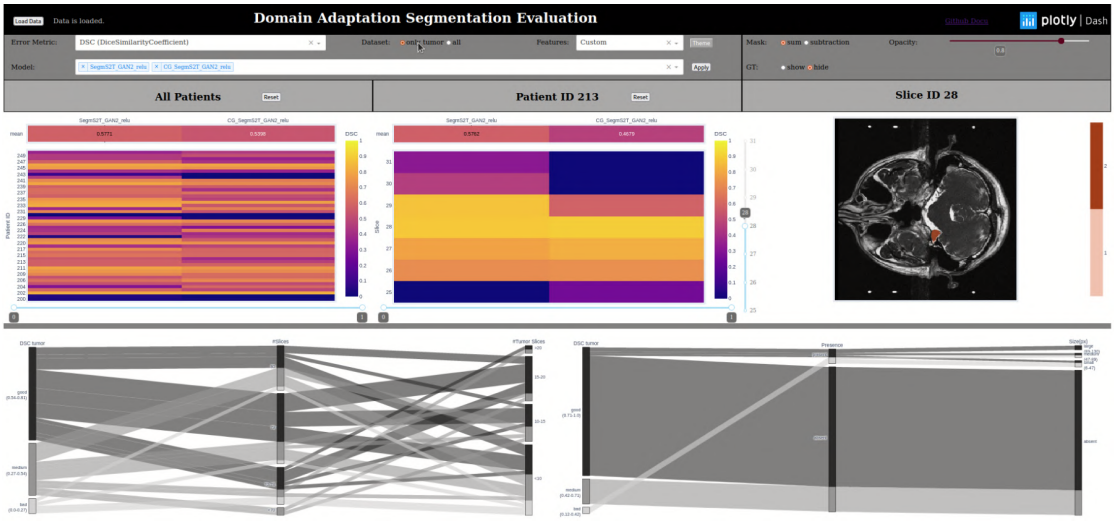
(f) Another example is patient ID 201. Zooming in on slice 34 shows inhomogeneous intensity values in the ROI.



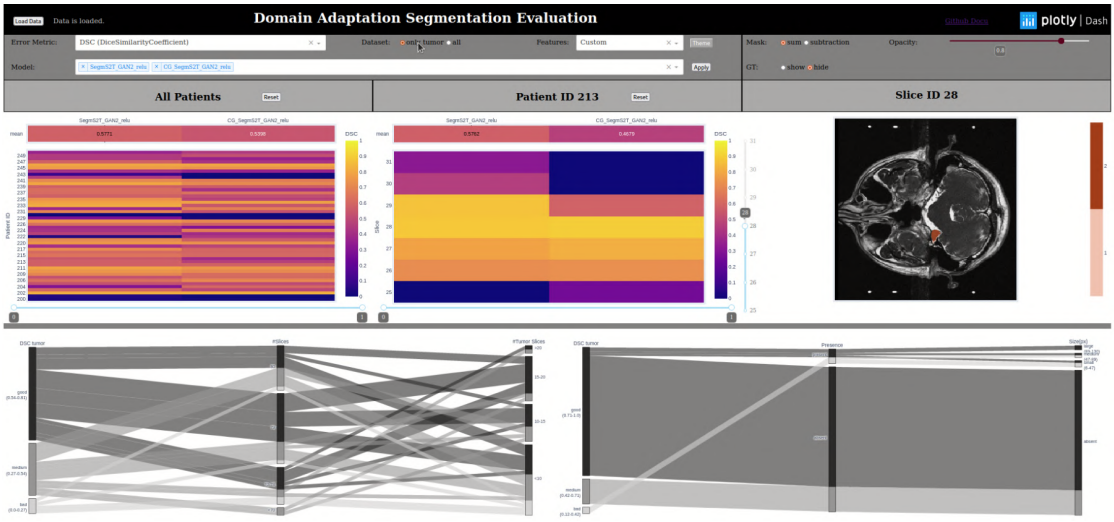
(g) As comparison, we also select subjects with good performance and analyze their subject PSD. The ROI zoom shows a more homogeneous intensity range.

Figure A.11: Workflow for **Q2** to analyze the relationship between first-order features and the performance scores. Steps (f)-(h) are repeated for multiple examples (patient and slice IDs).

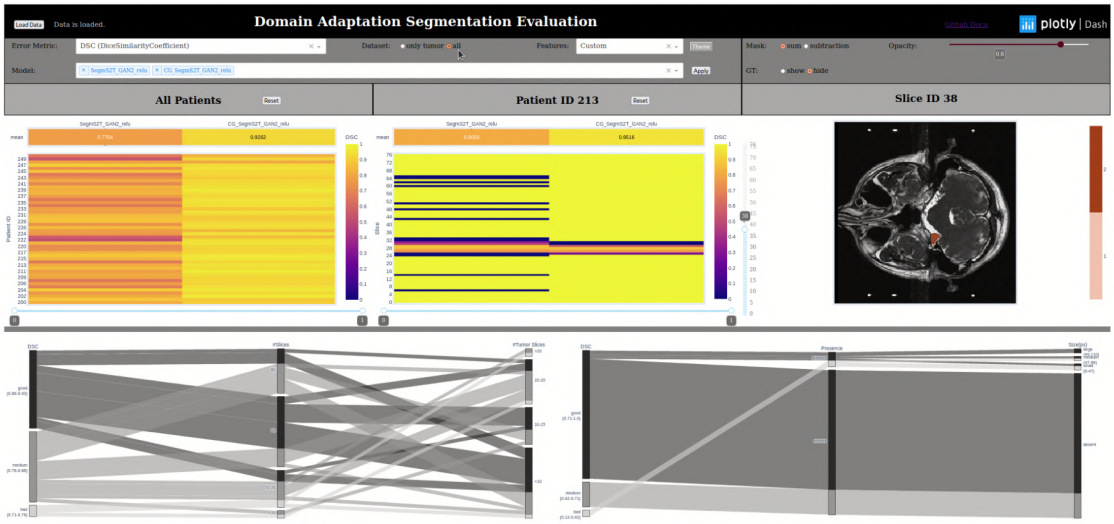
A.2.3 CG Module Analysis (Q3)



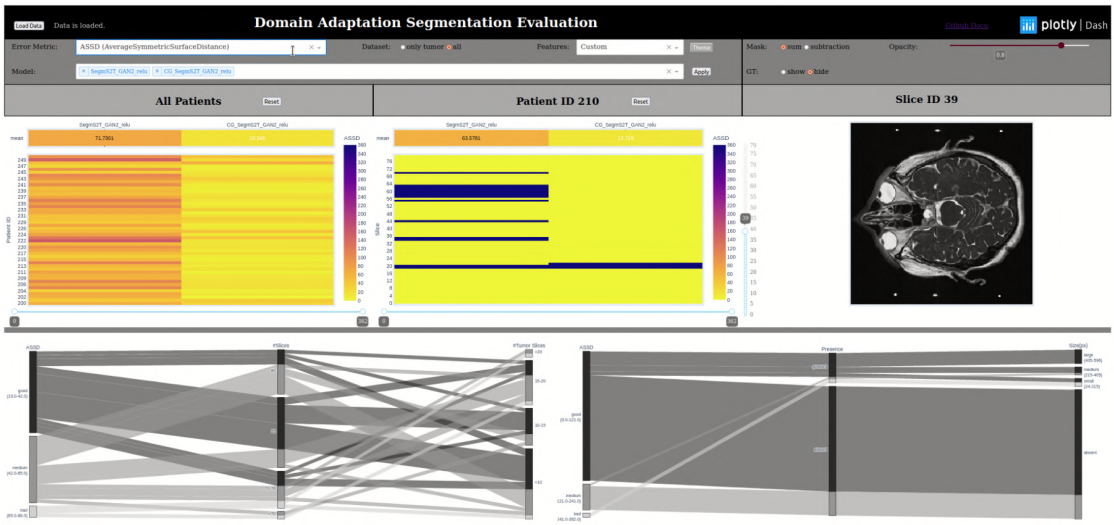
(a) We select the domain adaptation methods **X-Net S2T** `dstep=2` and **CG X-Net S2T** `dstep=2` for this showcase.



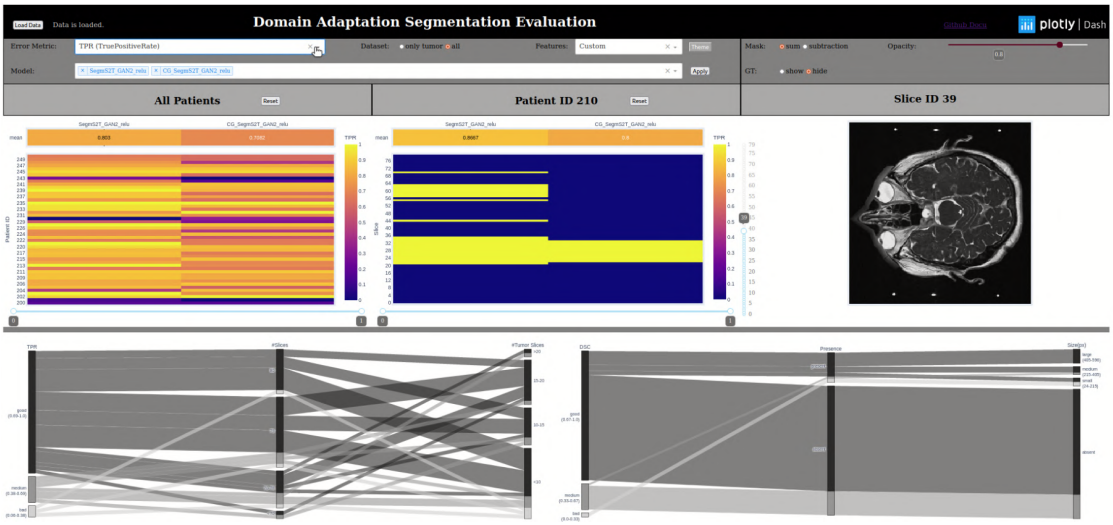
(b) The first overview refers only to tumor slices, where no clear difference can be determined.



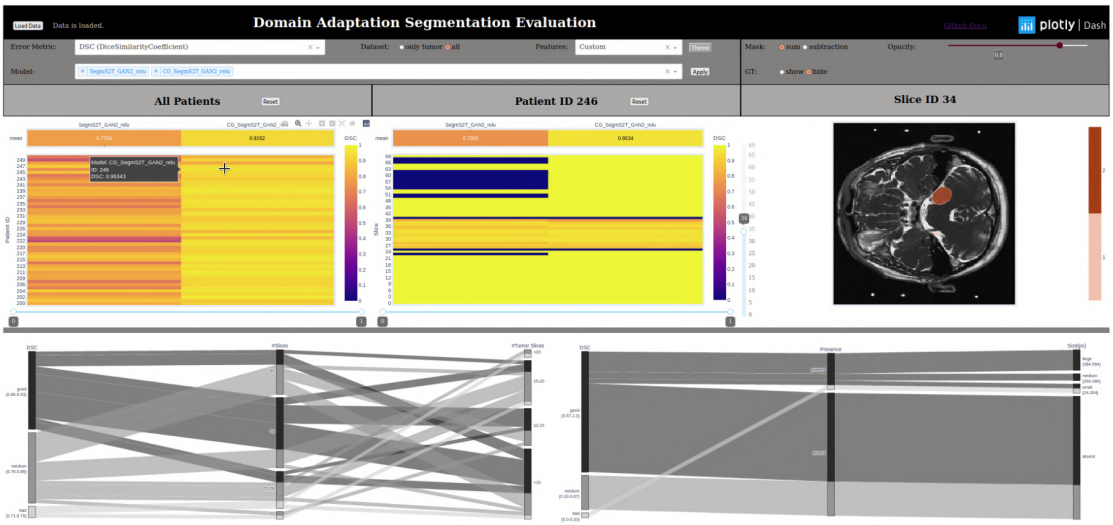
(c) If we switch to the view of the whole data set, the overall better results of the CG method become visible.



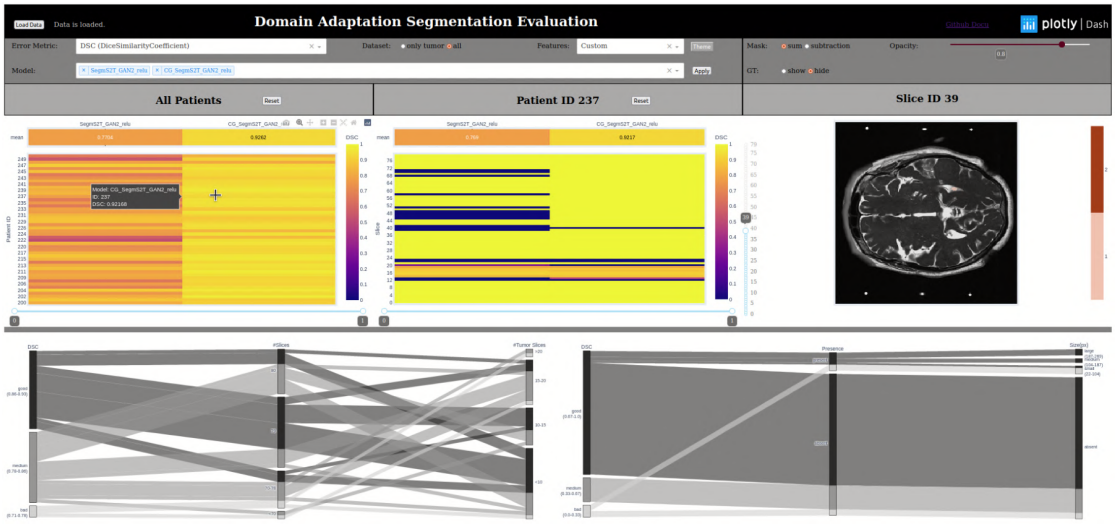
(d) Then, the performance heatmap is generated based on the ASSD values.



(e) TPR is only based on tumor slices, therefore the results are mixed as in the only tumor of other performance measures, such as DSC.



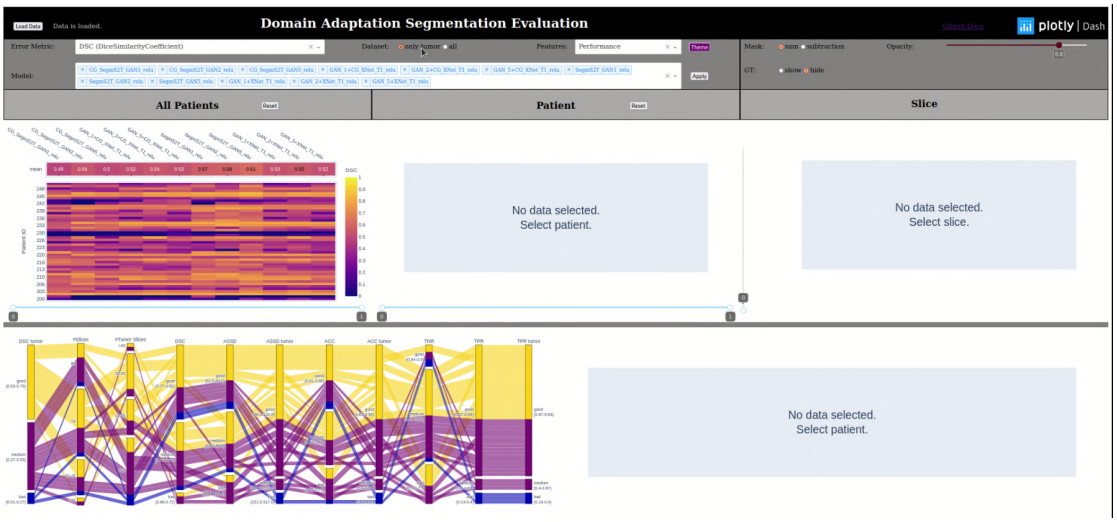
(f) The same behavior observed for the cohort is also found in the subject performance heatmaps. Patient ID 246 is the first example.



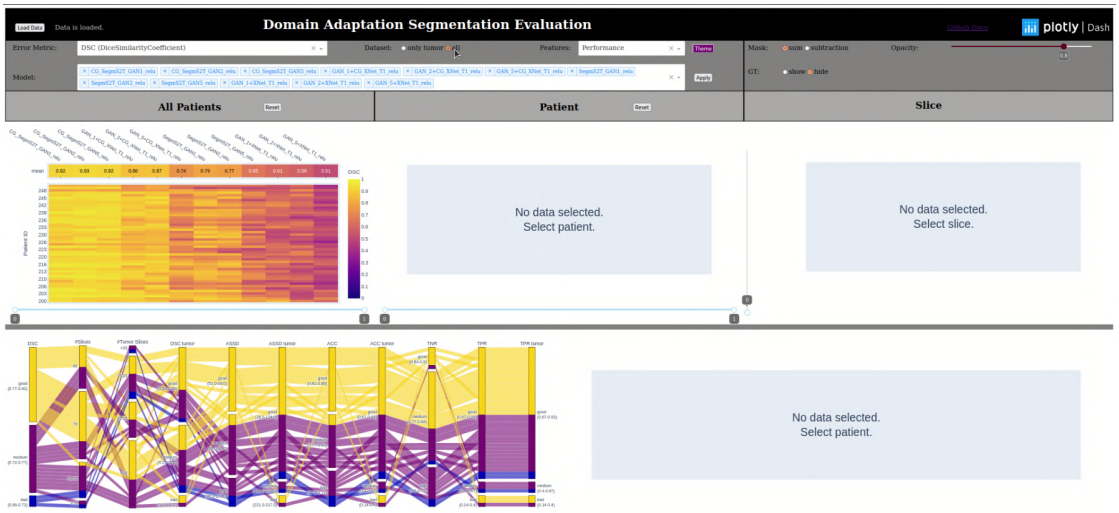
(g) Another example is given by patient ID 237.

Figure A.12: Workflow for **Q3** to find differences between the domain adaptation methods **X-Net S2T** `dstep=2` and **CG X-Net S2T** `dstep=2`. Steps (f) and (g) are repeated for multiple examples (patient and slice IDs).

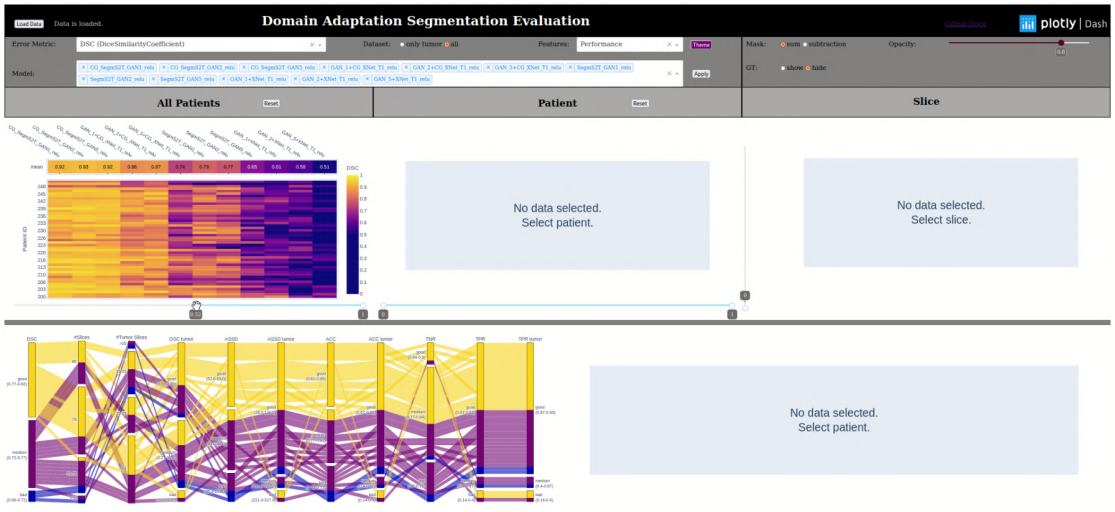
A.2.4 Segmentation Approach Analysis (Q4)



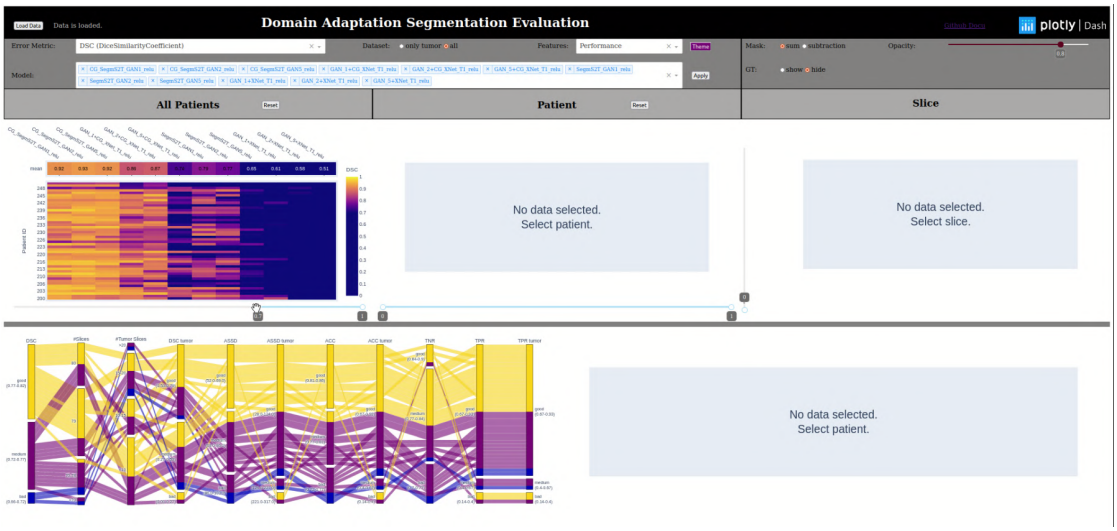
(a) The DA methods are selected and reordered. Only tumor slices are considered for the visualizations.



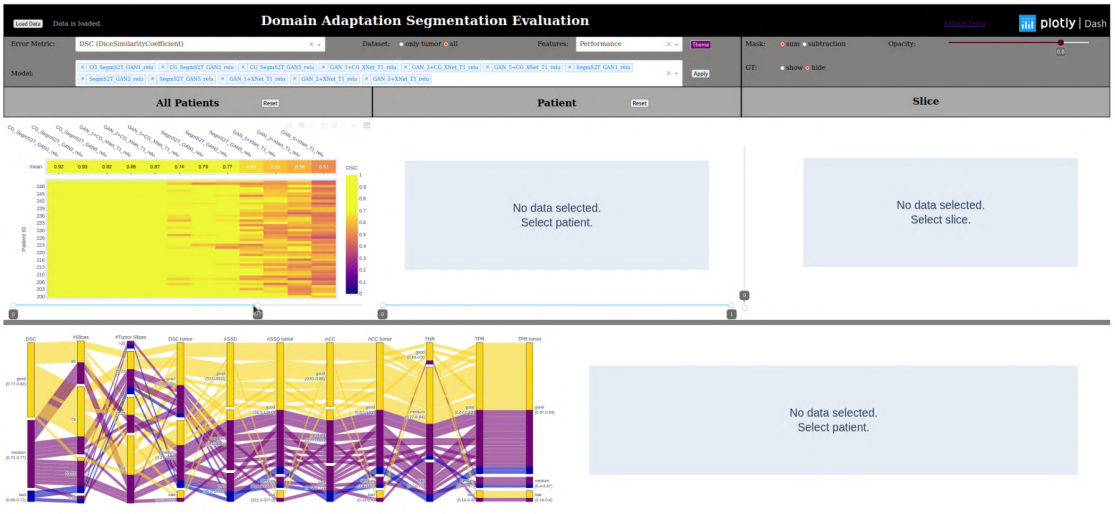
(b) Changing the dataset view to all shows a clear gradient of DSC values from left to right.



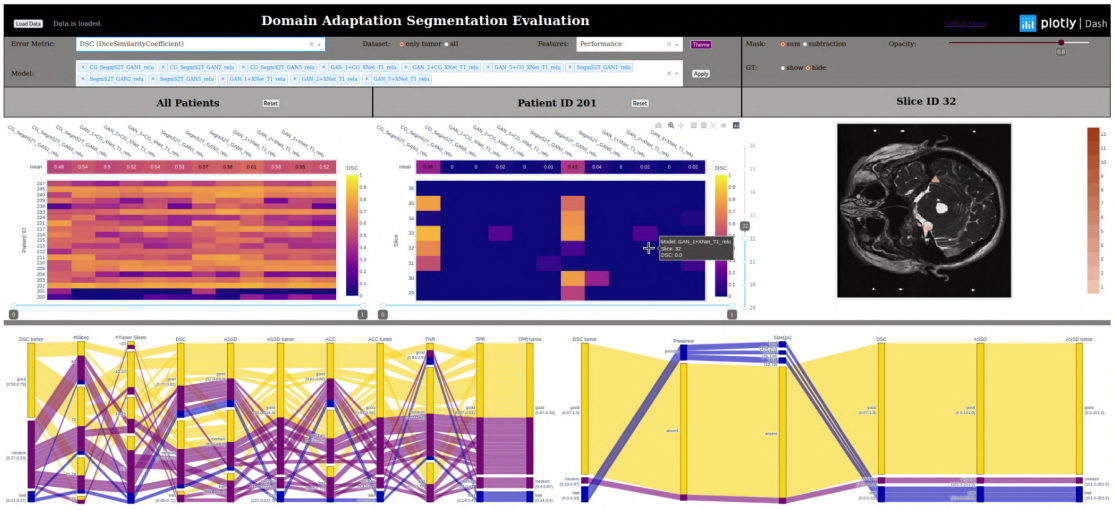
(c) Moving the slider for the heatmap colors to 0.52 for the minimal value highlights the color gradient.



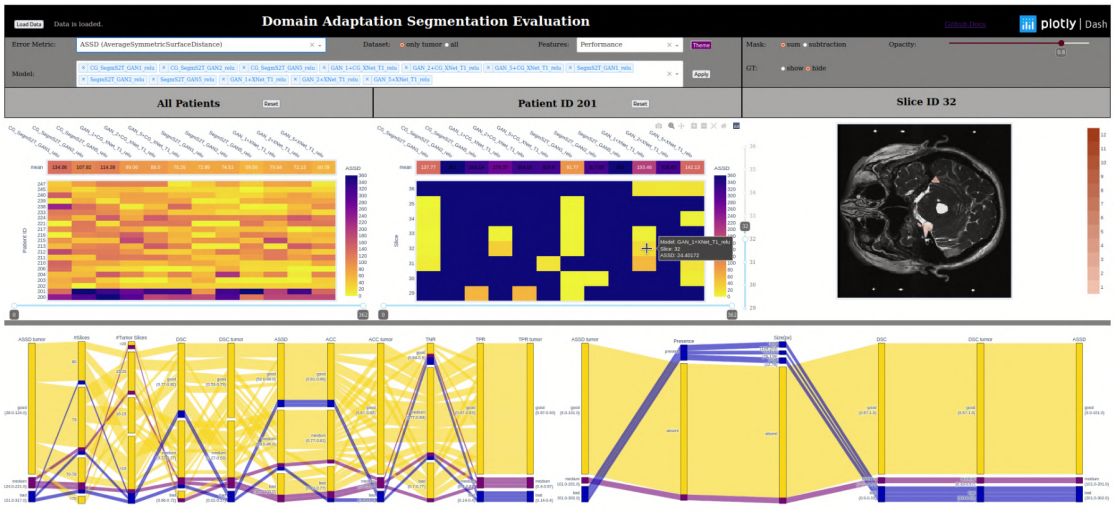
(d) Moving the slider for the heatmap colors further to 0.7 for the minimal value highlights the color gradient even more.



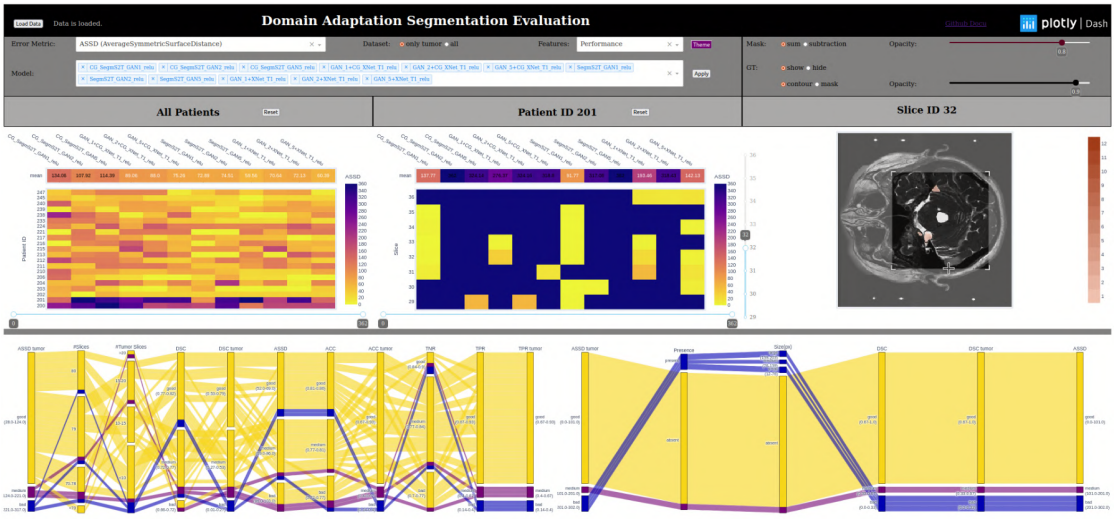
(e) Moving the slider for the heatmap colors to 0.7 for the maximal value results in a yellow color for the first rows from the left.



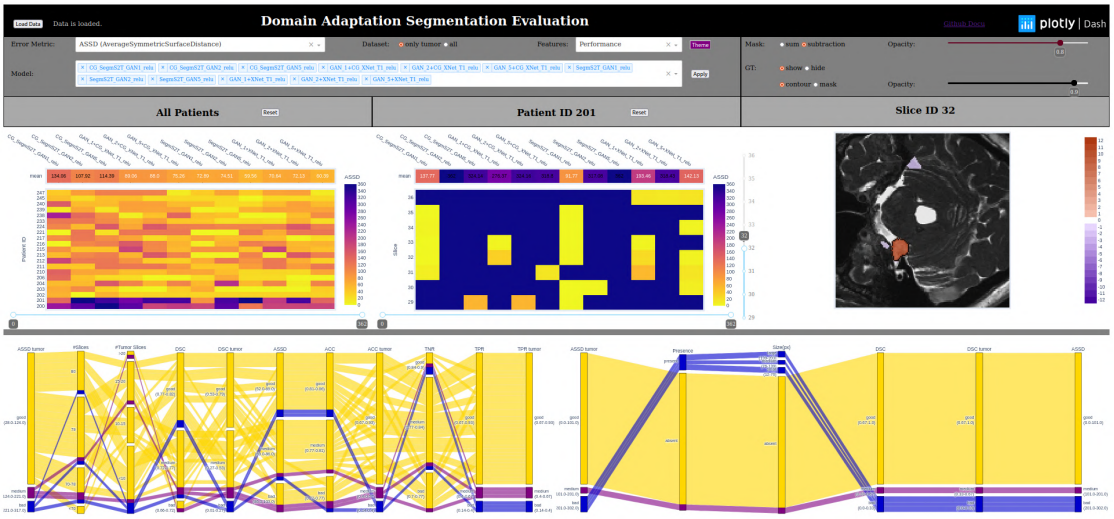
(f) The DSC performance heatmap of a subject example (patient ID 201) shows a lot of blue cells.



(g) Changing to the ASSD representation shows distance values between 0 and the maximum value, indicating the existing of predictions.



(h) We choose the slice with ID 32 and zoom in on the image center.



(i) Changing the encoding from sum to subtraction shows the blue area indicating predictions without overlap to the GT labels.

Figure A.13: Workflow for **Q4** to analyze the DA segmentation approaches. Steps (f)-(i) are repeated for multiple examples (patient and slice IDs).

List of Figures

List of Tables

Bibliography

[1] Anatomical Brain Barriers to Cancer Spread: Segmentation from CT and MR Images. https://abcs.mgh.harvard.edu/. (accessed 03.07.2021).

[2] Brain Tumor Segmentation (BraTS) Challenge 2021. http://braintumorsegmentation.org/. (accessed 22.08.2021).

[3] Cross-Modality Domain Adaptation for Medical Image Segmentation (crossMoDa). https://crossmoda-challenge.ml/. (accessed 20.11.2021).

[4] HEad and neCK TumOR segmentation challenge (HECKTOR). https://www.aicrowd.com/challenges/miccai-2020-hecktor. (accessed 21.08.2021).

[5] Vestibular Schwannoma (Acoustic Neuroma) and Neurofibromatosis. https://www.nidcd.nih.gov/health/vestibular-schwannoma-acoustic-neuroma-and-neurofibromatosis. (accessed 12.10.2021).

[6] Dash documentation. Available from: https://dash.plotly.com/, 2021. (accessed 20.11.2021).

[7] Plotly documentation. Available from: https://plotly.com/python/, 2021. (accessed 20.11.2021).

[8] Pyradiomics documentation. Available from: https://pyradiomics.readthedocs.io/en/latest/, 2021. (accessed 20.11.2021).

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from https://www.tensorflow.org/.

[10] R. R. Agravat and M. S. Raval. A Survey and Analysis on Automated Glioma Brain Tumor Segmentation and Overall Patient Survival Prediction. *Arch Computat Methods Eng*, 28:4117–4152, 2021.

[11] A. Al-Taie, H. K. Hahn, and L. Linsen. Uncertainty estimation and visualization in probabilistic segmentation. *Computers & Graphics*, 39:48–59, 2014.

[12] A. Al-Taie, H. K. Hahn, and L. Linsen. Uncertainty Estimation and Visualization for Multi-modal Image Segmentation. In *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2015.

[13] V. Andrearczyk, V. Oreiller, M. Vallières, J. Castelli, H. Elhalawani, M. Jreige, S. Boughdad, J. O. Prior, and A. Depeursinge. Automatic Segmentation of Head and Neck Tumors and Nodal Metastases in PET-CT scans. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, volume 121 of *Proceedings of Machine Learning Research*, pages 33–43. PMLR, 2020.

[14] R. A. Arun, S. Umamaheswari, and A. V. Jain. Reduced U-Net Architecture for Classifying Crop and Weed using Pixel-wise Segmentation. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pages 1–6, 2020.

[15] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[16] M. Blumenschein, M. Behrisch, S. Schmid, S. Butscher, D. R. Wahl, K. Villinger, B. Renner, H. Reiterer, and D. A. Keim. SMARTexplore: Simplifying High-Dimensional Data Analysis through a Table-Based Visual Analytics Approach. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 36–47, 2018.

[17] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised Pixel-Level Domain Adaptation With Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3722–3731, 2017.

[18] A. Buja, J. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Proceeding Visualization '91*, pages 156–163, 1991.

[19] J. Bullock, C. Cuesta-Lázaro, and A. Quera-Bofarull. XNet: a convolutional neural network (CNN) implementation for medical X-Ray image segmentation suitable for small datasets. In *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 10953, pages 453 – 463. International Society for Optics and Photonics, SPIE, 2019.

[20] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[21] M. D. Chan, C. L. Rogers, B. Anderson, and D. Khuntia. Chapter 28 - benign brain tumors: Meningiomas and vestibular schwannomas. In *Clinical Radiation Oncology (Fourth Edition)*, pages 483–501. Elsevier, fourth edition edition, 2016.

[22] A. Chartsias, T. Joyce, R. Dharmakumar, and S. A. Tsaftaris. Adversarial Image Synthesis for Unpaired Multi-modal Cardiac Data. In *Simulation and Synthesis in Medical Imaging*, pages 3–13, Cham, 2017. Springer International Publishing.

[23] C. Chen, Q. Dou, H. Chen, and P.-A. Heng. Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation in Chest X-Ray Segmentation. In *Machine Learning in Medical Imaging*, pages 143–151, Cham, 2018. Springer International Publishing.

[24] C. Chen, Q. Dou, H. Chen, J. Qin, and P.-A. Heng. Synergistic Image and Feature Adaptation: Towards Cross-Modality Domain Adaptation for Medical Image Segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):865–872, 2019.

[25] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng. Unsupervised Bidirectional Cross-Modality Adaptation via Deeply Synergistic Image and Feature Alignment for Medical Image Segmentation. *IEEE Transactions on Medical Imaging*, 39(7):2494–2505, 2020.

[26] H. Chen, H. Chen, and L. Wang. Iteratively Refine the Segmentation of Head and Neck Tumor in FDG-PET and CT Images. In *Head and Neck Tumor Segmentation*, pages 53–58, Cham, 2021. Springer International Publishing.

[27] H. Chen, D. Qian, W. Liu, H. Li, and L. Wang. An Enhanced Coarse-to-Fine Framework for the Segmentation of Clinical Target Volume. In *Segmentation, Classification, and Registration of Multi-modality Medical Imaging Data*, pages 34–39, Cham, 2021. Springer International Publishing.

[28] H. Chen, X. Wang, Y. Huang, X. Wu, Y. Yu, and L. Wang. Harnessing 2D Networks and 3D Features for Automated Pancreas Segmentation from Volumetric CT Images. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 339–347, Cham, 2019. Springer International Publishing.

[29] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing.

[30] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

[31] L. Datta. A Survey on Activation Functions and their relation with Xavier and He Normal Initialization. *arXiv:2004.06632 [cs.NE]*, 2020.

[32] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, 43:19–67, 2005.

[33] M. A. Deeley, A. Chen, R. D. Datteri, J. Noble, A. Cmelak, E. Donnelly, A. Malcolm, L. Moretti, J. Jaboin, K. Niermann, E. S. Yang, D. S. Yu, and B. M. Dawant. Segmentation editing improves efficiency while reducing inter-expert variation and maintaining accuracy for normal brain tissues in the presence of space-occupying lesions. *Phys Med Biol.*, 58(12):4071–4097, 2013.

[34] Q. Dou, C. Ouyang, C. Chen, H. Chen, B. Glocker, X. Zhuang, and P.-A. Heng. PnP-AdaNet: Plug-and-Play Adversarial Domain Adaptation Network at Unpaired Cross-Modality Cardiac Segmentation. *IEEE Access*, 7:99065–99076, 2019.

[35] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The Importance of Skip Connections in Biomedical Image Segmentation. In *Deep Learning and Data Labeling for Medical Applications*, pages 179–187, Cham, 2016. Springer International Publishing.

[36] G. E, P. JM, W. M, and Z. G. Deep learning enables reduced gadolinium dose for contrast-enhanced brain MRI. *J Magn Reson Imaging*, 48(2):330–340, 2018.

[37] N. J. Erickson, P. G. R. Schmalz, B. S. Agee, M. Fort, B. C. Walters, B. M. McGrew, and I. Fisher, Winfield S. Koos Classification of Vestibular Schwannomas: A Reliability Study. *Neurosurgery*, 85(3):409–414, 08 2018.

[38] B. Fröhler, T. Möller, and C. Heinzl. GEMSe: Visualization-Guided Exploration of Multi-channel Segmentation Algorithms. *Computer Graphics Forum*, 35(3):191–200, 2016.

[39] K. Furmanová, L. P. Muren, O. Casares-Magaz, V. Moiseenko, J. P. Einck, S. Pilskog, and R. G. Raidou. Previs: Predictive visual analytics of anatomical variability for radiotherapy decision support. *Computers & Graphics*, 97:126–138, 2021.

[40] Y. Ganin, E. Ustinova, H. Ajakan, and et al. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[41] S. S. Gay, C. Yu, D. J. Rhee, C. Sjogreen, R. P. Mumme, C. M. Nguyen, T. J. Netherton, C. E. Cardenas, and L. E. Court. A Bi-directional, Multi-modality Framework for Segmentation of Brain Structures. In *Segmentation, Classification, and Registration of Multi-modality Medical Imaging Data*, pages 49–57, Cham, 2021. Springer International Publishing.

[42] A. Geurts, G. Sakas, A. Kuijper, M. Becker, and T. v. Landesberger. Visual Comparison of 3D Medical Image Segmentation Algorithms Based on Statistical Shape Models. In *Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management: Ergonomics and Health*, pages 336–344, Cham, 2015. Springer International Publishing.

[43] K. Ghimire, Q. Chen, and X. Feng. Patch-Based 3D UNet for Head and Neck Tumor Segmentation with an Ensemble of Conventional and Dilated Convolutions. In *Head and Neck Tumor Segmentation*, pages 78–84, Cham, 2021. Springer International Publishing.

[44] C. Gillmann, D. Saur, T. Wischgoll, and G. Scheuermann. Uncertainty-aware Visualization in Medical Imaging - A Survey. *Computer Graphics Forum*, 40(3):665–689, 2021.

[45] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [stat.ML]*, 2014.

[46] H. Guan and M. Liu. Domain Adaptation for Medical Image Analysis: A Survey. *IEEE Transactions on Biomedical Engineering*, pages 1–1, 2021.

[47] M. Harrower and C. A. Brewer. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[48] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[49] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[50] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[51] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998. PMLR, July 2018.

[52] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861 [cs.CV]*, 2017.

[53] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, July 2017.

[54] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu. UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059, 2020.

[55] Y. Huo, Z. Xu, S. Bao, A. Assad, R. G. Abramson, and B. A. Landman. Adversarial Synthesis Learning Enables Segmentation Without Target Modality Ground Truth. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1217–1220, 2018.

[56] Y. Huo, Z. Xu, H. Moon, S. Bao, A. Assad, T. K. Moyo, M. R. Savona, R. G. Abramson, and B. A. Landman. SynSeg-Net: Synthetic Segmentation Without Target Modality Ground Truth. *IEEE Transactions on Medical Imaging*, 38(4):1016–1025, 2019.

[57] A. Iantsen, V. Jaouen, D. Visvikis, and M. Hatt. Squeeze-and-Excitation Normalization for Brain Tumor Segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 366–373, Cham, 2021. Springer International Publishing.

[58] A. Iantsen, D. Visvikis, and M. Hatt. Squeeze-and-Excitation Normalization for Automated Delineation of Head and Neck Primary Tumors in Combined PET and CT Images. In *Head and Neck Tumor Segmentation*, pages 37–43, Cham, 2021. Springer International Publishing.

[59] F. Isensee, P. F. Jäger, P. M. Full, P. Vollmuth, and K. H. Maier-Hein. nnU-Net for Brain Tumor Segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 118–132, Cham, 2021. Springer International Publishing.

[60] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, and K. H. Maier-Hein. nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation. *arXiv:1809.10486 [cs.CV]*, 2018.

[61] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[62] R. Jena and S. P. Awate. A Bayesian Neural Net to Segment Images with Uncertainty Estimates and Good Calibration. In A. C. S. Chung, J. C. Gee, P. A. Yushkevich, and S. Bao, editors, *Information Processing in Medical Imaging*, pages 3–15, Cham, 2019. Springer International Publishing.

[63] H. Jia, W. Cai, H. Huang, and Y. Xia. H^2NF-Net for Brain Tumor Segmentation Using Multimodal MR Imaging: 2nd Place Solution to BraTS Challenge 2020 Segmentation Task. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 58–68, Cham, 2021. Springer International Publishing.

[64] H. Jia, Y. Xia, W. Cai, and H. Huang. Learning High-Resolution and Efficient Non-local Features for Brain Glioma Segmentation in MR Images. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pages 480–490, Cham, 2020. Springer International Publishing.

[65] J. Jiang, Y.-C. Hu, N. Tyagi, P. Zhang, A. Rimner, G. S. Mageras, J. O. Deasy, and H. Veeraraghavan. Tumor-Aware, Adversarial Domain Adaptation from CT to MRI for Lung Cancer Segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 777–785, Cham, 2018. Springer International Publishing.

[66] T. Joyce, A. Chartsias, and S. A. Tsaftaris. Deep multi-class segmentation without ground- truth labels. *In International conference on Medical Imaging with Deep Learning (MIDL)*, 2018.

[67] K. Kamnitsas, C. Baumgartner, C. Ledig, V. Newcombe, J. Simpson, A. Kane, D. Menon, A. Nori, A. Criminisi, D. Rueckert, and B. Glocker. Unsupervised Domain Adaptation in Brain Lesion Segmentation with Adversarial Networks. In *Information Processing in Medical Imaging*, pages 597–609, Cham, 2017. Springer International Publishing.

[68] R. Karunamuni, H. Bartsch, N. S. White, V. Moiseenko, R. Carmona, D. C. Marshall, T. M. Seibert, C. R. McDonald, N. Farid, A. Krishnan, J. Kuperman, L. Mell, J. B. Brewer, A. M. Dale, and J. A. Hattangadi-Gluth. Dose-Dependent Cortical Thinning After Partial Brain Irradiation in High-Grade Glioma. *International Journal of Radiation Oncology Biology Physics*, 94(2):297–304, 2016.

[69] D. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[70] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. *Visual Analytics: Scope and Challenges*, pages 76–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[71] K. Kim, J. V. Carlis, and D. F. Keefe. Comparison techniques utilized in spatial 3D and 4D data visualizations: A survey and future directions. *Computers & Graphics*, 67:138–147, 2017.

[72] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs.LG]*, 2017.

[73] King's College London - School of Biomedical Engineering & Imaging Sciences. Automatic Segmentation of Vestibular Schwannoma with MONAI (PyTorch) - Data preprocessing. Available from: `https://github.com/KCL-BMEIS/VS_Seg/tree/master/preprocessing`, 2021. (accessed 05.07.2021).

[74] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-Normalizing Neural Networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 972–981. Curran Associates Inc., 2017.

[75] P. Klemm, S. Oeltze-Jafra, K. Lawonn, K. Hegenscheid, H. Völzke, and B. Preim. Interactive Visual Analysis of Image-Centric Cohort Study Data. *IEEE transactions on visualization and computer graphics*, 20(12):1673–1682, 2014.

[76] R. Kosara, F. Bendix, and H. Hauser. Parallel Sets: Interactive Exploration and Visual Analysis of Categorical Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, 2006.

[77] W. M. Kouw and M. Loog. A Review of Domain Adaptation without Target Labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):766–785, March 2021.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[79] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik. Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816, 2020.

[80] T. v. Landesberger, D. Basgier, and M. Becker. Comparative Local Quality Assessment of 3D Medical Image Segmentations with Focus on Statistical Shape Model-Based Algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2537–2549, 2016.

[81] M. Langhans, T. Fechter, D. Baltas, H. Binder, and T. Bortfeld. Automatic Segmentation of Brain Structures for Treatment Planning Optimization and Target Volume Definition. In *Segmentation, Classification, and Registration of Multimodality Medical Imaging Data*, pages 40–48, Cham, 2021. Springer International Publishing.

[82] K. Lawonn, N. Smit, K. Bühler, and B. Preim. A Survey on Multimodal Medical Data Visualization. *Computer Graphics Forum*, 37(1):413–438, 2018.

[83] C. Li, Y. Tan, W. Chen, X. Luo, Y. Gao, X. Jia, and Z. Wang. Attention Unet++: A Nested Attention-Aware U-Net for Liver CT Image Segmentation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 345–349, 2020.

[84] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes. *IEEE Transactions on Medical Imaging*, 37(12):2663–2674, 2018.

[85] Y. Li, T. Fujiwara, Y. K. Choi, K. K. Kim, and K.-L. Ma. A visual analytics system for multi-model comparison on clinical data predictions. *Visual Informatics*, 4(2):122–131, 2020. PacificVis 2020 Workshop on Visualization Meets AI.

[86] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015.

[87] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163—169, 1987.

[88] J. Ma, J. Chen, M. Ng, R. Huang, Y. Li, C. Li, X. Yang, and A. L. Martel. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71:102035, 2021.

[89] J. Ma and X. Yang. Combining CNN and Hybrid Active Contours for Head and Neck Tumor Segmentation in CT and PET Images. In *Head and Neck Tumor Segmentation*, pages 59–64, Cham, 2021. Springer International Publishing.

[90] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[91] T. Magadza and S. Viriri. Deep Learning for Brain Tumor Segmentation: A Survey of State-of-the-Art. *Journal of Imaging*, 7(2), 2021.

[92] J. McGlinchy, B. Johnson, B. Muller, M. Joseph, and J. Diaz. Application of UNet Fully Convolutional Neural Network to Impervious Surface Segmentation in Urban Environment from High Resolution Satellite Imagery. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3915–3918, 2019.

[93] F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016.

[94] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image Segmentation Using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[95] J. H. Moltz, S. Braunewell, J. Rühaak, F. Heckel, S. Barbieri, L. Tautz, H. K. Hahn, and H.-O. Peitgen. Analysis of variability in manual liver tumor delineation in CT scans. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1974–1977, 2011.

[96] K. Moreland. Diverging Color Maps for Scientific Visualization. In *Advances in Visual Computing*, pages 92–103, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[97] T. Nair, D. Precup, D. L. Arnold, and T. Arbel. Exploring uncertainty measures in deep networks for Multiple sclerosis lesion detection and segmentation. *Medical Image Analysis*, 59:101557, 2020.

[98] V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807—-814. Omnipress, 2010.

[99] M. Ning, C. Bian, C. Yuan, K. Ma, and Y. Zheng. Ensembled ResUnet for Anatomical Brain Barriers Segmentation. In *Segmentation, Classification, and Registration of Multi-modality Medical Imaging Data*, pages 27–33, Cham, 2021. Springer International Publishing.

[100] C. F. Njeh. Tumor delineation: The weakest link in the search for accuracy in radiotherapy. *J Med Phys.*, 33(4):136–140, 2008.

[101] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *arXiv:1811.03378 [cs.LG]*, 2018.

[102] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv:1804.03999 [cs.CV]*, 2018.

[103] B. Preim and K. Lawonn. A Survey of Visual Analytics for Public Health. *Computer Graphics Forum*, 39(1):543–580, 2020.

[104] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[105] R. G. Raidou. Uncertainty Visualization: Recent Developments and Future Challenges in Prostate Cancer Radiotherapy Planning. In *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)*. The Eurographics Association, 2018.

[106] R. G. Raidou, F. J. J. Marcelis, M. Breeuwer, E. Gröller, A. Vilanova, and H. M. M. v. d. Wetering. Visual Analytics for the Exploration and Assessment of Segmentation Errors. In *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2016.

[107] O. Reiter, M. Breeuwer, E. Gröller, and R. G. Raidou. Comparative Visual Analysis of Pelvic Organ Segmentations. In *EuroVis 2018—Short Papers*, pages 37–41. The Eurographics Association, 2018.

[108] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[109] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From Source to Target and Back: Symmetric Bi-Directional Adaptive GAN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8099–8108, June 2018.

[110] A. Saad, T. Möller, and G. Hamarneh. ProbExplorer: Uncertainty-guided Exploration and Editing of Probabilistic Medical Image Segmentation. *Computer Graphics Forum*, 29(3):1113–1122, 2010.

[111] M. Schlachter, R. Raidou, L. P. Muren, B. Preim, and K. Bühler. State-of-the-Art Report: Visual Computing in Radiation Therapy Planning. *Computer Graphics Forum*, 3(38):753–779, 2019.

[112] J. Schmidt, R. Preiner, T. Auzinger, M. Wimmer, M. E. Gröller, and S. Bruckner. YMCA - Your Mesh Comparison Application. In *IEEE Visual Analytics Science and Technology*, VAST. IEEE Computer Society, 2014.

[113] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[114] J. Shapey, A. Kujawa, R. Dorent, G. Wang, S. Bisdas, A. Dimitriadis, D. Grishchuck, I. Paddick, N. Kitchen, R. Bradford, S. Saeed, S. Ourselin, and T. Vercauteren. Segmentation of vestibular schwannoma from magnetic resonance imaging: An open annotated dataset and baseline algorithm [data set]. Available from: `https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70229053`, 2021. (accessed 05.07.2021).

[115] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[116] N. Shusharina, T. Bortfeld, C. Cardenas, B. De, K. Diao, S. Hernandez, Y. Liu, S. Maroongroge, J. Söderberg, and M. Soliman. Cross-Modality Brain Structures Image Segmentation for the Radiotherapy Target Definition and Plan Optimization. In *Segmentation, Classification, and Registration of Multi-modality Medical Imaging Data*, pages 3–15, Cham, 2021. Springer International Publishing.

[117] S. Silva, J. Madeira, and B. S. Santos. PolyMeCo—An integrated environment for polygonal mesh analysis and comparison. *Computers & Graphics*, 33(2):181–191, 2009.

[118] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, May 2015.

[119] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[120] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693, 2020.

[121] E. Tiu. Metrics to evaluate your semantic segmentation model. Available from: `https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2`, 2021. (accessed 19.07.2021).

[122] T. Torsney-Weir, A. Saad, T. Moller, H.-C. Hege, B. Weber, J.-M. Verbavatz, and S. Bergner. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.

[123] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial Discriminative Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, July 2017.

[124] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv:1607.08022 [cs.CV]*, 2017.

[125] A. Uosef, M. Villagran, J. Z. Kubiak, J. Wosik, R. M. Ghobrial, and M. Kloc. Side effects of gadolinium MRI contrast agents. *Pediatria I Medycyna Rodzinna-Paediatrics and Family Medicine*, 16(1):49–52, 2020.

[126] V. V. Valindria, N. Pawlowski, M. Rajchl, I. Lavdas, E. O. Aboagye, A. G. Rockall, D. Rueckert, and B. Glocker. Multi-modal Learning from Unpaired Images: Application to Multi-organ Segmentation in CT and MRI. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 547–556, 2018.

[127] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[128] J. J. van Griethuysen, A. Fedorov, C. Parmar, A. Hosny, N. Aucoin, V. Narayan, R. G. Beets-Tan, J.-C. Fillion-Robin, S. Pieper, and H. J. Aerts. Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer Research*, 77(21):e104–e107, 2017.

[129] I. Viola and E. Gröller. Smart Visibility in Visualization. In *Computational Aesthetics in Graphics, Visualization and Imaging*. The Eurographics Association, 2005.

[130] T. von Landesberger, G. Andrienko, N. Andrienko, S. Bremm, M. Kirschner, S. Wesarg, and A. Kuijper. Opening up the "black box" of medical image segmentation with statistical shape models. *Vis Comput*, 29:893–905, 2013.

[131] T. von Landesberger, S. Bremm, M. Kirschner, S. Wesarg, and A. Kuijper. Visual Analytics for model-based medical image segmentation: Opportunities and challenges. *Expert Systems with Applications*, 40(12):4934–4943, 2013.

[132] G. Wang, J. Shapey, W. Li, R. Dorent, A. Dimitriadis, S. Bisdas, I. Paddick, R. Bradford, S. Zhang, S. Ourselin, and T. Vercauteren. Automatic Segmentation of Vestibular Schwannoma from T2-Weighted MRI by Deep Spatial Attention with Hardness-Weighted Loss. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 264–72, Cham, 2019. Springer International Publishing.

[133] G. Wang, J. Shapey, W. Li, R. Dorent, A. Dimitriadis, S. Bisdas, I. Paddick, R. Bradford, S. Zhang, S. Ourselin, and T. Vercauteren. Automatic Segmentation of Vestibular Schwannoma from T2-Weighted MRI by Deep Spatial Attention with Hardness-Weighted Loss. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 264–272, Cham, 2019. Springer International Publishing.

[134] J. Wang, S. Hazarika, C. Li, and H.-W. Shen. Visualization and Visual Analysis of Ensemble Data: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 25(9):2853–2872, 2019.

[135] Y. Wang, Y. Zhang, F. Hou, Y. Liu, J. Tian, C. Zhong, Y. Zhang, and Z. He. Modality-Pairing Learning for Brain Tumor Segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 230–240, Cham, 2021. Springer International Publishing.

[136] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for Using Multiple Views in Information Visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pages 110–119, New York, NY, USA, 2000. Association for Computing Machinery.

[137] J. Xie and Y. Peng. The Head and Neck Tumor Segmentation Using nnU-Net with Spatial and Channel 'Squeeze & Excitation' Blocks. In *Head and Neck Tumor Segmentation*, pages 28–36, Cham, 2021. Springer International Publishing.

[138] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang. SegAN: Adversarial Network with Multi-scale L1 Loss for Medical Image Segmentation. *Neuroinformatics*, 16:383–392, 2018.

[139] W. Yan, Y. Wang, S. Gu, L. Huang, F. Yan, L. Xia, and Q. Tao. The Domain Shift Problem of Medical Image Segmentation and Vendor-Adaptation by Unet-GAN. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 623–631, Cham, 2019. Springer International Publishing.

[140] J. Yang, N. C. Dvornek, F. Zhang, J. Chapiro, M. Lin, and J. S. Duncan. Unsupervised Domain Adaptation via Disentangled Representations: Application to Cross-Modality Liver Segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 255–263, Cham, 2019. Springer International Publishing.

[141] F. Yousefirizi and A. Rahmim. GAN-Based Bi-Modal Segmentation Using Mumford-Shah Loss: Application to Head and Neck Tumors in PET-CT Images. In *Head and Neck Tumor Segmentation*, pages 99–108, Cham, 2021. Springer International Publishing.

[142] Y. Yuan. Automatic Brain Tumor Segmentation with Scale Attention Network. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 285–294, Cham, 2021. Springer International Publishing.

[143] Y. Yuan. Automatic Head and Neck Tumor Segmentation in PET/CT with Scale Attention Network. In *Head and Neck Tumor Segmentation*, pages 44–52, Cham, 2021. Springer International Publishing.

[144] Y. Zhang, S. Miao, T. Mansi, and R. Liao. Task Driven Generative Modeling for Unsupervised Domain Adaptation: Application to X-ray Image Segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 599–607, Cham, 2018. Springer International Publishing.

[145] Z. Zhang, Q. Liu, and Y. Wang. Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[146] Z. Zhang, L. Yang, and Y. Zheng. Translating and Segmenting Multimodal Medical Volumes With Cycle- and Shape-Consistency Generative Adversarial Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9242–9251, June 2018.

[147] T. Zhou, S. Ruan, and S. Canu. A review: Deep learning for medical image segmentation using multi-modality fusion. *Array*, 3-4:100004, 2019.

[148] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Transactions on Medical Imaging*, 39(6):1856–1867, 2020.

[149] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[150] S. Zhu, Z. Dai, and N. Wen. Two-stage approach for segmenting gross tumor volume in head and neck cancer with ct and pet imaging. In *Head and Neck Tumor Segmentation*, pages 22–27, Cham, 2021. Springer International Publishing.

[151] X. Zou and Q. Dou. Domain Knowledge Driven Multi-modal Segmentation of Anatomical Brain Barriers to Cancer Spread. In *Segmentation, Classification, and Registration of Multi-modality Medical Imaging Data*, pages 16–26, Cham, 2021. Springer International Publishing.

[152] A. Zwanenburg, M. Vallières, M. A. Abdalah, H. J. W. L. Aerts, V. Andrearczyk, A. Apte, S. Ashrafinia, S. Bakas, R. J. Beukinga, R. Boellaard, and et al. The Image Biomarker Standardization Initiative: Standardized Quantitative Radiomics for High-Throughput Image-based Phenotyping. *Radiology*, 295(2):328–338, May 2020.