Segment Anything Semi-automatically in Volumetric Medical Images Efficiently

Effizientes halbautomatisches Segmentieren von beliebigen Strukturen in volumetrischen medizinischen Bildern

Jonas Kordt

Universitätsmasterarbeit
zur Erlangung des akademischen Grades

Master of Science
(M. Sc.)

im Studiengang
IT Systems Engineering
eingereicht am 15. April 2024 am
Fachgebiet Digital Health - Machine Learning der
Digital-Engineering-Fakultät
der Universität Potsdam

Erstgutachter	Prof. Dr. Christoph Lippert
Zweitgutachter	Prof. Dr. Bert Arnrich
Betreuer	Dr. Sumit Shekhar

Abstract

Volumetric medical imaging, such as computed tomography (CT) and magnetic resonance imaging (MRI), is an important tool in both clinical and research settings. To label raw image data for analysis, regions of interest are often segmented in the images. This is useful for gathering data during research and for diagnosis and treatment planning. However, segmenting these volumetric images manually is time-consuming and requires medical expertise, which makes it expensive.

Recently, the Segment Anything Model (SAM) has been introduced as a foundation model for promptable (semi-automatic) segmentation, allowing segmentation through simple point and bounding box prompts. While SAM has been trained on 2D natural images, it shows surprising zero-shot performance for unseen tasks. To further improve the performance of SAM on medical images, it has been fine-tuned and adapted to create specialized versions for medical images, such as MedSAM.

Although MedSAM improves upon SAM, it is still a 2D model, resulting in a tedious slice-by-slice segmentation workflow for volumetric images. Thus, we design a novel 3D workflow breaking this slice-by-slice procedure. To achieve this workflow, we propose prompt engineering strategies to generate system-prompts for MedSAM on surrounding slices. This way, users can segment a whole volumetric image while working on only a subset of its slices.

We evaluate our prompt engineering strategies on the diverse set of medical image datasets from the Medical Segmentation Decathlon challenge. We find that they can, with interactive run-time, significantly reduce the segmentation effort while only marginally reducing the segmentation quality compared to applying MedSAM slice-by-slice. Breaking out of the inherently time-consuming slice-by-slice workflow with only a minor reduction in segmentation quality is a significant step in streamlining semi-automatic volumetric medical image segmentation. Looking forward, this 3D workflow could utilize other 2D models, offering medical experts a flexible and powerful tool for semi-automatic volumetric image segmentation across various applications.

Zusammenfassung

Volumetrische medizinische Bildgebung, z. B. Computertomographie (CT) und Magnetresonanztomographie (MRT), ist sowohl im klinischen Umfeld als auch in der Forschung ein wichtiges Werkzeug. Um die Rohbilddaten für die Analyse zu kategorisieren, werden häufig wichtige Bereiche der Bilder segmentiert. Dies ist nützlich sowohl für Datenerfassung in der Forschung als auch für Diagnose und Behandlungsplanung. Das manuelle Segmentieren dieser volumetrischen Bilder ist jedoch zeitaufwändig und erfordert medizinisches Fachwissen, und ist daher teuer.

Kürzlich wurde das Segment Anything Model (SAM) als Basismodell für *promptable* (halbautomatische) Segmentierung eingeführt, das die Segmentierung durch einfache Punkt- und Bounding-Box-Prompts ermöglicht. Obwohl SAM auf natürlichen 2D-Bildern trainiert wurde, zeigt es eine überraschende Zero-Shot-Leistung für neue Aufgaben. Um die Leistung von SAM auf medizinischen Bildern weiter zu verbessern, wurde es optimiert und angepasst, um spezielle Versionen für medizinische Bilder zu erstellen, wie z. B. MedSAM.

Obwohl MedSAM eine Verbesserung gegenüber SAM darstellt, handelt es sich immer noch um ein 2D-Modell, was zu einem mühsamen Schicht-für-Schicht-Segmentierungs-Workflow für volumetrische medizinische Bilder führt. Daher entwerfen wir einen neuartigen 3D-Workflow, der dieses reine Schicht-für-Schicht-Verfahren durchbricht. Um dies zu erreichen, schlagen wir Prompt-Engineering-Strategien vor, um Systemprompts für MedSAM auf umliegenden Schichten zu generieren. Auf diese Weise kann der Benutzer ein volumetrisches Bild segmentieren, während er nur an einer Teilmenge der Schichten des Bildes arbeitet.

Wir evaluieren unsere Prompt-Engineering-Strategien an verschiedenen medizinischen Bilddatensätzen aus dem Medical Segmentation Decathlon. Wir stellen fest, dass sie bei interaktiver Laufzeit den Prompting-Aufwand erheblich reduzieren können, während sie die Segmentierungsqualität im Vergleich zur Schicht-für-Schicht-Anwendung von MedSAM nur geringfügig verringern. Der Ausbruch aus dem inhärent zeitaufwändigen Schicht-für-Schicht-Workflow bei nur geringfügiger Verringerung der Segmentierungsqualität ist ein bedeutender Schritt zur Optimierung der halbautomatischen volumetrischen medizinischen Bildsegmentierung. Der 3D-Workflow könnte zukünftig auch andere 2D-Modelle verwenden und bietet Medizinern ein flexibles und leistungsstarkes Werkzeug für die halbautomatische volumetrische Bildsegmentierung in verschiedenen Anwendungen.

Acknowledgments

First of all, I want to thank my thesis advisor Prof. Dr. Christoph Lippert, and my supervisor Dr. Sumit Shekhar for their continued support and assistance. They allowed me to follow my passion by giving me the freedom to work on what I deemed interesting and always had wonderful ideas and helpful feedback for me. Sumit, in particular, was a tremendous help for brainstorming ideas and writing about my research in a clear and structured way. Additionally, I want to thank Christoph for supporting the VISIAN project year after year at the Digital Health - Machine Learning chair, giving me the opportunity to combine my interests for machine learning and computer graphics in a meaningful project. Moreover, I want to thank Prof. Dr. Bert Arnrich for agreeing to review this thesis.

Furthermore, I want to thank Richard Keil for initiating the SAM integration into VISIAN in his bachelor thesis on the AutoSeg tool as well as sharing ideas with me which eventually led to this thesis.

On the topic of VISIAN, I want to thank Paul Brachmann, Maximilian Lindner, and Clara Uktar for being an inspiring and talented team to work with. The amount of fun we had together while working on VISIAN made the whole project even more special to me. Paul and Clara were wonderful developers to learn from and collaborate with, always open to a casual chat about both mundane and important topics. Max had a huge impact on both VISIAN and myself with his design expertise. Without him, VISIAN would not have progressed this far and would most certainly look quite different, to say the least. Additionally, Max was a great help with the graphical figures in this thesis.

Last but not least, I want to thank my family and friends (Gian, Toni, Lara, Adam, Leonie, ...) for supporting me during all my studies, being there for me when I need them, and making me who I am. Without such a wonderful social environment and diverse range of opinions, my personal and professional development – and, well, my life – would not have been the same. I would also like to thank my brother for always being there for me when I need him, and always understanding my problems. The biggest thanks, however, go to my parents for raising me to be independent, rational yet emotional, and curious, by giving me the perfect mix of freedom, support, and love.

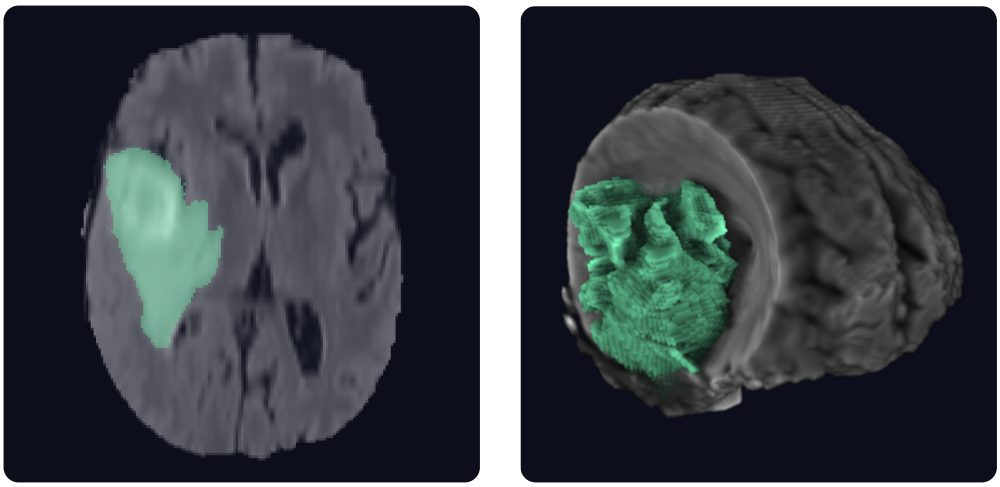
Contents

1 Introduction

Volumetric medical imaging, such as computed tomography (CT) and magnetic resonance imaging (MRI), is a crucial tool for medical practitioners in diagnosis and treatment planning processes [Wan+22]. It is extensively used by researchers to obtain statistical data about a population [Bai+20; Mil+20; Ren+20; Uhe+21; Wan+21]. The beauty of these images lies in the ability of physicians to gather data about the body's internal structures in a non-invasive way. CT imaging, through its X-ray-based technology, provides excellent clarity for dense tissues and can be pivotal in trauma cases where time is of the essence. MRI, using magnetic fields, excels in contrast resolution, making it particularly valuable for diagnosing conditions in the brain and spinal cord without exposing the subject to radiation risks. To properly analyze the images, it is often useful to segment the entire image or specific structures within the image. This means each voxel of the volume is assigned a label or class. That way, the raw image data is turned into labeled data, which is much easier to analyze.

Medical image segmentation is useful for a variety of use-cases. For example, it is used for counting and detecting cells [Fal+18], measuring and monitoring volumes of structures in the body [Kic+19; Mil+20], tumor analysis [Aer+14], treatment planning [Ros18], and diagnostic support systems [De +18]. While these use-cases are of very different nature, they all share a common need for high segmentation accuracy.

One use-case from the clinical setting which requires especially high segmentation accuracy is risk assessment and operation planning before brain tumor resection surgeries [Ros18]. During risk assessment, the tumor segmentation is put into the spatial context of important nerve strands in the brain. This way the risk of damaging these structures during the surgery can be estimated and the best angle for opening the skull can be found. Because it is hard to visually distinguish the tumor from the surrounding healthy brain tissue the segmentation itself is also used during the operation to guide the surgeon. For this guidance the segmentation can be viewed using traditional screens or the volume can be registered with the physical brain and viewed as a hologram in the actual brain by the surgeon using augmented reality glasses [SFN08]. Here, segmentation accuracy is crucial since even minimal removal or damage of healthy brain tissue during surgery can significantly affect a patient's mental and physical abilities.



(a) 2D slice-based view of one slice of the brain MRI and tumor segmentation; only a small part of the data is visible.

(b) 3D volumetric view of the whole brain MRI and tumor segmentation; a large part of the data is visible.

Figure 1.1: Brain tumor segmentation in a FLAIR MRI, rendered in 2D and 3D using VISIAN.

1.1 Motivation

Traditional manual segmentation of medical images is a tedious process that requires the expertise of a trained medical practitioner, such as a radiologist. Specifically, experts generally use drawing tools or outlining tools to color in specific parts (segments) of the image [Fed+12; Han+21; Kor+21; Wol+04; Yus+06]. Segmenting volumetric medical images manually is inherently time-consuming since the expert has to segment the image slice-by-slice in order to complete the volumetric segmentation (see Figure 1.1). While manual segmentation is the gold standard in terms of accuracy, it also introduces inter- and intra-observer variability [Cov+22; Ren+20], leading to inconsistencies in the data analysis and potentially affecting patient outcomes.

The advent of deep learning has revolutionized this task, offering automated solutions with the promise of consistency, speed, and accuracy [Ren+20]. However, while supervised learning approaches can achieve high-quality segmentation masks, training these models generally requires a large amount of segmented training data [SWS17; Wan+22]. Segmenting such a large amount of training data is time-consuming and expensive because it requires highly trained experts [Cru+21]. To avoid the need of large training datasets, semi-supervised and unsupervised

learning approaches have been applied to medical image segmentation [CBP19]. These approaches, however, come with their own challenges in the training process and often do not generalize well [CBP19]. Additionally, semi-supervised and unsupervised approaches usually cannot achieve the same high-quality segmentation results that make supervised models so desirable [Wan+22].

1.2 Problem Statement

Manual segmentation, though precise, has significant limitations: it requires considerable time (particularly for volumetric images that need segmentation on a slice-by-slice basis), incurs high costs, and is prone to inter- and intra-observer variability [Cov+22; Cru+21; Ren+20]. These factors can compromise the reliability of data and the quality of patient care. Automation through supervised learning approaches for segmentation, while addressing these time, cost, and consistency issues [Ren+20], still heavily relies on pre-existing segmented data for training [SWS17; Wan+22], creating a paradoxical situation in which the solution itself is part of the problem. Semi-supervised and unsupervised learning can reduce this need for training data, but struggle to deliver the necessary accuracy [CBP19].

Semi-automatic segmentation presents a practical pathway to efficiently acquire the necessary segmented data for supervised learning algorithms. By combining the precision and control of manual segmentation with the speed and consistency of automated algorithms, semi-automatic segmentation promises to expedite the process of dataset creation [Ram+18] without compromising on data quality. Semi-automatic segmentation tools which work in a volumetric fashion can even further decrease the manual effort, because volumetric medical images no longer have to be segmented slice-by-slice [Kor+21]. However, due to the particularly high variability in volumetric medical images [Ant+22], it can be difficult for semi-automatic segmentation tools to generalize well to many different segmentation tasks.

Overall, the three main requirements we identify for a semi-automatic volumetric medical image segmentation tool for both creation of large training datasets for automatic segmentation methods and for efficient single case segmentation are the following:

(R1) **Segment Anything.** Generalizing to a variety of different medical images and segmentation targets.

(R2) **Reduce Effort.** Requiring less time and effort from domain experts by breaking the typical slice-by-slice segmentation workflow.

(R3) **Maintain High Segmentation Quality.** At the same time, maintaining high segmentation quality which is crucial for medical applications.

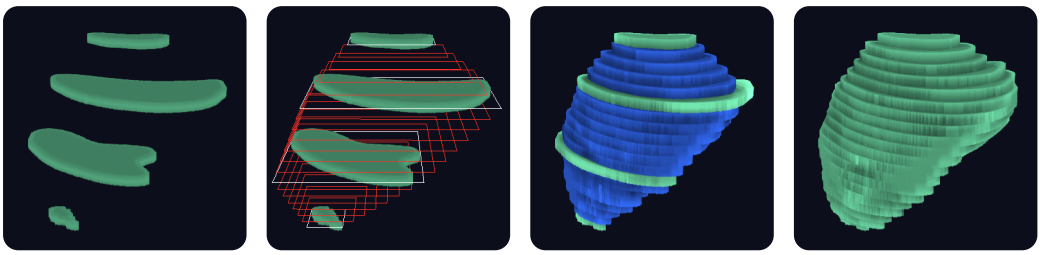
The challenge, therefore, lies in developing an accurate, semi-automatic segmentation tool that is versatile enough to adapt to a variety of medical imaging scenarios and that can break the slice-by-slice segmentation workflow in order to reduce the segmentation effort.

1.3 Approach and Contributions

Traditional semi-automatic segmentation of medical images relies on non learning-based algorithms, such as thresholding [Ram+18], region growing [AB94; Kor+21; LJT00], the watershed algorithm [BL79], and more [Ram+18]. However, recently the *Segment Anything Model* (SAM) [Kir+23] has been introduced. This promptable segmentation model allows the user to segment structures using point, bounding box, and text prompts.[1] Because of SAM's surprising zero-shot performance, it shows promise in solving many segmentation tasks. However, as SAM is trained on natural images, it tends to struggle with medical images where boundaries are typically softer than in natural images [Ma+24]. Nevertheless, Ma et al. [Ma+24] introduced a fine-tuned version called *MedSAM* which shows decent zero-shot performance across a variety of medical image datasets.

While MedSAM achieves relatively high accuracy on medical images, because of its 2D nature it still requires a slice-by-slice workflow where the expert must draw

1 Text prompts are mentioned in the paper but are still experimental and are currently not part of the published code.



(a) Initial slices confirmed by user. **(b)** Prompts generated by system (red). **(c)** Segmentation suggestions (blue). **(d)** Ground truth segmentation.

Figure 1.2: Steps of one iteration from applying our 3D workflow to spleen segmentation. Confirming more slices iteratively improves the suggestion quality in **(c)**.

a bounding box around the region of interest on every single slice. In this thesis, we design a workflow which breaks the traditional slice-by-slice segmentation procedure (3D workflow) and requires significantly less effort from experts. To enable that 3D workflow we propose a novel prompt engineering method which only requires the user to provide prompts on a subset of all slices of the volumetric image. The main steps of the 3D workflow are shown in Figure 1.2. We explain them in more detail in Chapter 4.

Our main contributions, addressing the identified requirements (R1), (R2), and (R3), are the following:

(C1) **Workflow Design.** We design a 3D workflow applying MedSAM [Ma+24] to volumetric medical image segmentation which requires significantly less effort from experts compared to using MedSAM slice-by-slice.

(C2) **Prompt Engineering.** We introduce a novel prompt engineering method based on bounding box interpolation to reduce effort in the 3D workflow applying MedSAM.

(C3) **Evaluation.** We demonstrate the feasibility of the 3D workflow, by showing that its resulting segmentation quality is comparable to state-of-the-art but with a better effort-quality trade-off. We show this by evaluating our method on the diverse volumetric medical image datasets used in the Medical Segmentation Decathlon challenge [Ant+22].[2]

1.4 Thesis Overview

The rest of the thesis is structured as follows. In Chapter 2, we give a general overview of the medical image segmentation background and review related work. Next, Chapter 3 further explores concepts relevant for our prompt engineering method. In Chapter 4, we design the 3D segmentation workflow and present our novel prompt engineering method to enable it. This prompt engineering method is then evaluated in Chapter 5. In Chapter 6, we discuss strengths and limitations of our method in order to establish guidelines for when to use it. We present future work in Chapter 7 and conclude this thesis in Chapter 8.

2 We evaluate the segmentation quality and runtime performance of our method in Chapter 5. User evaluation, which requires a full implementation of the workflow, remains future work (see Chapter 7).

2 Background and Related Work

To lay a foundation for a semi-automatic 3D segmentation workflow based on MedSAM and the corresponding prompt engineering method, we will start with a general overview of the medical image segmentation background. Additionally, we will review related work that resonates with our approach or attempts to solve a similar problem.

2.1 Volumetric Medical Imaging

In order to form a better understanding of the images we are working with, we will begin with an overview of volumetric medical images. Medical images in general have unique properties compared to natural images [Ma+24; She+21; Wen+21; XRV19]. This difference is even more pronounced in volumetric medical images, which introduce a whole third dimension to the image. While there are various types of volumetric medical images, such as magnetic resonance imaging (MRI), computed tomography (CT), and positron emission tomography (PET), we will focus on two very commonly used imaging types in this thesis: MRI and CT.

2.1.1 MRI and CT Acquisition

MRI scanners use a strong magnetic field to align atomic nuclei, such as hydrogen, within the human body (or any other subject) and then excite these nuclei using a radio-frequency pulse. When these nuclei return to their original state they emit energy which is detected and translated into a volumetric image using Fourier transformation [Gro+15]. Various different techniques including T1- and T2-weighted imaging, Magnetization Transfer (MT) Imaging, and Fluid Attenuated Inversion Recovery (FLAIR) can be used, each providing distinct image characteristics. For example, FLAIR images suppress the cerebrospinal fluid (CSF) signal to better reveal lesions near the ventricles, unlike T1 images where CSF appears similar to brain tissue [Mus+21].

In contrast, CT scanners utilize a series of x-ray images taken from all angles around the subject. From this series of 2D images, a volumetric image is recon-

(a) No windowing is used. The large range of values results in little contrast overall.

(b) Windowing is used to enhance contrast in soft tissue. Little contrast in the bones.

(c) Windowing is used to highlight bones. Soft tissue is not visible anymore.

Figure 2.1: Abdomen CT with different window settings. The raw image comes from the Medical Segmentation Decathlon dataset (task 09) and is rendered using VISIAN.

structed slice-by-slice, by optimizing all the attenuation levels of the relevant voxels to match the measurements from the x-ray images as closely as possible [Gol07].

2.1.2 MRI and CT Properties

One common property of both MRI and CT images is their variable resolution. The resolution not only varies between images, but can also vary within an image, such that the voxel size is anisotropic. This results in a higher image resolution on a slice than along the slicing direction and in turn in cuboid-shaped voxels instead of cube-shaped voxels.

However, due to MRI and CT measuring different properties of the subject and using different technology to do so, the resulting images have different properties too. MRI, in particular, does not have a standardized measurement scale. This means that MRI intensities of the same subject can vary across scans, even if the same scanner is used [NUZ00].

CT, on the other hand, measures attenuation in the standardized Hounsfield unit. Attenuation values are relative to the one of water which has a value of zero; for example, air has an attenuation value of −1000, while bone has values of +1000 and more [Gol07]. Due to this large range of values, a lot of detail is lost when the range is squashed into the common 8-bit gray-scale range of typical monitors, which only support 256 different gray levels. To counteract this, radiologists use a process known as *windowing*. Essentially, windowing applies a min-max-clamping to the native CT value range. First, the user selects a minimum value and a maximum value of interest. In practice, the user usually selects the window level (middle of

the window) and the window width, both of which are then used to calculate the minimum and maximum values. Then, all values below the minimum value are clamped to the minimum, and all values above the maximum value are clamped to the maximum. The resulting range from the minimum to the maximum, is then squashed into the 8-bit gray-scale range. The result is more detail and contrast in the range of interest. For an example of different window settings, see Figure 2.1. Because of the standardization of the Hounsfield scale, standard window settings (window level and width or minimum and maximum value respectively) per tissue type are often used across scans and scanners [BM17]. As mentioned, the MRI scale lacks this sort of standardization. Thus, standard windows for different tissue types are not possible [Wah+21].

This difference in standardisation highlights the complex variability in medical images and the resulting need for robust segmentation techniques.

2.2 Volumetric Medical Image Segmentation

In the field of volumetric image segmentation there are three overarching approaches. Namely manual segmentation, semi-automatic segmentation, and automatic segmentation.

2.2.1 Manual and Semi-automatic Segmentation

Many different manual and semi-automatic tools are usually combined in a single segmentation application. The landscape of segmentation applications includes both free-to-use software and commercial software. Among free-to-use software, options include ITK-SNAP[3] [Yus+06], MITK[4] [Wol+04], 3D-Slicer[5] [Fed+12], and VISIAN[6] [Kor+21]. A commercial alternative is Encord[7] [Han+21]. For screenshots of VISIAN see Figure 3.1. Screenshots of ITK-SNAP, MITK, and 3D-Slicer are included in Appendix A.

Popular manual segmentation tools that these applications include are simple pixel brushes and outlining tools. Semi-automatic segmentation tools, which reduce manual effort, combine user input with various automated algorithms, such as

3 http://www.itksnap.org/
4 https://www.mitk.org/
5 https://www.slicer.org
6 VISIAN is developed by us and our colleagues at HPI and is available at https://visian.org.
7 https://encord.com/

region-growing, thresholding, dilation and erosion, or even more complex machine learning approaches.

While specifically the semi-automatic tools can satisfy some of the requirements identified in Section 1.2, they usually fall short in at least one of them. Region-growing tools, for example, struggle with soft borders and thus don't generalize very well (R1). They can, however, reduce effort and even break the slice-by-slice workflow (R2) if the region growing is applied in a 3D manner [Kor+21], but can also struggle to maintain high segmentation quality (R3) resulting in a need for significant manual cleanup. Other semi-automatic tools come with similar trade-offs.

In Section 3.1 we explore the specific tools available in VISIAN in more detail and also highlight relevant tools from other applications where appropriate. Additionally, we relate the semi-automatic tools to the requirements (R1) to (R3).

However, the field of medical image segmentation does not end with manual and semi-automatic tools. Of course, there are plenty of fully automatic segmentation approaches which try to eliminate manual segmentation effort completely.

2.2.2 Automatic Segmentation

Automatic segmentation of medical images initially started out with manually defined algorithms which incorporated mathematical models and rules based on domain knowledge [WK07]. A prominent example is FreeSurfer,[8] which is an open-source software suite designed for analyzing and visualizing brain MRI. It offers a processing pipeline which, among other steps, includes segmentation [Fis12], and which has been widely used by researchers [Dew+10; Egg+12; May+16; Son+23].

More recently, deep learning techniques have been used for automatic medical image segmentation, and outperformed the traditional algorithms, such as FreeSurfer [Son+23]. Among deep learning techniques for medical image segmentation, Convolutional Neural Networks (CNNs) initially became popular and are still one of the state of the art techniques [Wan+22]. A notable CNN architecture in this context is U-Net [RFB15]. It encodes an image through contracting layers, then up-samples it and uses skip-connections for more detailed segmentation (see Figure 2.2). Isensee et al. [Ise+20] introduced nnU-Net, a self-configuring framework based on 2D and 3D versions of U-Net, and won the Medical Segmentation Decathlon challenge [Ant+22] with it.

Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) networks [HS97], have been adapted for medical imaging tasks involving sequential

8 https://surfer.nmr.mgh.harvard.edu

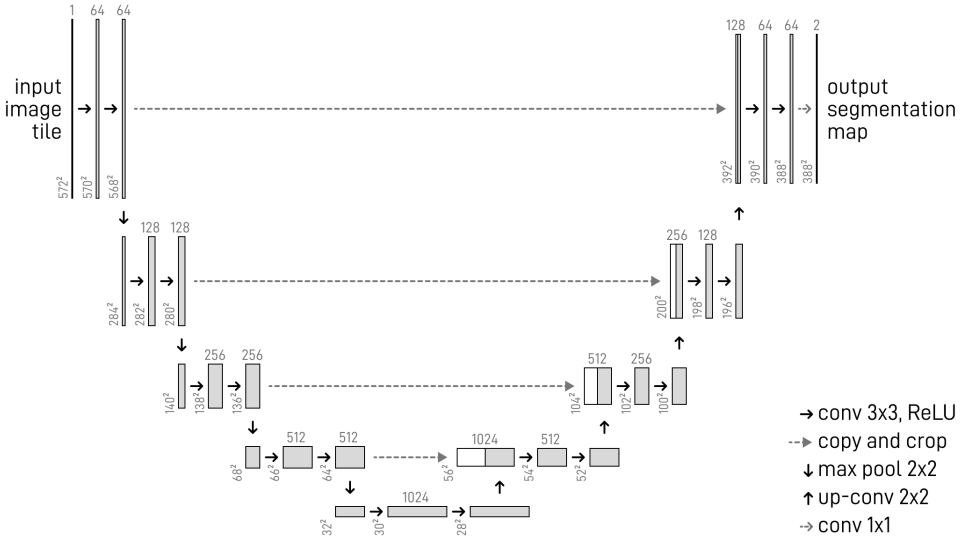


Figure 2.2: Architecture of U-Net. Note the combination of down-sampling and up-sampling layers leading to a U-shaped architecture. The skip-connections (copy and crop) introduce fine-grained detail into the segmentation in the up-sampling process. (Figure adapted from [RFB15].)

data [Wan+22]. This is particularly relevant in dynamic MRI studies, such as functional MRI (fMRI), where capturing temporal changes is crucial [Ang+18; LLC20; LSQ22]. LSTMs are adept at processing these time-related variations, offering insights into temporal aspects of the medical images [Gao+18].

Transformers, which were initially developed for natural language processing by Vaswani et al. [Vas+17], have also been applied to medical image segmentation [Che+21; Xia+23]. The Vision Transformer (ViT), which was developed for image classification (see Figure 2.3), for example, employs self-attention mechanisms to effectively manage spatial relationships in images [Dos+20].

In addition to the supervised learning methods presented above, semi-supervised and unsupervised learning methods are increasingly relevant, especially in situations where annotated data is scarce [Wan+22]. Semi-supervised approaches, blending labeled and unlabeled data, have shown promise in medical image segmentation, offering a balance between supervised and unsupervised methods [CBP19]. Similarly, unsupervised learning techniques, such as autoencoders, are being explored for their capability to derive meaningful features from unlabeled medical images, which can then be applied in segmentation tasks [RS21]. However, semi-supervised and unsupervised approaches can usually not achieve the same high quality segmentation results which supervised models achieve [Wan+22].

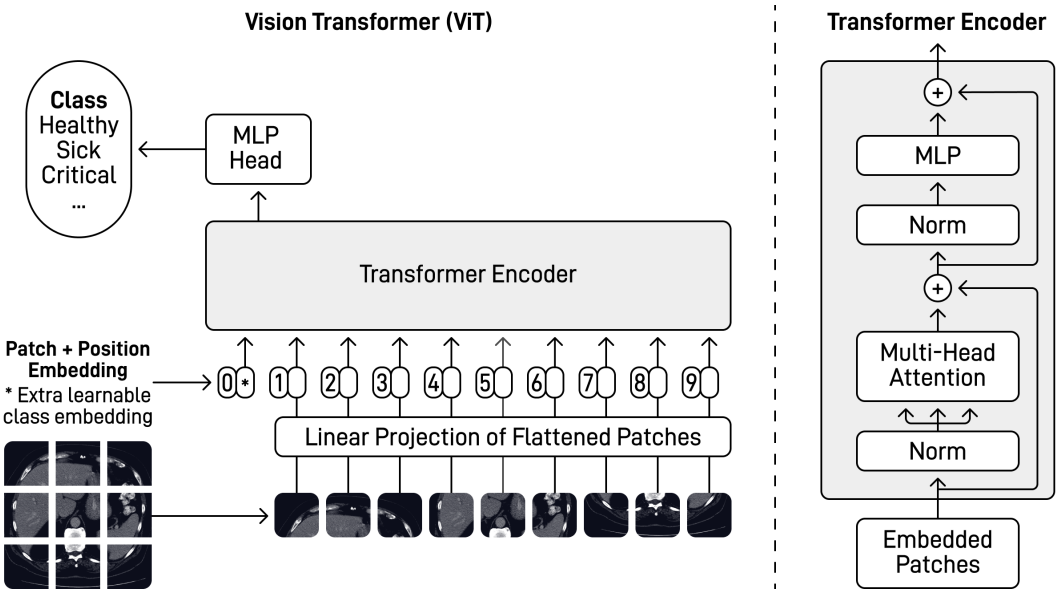


Figure 2.3: Vision Transformer (ViT) architecture with an attached multi-layer perceptron (MLP) head for image classification. Alternatively, a segmentation head can be attached. (Figure adapted from [Dos+20].)

All these automatic segmentation methods have in common that they reduce the segmentation effort for experts in day to day use drastically. Instead of segmenting the whole volume manually or semi-automatically, the results from the automatic approaches only have to be manually checked and possibly corrected. However, apart from semi-supervised and unsupervised methods, which lack the desired quality [Wan+22], the effort is at least partly shifted to the algorithm creation. In particular, supervised learning approaches usually require a lot of segmented training data which has to be created manually or semi-automatically. Thus, semi-automatic segmentation methods continue to stay relevant in parallel with the rise of fully automatic deep learning methods. More recently, the effort to create powerful and semi-automatic segmentation tools has led to promptable, deep-learning-based segmentation methods.

2.3 Promptable Segmentation Using Deep Learning

In the surge of large language models, in addition to the transformer-based architecture, the prompting approach has been transferred to segmentation. The idea is to interactively give hints (so called prompts) to the model about what it should

segment. Additionally, prompt engineering can be used to further automate the segmentation process by automatically generating all or some of the prompts.

In this realm, Kirillov et al. [Kir+23] introduced the Segment Anything Model (SAM). It is trained on more than 1 billion segmentation masks for natural images and consists of 3 main parts: the image encoder, the prompt encoder, and the mask decoder (see Figure 3.4).

Kirillov et al. [Kir+23] present remarkable zero-shot performance of SAM on variety of different segmentation tasks. However, they also identify limitations, such as missing fine structures, hallucinating small disconnected components, and having less crisp borders than other methods.

SAM has also been applied to medical images even though it was trained on natural images. However, despite good zero-shot performance on natural images, multiple studies have found the zero-shot performance on medical images to be generally lower than state of the art deep learning models and varying a lot depending on the dataset [He+23; Maz+23; Roy+23], i.e. SAM struggles with generalization (R1) and segmentation quality (R3) when applied to medical images, and naturally cannot break the slice-by-slice workflow for volumetric medical images (R2), as it is a 2D model.

To improve SAM's performance and generalization on medical images ((R1) and (R3)), various fine-tuning and adaptation approaches have been proposed. MedSAM [Ma+24] fine-tunes SAM's image encoder and prompt decoder using over 1.5 million medical image masks and applies specific preprocessing for different image types. Medical SAM Adapter (MSA) [Wu+23] introduces Low-Rank Adaptation (LoRA) with additional adapter blocks in SAM's architecture, enhancing 3D image processing. SAM-Med2D [Che+23a], an enhanced version of SAM with adapter layers, is fine-tuned with a very large dataset of 2D medical images and segmentation masks [Ye+23], yet shows lower dice scores than other methods. SAM-Med3D [Wan+23] extensively modifies SAM for 3D images, trained from scratch for improved segmentation quality. It lifts SAM's architecture to the third dimension, inherently reducing effort (R2) by generating 3D segmentations from single prompts, however, as we will see in Chapter 5, it struggles to maintain high segmentation quality (R3). 3DSAM-adapter [Gon+23], tailored for 3D images, reuses and adapts SAM's components, showing superior performance on most datasets compared to SAM and other benchmarks.

The idea of promptable segmentation has also been transferred to other deep learning approaches which are not based on SAM. One example is the One-Prompt Segmentation [WX23], which combines the strengths of one-shot methods and

prompting. During inferencing, the model is capable of segmenting an unseen task with only one prompted example in a single forward pass.

Additionally, Ma et al. [Ma+24] suggest a promptable version of nnU-Net [Ise+20] and use it as a benchmark for MedSAM. The promptable nnU-Net encodes a bounding box prompt in a binary mask and uses this mask as a second image channel during training and inferencing.

3 Conceptual Overview

After our more general presentation of the medical image segmentation background and review of related work in Chapter 2, we will now explore the concepts which are relevant for our prompt engineering method and its integration into a segmentation application in more detail.

3.1 Volumetric Medical Image Segmentation

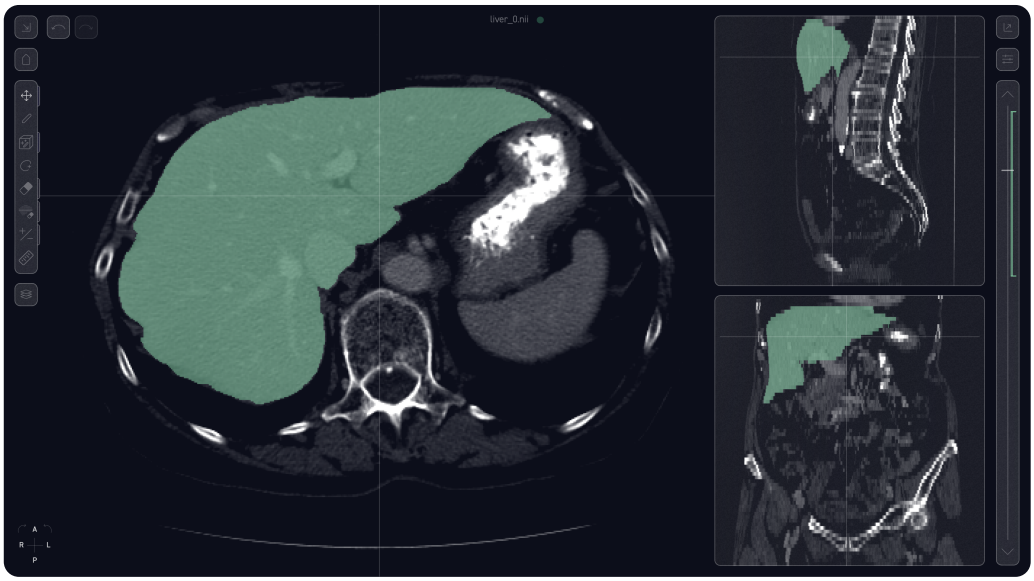
To segment volumetric medical images medical domain experts use a variety of different applications, such as ITK-SNAP [Yus+06], MITK [Wol+04], 3D-Slicer [Fed+12], VISIAN [Kor+21], and Encord [Han+21].

VISIAN in particular, is a modern, browser-based segmentation editor for volumetric medical images. It was developed with a focus on an intuitive user experience with the goal of reducing segmentation effort for medical experts. While it can be accessed with little effort using a normal web browser,[9] the sensitive medical data stays on the users machine as long as no machine learning integration is used. In addition to using VISIAN on a computer or laptop, it can be used on a tablet. For this, it has full multi-touch and pen support, allowing for a very natural and precise pen drawing interaction during the segmentation process. VISIAN's 2D view is comprised of a large main view used for segmentation and two optional smaller view panels (see Figure 3.1 (a)).[10] The transverse, sagittal, and coronal views of a volumetric medical image can be arranged across these three panels. Additionally, a 3D view with a variety of rendering options is available, allowing the user to explore the image and the segmentation as a whole (see Figure 3.1 (b) to 3.1 (d)).

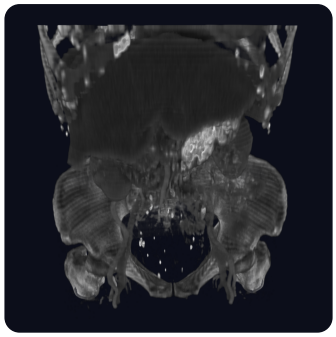
While each of the mentioned applications has its own advantages, we will now explore the tools available in VISIAN. Additionally, we will reference relevant tools from other applications where appropriate.

9 VISIAN can be used online, free of charge at https://app.visian.org.

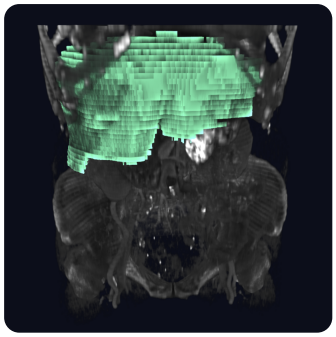
10 For comparison, screenshots of the ITK-SNAP, MITK, and 3D-Slicer interfaces are available in Appendix A.



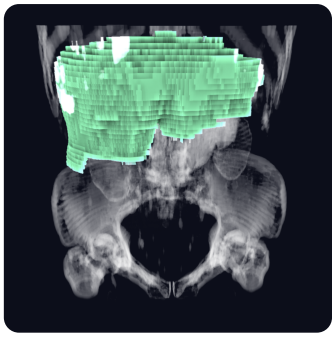
(a) 2D user interface in VISIAN. The most space is given to the main view for segmentation editing. The side panels show the image from different view directions. The interface includes a toolbar and action buttons in the top left, a coordinate system for orientation in the bottom left, and a slider to navigate through the slices of the volume on the right. The slider highlights the range of slices including a part of the segmentation volume.



(b) Rendering the image intensities without the segmentation in 3D.

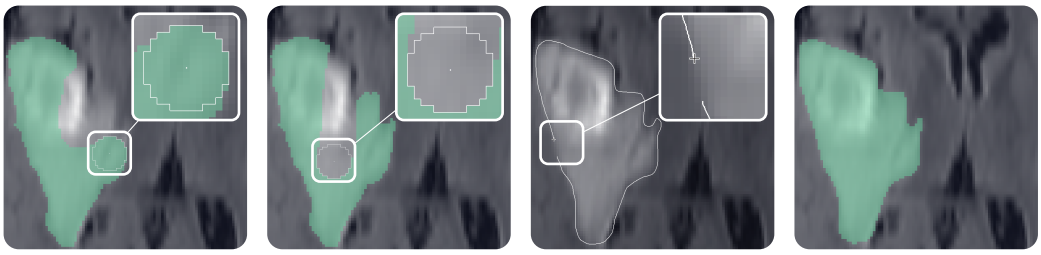


(c) Combining the image intensities with the segmentation in 3D.



(d) Rendering edges of the image intensity volume around the segmentation in 3D.

Figure 3.1: VISIAN user interface showing an abdomen CT with a liver segmentation in the 2D view and in the different rendering modes of the 3D view. The 3D view is separate from the 2D view. The different 3D views can be individually customized to reveal relevant parts of the image depending on the use-case.



(a) The *Brush*. **(b)** The *Eraser*. **(c)** The *Outline Tool*. **(d)** The outline result.

Figure 3.2: Segmenting a brain tumor using different manual tools in VISIAN.

3.1.1 Manual Segmentation Tools

Manual segmentation of volumetric medical images is done slice-by-slice. This means, that the expert views one 2D slice of the volumetric image at a time (see Figure 3.1 (a)).

The tools for manual image segmentation are quite intuitive. In VISIAN, two manual tools are available, the *Brush* and the *Outline Tool*. The *Brush* can be set to different sizes and is then used to *color in* or *erase* (see Figure 3.2 (a) and 3.2 (b)). The *Outline Tool* is used to draw an outline around the region of interest (see Figure 3.2 (c)). This outline is then converted to a pixel-accurate binary mask (see Figure 3.2 (d)). Additionally, shortcuts and buttons are available to clear a whole slice of the segmentation or the whole volume.

Other applications have very similar manual tools, some with slight alterations, for example in the brush shape. However, the landscape of tools becomes more diverse with semi-automatic tools.

3.1.2 Semi-automatic Segmentation Tools

VISIAN has various different semi-automatic segmentation tools, designed to reduce the segmentation effort for users. The first group of semi-automatic segmentation tools builds on region growing.

The basic 2D region growing tool in VISIAN is called *Smart Brush*. Essentially, the user configures a region growing threshold and selects a region growing seed by drawing with a normal pixel brush. The region is then grown from the seed based on the configured threshold.

The *Bounded Smart Brush* expands on this functionality. It introduces a configurable bounded region growing area which moves with the cursor (see Figure 3.3 (a)). The user still places a seed by drawing, but every time the mouse is moved while

drawing the seed, the region is immediately grown within the configured bounds. This gives a bit more control over the region growing process.

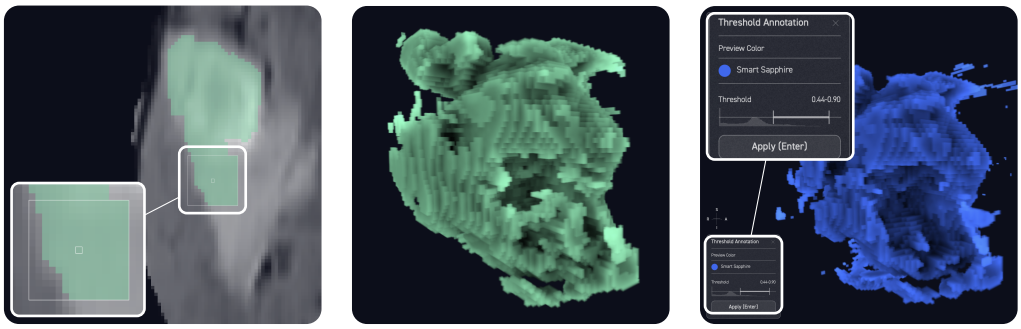
While most segmentation applications mainly include slice-by-slice tools, such as the Smart Brush and Bounded Smart Brush, they usually also include 3D tools. In the case of VISIAN, we introduced the *3D Smart Brush* in Kordt et al. [Kor+21], which expands the region growing process to the third dimension. The user still configures a seed and threshold but the region can then grow in all three dimensions (see Figure 3.3 (b)). While this allows to segment a whole volumetric structure in an image all at once, it can lead to more significant region growing errors where the grown region escapes the desired structure in the image. Thus, an interactive tuning process allows the user to scale back how far the region may grow from the seed based on a preview in both a 2D and 3D view. This is explained in more detail in [Kor+21].

As mentioned in Section 2.2.1, these region-growing tools, struggle with soft borders and thus don't generalize very well (R1). They can, however, reduce effort and the 3D-Smart-Brush can break the slice-by-slice workflow (R2), but struggles to maintain high segmentation quality (R3) resulting in a need for significant manual cleanup.

The second group of semi-automatic segmentation tools in VISIAN generates and manipulates segmentations procedurally. The *Threshold* tool, allows the user to select upper and lower image intensity thresholds (see Figure 3.3 (c)). All voxels between these thresholds are then highlighted in a preview which allows the user to judge the resulting segmentation before accepting the result. This thresholding approach can break the slice-by-slice workflow (R2), however, it is prone to produce noisy segmentations with small holes or small artefacts outside the desired segmentation, i.e. it struggles with segmentation quality (R3).

The *Dilate/Erode* tool is useful to clean up this noise. It works by applying an iterative dilation followed by the same amount of erosion (or the other way around) to the whole volumetric segmentation. If the dilation is applied first, holes in the segmentation are filled in and the whole segmentation expands outwards. The erosion step then does not reintroduce the holes because there are no adjacent background voxels anymore, but reverses the expansion on the edges of the segmentation. If applied the other way around, the erosion first removes small disconnected structures from the segmentation but also shrinks the larger structures. The following dilation then expands these larger structures back to their original size.

ITK-SNAP, for example, has a more advanced semi-automatic segmentation tool based on thresholding and active contours [Yus+06]. The user first isolates the



(a) Bounded Smart Brush. Region growing only happens within the outer bound of the brush cursor.

(b) Result of 3D Smart Brush. The segmentation has holes and rough around the edges, but does capture detail.

(c) Preview and interface of the Threshold tool. The segmentation is noisy and has prominent holes.

Figure 3.3: Semi-automatic segmentation tools in VISIAN. In particular, Bounded Smart Brush, 3D Smart Brush, and Threshold tool. Note the noisy results for the 3D tools.

structure of interest as best as possible using upper and lower thresholds. Then, the active contours are initialized with manually placed seeds which are interactively grown to fill the structure of interest. While this tool breaks the slice-by-slice workflow (R2), its application can be a bit cumbersome and is not easy to learn. Additionally, while it often produces high quality segmentation (R3), it can struggle to generalize to more complicated structures (R1).

The next step beyond semi-automatic segmentation tools are fully automatic methods, which can reduce segmentation effort even further. How these can be used by medical domain experts in practice is what we discuss next.

3.1.3 Integration of Automatic Segmentation Methods

To make automatic segmentation methods available to medical experts in a convenient way, they too have been integrated into segmentation applications. This tight integration has the benefit that manual segmentation tools can easily be used to correct the output from automatic segmentation. This is necessary because even though automatic segmentation methods have become surprisingly accurate, the resulting accuracy is not always sufficient for medical applications [Fu+21].

For instance, VISIAN has been integrated with MIA[11] (Medical Image Annotation Platform). MIA uses machine learning models to provide segmentation recommen-

11 https://mia-ai.vercel.app

dations to users. These recommendations can be examined and corrected in VISIAN before being confirmed by the user. Confirmed segmentations are then used by MIA to improve the existing recommendation model.

Encord, built for creating large training datasets for machine learning, naturally also includes an extensive AI integration. A variety of *micro models* can be created by the user to solve annotation tasks, such as instance segmentation. Additionally, an active learning workflow similar to the one in MIA is available.

Now that we have explored the landscape of volumetric medical image segmentation, including manual, semi-automatic, and fully automatic methods, in the next section we move on to the Segment Anything Model which, while being targeted at natural images, also marks the most recent development in the medical image segmentation domain.

3.2 Segment Anything Model

The Segment Anything Model (SAM) was introduced by Kirillov et al. [Kir+23] and is a foundation model for promptable segmentation. It is trained on more than 1 billion segmentation masks for natural images. With the resulting astonishing zero-shot performance in natural image segmentation, it did not take long for SAM to be applied to medical images as well. Before we explore SAM's generalization abilities to medical images, we will begin with an overview of its architecture and the workflow it enables.

3.2.1 Architecture and Workflow

SAM consists of 3 main parts: the image encoder, the prompt encoder, and the mask decoder (see Figure 3.4).

The image encoder is responsible for the image embedding. This embedding is generated using a pre-trained ViT. The embedding only has to be generated once per image and can then be continuously reused for multiple prompts.

The prompts are processed by the prompt encoder. Possible prompts are points (foreground and background), a bounding box, text, and a preexisting segmentation mask to be refined. Points and bounding boxes are represented by positional encodings [Tan+20] and summed with a learned embedding for each type of prompt. Text is encoded with a CLIP encoder [Rad+21], however, in the released code, text prompts are excluded as they remain experimental. Lastly, convolutions are used

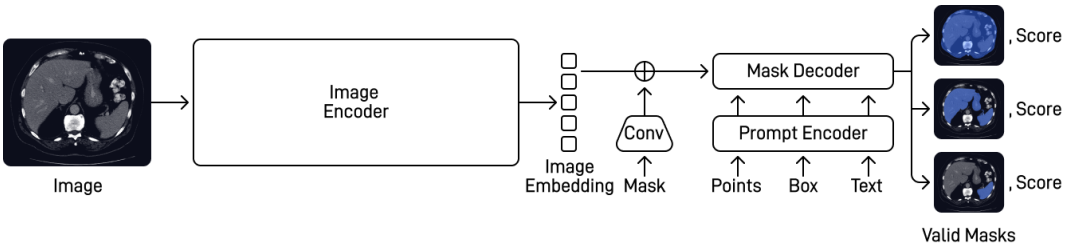


Figure 3.4: Segment Anything Model (SAM) architecture for promptable segmentation. The tree main parts are the image encoder, the prompt encoder, and the mask decoder. (Figure adapted from [Kir+23].)

to embed preexisting segmentation masks. The resulting mask embedding is then element-wise added to the image embedding.

To compute the resulting segmentation mask, the mask decoder interprets the image embedding (with the optionally included preexisting mask embedding) and the prompt encodings. The mask decoder consists of a modified Transformer decoder block [Vas+17] and a dynamic mask prediction head. In order to resolve ambiguity, the mask decoder actually predicts 3 masks and ranks them with a score. Otherwise, multiple valid masks could end up being averaged, resulting in lower mask quality.

The split between heavyweight image encoder and lightweight prompt encoder and mask decoder allows for an interesting workflow. The heavyweight image encoder has to run on a high performance GPU in order to run efficiently. On the other hand, the lightweight prompt encoder and mask decoder can run in a web browser on a consumer-grade CPU. This means that given a GPU server, users can interactively use SAM in a web browser on consumer-grade machines.[12]. This is particularly powerful because the same image encoding can be used with many different prompts, allowing for a tight feedback loop on the consumer machine without needing the high performance GPU server in the feedback loop. This split between server and consumer machine is a strong contrast to models, such as U-Net or a basic Vit, where the architectures consist of just one heavyweight part (see Figure 2.2 and 2.3) which is evaluated all at once, usually on a high performance GPU. Next we explore how this unique workflow translates to applying SAM to medical images.

12 A comprehensive demonstration of the described workflow is available at https://segment-anything.com/demo.

3.2.2 Application to Medical Images

Due to the convenient interactive workflow and the promising zero-shot performance on natural images, SAM has also been applied to medical images. However, studies have found varying levels of accuracy. For instance, Mazurowski et al. [Maz+23] apply SAM to 19 different medical image datasets and get a lot better results on structures with clearly defined borders, such as organs in CT images, and worse results on structures with less defined borders, such as brain tumors. Additionally, they find that bounding box prompts perform much better than point prompts and see little improvement when iteratively adding more point prompts. Similarly, He et al. [He+23] test SAM on 12 medical image datasets and find SAM's segmentation quality to be significantly lower than state of the art deep learning models. Roy et al. [Roy+23] limit their experiments with SAM on medical images to CT images. However, they test 14 different CT datasets and get rather good results when using bounding box prompts. Additionally, they discover that small inaccuracies in the bounding box prompt only result in a small reduction in segmentation quality. However, they find that SAM cannot match the results achievable with nnU-Net.

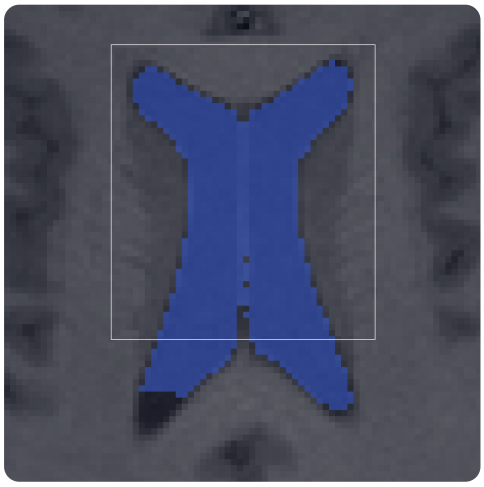
Nevertheless, a big strength of SAM is the fact that, while other deep learning alternatives have to be trained for the specific use-case, SAM is able to generalize to unseen tasks surprisingly well. This generalisation ability and the rather good results especially on CT images have prompted segmentation application developers to integrate SAM-based semi-automatic segmentation tools. For instance, the AutoSeg tool allowing a combination of bounding box and point prompts for SAM has been integrated into VISIAN [Kei23] (see Figure 3.5). Similarly, SAM-based tools have been integrated into 3D-Slicer [Liu+23] and Encord.[13] These SAM-based tools, however, struggle with breaking the slice-by-slice workflow for volumetric medical images (R2) and inherit SAM's struggles with generalizing to a variety of medical image segmentation tasks (R1).

3.2.3 Fine-tuning and Adaptation for Medical Images

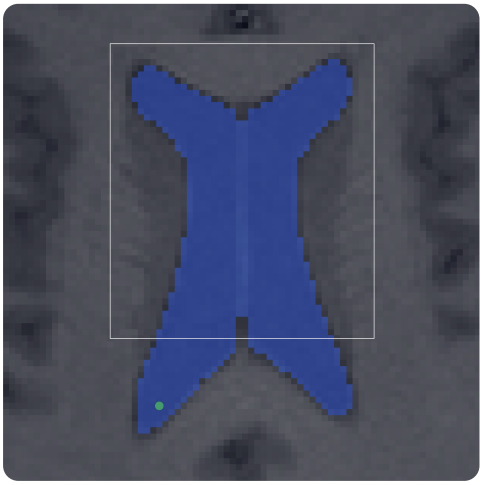
Since SAM still struggles with more difficult medical images [He+23; Maz+23; Roy+23] many approaches for fine-tuning and adaptation have been proposed. One popular fine-tuning approach is MedSAM [Ma+24], for which the image encoder and mask decoder were fine-tuned using over 1.5 million medical image segmentation masks for a variety of different images. The prompt encoder remains unchanged from SAM. During fine-tuning, only bounding box prompts were used.

13 https://encord.com/blog/segment-anything-live-in-encord/

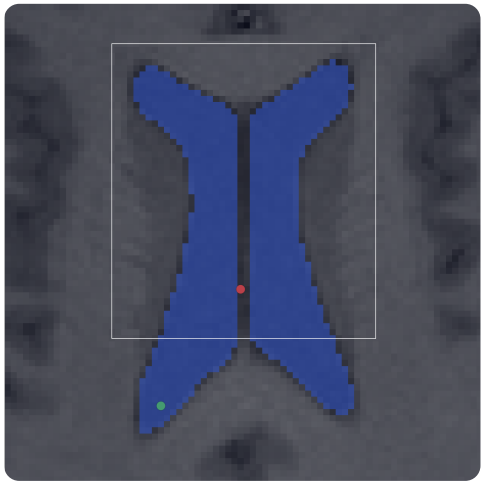
(a) The input slice of the brain MRI.



(b) Prompting with a flawed bounding box.



(c) Prompt refinement with a positive point.



(d) Prompt refinement with a negative point.

Figure 3.5: Brain ventricle segmentation in VISIAN using the SAM-based AutoSeg tool introduced in [Kei23]. Note the iterative improvement achieved by adding additional point prompts.

In addition to the fine-tuned model, MedSAM also uses specific pre-processing for the medical images. Firstly, for CT images, the standard windows discussed in Section 2.1.2 are applied to achieve more contrast. Since the same is not possible for MRI images (see Section 2.1.2), a more general percentile-based normalization is applied. In particular, the top and bottom 0.5 percentiles of the MRI intensity values are clamped. In a similar fashion to the CT windowing, this increases the contrast in the image, though with a smaller effect. Before a single image slice is processed by the image encoder, the slice itself is again normalized to a range of 0 to 255 and scaled to a size of 1024×1024 pixels.[14] The authors show internal and external validation results which outperform plain SAM and a promptable version of nnU-Net (see Section 5.3). However, they claim that for small segmentation targets the main challenge lies in detection and not in segmentation, and thus remove disconnected structures in 3D with less than 1000 voxels and disconnected structures in 2D with less than 100 pixels from both the training and testing data. We found that this simplification increases performance significantly (see Section 5.3).

Cheng et al. [Che+23a] introduce SAM-Med2D, which has adapter layers in the image encoder while both the prompt encoder and mask decoder are fully fine-tuned. The model is fine-tuned on a large medical image dataset encompassing 4.6 million 2D medical images and 19.7 million corresponding segmentation masks [Ye+23]. During the fine-tuning process, point prompts and bounding box prompts are used. SAM-Med2D is evaluated against plain SAM and a version of SAM with a fine-tuned (SAM-FT) mask decoder and shows a significant increase of segmentation quality compared to both. However, reported dice scores are comparatively lower than those reported by other fine-tuning and adaptation approaches, such as MedSAM.

A more drastic alteration to SAM is SAM-Med3D [Wan+23]. It has a model structure similar to SAM, but every 2D component is replaced with a corresponding 3D component. This means that a whole 3D medical image is encoded by the image encoder at once, one or more 3D points (and an optional 3D preexisting mask) are encoded by the prompt encoder,[15] and the mask decoder outputs a whole 3D segmentation mask. However, because of this full redesign no parameters from SAM are reused. Instead, the model is trained from scratch on 21 thousand 3D medical images with 131 thousand 3D segmentation masks. SAM-Med3D shows increased segmentation quality over SAM-Med2D and SAM. More recently, after the release of the SAM-Med3D paper [Wan+23], a further fine-tuned version of the model, called SAM-Med3D-turbo, has been published.

14 We found that even small deviations from this exact pre-processing approach result in significantly lower segmentation quality.

15 3D bounding box prompts and text prompts are not supported by SAM-Med3D.

4 Workflow and Method

In this chapter we design a new 3D workflow applying MedSAM to volumetric medical image segmentation that can be implemented in VISIAN. While we use VISIAN as an example in this thesis, the workflow is general enough to be integrated into other medical image segmentation applications too. After presenting the workflow, we propose a novel prompt engineering method which allows the 3D workflow to reduce manual prompting effort. Finally, we explore the details of how to apply the prompt engineering methods to the workflow.

4.1 Workflow in VISIAN

The traditional manual workflow for segmenting volumetric medical images has already improved significantly through semi-automatic tools. VISIAN, in particular, has had smart region growing tools which reduce the segmentation effort from the start. More recently, this semi-automation has seen a leap forward through the integration of the AutoSeg tool [Kei23] based on SAM (see Section 3.2.2 and Figure 3.5). Due to this integration, VISIAN already supports the server-browser architecture split unique to SAM (see Section 3.2.1). Thus, the infrastructure for a more advanced workflow based on SAM or MedSAM is conveniently already in place. Such a more advanced workflow is necessary as the AutoSeg tool still requires a slice-by-slice workflow, which ends up being cumbersome, especially for volumetric images with many slices. We already identified the need for a 3D workflow in [Kor+21], however, the proposed 3D region-growing tool struggles with more complicated segmentation tasks. Thus, we now want to bring the generalisation abilities of MedSAM to a 3D workflow, in order to reduce the manual segmentation and prompting effort compared to the simple slice-by-slice workflow.

To make a meaningful improvement compared to the simple slice-by-slice workflow using the AutoSeg tool in VISIAN, it is important to design an intuitive and user-centered workflow that is easy to learn. Otherwise, the effort reduction of our workflow could be cancelled out by the additional hassle of dealing with the confusion. To avoid this, we propose a combination of an easy-to-learn, system-guided workflow and a user-guided workflow, which allows for more control.

Both workflow approaches build upon VISIAN's AutoSeg tool (see Section 3.2.2 and Figure 3.5 which should still be available. However, it should be updated to use MedSAM and the corresponding image normalization method. Additionally, to improve the workflow of segmenting volumetric images in VISIAN in general, we propose to combine the 2D and 3D views of VISIAN. Currently, it is possible to either view the image in a 2D view (optionally with two additional 2D views on the side) or in a 3D view. However, combining the two views by allowing the 3D view to be rendered in one of the side views of the 2D view (see Figure 4.1 (a)), would allow the user to see a volumetric representation of the segmentation while editing in the 2D view, giving them access to more spatial context information.

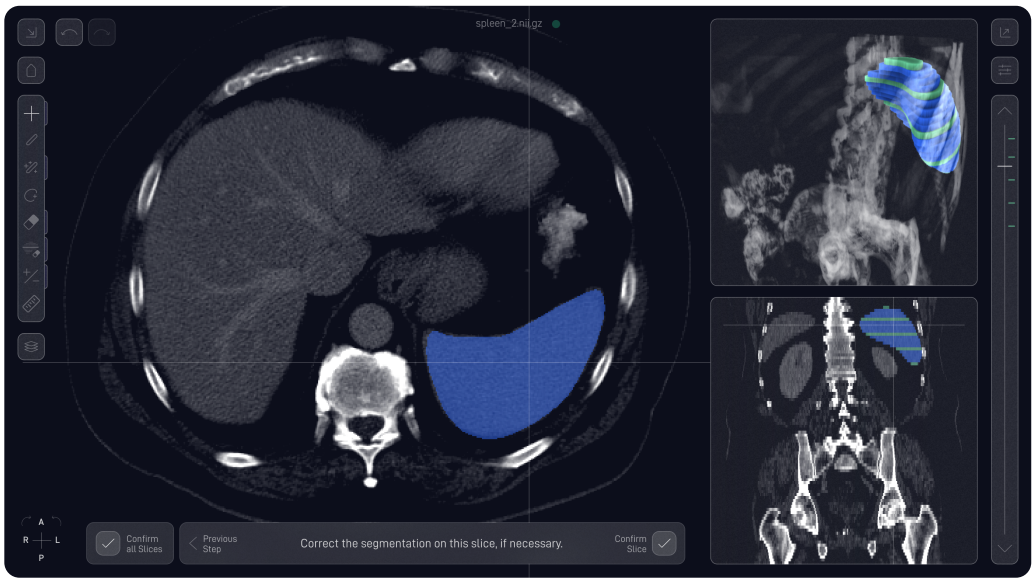
The system-guided workflow consists of a series of clearly defined steps, each guiding the user towards the desired outcome without having to make decisions outside the realm of segmentation. The steps are communicated to the user using a task bar which allows the user to confirm the current task (which will move them on to the next task) or to go back to previous tasks (see Figure 4.1 (b)).

1. **Select Top and Bottom Slices:** The user begins by selecting the top and bottom slices of the volume of interest. This initial selection sets the range for the subsequent segmentation process.

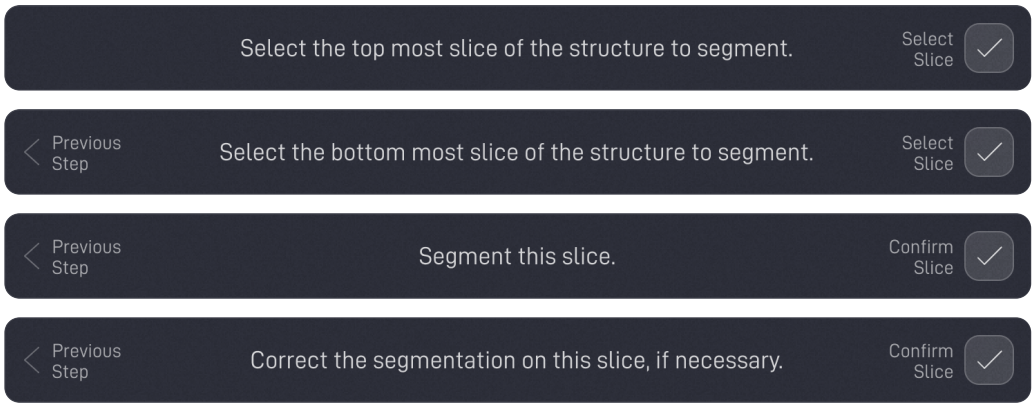
2. **Segmentation on Some of the Selected Slices:** Once the top and bottom slices are selected, VISIAN asks the user to segment specific slices. To do this, the user can use the updated AutoSeg tool using MedSAM or any other tools available in VISIAN (see Section 3.1). Initially, VISIAN asks the user to segment the top, middle, and bottom slices. Once the user has confirmed these segmentations, VISIAN shows segmentation suggestions for the other slices in between.

3. **Iterative Refinement:** The process continues by VISIAN asking the user to segment more slices. Instead of having to start the segmentation from scratch, the segmentation suggestions are given to the user as a starting point. They can choose to manually correct it or choose to again use the AutoSeg tool. Once the user has confirmed the slice, VISIAN uses the new information to improve surrounding segmentation suggestions and redirects the user to the next slice. Users also have the option to manually navigate to another slice in which case they can also segment and confirm it (see Figure 4.1 (c)).

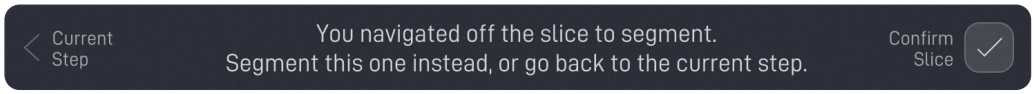
Once the user is satisfied with the segmentation quality on the interpolated slices, they can confirm the remaining suggestions all at once. To assist with judging



(a) User interface mock-up of VISIAN with the task bar for the system-guided workflow and the 3D view in the top side-view. Blue is used to show segmentation suggestions and green (see the side-views) shows segmentations confirmed by the user. Shown is an abdomen CT scan with a spleen segmentation.



(b) Task bars for the different steps of the system-guided workflow.



(c) Task bar for when the user navigates away from their current task-slice.

Figure 4.1: User interface mock-ups for the system-guided workflow in VISIAN.

the segmentation quality, a 3D view in one of the side view panels can be used (see Figure 4.1 (a)). Alternatively, users can also navigate to different slices in the 2D view to judge segmentation quality in more detail and can easily return to their next task slice, using a dedicated button in the task bar which appears if they navigate away from their current task (see Figure 4.1 (c)). This user control within the system-guided workflow allows the user to make case-by-case decisions while still using the more general framework provided by the system-guided workflow. Because of the high variability in medical images this is necessary to make a general workflow work for special cases as well.

The user-guided workflow, in contrast, is less structured and requires more decisions from the user. However, it is also less confining as it does not include specific tasks provided by the system. Instead, the user simply turns on segmentation suggestions and can then freely choose which slices to segment. Similarly to the system-guided workflow, the user can freely choose how to segment these slices. For this, all available manual as well as semi-automatic tools, including the MedSAM-based AutoSeg tool for a single slice, can be used. Once they have segmented one or more slices, VISIAN should provide segmentation suggestions for the surrounding slices.

The suggestions can then be corrected or confirmed by the user slice-by-slice or all at once. Just like in the system-guided workflow, if the user corrects or confirms a segmentation the new information is used to improve surrounding segmentation suggestions. The main difference to the system-guided workflow is that the user is responsible for choosing which slices to segment. This requires more thought from the user, but also removes the forced navigation from slice to slice, which happens in the system-guided workflow when a task is completed. Instead, the user navigates from slice to slice manually, which avoids spatial confusion.

4.2 Prompt Engineering Methods

The typical segmentation workflow which applies a 2D promptable model, such as SAM or MedSAM, to volumetric medical images is a slice-by-slice procedure. In the 3D workflow that we envision, the user should no longer have to segment the volumetric image on every single slice. Instead, from sparse user-provided segmentation across the slices, we have to estimate prompts for MedSAM on the other slices. Since bounding box prompts achieve generally better results than point prompts [Ma+24; Maz+23; Roy+23], we focus on bounding box prompts and leave experimentation with point prompts for future work (see Chapter 7).

Thus, the goal of our prompt engineering methods is to estimate bounding boxes for prompting MedSAM on the slices which were not yet segmented by the user in order to generate segmentation suggestions.

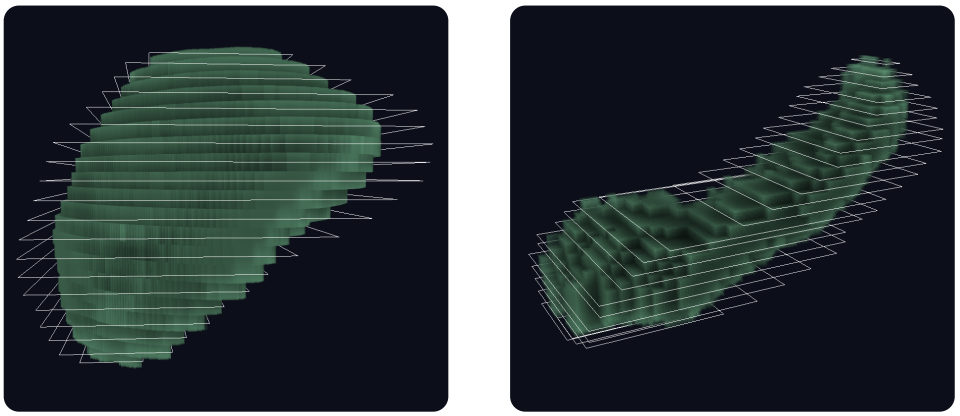
4.2.1 Nearest Neighbor Bounding Box Interpolation

The most naive way to estimate additional bounding box prompts from user-prompts is to propagate the same exact user-provided bounding box from one slice to the adjacent slices of the volume. By propagating the prompt to one adjacent slice on both sides of the original slice, we already reduce the prompting-effort by 66% because one prompt is now used for 3 slices instead of just 1 slice. However, we can extrapolate this even further by propagating the prompt to n slices on both sides of the original slice instead of just 1 slice on each side (i.e. nearest neighbor interpolation). For $n = 5$, for example, a single prompt is then used for 11 slices, reducing the prompting-effort by more than 90%. However, this of course comes with a trade off. The further we propagate a prompt from a slice, the greater becomes the inaccuracy of the propagated prompts. This, however, depends on the structure that is being segmented and the physical distance between slices in the image. If the structure which is being segmented is regularly shaped and aligned with the slicing direction of the image, such as the spleen, far propagation has less of a negative effect on bounding box quality than for structures with an irregular shape or a diagonal alignment with the slicing direction, such as the hippocampus (see Figure 4.2 (c) and 4.2 (d)).

4.2.2 Bilinear Bounding Box Interpolation

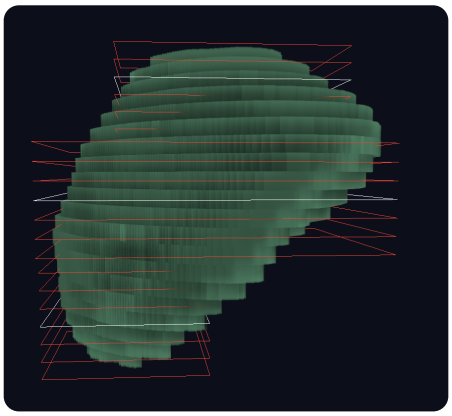
Especially misalignment of a structure with the slicing direction of the image can be mitigated by bilinear interpolating bounding boxes instead of simply propagating them (nearest neighbor interpolation). Here, the two bounding boxes at the edges of an interval of slices are provided by the user. The bounding boxes between the two inputs are then bilinearly interpolated in the following manner. A bounding box B_s on a slice with index $s \in \mathbb{N}$ is defined by two points (x_s^{min}, y_s^{min}) and (x_s^{max}, y_s^{max}). Let the interpolation interval go from slice $a \in \mathbb{N}$ to slice $b \in \mathbb{N}$. The bounding boxes B_a and B_b on slices a and b are provided by the user. For a slice with index $i \in \mathbb{N}$ with $a < i < b$, we calculate x_i^{max} like this:

$$x_i^{max} = \left(\frac{i-a}{b-a}\right) \cdot x_b^{max} + \left(1 - \frac{i-a}{b-a}\right) \cdot x_a^{max} \tag{4.1}$$

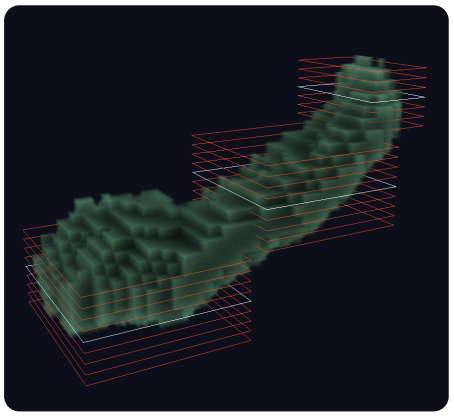


(a) Spleen, perfect boxes.

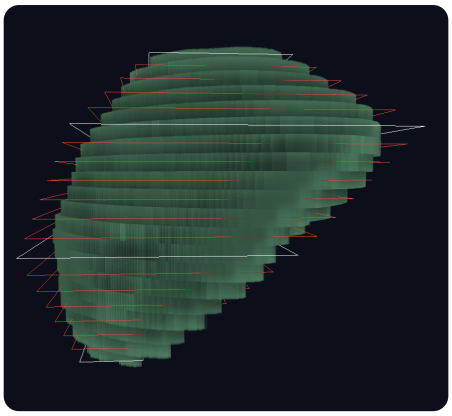
(b) Hippocampus, perfect boxes.



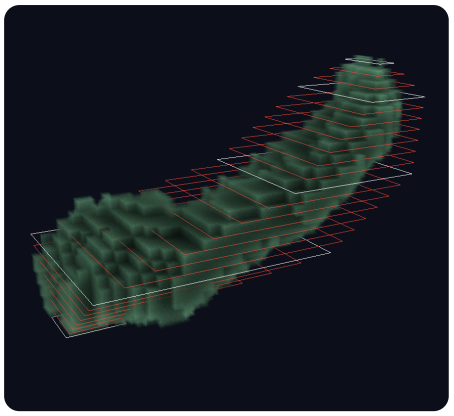
(c) Spleen, nearest neighbor.



(d) Hippocampus, nearest neighbor.



(e) Spleen, bilinear interpolation.



(f) Hippocampus, bilinear interpolation.

Figure 4.2: Spleen and Hippocampus volumes rendered in 3D with slice-wise bounding boxes. White bounding boxes are user-provided (simulated from ground truth) and red bounding boxes are estimated by nearest neighbor or bilinear interpolation. Note the better bounding box quality resulting from bilinear interpolation compared to nearest neighbor interpolation.

y_i^{max}, x_i^{min}, and y_i^{min} are calculated analogously. Mathematically, this is the same as interpolating the center, height, and width of the bounding boxes.

Adjacent interpolation intervals can share a common slice on which the user inputs the slice. For example, if the first interval is $[a, b]$, then the next interval can be $[b, c]$. This way, with many adjacent intervals, we get a prompting-effort reduction of 66% if the user prompts on every third slice and the bounding boxes are bilinearly interpolated for the slices in between. Similarly to the bounding box propagation distance, i.e. the nearest neighbor interpolation interval size, the interpolation interval size for bilinear interpolation can also be increased to further reduce the manual effort.

While the bilinear interpolation can better represent structures which are aligned diagonally to the slicing direction of the image than the nearest neighbor interpolation (see Figure 4.2 (e) and 4.2 (f)), highly irregular structures are still hard to capture. Thus, the trade off between reducing effort and reducing bounding box quality still exists. However, as we will see in Section 5.4 the bilinear interpolation strategy results in better segmentation results than the nearest neighbor bounding box interpolation, especially for greater interpolation distances.

In an effort to further improve upon nearest neighbor and bilinear bounding box interpolation, we tried adding bounding box search based on various image similarity metrics. However, instead of an improvement, we actually saw a reduction in the resulting segmentation quality. Nevertheless, we document our approach and the corresponding results in Appendix B.

4.3 Fusion of Workflow and Prompt Engineering

After discussing the envisioned 3D workflow for volumetric medical image segmentation and the corresponding prompt engineering methods, we will now explore the specifics of applying prompt engineering within this workflow to generate segmentation suggestions. The whole workflow with all its components and its split between GPU server and consumer machine is shown in Figure 4.3.

One particular observation is that it is difficult for users to draw perfect bounding boxes on a 2D slice. Other authors use random bounding box perturbation during their evaluation in order to simulate this human error [Ma+24; Roy+23]. Usually this is not a big problem because small inaccuracies in the bounding box prompts only result in a minor reduction in segmentation quality [Roy+23]. However, both the nearest neighbor bounding box interpolation and the bilinear bounding

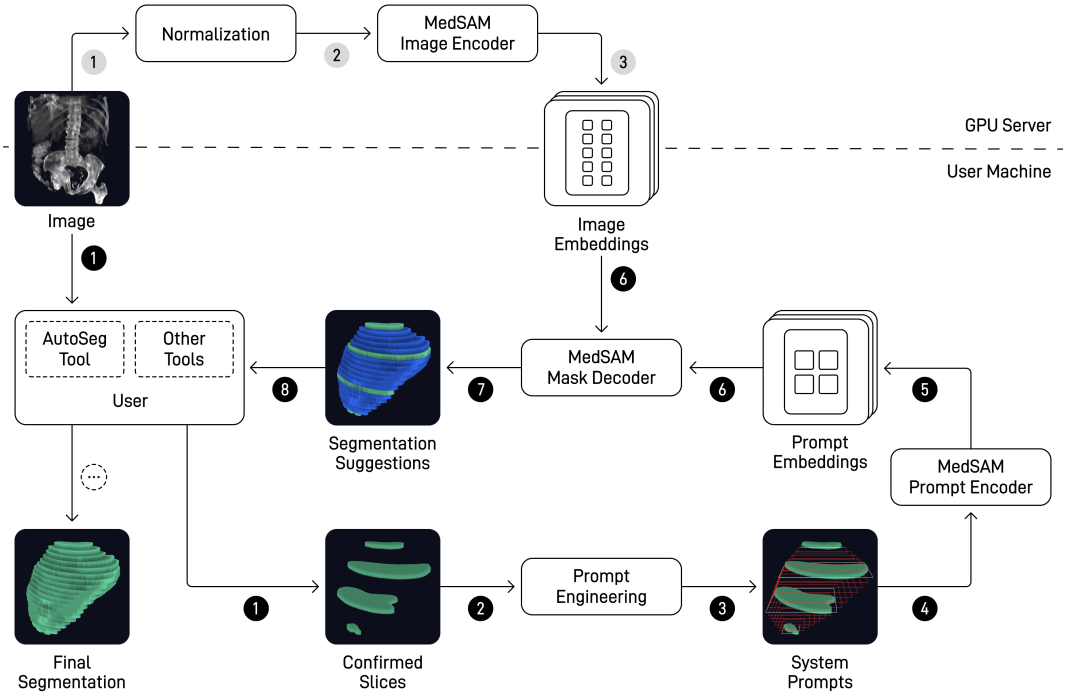


Figure 4.3: Flow of the presented workflow between high performance GPU server and user machine. The image is accessible to both the user machine and the GPU server. The GPU server handles image normalization and generates image embeddings for user machine to download and cache (1 to 3). The segmentation suggestion iteration (❶ to ❽) starts when the user confirms segmentation slices (❶) on their machine, using the AutoSeg tool or other tools. Confirmed slices are input for prompt engineering (❷) which creates system prompts (❸). These prompts are encoded and combined with corresponding image embeddings provided by the GPU server to generate new segmentation suggestions (❹ to ❼). The user checks and corrects these suggestions (❽), creating more confirmed slices (❶) for further iterations resulting in improved segmentation suggestions. Once satisfied with the suggestion quality, the user stops the iteration and confirms all suggestions (⋯) for the final segmentation.

box interpolation already naturally introduce inaccuracy and would additionally propagate the human-error. To circumvent this problem, VISIAN can extract the bounding box from the corrected segmentation after the user confirms a slice, effectively eliminating the human error from the bounding box. This applies for both the user-guided and system-guided workflows and regardless of whether nearest neighbor or bilinear interpolation is used.

Moreover, an important aspect of simplifying the user experience involves concealing the generation of image embeddings. This is relevant to both the system-guided workflow and the user-guided workflow, as well as the updated AutoSeg tool. Generating the image embeddings should happen in the background instead. That way, the user has one less thing to worry about and can fully focus on the segmentation and the bounding boxes. To make this work smoothly, the image embeddings could be pre-computed for every single slice when the image is loaded. Alternatively, the required embeddings could be computed once the user has selected the top and bottom slices in the system-guided workflow. In order to have the embeddings available to instantly generate a segmentation when the user provides a bounding box prompt on one of the slices during the system-guided workflow, the image embedding computation should be done in the order in which the system asks the user to provide prompts. As this is not possible in the user-guided workflow or when the AutoSeg tool is used by itself, the image embedding for the slice which the user is currently viewing should be prioritized to achieve interactive feedback for user-provided prompts.

The system-guided workflow can always use the bilinear bounding box interpolation and does not need the nearest neighbor bounding box interpolation. After the first three slices are segmented by the user, two interpolation intervals can be used to calculate bounding box estimations for the segmentation suggestions. During the iterative refinement phase, VISIAN should always ask the user to segment a new slice which halves an existing interpolation interval. Once the user has confirmed the slice, VISIAN can then update the other interpolated bounding boxes by computing the bounding box of the confirmed segmentation and splitting the interpolation interval into two.

During the user-guided workflow, once two non-neighboring slices are segmented, VISIAN should automatically compute the bounding boxes of the segmentation on each of these slices and bilinearly interpolate them resulting in bounding box prompts which are used to generate segmentation suggestions for slices in between using MedSAM. Additionally, if only a single slice was segmented, the user should be able to manually request segmentation suggestions for the surrounding slices, in which case the nearest neighbor bounding box interpolation can be used,

essentially copying the user-provided bounding box. Just like in the system-guided workflow, if the user corrects or confirms a segmentation in the middle of an interpolation interval, the resulting bounding box should be used to update the other interpolated bounding box prompts and the resulting segmentation suggestions.

Overall, the proposed prompt engineering methods are able to make the 3D workflow for volumetric medical image segmentation work, even using a 2D model like MedSAM. In the next chapter, we evaluate the segmentation quality resulting from the proposed prompt engineering methods in detail. We compare our results with using SAM and MedSAM slice-by-slice. Additionally, we compare against SAM-Med3D, a 3D model focused on medical images which was inspired by SAM's architecture. Especially the comparison with the 3D model will show, why a 3D workflow based on a 2D model is advantageous.

5 Evaluation

We now present experimental results of our prompt engineering method applied to the Medical Segmentation Decathlon [Ant+22] datasets. Before we dive into quantitative and qualitative results, we give an overview of the Medical Segmentation Decathlon datasets and review some details of the models we compare our method to. We finish up with an ablation study analyzing the difference between SAM and MedSAM when applying the presented prompt engineering methods.

5.1 Dataset

The Medical Segmentation Decathlon [Ant+22] is a medical image segmentation challenge aimed at generalization. Contenders have to design one segmentation algorithm which can handle a multitude of segmentation tasks and image modalities. No manual configuration for the specific tasks is allowed. Additionally, during the algorithm development phase of the original, time-constraint challenge, only 7 of 10 datasets were available. The other 3 *mystery tasks* were only available for final testing of the algorithms, thus making sure that the algorithms had to generalize to unseen tasks. Now, the challenge continues on with all tasks publicly available.

The 10 volumetric medical image datasets include various structures such as organs and cancers captured in different image modalities. In particular, they are comprised of 4 MRI datasets and 6 CT datasets. All 10 datasets are split into a training set and a testing set. As the Medical Segmentation Decathlon challenge is still ongoing, the ground truth segmentations of the testing sets are not publicly available. However, the training sets include ground truth segmentations and, with differing sizes, include 1741 volumetric images with segmentations in total (see Table 5.1). Of the 4 MRI datasets 2 include multiple modalities. 4 of the 10 datasets include a single foreground class, 5 datasets include 2 separate foreground classes, and one dataset, the brain tumor dataset, includes 3 foreground classes.

In order to better match the common real world application scenario of semi-automatically segmenting one structure at a time while viewing one image modality at a time, we converted the Medical Segmentation Decathlon datasets to this scenario as follows. As the brain tumor segmentations include different sub-classes of the tumor, we combine them all to a single class. We use the FLAIR-MRI channel,

Table 5.1: Overview of the 10 Medical Segmentation Decathlon datasets. The number of images only includes the training dataset which includes segmentations. During our evaluation we simplify each dataset to a single image channel and a single segmentation class in order to better match the common real world application scenario of semi-automatically segmenting one structure at a time while viewing one image modality at a time.

Dataset	Modality	Channels	Seg. Classes	Images
Brain Tumor	MRI	4	3	484
Heart	MRI	1	1	20
Liver	CT	1	2	131
Hippocampus	MRI	1	2	260
Prostate	MRI	2	2	32
Lung Tumor	CT	1	1	63
Pancreas	CT	1	2	281
Hepatic Tumor	CT	1	2	303
Spleen	CT	1	1	41
Colon Tumor	CT	1	1	126
Total				**1741**

as FLAIR is commonly used for tumor segmentation and produces good contrast [Hav+17; SJ15], and disregard the other channels. The liver segmentation include the healthy liver and small tumors within the liver. We combine the classes to result in one class for the whole liver. The hippocampus segmentations are split into anterior and posterior hippocampus. We combine both classes to result in one segmentation for the whole hippocampus. Similarly, prostate segmentation are split into the peripheral zone and the transition zone. We again combine them to one class of the whole prostate. Additionally, we choose the T2-MRI channel and disregard the ADC-MRI channel. The pancreas segmentations include one class for the healthy pancreas and one class for small tumors. We combine the classes into one class of the whole pancreas. The hepatic tumor dataset includes one class for hepatic vessels and one class for tumors. As the hepatic vessels are many small disconnected structures which are entirely distinct from the tumor, we choose the tumor class and disregard the vessel class.[16] All other datasets already have a single channel and single foreground segmentation class and remain unchanged.

16 The vessel dataset was also found to be *non-optimal* [Ant+22] by the Medical Segmentation Decathlon authors after releasing it to the challenge participants. However, in order to not disrupt the ongoing challenge, it remained part of the challenge.

Overall, even with our simplification to a single image channel and a single foreground segmentation class, the Medical Segmentation Decathlon datasets remain a diverse set of volumetric medical image segmentation tasks.

Next we review the the metric we use for our evaluation and present some details of the models we compare our method to, before we move on to our experiments and the results.

5.2 Dice Similarity Coefficient

For quantitative evaluation we use the Dice Similarity Coefficient (DSC) [Dic45], which is commonly used in medical image segmentation [TH15]. The DSC is defined as

$$DSC = \frac{2 \cdot |G \cap P|}{|G| + |P|} \tag{5.1}$$

where G is the set of ground truth segmentation voxels and P is the set of predicted segmentation voxels. The result is a value between 0 and 1, with 0 indicating no overlap between ground truth and prediction and 1 indicating exact overlap. In all cases, as we are interested in the volumetric segmentation, we report the full 3D DSC, instead of averaging the 2D DSC across all slices of a volume.

Note that the DSC is a sensible metric for medical image segmentation due to the fact that it does not consider true negatives and instead only focuses on overlap of prediction and ground truth [TH15]. This is important due to the common imbalance between background and foreground in medical image segmentation tasks.

With this metric in mind, we now move on the benchmark models we compare our method to.

5.3 Benchmark Models

In order to meaningfully contextualize the results of our method we compare it to the following models: SAM [Kir+23], MedSAM [Ma+24], and SAM-Med3D (and SAM-Med3D-turbo) [Wan+23]. Using SAM and MedSAM we simulate the normal slice-by-slice workflow, where the model is prompted with a bounding box on every single slice. SAM-Med3D and SAM-Med3D-turbo, on the other hand, constitute an alternate 3D workflow based on point prompts. Here, we simulate a 3D workflow where 10 3D point prompts are given to the models.

We already presented SAM, MedSAM, and SAM-Med3D in Section 3.2. However,

Table 5.2: Average DSC (and standard deviation) of MedSAM [Ma+24] with and without simplifying datasets by removing small structures (up to 1000 voxels in 3D an 100 pixels in 2D) from the ground truth. In both cases, MedSAM is applied slice-by-slice with a perfect bounding box prompt, computed from the ground truth. Note the generally higher DSC for the simplified ground truth.

Ground Truth	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.	**Average**
Simplified	0.865 (0.07)	0.890 (0.02)	0.956 (0.01)	0.849 (0.02)	0.903 (0.03)	0.799 (0.08)	0.827 (0.05)	0.856 (0.11)	0.950 (0.01)	0.794 (0.08)	**0.869 (0.05)**
Original	0.830 (0.11)	0.846 (0.03)	0.956 (0.01)	0.822 (0.02)	0.903 (0.03)	0.791 (0.10)	0.822 (0.06)	0.782 (0.18)	0.950 (0.01)	0.782 (0.08)	**0.848 (0.06)**

to bring MedSAM into perspective, we first discuss a simplification used by Ma et al. [Ma+24] when evaluating MedSAM. Additionally, we showcase how the promptable version of nnU-Net presented by Ma et al. [Ma+24] works and why we do not adopt it as a meaningful benchmark.

Ma et al. [Ma+24] present an outstandingly high DSC for MedSAM applied to many different medical image segmentation datasets. When applying MedSAM slice-by-slice to the Medical Image Segmentation datasets, we were not able to reproduce quite the same results, even using the same normalization techniques during preprocessing. We find that the reason for this is additional simplification during data preprocessing which is not mentioned in the paper. Specifically, 3D objects with less than 1000 voxels and 2D objects with less than 100 voxels are removed from the ground truth segmentations before the slice-by-slice prompting is simulated.[17] We present the difference between applying this simplification and using the original ground truth in Table 5.2. As the simplification is significant and, in our opinion, not realistic in a real world scenario where usually the whole structure needs to be segmented, we stick with the original ground truth segmentations and do not use the same simplification in the rest of our evaluation.

Ma et al. [Ma+24] also compare MedSAM to the state of the art deep learning model nnU-Net [Ise+20]. However, since MedSAM is a single model for many different datasets at once they do not train a separate nnU-Net for every single dataset either. Instead, they train one model per image kind, such as MRI and CT. This comes with a new problem though: it is no longer obvious from the image alone, which structure the nnU-Net should segment. For example, if the model is presented with an abdomen CT scan it is not obvious whether it should segment the

17 In the published preprocessing code the assumption that for such small structures the main challenge would be detection and not segmentation is given as a reason.

Table 5.3: Average DSC of promptable 2D nnU-Net [Ise+20] trained on all 10 Medical Segmentation Decathlon datasets and trained on only 7 datasets, excluding Brain Tumor, Liver, and Hepatic Tumor. Note that the hepatic tumors are relatively simple and similar in structure to the lung tumors which are included in the training dataset for both models.

	Brain T.	Heart	**Liver**	Hippoca.	Prostate	Lung T.	Pancreas	**Hepatic T.**	Spleen	Colon T.
nnU-Net (all)	**0.921**	0.948	**0.974**	0.918	0.962	0.919	0.924	**0.939**	0.981	0.908
nnU-Net (reduced)	**0.635**	0.949	**0.690**	0.921	0.963	0.914	0.927	**0.930**	0.982	0.914
MedSAM	**0.830**	0.846	**0.955**	0.822	0.903	0.792	0.822	**0.782**	0.949	0.783

spleen or the liver. To solve this, Ma et al. [Ma+24] make the nnU-Net promptable by adding an additional prompt channel to the image. In this prompt channel, the same bounding box which is used to prompt MedSAM is encoded as a binary mask.

The result is a model which can learn to segment multiple different structures very well (see Table 5.3 (all)). However, the generalization abilities of the promptable nnU-Net are subpar. For instance, we trained one promptable nnU-Net on all 10 Medical Segmentation Decathlon datasets and one nnU-Net on a reduced set of 7 datasets, excluding Brain Tumor, Liver, and Hepatic Tumor.[18] The model trained on all 10 datasets showed a very good DSC during testing across all datasets (see Table 5.3 (all)). In contrast, the model which was trained on the reduced dataset, did not generalize well to the Brain Tumor and Liver datasets, but was able to generalize well to the Hepatic Tumor dataset (see Table 5.3 (reduced)) which is is comparatively simple and similar in structure to the Lung Tumor dataset which was part of the training data for both models. Notable is that the purpose trained nnU-Nets achieve a higher segmentation quality as MedSAM, but MedSAM is significantly superior when the nnU-Nets have to segment an unseen task (see Table 5.3).

Because of this lack of generalization, which is a strong contrast to MedSAM's generalization abilities [Ma+24], a promptable nnU-Net would have to be purpose trained for a specific set of use-cases before it could be used in a semi-automatic segmentation tool. Thus, we disregard this approach as the model for the proposed 3D workflow and do not adopt it as a benchmark in the upcoming experiments.

18 Both models were trained for 1000 epochs using the available training scripts with a random split into 80% training data and 20% testing data. Additionally, the same normalization approach detailed in Section 5.4 was used.

5.4 Experiments

For our experiments we adopt the data preprocessing and normalization method which MedSAM was trained with [Ma+24] (see Section 3.2.3). In particular, for MRI we clamp the top and bottom 0.5 percentiles. For lung CT images, we apply the standard lung windowing with window width 1500 and window level −600 [BM17]. For all other CT datasets we use the standard abdomen windowing with window width 400 and window level 50 [BM17]. For the 2D models, we additionally normalize each slice to a range of 0 to 255 and resample to a size of 1024×1024 pixels.

To evaluate SAM, we use the most powerful ViT-H image encoder version, while MedSAM is based on the smaller ViT-B image encoder [Ma+24].

As we have discussed in Section 4.3, in the workflow we propose, the exact bounding boxes from the segmentation slices which the user has confirmed can be computed before using them for interpolation. Thus, in our experiments we also use the exact bounding boxes for interpolation. We use the ground truth segmentations to simulate a user confirming a segmentation slice. To keep the comparison fair, we also use the exact bounding boxes when simulating the slice-by-slice workflow with both SAM and MedSAM and disregard random perturbation.

For both the nearest neighbor and bilinear bounding box interpolation we simulate various amounts of user input. Specifically, we use approximately 33%, 20%, 14%, 11%, and 9% user prompts for both the bilinear and nearest neighbor interpolation. Additionally, to avoid bilinearly interpolating between the usually small edge bounding boxes of a structure, we ensure that at least two interpolation intervals are used. This also reflects the system-guided workflow, where the user is first asked to segment and confirm the top, middle, and bottom slices of the structure, resulting in at least two interpolation intervals for bilinear interpolation (see Section 4.1). For nearest neighbor interpolation we sample the user-provided bounding box in the center of each interpolation interval instead of at the edges (see Figure 4.2), which ensures that the small edge bounding boxes of a structure are not copied further towards the center.

For the SAM-Med3D and SAM-Med3D-turbo benchmarks, we use the available evaluation script, with 10 point prompts. The first point prompt is a random point from the foreground region. The following points are iteratively placed in a random position within the error region [Wan+23].

5.4.1 Quantitative Results

The quantitative results of our experiments are presented in Table 5.4. First of all, it becomes apparent that, at least on the Medical Segmentation Decathlon datasets, the difference between SAM and MedSAM, while significant, is smaller than claimed by Ma et al. [Ma+24]. Secondly, the alternate 3D workflow using point prompts and the 3D models SAM-Med3D and SAM-Med3D-turbo, while arguably requiring less user effort, is not competitive in regard to segmentation quality. This also remains true, when comparing it to our prompt engineering methods instead of the slice-by-slice workflow using SAM or MedSAM. Even when using only 9% user prompts, both the nearest neighbor and bilinear bounding box interpolation by far outperform SAM-Med3D and SAM-Med3D-turbo. The only exceptions are brain tumor and heart segmentation.

Table 5.4: Average DSC (and standard deviation) of bilinear (BL) and nearest neighbor (NN) bounding box interpolation used with MedSAM compared to applying SAM [Kir+23] and MedSAM [Ma+24] slice-by-slice, and SAM-Med3D and SAM-Med3D-turbo [Wan+23] prompted with 10 3D point prompts, all applied to the Medical Segmentation Decathlon datasets. For both the bilinear and nearest neighbor bounding box interpolation, user input was simulated on a varying percentage of slices. Note that the DSC for bilinear interpolation with 33% user prompts comes rather close to the DSC for using MedSAM with 100% user prompts (~ 0.017 difference).

	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.	**Average**
MedSAM	0.830 (0.11)	0.846 (0.03)	0.956 (0.01)	0.822 (0.02)	0.903 (0.03)	0.791 (0.10)	0.822 (0.06)	0.782 (0.18)	0.950 (0.01)	0.782 (0.08)	**0.848 (0.06)**
SAM	0.756 (0.13)	0.839 (0.03)	0.908 (0.05)	0.778 (0.03)	0.930 (0.02)	0.805 (0.09)	0.750 (0.09)	0.764 (0.18)	0.936 (0.02)	0.736 (0.12)	0.820 (0.08)
SAM-Med3D	0.753 (0.18)	0.695 (0.04)	0.876 (0.15)	0.355 (0.06)	0.426 (0.10)	0.486 (0.31)	0.345 (0.15)	0.336 (0.28)	0.707 (0.19)	0.389 (0.20)	0.537 (0.17)
SAM-Med3D-turbo	0.862 (0.07)	0.887 (0.02)	0.904 (0.13)	0.116 (0.11)	0.644 (0.13)	0.741 (0.11)	0.623 (0.16)	0.339 (0.28)	0.824 (0.11)	0.512 (0.18)	0.645 (0.13)
NN Interpolation											
33% user prompts	0.828 (0.11)	0.843 (0.03)	0.953 (0.01)	0.783 (0.02)	0.874 (0.04)	0.769 (0.11)	0.800 (0.06)	0.746 (0.18)	0.929 (0.02)	0.724 (0.09)	**0.825 (0.07)**
20% user prompts	0.824 (0.10)	0.839 (0.03)	0.950 (0.02)	0.728 (0.03)	0.840 (0.05)	0.759 (0.11)	0.771 (0.06)	0.727 (0.17)	0.901 (0.04)	0.701 (0.09)	0.804 (0.07)
14% user prompts	0.817 (0.11)	0.833 (0.03)	0.945 (0.02)	0.676 (0.04)	0.814 (0.05)	0.750 (0.11)	0.745 (0.07)	0.717 (0.17)	0.871 (0.06)	0.691 (0.09)	0.786 (0.07)
11% user prompts	0.812 (0.11)	0.826 (0.03)	0.941 (0.02)	0.618 (0.05)	0.793 (0.06)	0.743 (0.11)	0.718 (0.07)	0.711 (0.17)	0.854 (0.06)	0.688 (0.09)	0.770 (0.08)
9% user prompts	0.804 (0.11)	0.822 (0.03)	0.936 (0.03)	0.586 (0.04)	0.785 (0.06)	0.741 (0.10)	0.697 (0.08)	0.706 (0.17)	0.831 (0.07)	0.688 (0.09)	0.760 (0.08)
BL Interpolation											
33% user prompts	0.830 (0.11)	0.843 (0.03)	0.955 (0.01)	0.808 (0.02)	0.870 (0.04)	0.781 (0.10)	0.804 (0.06)	0.752 (0.18)	0.939 (0.02)	0.729 (0.09)	**0.831 (0.07)**
20% user prompts	0.829 (0.10)	0.842 (0.03)	0.954 (0.01)	0.790 (0.03)	0.836 (0.04)	0.777 (0.10)	0.774 (0.06)	0.733 (0.17)	0.917 (0.04)	0.706 (0.09)	0.816 (0.07)
14% user prompts	0.827 (0.10)	0.841 (0.03)	0.952 (0.01)	0.770 (0.03)	0.783 (0.07)	0.768 (0.10)	0.742 (0.08)	0.716 (0.17)	0.887 (0.06)	0.683 (0.09)	0.797 (0.08)
11% user prompts	0.823 (0.11)	0.840 (0.04)	0.949 (0.02)	0.749 (0.04)	0.747 (0.07)	0.769 (0.10)	0.708 (0.08)	0.705 (0.17)	0.861 (0.06)	0.678 (0.09)	0.783 (0.08)
9% user prompts	0.820 (0.10)	0.838 (0.03)	0.945 (0.03)	0.717 (0.07)	0.738 (0.07)	0.764 (0.10)	0.670 (0.10)	0.696 (0.17)	0.828 (0.07)	0.676 (0.09)	0.769 (0.08)

Bilinear and nearest neighbor bounding box interpolation result in lower segmentation quality than applying MedSAM slice-by-slice. This is of course to be expected, considering that the same model is used and the bounding boxes resulting from interpolation are less accurate than using a perfect bounding box on every single slice. However, the segmentation quality, especially for 33% user prompts, is only slightly worse than the slice-by-slice approach using MedSAM and still outperforms SAM.

For both bilinear and nearest neighbor interpolation we see a relatively even reduction in segmentation quality as the amount of user prompts is reduced. Between the two we see superior segmentation quality using bilinear interpolation, with an improvement in DSC of around 0.01 (see Table 5.4).

Additionally, both interpolation methods result in similar segmentation-consistency as applying MedSAM slice-by-slice, as indicated by only slightly higher standard deviations. SAM-Med3D and SAM-Med3D-turbo, on the other hand, produce much more inconsistent results (see Table 5.4).

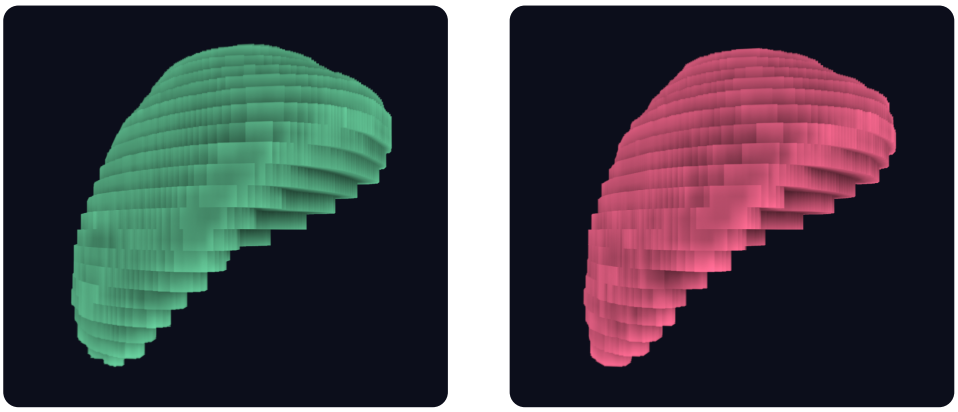
5.4.2 Qualitative Results

To qualitatively analyze the segmentation results of the bilinear and nearest neighbor bounding box interpolation, we present examples for spleen (see Figure 5.1) and hippocampus (see Figure 5.2) segmentation. The shown cases are the same used for showing slice-wise bounding boxes in Figure 4.2.

First of all, we observe that the slice-by-slice application of MedSAM (see Figure 5.1 (b) and 5.2 (b)) is quite close to the ground truth segmentations. Thus, the result only requires minor manual clean up. We note, however, that the result is not perfect either. For example, the hippocampus segmentation (see Figure 5.2 (b)) is missing a bulk of the structure in the bottom left of the image compared to the ground truth.

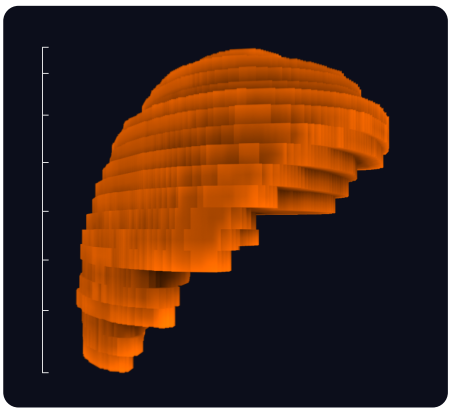
The nearest neighbor bounding box interpolation, as expected, results in edgy segmentations where the chunks of slices sharing the same bounding box prompts are clearly visible, even with small interpolation distances (see Figure 5.1 (c) and 5.2 (c)). These chunks become even more obvious when using a larger interpolation distance. For the spleen, this results in especially low quality segmentations at the top and bottom of the structure (see Figure 5.1 (d)). For the hippocampus, which is miss-aligned with the slicing direction, the result is even more extreme with very clear chunks which are even disconnected from each other (see Figure 5.2 (d)).

The bilinear bounding box interpolation, on the other hand, results in smoother segmentations. With small interpolation intervals the result for the spleen (see Figure 5.1 (e)) is almost exactly the same as the slice-by-slice method (see Fig-

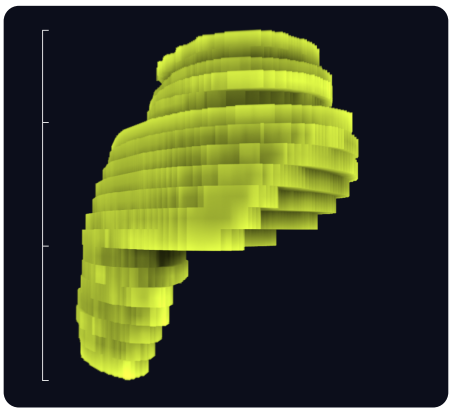


(a) Ground truth

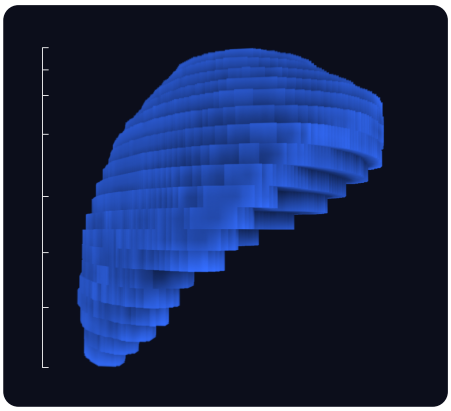
(b) Slice-by-slice



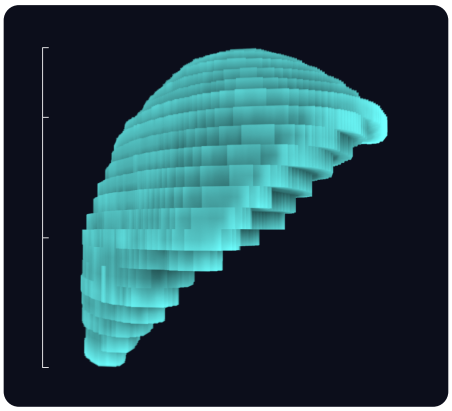
(c) Nearest neighbor (33% user-prompts)



(d) Nearest neighbor (9% user-prompts)

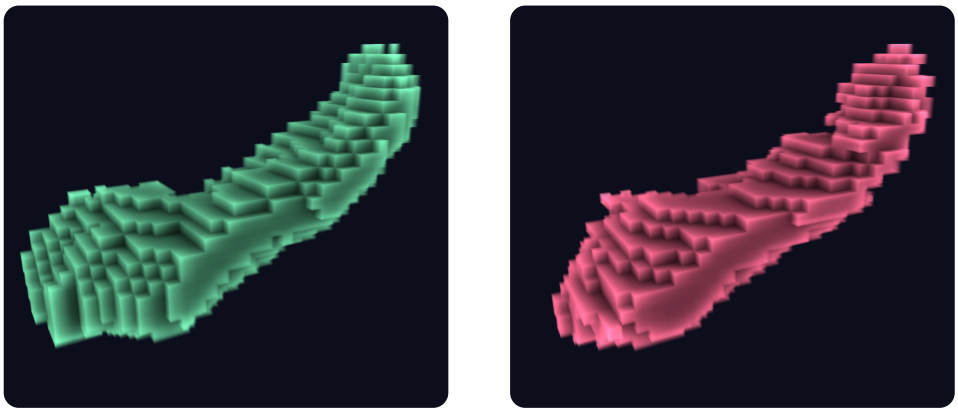


(e) Bilinear (33% user-prompts)



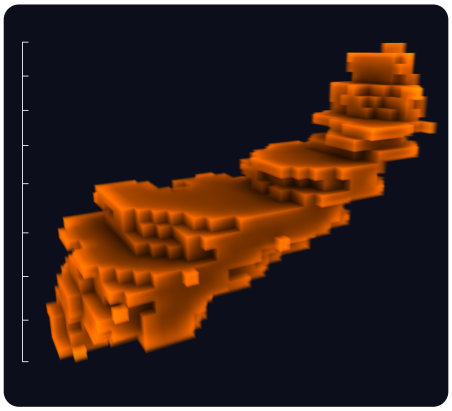
(f) Bilinear (9% user-prompts)

Figure 5.1: Spleen segmentations generated with different prompting methods, all using MedSAM, compared to ground truth. In all images the individual slices are quite visible due to the low image resolution in the slicing dimension resulting in high slice thickness (5mm). Interpolation intervals are marked in white (intervals towards the top of the scale appear smaller due to the 3D perspective). Note the chunks corresponding with the intervals used for nearest neighbor bounding box interpolation and the smoother results from bilinear bounding box interpolation.

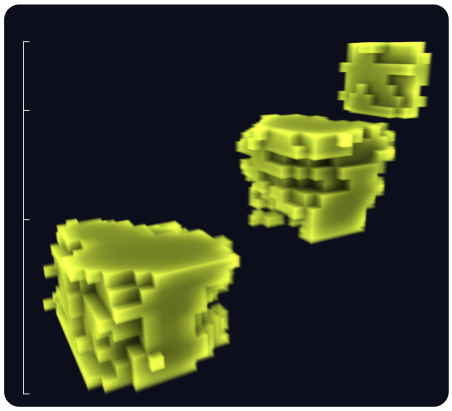


(a) Ground truth

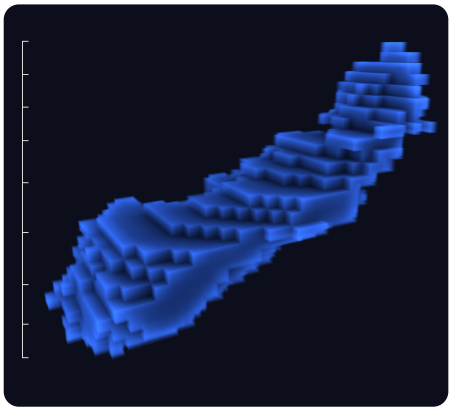
(b) Slice-by-slice



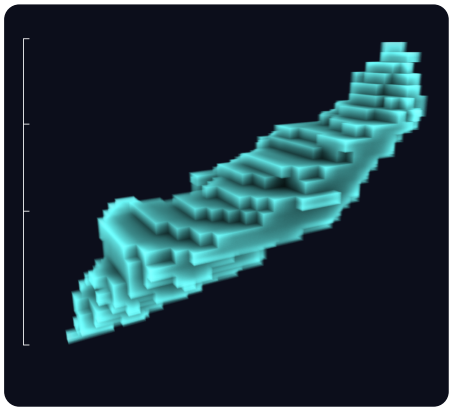
(c) Nearest neighbor (33% user-prompts)



(d) Nearest neighbor (9% user-prompts)

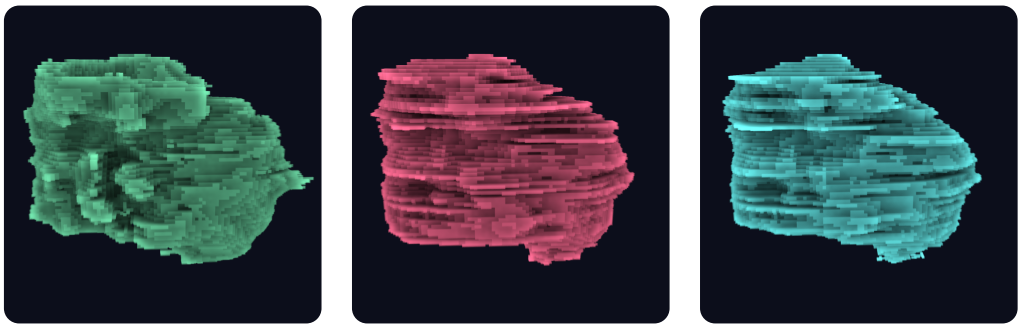


(e) Bilinear (33% user-prompts)



(f) Bilinear (9% user-prompts)

Figure 5.2: Hippocampus segmentations generated with different prompting methods, all using MedSAM, compared to ground truth. In all images the individual voxels are quite visible due to low resolution (1 voxel per *mm*) compared to the size of the hippocampus ($\sim 37mm$ in length). Interpolation intervals are marked in white (intervals towards the edges of the scale appear smaller due to the 3D perspective). Note the (disconnected) chunks resulting from nearest neighbor bounding box interpolation and the loss of detail but preservation of shape resulting from bilinear bounding box interpolation.



(a) Ground truth **(b)** Slice-by-slice **(c)** Bilinear interpolation

Figure 5.3: Brain tumor segmentations generated with slice-by-slice prompts and bilinear bounding box interpolation (9% user-prompts), all using MedSAM, compared to ground truth. Note how finer details visible at the front and the right of the ground truth get lost with both the slice-by-slice and the bilinear interpolation approaches.

ure 5.1 (b)). The hippocampus segmentation resulting from small interpolation intervals (see Figure 5.2 (e)) is not quite as good, but still only slightly deviates from the ground truth (see Figure 5.2 (b)). With large interpolation intervals the result is only slightly worse for the relatively smooth spleen (see Figure 5.1 (f)). The bilinear bounding box interpolation with large interpolation intervals is still able to capture the general shape of the more complicated hippocampus (see Figure 5.2 (f)). Compared to the disconnected chunks resulting from the nearest neighbor bounding box interpolation (see Figure 5.2 (d)), this is a large improvement. However, the difference from the slice-by-slice result and the ground truth becomes larger. The main reason for this is the comparatively small bounding box from the bottom most slice of the hippocampus. This results in sub-optimal bounding boxes (see Figure 4.2 (f)) which in turn result in similarly sub-optimal segmentations for the bottom slices of the hippocampus. The same effect can be observed for the top slices of the brain tumor segmentation in Figure 5.3 (c).

Lastly, when looking at more complex structures, such as the brain tumor in Figure 5.3, we see that both the slice-by-slice method and the bilinear bounding box interpolation method tend to loose some of the finer details visible in the ground truth while capturing the main shape of the structure very well. This shows the need for manual clean up.

5.4.3 Run-time Performance

In addition to high quality segmentations it is also important for a semi-automatic segmentation tool to achieve interactive performance. Thus, we will now evaluate the run-time performance of our method.

For performance evaluation we ran jobs on a SLURM cluster with the following job specification: 1 NVIDIA GeForce RTX 2080 Ti GPU with 11 GB of VRAM, 6 Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz, and 24 GB of RAM.

We measured the performance for the following steps in the process:

(S1) Normalization of 3D images (windowing for CT, percentile clipping for MRI)

(S2) Re-scaling and normalization of 2D slices

(S3) Image embedding generation (MedSAM image encoder, includes moving the image data to GPU memory)

(S4) Extraction of bounding boxes from user-provided segmentation

(S5) Mask prediction (MedSAM prompt encoder and mask decoder)

We exclude the execution time of our prompt engineering calculations as they are non existent for nearest neighbor interpolation (the extracted bounding box is simply reused) and negligible for bilinear bounding box interpolation.

From each Medical Segmentation Decathlon dataset we use 20 images. For each image, we use only the slices containing a part of the ground truth segmentation for the per-slice steps (S2) to (S5).

The average run-time of each step for each dataset is shown in Table 5.5. It becomes clear that the most expensive operation is the normalization of the 3D image (S1), because this is the only step that is performed once on the whole volumetric

Table 5.5: Average run-time in seconds for steps (S1) to (S5) for each Medical Segmentation Decathlon [Ant+22] dataset, running on an NVIDIA GeForce RTX 2080 Ti GPU with 11 GB of VRAM with 6 Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz, and 24 GB of RAM. We use 20 images per dataset. For each image, we use only the slices containing a part of the ground truth segmentation for the per-slice steps (S2) to (S5). Note that the average run-time of (S1) is skewed due to the large values for the liver and lung tumor datasets.

Step	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.	**Average**
(S1)	0.3186	0.6462	39.7367	0.0043	0.1160	3.2572	0.7843	0.3532	0.9820	0.6980	**4.6897**
(S2)	0.1118	0.1176	0.1272	0.1245	0.1205	0.1258	0.1283	0.1287	0.1286	0.1274	**0.1240**
(S3)	0.1387	0.1394	0.1402	0.1411	0.1419	0.1410	0.1403	0.1424	0.1401	0.1433	**0.1408**
(S4)	0.0002	0.0004	0.0011	0.0001	0.0005	0.0010	0.0009	0.0010	0.0010	0.0009	**0.0007**
(S5)	0.0237	0.0241	0.0247	0.0244	0.0248	0.0249	0.0250	0.0252	0.0250	0.0252	**0.0247**

Table 5.6: Average size of the 20 images of each Medical Segmentation Decathlon [Ant+22] dataset used for performance evaluation. X and Y make up the size of a slice, Z makes up the number of slices in an image. Liver and lung tumor images are large due to their high slice count (Z), hippocampus images are small, all other images have medium size. All MRI images, i.e. brain tumor, heart, hippocampus, prostate (compare Table 5.1), have smaller slices compared to the CT images.

Size	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.
X	240	320	512	36	320	512	512	512	512	512
Y	240	320	512	49	320	512	512	512	512	512
Z	155	114	543	36	19	334	95	46	92	91

image instead of separately for every slice. Additionally, we see that (S1) is only very expensive for large images, i.e liver and lung tumor images (see Table 5.6). For medium size images, (S1) takes up to 1 second, and for small images, such as from the hippocampus dataset (see Table 5.6), the run-time is negligible. The next observation is that the run-time of the per-slice steps (S2), (S3), and (S5) is independent of the slice size. The reason for this is that in (S2) the slice is first re-scaled to 1024×1024 pixels, eliminating any further dependence on the slice size. We find that re-scaling and normalizing as slice (S2) takes about 0.12 seconds, generating the image embedding (S3) takes about 0.14 seconds, and predicting a mask (S5) takes around 0.02 seconds. The only per-slice step which depends on the size of the slices is extracting the bounding box from a user-provided segmentation (S4). However, with run-times in the millisecond range, this is negligible.

From these results we can calculate the run-time for generating a segmentation suggestion on a single slice t_S. We exclude (S1) as it is a per-image operation and (S4) as the run-time for this is negligible. To calculate t_S we add up the average run-times of (S2), (S3), and (S5) and end up with about 0.29 seconds excluding network delays caused by the separation between server and browser. This means that given the 3D normalization, which can happen as soon as the image file is uploaded to the server, all other steps of the workflow have interactive run-time which allows for quick feedback loops for users.

5.5 Ablation Study

To finish up the evaluation of the nearest neighbor and bilinear bounding box interpolation methods, we present an ablation study comparing both methods applied with SAM and MedSAM. The results for the nearest neighbor bounding

box interpolation are shown in Table 5.7 and the result for the bilinear bounding box interpolation are shown in Table 5.8.

For nearest neighbor interpolation we see a slightly smaller reduction in DSC with smaller amounts of user prompts for SAM compared to MedSAM. This effect becomes even more noticeable with the bilinear bounding box interpolation. The DSC reduction for larger interpolation intervals is smaller for SAM than for MedSAM, to the point where SAM matches the results of MedSAM for 14% user prompts, and outperforms MedSAM for 11% and 9% user prompts. This indicates, that SAM can handle bounding boxes which are aligned correctly but might be too small or too large (a common result from bilinear bounding box interpolation) better than MedSAM. MedSAM, however, produces slightly more consistent results with slightly lower standard deviations in the DSC. Both prompt engineering methods, however, work well regardless of the used model.

Table 5.7: Average DSC (and standard deviation) of nearest neighbor (NN) bounding box interpolation with MedSAM [Ma+24] compared to SAM [Kir+23]. Note that the average DSC is always larger for MedSAM compared to SAM. However, the difference with many user prompts (33%: ~ 0.025) is larger than for few user prompts (9%: ~ 0.011).

NN Interpolation	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.	**Average**
MedSAM											
33% user prompts	0.828 (0.11)	0.843 (0.03)	0.953 (0.01)	0.783 (0.02)	0.874 (0.04)	0.769 (0.11)	0.800 (0.06)	0.746 (0.18)	0.929 (0.02)	0.724 (0.09)	**0.825 (0.07)**
20% user prompts	0.824 (0.10)	0.839 (0.03)	0.950 (0.02)	0.728 (0.03)	0.840 (0.05)	0.759 (0.11)	0.771 (0.06)	0.727 (0.17)	0.901 (0.04)	0.701 (0.09)	**0.804 (0.07)**
14% user prompts	0.817 (0.11)	0.833 (0.03)	0.945 (0.02)	0.676 (0.04)	0.814 (0.05)	0.750 (0.11)	0.745 (0.07)	0.717 (0.17)	0.871 (0.06)	0.691 (0.09)	**0.786 (0.07)**
11% user prompts	0.812 (0.11)	0.826 (0.03)	0.941 (0.02)	0.618 (0.05)	0.793 (0.06)	0.743 (0.11)	0.718 (0.07)	0.711 (0.17)	0.854 (0.06)	0.688 (0.09)	**0.770 (0.08)**
9% user prompts	0.804 (0.11)	0.822 (0.03)	0.936 (0.03)	0.586 (0.04)	0.785 (0.06)	0.741 (0.10)	0.697 (0.08)	0.706 (0.17)	0.831 (0.07)	0.688 (0.09)	**0.760 (0.08)**
SAM											
33% user prompts	0.755 (0.13)	0.838 (0.03)	0.907 (0.05)	0.758 (0.04)	0.882 (0.03)	0.785 (0.09)	0.736 (0.09)	0.724 (0.18)	0.929 (0.03)	0.689 (0.13)	**0.800 (0.08)**
20% user prompts	0.752 (0.12)	0.838 (0.03)	0.905 (0.05)	0.721 (0.04)	0.856 (0.04)	0.776 (0.10)	0.716 (0.10)	0.707 (0.17)	0.912 (0.05)	0.670 (0.13)	**0.785 (0.08)**
14% user prompts	0.747 (0.13)	0.832 (0.03)	0.904 (0.05)	0.680 (0.05)	0.832 (0.05)	0.764 (0.10)	0.698 (0.10)	0.698 (0.17)	0.896 (0.06)	0.656 (0.13)	**0.771 (0.09)**
11% user prompts	0.745 (0.13)	0.828 (0.03)	0.901 (0.05)	0.629 (0.05)	0.816 (0.05)	0.756 (0.11)	0.677 (0.11)	0.695 (0.17)	0.883 (0.07)	0.651 (0.13)	**0.758 (0.09)**
9% user prompts	0.739 (0.13)	0.831 (0.03)	0.897 (0.05)	0.597 (0.04)	0.812 (0.05)	0.752 (0.11)	0.657 (0.11)	0.690 (0.17)	0.861 (0.07)	0.650 (0.13)	**0.749 (0.09)**

Table 5.8: Average DSC (and standard deviation) of bilinear (BL) bounding box interpolation with MedSAM [Ma+24] compared to SAM [Kir+23]. Note that the average DSC for MedSAM is larger than for SAM with many user prompts (33% and 20%), about equal for a medium amount of user prompts (20%). For few user prompts (11% and 9%), however, the average DSC is larger for SAM.

BL Interpolation	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.	**Average**
MedSAM											
33% user prompts	0.830 (0.11)	0.843 (0.03)	0.955 (0.01)	0.808 (0.02)	0.870 (0.04)	0.781 (0.10)	0.804 (0.06)	0.752 (0.18)	0.939 (0.02)	0.729 (0.09)	**0.831 (0.07)**
20% user prompts	0.829 (0.10)	0.842 (0.03)	0.954 (0.01)	0.790 (0.03)	0.836 (0.04)	0.777 (0.10)	0.774 (0.06)	0.733 (0.17)	0.917 (0.04)	0.706 (0.09)	**0.816 (0.07)**
14% user prompts	0.827 (0.10)	0.841 (0.03)	0.952 (0.01)	0.770 (0.03)	0.783 (0.07)	0.768 (0.10)	0.742 (0.08)	0.716 (0.17)	0.887 (0.06)	0.683 (0.09)	**0.797 (0.08)**
11% user prompts	0.823 (0.11)	0.840 (0.04)	0.949 (0.02)	0.749 (0.04)	0.747 (0.07)	0.769 (0.10)	0.708 (0.08)	0.705 (0.17)	0.861 (0.06)	0.678 (0.09)	**0.783 (0.08)**
9% user prompts	0.820 (0.10)	0.838 (0.03)	0.945 (0.03)	0.717 (0.07)	0.738 (0.07)	0.764 (0.10)	0.670 (0.10)	0.696 (0.17)	0.828 (0.07)	0.676 (0.09)	**0.769 (0.08)**
SAM											
33% user prompts	0.757 (0.13)	0.838 (0.03)	0.908 (0.05)	0.774 (0.04)	0.899 (0.02)	0.796 (0.09)	0.745 (0.09)	0.743 (0.17)	0.937 (0.02)	0.710 (0.13)	**0.811 (0.08)**
20% user prompts	0.758 (0.12)	0.838 (0.03)	0.908 (0.04)	0.767 (0.04)	0.892 (0.02)	0.794 (0.09)	0.732 (0.09)	0.734 (0.17)	0.937 (0.02)	0.696 (0.14)	**0.806 (0.08)**
14% user prompts	0.759 (0.12)	0.840 (0.03)	0.909 (0.04)	0.759 (0.03)	0.866 (0.05)	0.786 (0.09)	0.713 (0.10)	0.725 (0.17)	0.933 (0.03)	0.683 (0.14)	**0.797 (0.08)**
11% user prompts	0.757 (0.13)	0.844 (0.03)	0.908 (0.04)	0.750 (0.04)	0.849 (0.05)	0.788 (0.09)	0.694 (0.11)	0.718 (0.17)	0.931 (0.03)	0.680 (0.14)	**0.792 (0.08)**
9% user prompts	0.759 (0.13)	0.844 (0.03)	0.907 (0.04)	0.732 (0.05)	0.844 (0.05)	0.782 (0.09)	0.669 (0.12)	0.713 (0.17)	0.922 (0.03)	0.679 (0.14)	**0.785 (0.09)**

6 Discussion

In this chapter, we discuss the results presented in Chapter 5. We highlight strengths and weaknesses of the presented prompt engineering approaches and discuss when the 3D workflow using the prompt engineering approaches is applicable. Additionally, we explore how much the workflow could improve current solutions and where we need to go from here to make it a reality.

Firstly, the comparison between MedSAM and SAM is vital as MedSAM forms the basis of our proposed prompt engineering methods and the 3D workflow. Our findings suggest that MedSAM does outperform SAM, but not as significantly as reported by Ma et al. [Ma+24]. This discrepancy likely stems from the simplification in earlier studies, where smaller structures were excluded from ground truth segmentations and comparisons were made against ViT-B SAM instead of the more powerful ViT-H SAM.

Our exploration of SAM-Med3D models as an alternative foundation for a different kind of 3D workflow reveals a notable deficiency in segmentation quality, likely caused by the small amount of user input given to the model. Although the further fine-tuned version, SAM-Med3D-turbo, shows some improvement, it fails to compete effectively in more than two datasets, specifically the brain tumor and heart datasets. This highlights the critical balance between minimizing user input and maintaining segmentation quality. It remains to be seen if further fine-tuning or architectural enhancements could make these 3D models more competitive.

In contrast, our approach, leveraging MedSAM coupled with the presented prompt engineering methods, strikes a more advantageous balance. It significantly reduces user effort compared to applying MedSAM slice-by-slice, while only marginally compromising segmentation quality. Interestingly, for smooth structures, such as the heart or liver, the reduction in user prompts can be quite substantial without significantly sacrificing segmentation quality. However, there is still room for improvement. For instance, small bounding boxes at the volume edges can lead to low-quality interpolation results. It remains future work to investigate if machine learning models can do a better job at tracking 2D bounding boxes from slice to slice within the volumetric image (see Chapter 7). Additionally, fine details can sometimes be lost, a limitation not unique to our approach but inherent to MedSAM, even in its slice-by-slice application. Improving MedSAM or potentially

replacing it with another 2D model, such as SAM-Med2D, could be viable solutions, especially considering our ablation study's findings that the prompt engineering approaches are not strictly dependent on MedSAM (see Section 5.5).

Regarding the comparison between bilinear and nearest neighbor bounding box interpolation, bilinear interpolation emerges as the more effective technique. It yields smoother results, which is particularly beneficial for organs that are mostly smooth in nature. Moreover, bilinear interpolation demonstrates superior segmentation quality for more complex structures. These findings align with expectations and reinforce the suitability of bilinear bounding box interpolation for our use-case.

From the presented results we can establish guidelines for when and how to use our proposed methods and workflow:

1. The proposed prompt engineering methods combined with MedSAM work well for single-class segmentation[19] of relatively smooth structures in both MRI and CT images. The segmentation quality for organs with clear boundaries is better compared to tumors with less clear boundaries.

2. Bilinear bounding box interpolation should be used, nearest neighbor bounding box interpolation should remain a fallback option if user input is only available on a single slice.

3. The optimal interpolation distance balancing user effort and segmentation quality depends on the variability of the structure and the physical distance between slices. If slices are closer together in physical space, increasing interpolation distances can still achieve high quality segmentation results.

4. A manual check and clean up step after using the method is important to ensure high quality results, especially for structures with fine details.

While we have presented an extensive evaluation of the proposed prompt engineering methods, the 3D workflow we designed in Section 4.1 has not been tested with actual users. Even though we believe the presented ideas for the 3D workflow are promising it remains to be seen how the workflow will be received by medical domain experts. However, with the presented prompt engineering methods we have laid out the necessary foundation for implementing the workflow both in VISIAN and other medical image segmentation applications.

19 Multi-class segmentation can be performed with the same methods one class at a time.

Additionally, we want to comment on the fact that large parts of the related work we build upon has not yet been peer-reviewed by the community. This trend starts with SAM itself, which was published by researchers working at Meta, but has not undergone the traditional peer-review of a scientific conference or journal. With a large corporation, such as Meta, we have a certain amount of trust that an internal review process tested the quality of the methods and results. Other findings, such as MSA, SAM-Med2d, and SAM-Med3D, however, have been presented by smaller research groups and have not yet been peer reviewed either. MedSAM, in particular, was originally also published without peer review [Ma+23] and only recently published in Nature Communications [Ma+24]. This fact prompted us to conduct a more detailed evaluation of MedSAM, where we found less significant improvements compared to SAM than expected. This discrepancy in particular seems to stem from an oversimplification of datasets (removing small structures from ground truth segmentations) and not comparing to the most powerful SAM model available Section 5.3. We expected that both of these problems, might be brought to light in a proper peer-review, however, even in the peer reviewed version of the paper, this is not discussed.

Overall, we have demonstrated the potential and limitations of prompt engineering methods in the context of volumetric medical image segmentation, particularly using MedSAM. Our methods balance user input with segmentation quality. While our methods have shown promise, particularly in reducing user effort for smoother structures such as the heart or liver, challenges such as handling fine details and inaccuracies at the edges of the structure persist. These issues highlight the need for ongoing refinement of these techniques and potentially exploring alternative models to enhance segmentation quality.

7 Future Work

In this chapter, we discuss future work with a focus on testing the presented 3D workflow in user studies. Additionally, we present ideas for improving the prompt engineering methods and discuss improving the whole method with better underlying 2D models.

The goal of this thesis is to improve semi-automatic volumetric medical image segmentation. For this, we expand upon the AutoSeg tool in VISIAN, which allows using SAM on a slice-by-slice basis and design a new 3D workflow based on MedSAM. In this thesis, we develop and thoroughly evaluate the prompt engineering methods necessary to make this workflow work.

The next step is to conduct a user study testing the 3D workflow with real world data and actual medical domain experts. This user study will yield valuable feedback and ways for improvement. This will, at the same time, test the usability and user-friendliness of our 3D workflow.

For this user study, we plan to integrate the workflow in VISIAN, building upon the browser-server architecture which is already present for the AutoSeg tool.

With regards to improving the prompt engineering methods, we have two main ideas for the future.

Firstly, we can borrow ideas from object tracking and segmentation in videos, which has been done in a variety of ways [Yao+20]. Recently, SAM has also been used for this [Yan+23]. Volumetric medical images, as a series of 2D slices, can also be interpreted like a video where objects, such as organs or tumors, could be tracked and segmented. Initial research in this field, applying SAM, has already shown promise [Che+23b]. We intend to explore this further.

Secondly, we have only experimented with prompt engineering for bounding boxes so far. While bounding boxes have shown better segmentation results than point prompts, the latter are also an option. More specifically, we intend to exploit the 3D nature of volumetric medical images allowing us to slice the volume from three different directions. Thus, from one (or multiple) segmented slices from a specific slicing direction point prompts for many slices from a different slicing direction could be extracted. The segmentation quality of the resulting volumetric

segmentations, of course, remains to be evaluated.

To improve the segmentation quality of our whole method, another approach could be to simply improve the underlying 2D model. As our ablation study suggests that the presented prompt engineering methods are not only suited for MedSAM, other 2D models might perform better, especially regarding finer details of more complex structures. In particular, SAM-Med2D and MSA are viable options. Further, it remains to be seen if 3D models, such as SAM-Med3D, improve more rapidly and become competitive.

With these future directions in mind, we now turn to the conclusion of this thesis, where we will summarize our key contributions, reflect on the implications of our work, and consider its impact on the field of medical image segmentation.

8 Conclusion

To conclude this thesis, in this chapter, we provide a summary of our main contributions, reflect on the implications of our research, and discuss its influence on the medical image segmentation field.

Manual segmentation of medical images, performed by medical domain experts, is a precise but time-consuming process involving tools for delineating specific image segments. For volumetric images, this approach requires laborious slice-by-slice segmentation, which is even more time-consuming and, thus, expensive and can be inconsistent due to inter- and intra-observer variability.

Deep learning has revolutionized this field, offering automated segmentation solutions that promise efficiency and accuracy. However, supervised learning models which can produce high quality segmentations require an extensive amount of segmented training data, while semi-supervised and unsupervised methods can work with less segmented data but often fall short in accuracy. In response, semi-automatic segmentation has emerged as a viable alternative, combining the precision of manual methods with the speed of automation. Additionally, semi-automatic segmentation presents a valuable pathway towards obtaining enough segmented training data for supervised learning models.

Thus, we set out to develop a semi-automatic segmentation tool that is versatile enough to adapt to a variety of medical imaging scenarios and can break the slice-by-slice segmentation workflow. In the rise of powerful promptable segmentation with high zero-shot performance, we designed a 3D workflow applying MedSAM to volumetric medical image segmentation. Our workflow requires significantly less effort from experts compared to using MedSAM slice-by-slice. With this workflow in mind, the focus of this thesis was the design and evaluation of novel prompt engineering methods which allow the 3D workflow to reduce the manual effort for medical domain experts. While the prompt engineering methods are rather straight forward, being based on nearest neighbor and bilinear bounding box interpolation, they are surprisingly powerful. In particular, the bilinear bounding box interpolation showed the ability to reduce the manual prompting input to 33% with only a minor reduction in DSC (0.0175) on average. For some datasets, including brain tumor, heart, and liver segmentation, it is even possible to reduce manual prompting to around 9% with a DSC reduction of only around 0.01. Nearest

neighbor bounding box interpolation presents a slightly less powerful but more flexible alternative, as it only requires user input on a single slice instead of two slices. The simplicity of the proposed prompt engineering methods and the early stage of SAM-based models for medical image segmentation, however, leave room for future improvement. Additionally, the subjective evaluation of the 3D workflow remains for future work, which we outlined in Chapter 7.

Our work highlights an important direction for volumetric medical image segmentation: breaking out of the slice-by-slice workflow. This is an important step as this slice-by-slice approach is what makes volumetric medical image segmentation inherently time-consuming. Due to the high variability of medical images and corresponding segmentation tasks, powerful general purpose segmentation tools are important yet difficult to achieve. The surprising zero-shot performance of SAM and especially of models adapted and fine-tuned for medical images, such as MedSAM, present a unique opportunity in this space. The combination of this and the aforementioned break of the slice-by-slice workflow has the potential to transform semi-automatic volumetric medical image segmentation.

At the same time, while we propose straight forward prompt engineering strategies, it leaves room for future improvement. Additionally, we can conclude that enabling the 3D workflow with the presented prompt engineering methods is definitely feasible.

In summary, we have shown that bilinear bounding box interpolation can be used to reduce manual prompting effort when applying 2D promptable segmentation models, such as SAM and MedSAM, to volumetric medical image segmentation. While the reduction in prompting effort is significant, the reduction in segmentation quality is minor. This allows breaking out of the traditional, time-consuming slice-by-slice workflow when segmenting volumetric medical images, which is a significant step in streamlining semi-automatic volumetric medical image segmentation.

Bibliography

[AB94] R. Adams and L. Bischof. **Seeded region growing**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16:6 (June 1994), 641–647. DOI: 10.1109/34.295913. URL: https://doi.org/10.1109/34.295913 (see page 4).

[Aer+14] Hugo J. W. L. Aerts, Emmanuel Rios Velazquez, Ralph T. H. Leijenaar, Chintan Parmar, Patrick Grossmann, Sara Carvalho, Johan Bussink, René Monshouwer, Benjamin Haibe-Kains, Derek Rietveld, Frank Hoebers, Michelle M. Rietbergen, C. René Leemans, Andre Dekker, John Quackenbush, Robert J. Gillies, and Philippe Lambin. **Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach**. *Nature Communications* 5:1 (June 2014). ISSN: 2041-1723. DOI: 10.1038/ncomms5006. URL: http://dx.doi.org/10.1038/ncomms5006 (see page 1).

[Ang+18] Sui Paul Ang, Son Lam Phung, Mark Matthias Schira, Abdesselam Bouzerdoum, and Soan Thi Minh Duong. **Human Brain Tissue Segmentation in fMRI using Deep Long-Term Recurrent Convolutional Network**. In: *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, Dec. 2018. DOI: 10.1109/dicta.2018.8615850. URL: http://dx.doi.org/10.1109/DICTA.2018.8615850 (see page 11).

[Ant+22] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Bram van Ginneken, Michel Bilello, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc J. Gollub, Stephan H. Heckers, Henkjan Huisman, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Jennifer S. Golia Pernicka, Kawal Rhode, Catalina Tobon-Gomez, Eugene Vorontsov, James A. Meakin, Sebastien Ourselin, Manuel Wiesenfarth, Pablo Arbeláez, Byeonguk Bae, Sihong Chen, Laura Daza, Jianjiang Feng, Baochun He, Fabian Isensee, Yuanfeng Ji, Fucang Jia, Ildoo Kim, Klaus Maier-Hein, Dorit Merhof, Akshay Pai, Beomhee Park, Mathias Perslev, Ramin Rezaiifar, Oliver Rippel, Ignacio Sarasua, Wei Shen, Jaemin Son, Christian Wachinger, Liansheng Wang, Yan Wang, Yingda Xia, Daguang Xu, Zhanwei Xu, Yefeng Zheng, Amber L. Simpson, Lena Maier-Hein, and M. Jorge Cardoso. **The Medical Segmentation Decathlon**. *Nature Communications* 13:1 (July 2022). DOI: 10.1038/s41467-022-30695-9. URL: https://doi.org/10.1038/s41467-022-30695-9 (see pages 3, 5, 10, 35, 36, 46, 47).

[Bai+20] Wenjia Bai, Hideaki Suzuki, Jian Huang, Catherine Francis, Shuo Wang, Giacomo Tarroni, Florian Guitton, Nay Aung, Kenneth Fung, Steffen E. Petersen, Stefan K. Piechnik, Stefan Neubauer, Evangelos Evangelou, Abbas Dehghan, Declan P. O'Regan, Martin R. Wilkins, Yike Guo, Paul M. Matthews, and Daniel Rueckert. **A population-based phenome-wide association study of cardiac and aortic structure and function**. *Nature Medicine* 26:10 (Aug. 2020), 1654–1662. DOI: 10.1038/s41591-020-1009-y. URL: https://doi.org/10.1038/s41591-020-1009-y (see page 1).

[BL79] Serge Beucher and Christian Lantuéjoul. **Use of Watersheds in Contour Detection**. In: vol. 132. Jan. 1979 (see page 4).

[BM17] Yahya Baba and Andrew Murphy. *Windowing (CT)*. Mar. 2017. DOI: 10.53347/rid-52108. URL: https://doi.org/10.53347/rid-52108 (see pages 9, 40).

[CBP19] Veronika Cheplygina, Marleen de Bruijne, and Josien P.W. Pluim. **Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis**. *Medical Image Analysis* 54 (May 2019), 280–296. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.03.009. URL: http://dx.doi.org/10.1016/j.media.2019.03.009 (see pages 3, 11).

[Che+21] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*. 2021. DOI: 10.48550/ARXIV.2102.04306. URL: https://arxiv.org/abs/2102.04306 (see page 11).

[Che+23a] Junlong Cheng, Jin Ye, Zhongying Deng, Jianpin Chen, Tianbin Li, Haoyu Wang, Yanzhou Su, Ziyan Huang, Jilong Chen, Lei Jiang, Hui Sun, Junjun He, Shaoting Zhang, Min Zhu, and Yu Qiao. *SAM-Med2D*. 2023. DOI: 10.48550/ARXIV.2308.16184. URL: https://arxiv.org/abs/2308.16184 (see pages 13, 24).

[Che+23b] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. *Segment and Track Anything*. 2023. DOI: 10.48550/ARXIV.2305.06558. URL: https://arxiv.org/abs/2305.06558 (see page 55).

[Cov+22] Elise C. Covert, Kellen Fitzpatrick, Justin Mikell, Ravi K. Kaza, John D. Millet, Daniel Barkmeier, Joseph Gemmete, Jared Christensen, Matthew J. Schipper, and Yuni K. Dewaraja. **Intra- and inter-operator variability in MRI-based manual segmentation of HCC lesions and its impact on dosimetry**. *EJNMMI Physics* 9:1 (Dec. 2022). ISSN: 2197-7364. DOI: 10.1186/s40658-022-00515-6. URL: http://dx.doi.org/10.1186/s40658-022-00515-6 (see pages 2, 3).

[Cru+21] Leonardo da Cruz, César Sierra-Franco, Greis Silva-Calpa, and Alberto Raposo. **Enabling Autonomous Medical Image Data Annotation: A human-in-the-loop Reinforcement Learning Approach**. In: *Annals of Computer Science and Information Systems*. IEEE, Sept. 2021. DOI: 10.15439/2021f86. URL: https://doi.org/10.15439/2021f86 (see pages 2, 3).

[De +18] Jeffrey De Fauw, Joseph R. Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, George van den Driessche, Balaji Lakshminarayanan, Clemens Meyer, Faith Mackinder, Simon Bouton, Kareem Ayoub, Reena Chopra, Dominic King, Alan Karthikesalingam, Cían O. Hughes, Rosalind Raine, Julian Hughes, Dawn A. Sim, Catherine Egan, Adnan Tufail, Hugh Montgomery, Demis Hassabis, Geraint Rees, Trevor Back, Peng T. Khaw, Mustafa Suleyman, Julien Cornebise, Pearse A. Keane, and Olaf Ronneberger. **Clinically applicable deep learning for diagnosis and referral in retinal disease**. *Nature Medicine* 24:9 (Aug. 2018), 1342–1350. ISSN: 1546-170X. DOI: 10.1038/s41591-018-0107-6. URL: http://dx.doi.org/10.1038/s41591-018-0107-6 (see page 1).

[Dew+10] Jeffrey Dewey, George Hana, Troy Russell, Jared Price, Daniel McCaffrey, Jaroslaw Harezlak, Ekta Sem, Joy C. Anyanwu, Charles R. Guttmann, Bradford Navia, Ronald Cohen, and David F. Tate. **Reliability and validity of MRI-based automated volumetry software relative to auto-assisted manual measurement of subcortical structures in HIV-infected patients from a multisite study**. *NeuroImage* 51:4 (July 2010), 1334–1344. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2010.03.033. URL: http://dx.doi.org/10.1016/j.neuroimage.2010.03.033 (see page 10).

[Dic45] Lee R. Dice. **Measures of the Amount of Ecologic Association Between Species**. *Ecology* 26:3 (July 1945), 297–302. ISSN: 1939-9170. DOI: 10.2307/1932409. URL: http://dx.doi.org/10.2307/1932409 (see page 37).

[Dos+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929 (see pages 11, 12).

[Egg+12] Lucas D. Eggert, Jens Sommer, Andreas Jansen, Tilo Kircher, and Carsten Konrad. **Accuracy and Reliability of Automated Gray Matter Segmentation Pathways on Real and Simulated Structural Magnetic Resonance Images of the Human Brain**. *PLoS ONE* 7:9 (Sept. 2012). Ed. by Yong Fan, e45081. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0045081. URL: http://dx.doi.org/10.1371/journal.pone.0045081 (see page 10).

[Fal+18] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün undefinediçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, Alexander Dovzhenko, Olaf Tietz, Cristina Dal Bosco, Sean Walsh, Deniz Saltukoglu, Tuan Leng Tay, Marco Prinz, Klaus Palme, Matias Simons, Ilka Diester, Thomas Brox, and Olaf Ronneberger. **U-Net: deep learning for cell counting, detection, and morphometry**. *Nature Methods* 16:1 (Dec. 2018), 67–70. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0261-2. URL: http://dx.doi.org/10.1038/s41592-018-0261-2 (see page 1).

[Fed+12] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V. Miller, Steve Pieper, and Ron Kikinis. **3D Slicer as an image computing platform for the Quantitative Imaging Network**. *Magnetic Resonance Imaging* 30:9 (Nov. 2012), 1323–1341. DOI: 10.1016/j.mri.2012.05.001. URL: https://doi.org/10.1016/j.mri.2012.05.001 (see pages 2, 9, 15, 73).

[Fis12] Bruce Fischl. **FreeSurfer**. *NeuroImage* 62:2 (Aug. 2012), 774–781. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2012.01.021. URL: http://dx.doi.org/10.1016/j.neuroimage.2012.01.021 (see page 10).

[Fu+21] Yabo Fu, Yang Lei, Tonghe Wang, Walter J. Curran, Tian Liu, and Xiaofeng Yang. **A review of deep learning based methods for medical image multi-organ segmentation**. *Physica Medica* 85 (May 2021), 107–122. ISSN: 1120-1797. DOI: 10.1016/j.ejmp.2021.05.003. URL: http://dx.doi.org/10.1016/j.ejmp.2021.05.003 (see page 19).

[Gao+18] Yang Gao, Jeff M. Phillips, Yan Zheng, Renqiang Min, P. Thomas Fletcher, and Guido Gerig. **Fully convolutional structured LSTM networks for joint 4D medical image segmentation**. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, Apr. 2018. DOI: 10.1109/isbi.2018.8363764. URL: http://dx.doi.org/10.1109/ISBI.2018.8363764 (see page 11).

[Gol07] L. W. Goldman. **Principles of CT and CT Technology**. *Journal of Nuclear Medicine Technology* 35:3 (Sept. 2007), 115–128. DOI: 10.2967/jnmt.107.042978. URL: https://doi.org/10.2967/jnmt.107.042978 (see page 8).

[Gon+23] Shizhan Gong, Yuan Zhong, Wenao Ma, Jinpeng Li, Zhao Wang, Jingyang Zhang, Pheng-Ann Heng, and Qi Dou. *3DSAM-adapter: Holistic Adaptation of SAM from 2D to 3D for Promptable Medical Image Segmentation*. 2023. DOI: 10.48550/ARXIV.2306.13465. URL: https://arxiv.org/abs/2306.13465 (see page 13).

[Gro+15] Vijay P.B. Grover, Joshua M. Tognarelli, Mary M.E. Crossey, I. Jane Cox, Simon D. Taylor-Robinson, and Mark J.W. McPhail. **Magnetic Resonance Imaging: Principles and Techniques: Lessons for Clinicians**. *Journal of Clinical and Experimental Hepatology* 5:3 (Sept. 2015), 246–255. DOI: 10.1016/j.jceh.2015.08.001. URL: https://doi.org/10.1016/j.jceh.2015.08.001 (see page 7).

[Han+21] Ulrik Stig Hansen, Eric Landau, Mehul Patel, and Bu'Hussain Hayee. **Novel artificial intelligence-driven software significantly shortens the time required for annotation in computer vision projects**. *Endoscopy International Open* 09:04 (Apr. 2021), E621–E626. DOI: 10.1055/a-1341-0689. URL: https://doi.org/10.1055/a-1341-0689 (see pages 2, 9, 15).

[Hav+17] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. **Brain tumor segmentation with Deep Neural Networks**. *Medical Image Analysis* 35 (Jan. 2017), 18–31. ISSN: 1361-8415. DOI: 10.1016/j.media.2016.05.004. URL: http://dx.doi.org/10.1016/j.media.2016.05.004 (see page 36).

[He+23] Sheng He, Rina Bao, Jingpeng Li, Jeffrey Stout, Atle Bjornerud, P. Ellen Grant, and Yangming Ou. *Computer-Vision Benchmark Segment-Anything Model (SAM) in Medical Images: Accuracy in 12 Datasets*. 2023. DOI: 10.48550/ARXIV.2304.09324. URL: https://arxiv.org/abs/2304.09324 (see pages 13, 22).

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. **Long Short-Term Memory**. *Neural Computation* 9:8 (Nov. 1997), 1735–1780. ISSN: 1530-888X. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735 (see page 10).

[Ise+20] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. **nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation**. *Nature Methods* 18:2 (Dec. 2020), 203–211. DOI: 10.1038/s41592-020-01008-z. URL: https://doi.org/10.1038/s41592-020-01008-z (see pages 10, 14, 38, 39).

[Kei23] Richard Keil. **AutoSeg: Implementation of a Machine-Learning based Medical Image Segmentation Tool in the Visian Editor**. Bachelor Thesis. Aug. 2023 (see pages 22, 23, 25).

[Kic+19] Philipp Kickingereder, Fabian Isensee, Irada Tursunova, Jens Petersen, Ulf Neuberger, David Bonekamp, Gianluca Brugnara, Marianne Schell, Tobias Kessler, Martha Foltyn, Inga Harting, Felix Sahm, Marcel Prager, Martha Nowosielski, Antje Wick, Marco Nolden, Alexander Radbruch, Jürgen Debus, Heinz-Peter Schlemmer, Sabine Heiland, Michael Platten, Andreas von Deimling, Martin J van den Bent, Thierry Gorlia, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein. **Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural**

networks: a multicentre, retrospective study. *The Lancet Oncology* 20:5 (May 2019), 728–740. ISSN: 1470-2045. DOI: 10.1016/s1470-2045(19)30098-1. URL: http://dx.doi.org/10.1016/S1470-2045(19)30098-1 (see page 1).

[Kir+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. *Segment Anything*. 2023. DOI: 10.48550/ARXIV.2304.02643. URL: https://arxiv.org/abs/2304.02643 (see pages 4, 13, 20, 21, 37, 41, 48, 49).

[Kor+21] Jonas Kordt, Paul Brachmann, Daniel Limberger, and Christoph Lippert. **Interactive Volumetric Region Growing for Brain Tumor Segmentation on MRI using WebGL**. In: *The 26th International Conference on 3D Web Technology*. ACM, Nov. 2021. DOI: 10.1145/3485444.3487640. URL: https://doi.org/10.1145/3485444.3487640 (see pages 2–4, 9, 10, 15, 18, 25).

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. **ImageNet classification with deep convolutional neural networks**. *Communications of the ACM* 60:6 (May 2017), 84–90. ISSN: 1557-7317. DOI: 10.1145/3065386. URL: http://dx.doi.org/10.1145/3065386 (see page 78).

[Liu+23] Yihao Liu, Jiaming Zhang, Zhangcong She, Amir Kheradmand, and Mehran Armand. *SAMM (Segment Any Medical Model): A 3D Slicer Integration to SAM*. 2023. DOI: 10.48550/ARXIV.2304.05622. URL: https://arxiv.org/abs/2304.05622 (see page 22).

[LJT00] Zheng Lin, Jesse Jin, and Hugues Talbot. **Unseeded Region Growing for 3D Image Segmentation.** In: *ACM International Conference Proceeding Series*. Vol. 9. 2000, 31–37 (see page 4).

[LLC20] Wei Li, Xuefeng Lin, and Xi Chen. **Detecting Alzheimer's disease Based on 4D fMRI: An exploration under deep learning framework**. *Neurocomputing* 388 (May 2020), 280–287. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.01.053. URL: http://dx.doi.org/10.1016/j.neucom.2020.01.053 (see page 11).

[LSQ22] Jiyun Li, Binbin Song, and Chen Qian. **Diagnosis of Alzheimer's disease by feature weighted-LSTM: a preliminary study of temporal features in brain resting-state fMRI**. *Journal of Integrative Neuroscience* 21:2 (Mar. 2022), 056. ISSN: 0219-6352. DOI: 10.31083/j.jin2102056. URL: http://dx.doi.org/10.31083/j.jin2102056 (see page 11).

[Ma+23] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. *Segment Anything in Medical Images*. 2023. DOI: 10.48550/ARXIV.2304.12306. URL: https://arxiv.org/abs/2304.12306 (see page 53).

[Ma+24] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. **Segment anything in medical images**. *Nature Communications* 15:1 (Jan. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-44824-z. URL: http://dx.doi.org/10.1038/s41467-024-44824-z (see pages 4, 5, 7, 13, 14, 22, 28, 31, 37–41, 48, 49, 51, 53).

[May+16] Kristina N. Mayer, Beatrice Latal, Walter Knirsch, Ianina Scheer, Michael von Rhein, Bettina Reich, Jürgen Bauer, Kerstin Gummel, Neil Roberts, and Ruth O'Gorman Tuura. **Comparison of automated brain volumetry methods with stereology in children aged 2 to 3 years**. *Neuroradiology* 58:9 (July 2016), 901–910. ISSN: 1432-1920. DOI: 10.1007/s00234-016-1714-x. URL: http://dx.doi.org/10.1007/s00234-016-1714-x (see page 10).

[Maz+23] Maciej A. Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. **Segment anything model for medical image analysis: An experimental study**. *Medical Image Analysis* 89 (Oct. 2023), 102918. ISSN: 1361-8415. DOI: 10.1016/j.media.2023.102918. URL: http://dx.doi.org/10.1016/j.media.2023.102918 (see pages 13, 22, 28).

[Mil+20] Jason M. Millward, Paula Ramos Delgado, Alina Smorodchenko, Laura Boehmert, Joao Periquito, Henning M. Reimann, Christian Prinz, Antje Els, Michael Scheel, Judith Bellmann-Strobl, Helmar Waiczies, Jens Wuerfel, Carmen Infante-Duarte, Andreas Pohlmann, Frauke Zipp, Friedemann Paul, Thoralf Niendorf, and Sonia Waiczies. **Transient enlargement of brain ventricles during relapsing-remitting multiple sclerosis and experimental autoimmune encephalomyelitis**. *JCI Insight* 5:21 (Nov. 2020). DOI: 10.1172/jci.insight.140040. URL: https://doi.org/10.1172/jci.insight.140040 (see page 1).

[Mus+21] Wessam Mustafa, Sherif Ali, Nadia Elgendy, Samer Salama, Lamiaa El Sorogy, and Mohamed Mohsen. **Role of contrast-enhanced FLAIR MRI in diagnosis of intracranial lesions**. *The Egyptian Journal of Neurology, Psychiatry and Neurosurgery* 57:1 (Aug. 2021). DOI: 10.1186/s41983-021-00360-x. URL: https://doi.org/10.1186/s41983-021-00360-x (see page 7).

[NUZ00] L.G. Nyul, J.K. Udupa, and Xuan Zhang. **New variants of a method of MRI scale standardization**. *IEEE Transactions on Medical Imaging* 19:2 (2000), 143–150. DOI: 10.1109/42.836373. URL: https://doi.org/10.1109/42.836373 (see page 8).

[Rad+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. **Learning Transferable Visual Models From Natural Language Supervision**. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July

2021, 8748–8763. URL: https://proceedings.mlr.press/v139/radford21a.html (see page 20).

[Ram+18] K.K.D. Ramesh, G. Kumar, K. Swapna, Debabrata Datta, and S. Rajest. **A Review of Medical Image Segmentation Algorithms**. *EAI Endorsed Transactions on Pervasive Health and Technology* (July 2018), 169184. DOI: 10.4108/eai.12-4-2021.169184. URL: https://doi.org/10.4108/eai.12-4-2021.169184 (see pages 3, 4).

[Ren+20] Félix Renard, Soulaimane Guedria, Noel De Palma, and Nicolas Vuillerme. **Variability and reproducibility in deep learning for medical image segmentation**. *Scientific Reports* 10:1 (Aug. 2020). DOI: 10.1038/s41598-020-69920-0. URL: https://doi.org/10.1038/s41598-020-69920-0 (see pages 1–3).

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, 234–241. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015. ISBN: 9783319245744. DOI: 10.1007/978-3-319-24574-4_28. URL: http://dx.doi.org/10.1007/978-3-319-24574-4_28 (see pages 10, 11).

[Ros18] Tizian Rosenstock. **Risk stratification in motor area-related glioma surgery based on navigated transcranial magnetic stimulation data**. PhD thesis. Charité - Universitätsmedizin Berlin, 2018. DOI: 10.17169/REFUBIUM-6070. URL: https://refubium.fu-berlin.de/handle/fub188/1868 (see page 1).

[Roy+23] Saikat Roy, Tassilo Wald, Gregor Koehler, Maximilian R. Rokuss, Nico Disch, Julius Holzschuh, David Zimmerer, and Klaus H. Maier-Hein. *SAM.MD: Zero-shot medical image segmentation capabilities of the Segment Anything Model*. 2023. DOI: 10.48550/ARXIV.2304.05396. URL: https://arxiv.org/abs/2304.05396 (see pages 13, 22, 28, 31).

[RS21] Khalid Raza and Nripendra Kumar Singh. **A Tour of Unsupervised Deep Learning for Medical Image Analysis**. *Current Medical Imaging Formerly Current Medical Imaging Reviews* 17:9 (Sept. 2021), 1059–1077. ISSN: 1573-4056. DOI: 10.2174/1573405617666210127154257. URL: http://dx.doi.org/10.2174/1573405617666210127154257 (see page 11).

[SB06] H.R. Sheikh and A.C. Bovik. **Image information and visual quality**. *IEEE Transactions on Image Processing* 15:2 (Feb. 2006), 430–444. ISSN: 1057-7149. DOI: 10.1109/tip.2005.859378. URL: http://dx.doi.org/10.1109/TIP.2005.859378 (see pages 77, 79, 80).

[SFN08] Tobias Sielhorst, Marco Feuerstein, and Nassir Navab. **Advanced Medical Displays: A Literature Review of Augmented Reality**. *Journal of Display Technology* 4:4 (Dec. 2008), 451–467. DOI: 10.1109/jdt.2008.2001575. URL: https://doi.org/10.1109/jdt.2008.2001575 (see page 1).

[She+21] Yiqiu Shen, Nan Wu, Jason Phang, Jungkyu Park, Kangning Liu, Sudarshini Tyagi, Laura Heacock, S. Gene Kim, Linda Moy, Kyunghyun Cho, and Krzysztof J. Geras. **An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization**. *Medical Image Analysis* 68 (Feb. 2021), 101908. ISSN: 1361-8415. DOI: 10.1016/j.media.2020.101908. URL: http://dx.doi.org/10.1016/j.media.2020.101908 (see page 7).

[SJ15] V.R. Simi and Justin Joseph. **Segmentation of Glioblastoma Multiforme from MR Images – A comprehensive review**. *The Egyptian Journal of Radiology and Nuclear Medicine* 46:4 (Dec. 2015), 1105–1110. ISSN: 0378-603X. DOI: 10.1016/j.ejrnm.2015.08.001. URL: http://dx.doi.org/10.1016/j.ejrnm.2015.08.001 (see page 36).

[Son+23] Joomee Song, Juyoung Hahm, Jisoo Lee, Chae Yeon Lim, Myung Jin Chung, Jinyoung Youn, Jin Whan Cho, Jong Hyeon Ahn, and Kyungsu Kim. **Comparative validation of AI and non-AI methods in MRI volumetry to diagnose Parkinsonian syndromes**. *Scientific Reports* 13:1 (Mar. 2023). ISSN: 2045-2322. DOI: 10.1038/s41598-023-30381-w. URL: http://dx.doi.org/10.1038/s41598-023-30381-w (see page 10).

[SWS17] Dinggang Shen, Guorong Wu, and Heung-Il Suk. **Deep Learning in Medical Image Analysis**. *Annual Review of Biomedical Engineering* 19:1 (June 2017), 221–248. DOI: 10.1146/annurev-bioeng-071516-044442. URL: https://doi.org/10.1146/annurev-bioeng-071516-044442 (see pages 2, 3).

[Tan+20] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. **Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains**. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, 7537–7547. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf (see page 20).

[TH15] Abdel Aziz Taha and Allan Hanbury. **Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool**. *BMC Medical Imaging* 15:1 (Aug. 2015). ISSN: 1471-2342. DOI: 10.1186/s12880-015-0068-x. URL: http://dx.doi.org/10.1186/s12880-015-0068-x (see page 37).

[Uhe+21] Tomas Uher, Jan Krasensky, Charles Malpas, Niels Bergsland, Michael G. Dwyer, Eva Kubala Havrdova, Manuela Vaneckova, Dana Horakova, Robert Zivadinov, and Tomas Kalincik. **Evolution of Brain Volume Loss Rates in Early Stages of Multiple Sclerosis**. *Neurology - Neuroimmunology Neuroinflammation* 8:3 (Mar. 2021), e979. DOI: 10.1212/nxi.0000000000000979. URL: https://doi.org/10.1212/nxi.0000000000000979 (see page 1).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. **Attention is All you Need**. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (see pages 11, 21).

[Wah+21] Kareem A. Wahid, Renjie He, Brigid A. McDonald, Brian M. Anderson, Travis Salzillo, Sam Mulder, Jarey Wang, Christina Setareh Sharafi, Lance A. McCoy, Mohamed A. Naser, Sara Ahmed, Keith L. Sanders, Abdallah S.R. Mohamed, Yao Ding, Jihong Wang, Kate Hutcheson, Stephen Y. Lai, Clifton D. Fuller, and Lisanne V. van Dijk. **Intensity standardization methods in magnetic resonance imaging of head and neck cancer**. *Physics and Imaging in Radiation Oncology* 20 (Oct. 2021), 88–93. DOI: 10.1016/j.phro.2021.11.001. URL: https://doi.org/10.1016/j.phro.2021.11.001 (see page 9).

[Wan+04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. **Image Quality Assessment: From Error Visibility to Structural Similarity**. *IEEE Transactions on Image Processing* 13:4 (Apr. 2004), 600–612. ISSN: 1057-7149. DOI: 10.1109/tip.2003.819861. URL: http://dx.doi.org/10.1109/TIP.2003.819861 (see pages 76, 79, 80).

[Wan+21] Shanshan Wang, Cheng Li, Rongpin Wang, Zaiyi Liu, Meiyun Wang, Hongna Tan, Yaping Wu, Xinfeng Liu, Hui Sun, Rui Yang, Xin Liu, Jie Chen, Huihui Zhou, Ismail Ben Ayed, and Hairong Zheng. **Annotation-efficient deep learning for automatic medical image segmentation**. *Nature Communications* 12:1 (Oct. 2021). DOI: 10.1038/s41467-021-26216-9. URL: https://doi.org/10.1038/s41467-021-26216-9 (see page 1).

[Wan+22] Risheng Wang, Tao Lei, Ruixia Cui, Bingtao Zhang, Hongying Meng, and Asoke K. Nandi. **Medical image segmentation using deep learning: A survey**. *IET Image Processing* 16:5 (Jan. 2022), 1243–1267. DOI: 10.1049/ipr2.12419. URL: https://doi.org/10.1049/ipr2.12419 (see pages 1–3, 10–12).

[Wan+23] Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, Bin Fu, Shaoting Zhang, Junjun He, and Yu Qiao. *SAM-Med3D*. 2023. DOI: 10.48550/ARXIV.2310.15161. URL: https://arxiv.org/abs/2310.15161 (see pages 13, 24, 37, 40, 41).

[Wen+21] Yang Wen, Leiting Chen, Yu Deng, and Chuan Zhou. **Rethinking pre-training on medical imaging**. *Journal of Visual Communication and Image Representation* 78 (July 2021), 103145. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2021.103145. URL: http://dx.doi.org/10.1016/j.jvcir.2021.103145 (see page 7).

[WK07] D.J. Withey and Z.J. Koles. **Medical Image Segmentation: Methods and Software**. In: *2007 Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging*. IEEE, Oct. 2007. DOI: 10.1109/nfsi-icfbi.2007.4387709. URL: http://dx.doi.org/10.1109/NFSI-ICFBI.2007.4387709 (see page 10).

[Wol+04] Ivo Wolf, Marcus Vetter, Ingmar Wegner, Marco Nolden, Thomas Bottger, Mark Hastenteufel, Max Schobinger, Tobias Kunert, and Hans-Peter Meinzer. **The medical imaging interaction toolkit (MITK): a toolkit facilitating the creation of interactive software by extending VTK and ITK**. In: *SPIE Proceedings*. Ed. by Jr. Robert L. Galloway. SPIE, May 2004. DOI: 10.1117/12.535112. URL: https://doi.org/10.1117/12.535112 (see pages 2, 9, 15, 72).

[WSB03] Z. Wang, E.P. Simoncelli, and A.C. Bovik. **Multiscale structural similarity for image quality assessment**. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216 (see pages 77, 79, 80).

[Wu+23] Junde Wu, Yu Zhang, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, and Yueming Jin. *Medical SAM Adapter: Adapting Segment Anything Model for Medical Image Segmentation*. 2023. DOI: 10.48550/ARXIV.2304.12620. URL: https://arxiv.org/abs/2304.12620 (see page 13).

[WX23] Junde Wu and Min Xu. *One-Prompt to Segment All Medical Images*. 2023. DOI: 10.48550/ARXIV.2305.10300. URL: https://arxiv.org/abs/2305.10300 (see page 13).

[Xia+23] Hanguang Xiao, Li Li, Qiyuan Liu, Xiuhong Zhu, and Qihang Zhang. **Transformers in medical image segmentation: A review**. *Biomedical Signal Processing and Control* 84 (July 2023), 104791. ISSN: 1746-8094. DOI: 10.1016/j.bspc.2023.104791. URL: http://dx.doi.org/10.1016/j.bspc.2023.104791 (see page 11).

[XRV19] Yueyang Xu, Ashish Raj, and Jonathan D. Victor. **Systematic Differences Between Perceptually Relevant Image Statistics of Brain MRI and Natural Images**. *Frontiers in Neuroinformatics* 13 (June 2019). ISSN: 1662-5196. DOI: 10.3389/fninf.2019.00046. URL: http://dx.doi.org/10.3389/fninf.2019.00046 (see page 7).

[Yan+23] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. *Track Anything: Segment Anything Meets Videos*. 2023. DOI: 10.48550/ARXIV.2304.11968. URL: https://arxiv.org/abs/2304.11968 (see page 55).

[Yao+20] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. **Video Object Segmentation and Tracking: A Survey**. *ACM Transactions on Intelligent Systems and Technology* 11:4 (May 2020), 1–47. ISSN: 2157-6912. DOI: 10.1145/3391743. URL: http://dx.doi.org/10.1145/3391743 (see page 55).

[Ye+23] Jin Ye, Junlong Cheng, Jianpin Chen, Zhongying Deng, Tianbin Li, Haoyu Wang, Yanzhou Su, Ziyan Huang, Jilong Chen, Lei Jiang, Hui Sun, Min Zhu, Shaoting Zhang, Junjun He, and Yu Qiao. *SA-Med2D-20M Dataset: Segment Anything in 2D Medical Imaging with 20 Million masks*. 2023. DOI: 10.48550/ARXIV.2311.11969. URL: https://arxiv.org/abs/2311.11969 (see pages 13, 24).

[Yus+06] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, et al. **User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability**. *NeuroImage* 31:3 (2006), 1116–1128. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2006.01.015. URL: https://www.sciencedirect.com/science/article/pii/S1053811906000632 (see pages 2, 9, 15, 18, 71).

[Zha+18] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. **The Unreasonable Effectiveness of Deep Features as a Perceptual Metric**. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018. DOI: 10.1109/cvpr.2018.00068. URL: http://dx.doi.org/10.1109/CVPR.2018.00068 (see pages 77, 79, 80).

A Additional Images

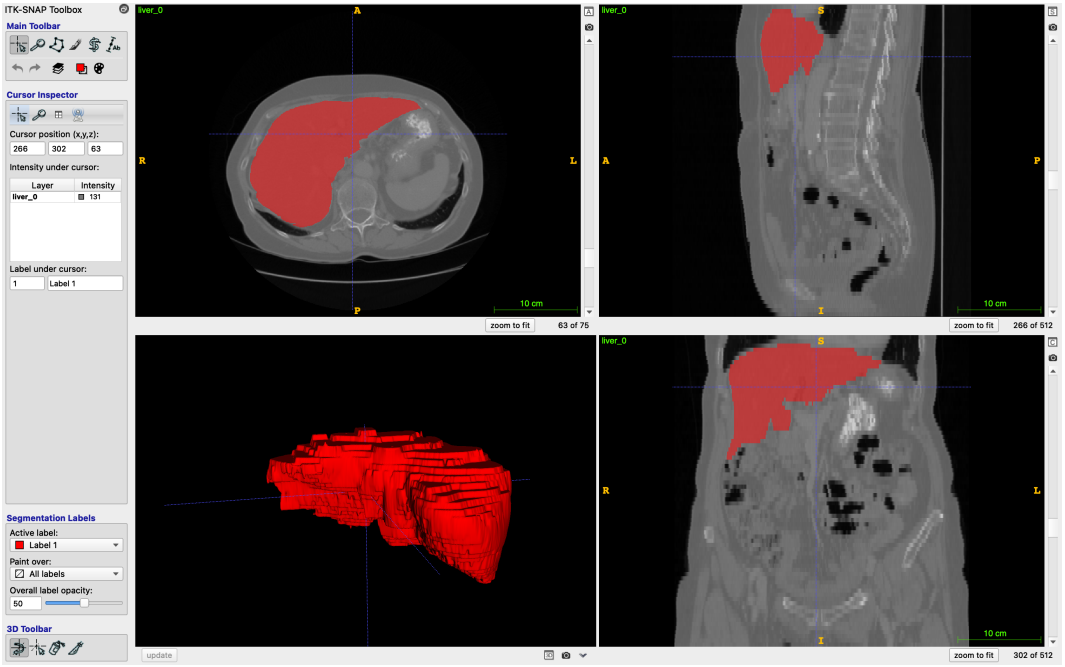


Figure A.1: ITK-SNAP [Yus+06] user interface showing an abdomen CT image with liver segmentation. The default view gives an equal amount of space to the 3 2D views and the 3D view. Any view can be enlarged to full screen, which hides the other views. The 3D view shows only the segmentation without the image context around it.

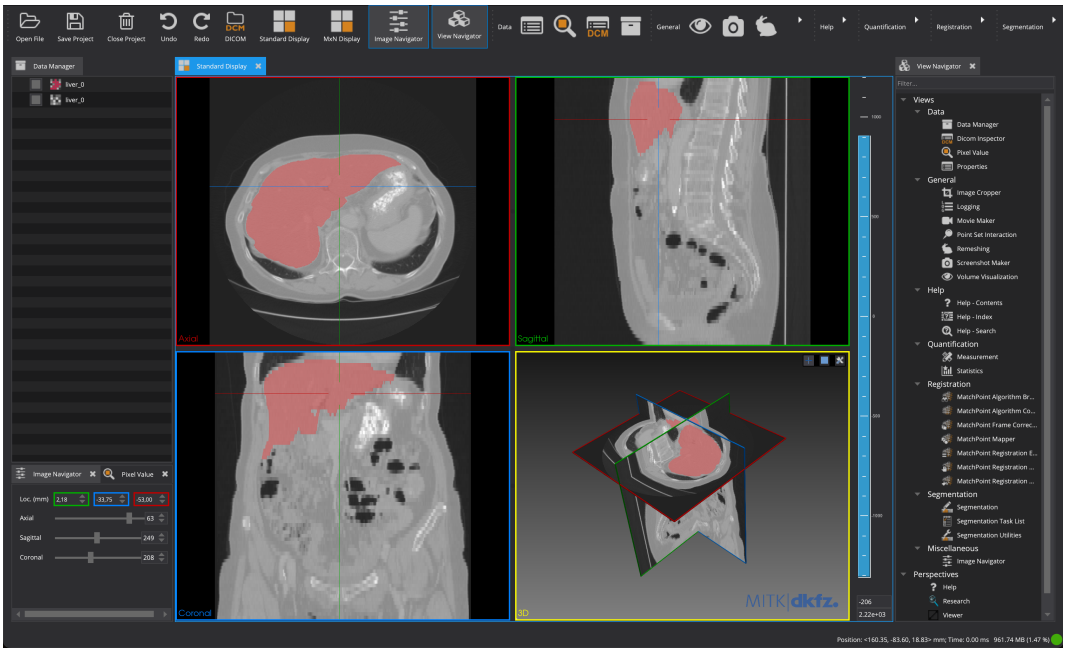


Figure A.2: MITK Workbench [Wol+04] user interface showing an abdomen CT image with liver segmentation. The default view gives an equal amount of space to the 3 2D views and the 3D view. Any view can be enlarged to full screen, which hides the other views. The 3D view does not render an actual volume, instead the 2D slices are intersected in 3D and move through the volume according to which slices are shown in the 2D views.

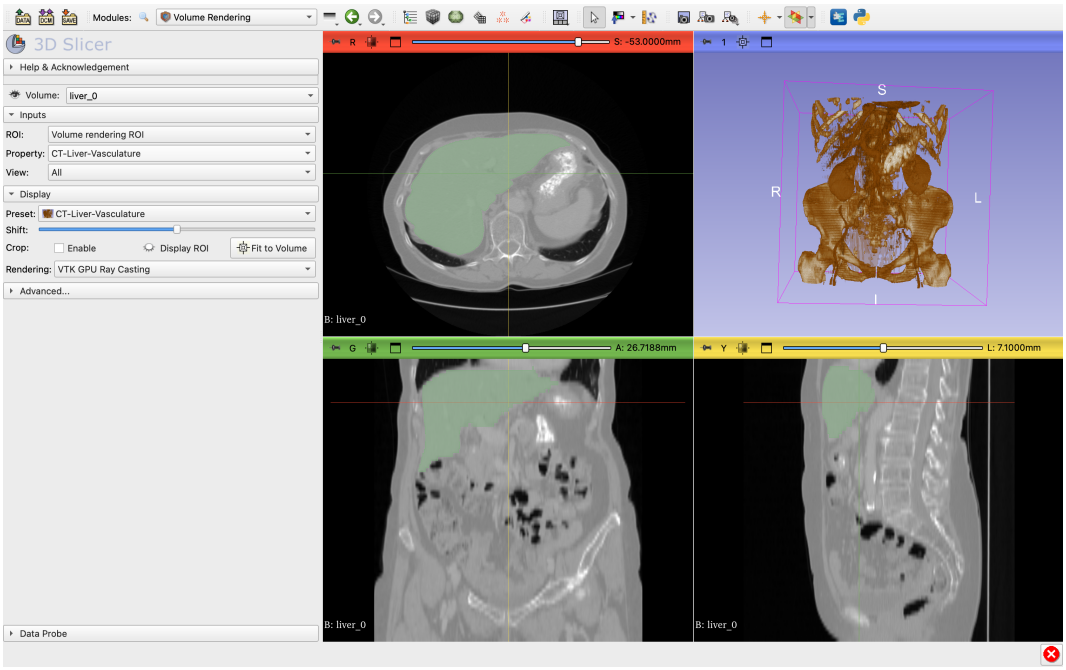


Figure A.3: 3D-Slicer [Fed+12] user interface showing an abdomen CT image with liver segmentation. The default view gives an equal amount of space to the 3 2D views and the 3D view. Any view can be enlarged to full screen, which hides the other views. The 3D view renders the image volume. For this, many rendering presets are available but the user can fully customize the rendering logic. In this screenshot the *CT-Liver-Vasculature* preset is shown.

B Bounding Box Similarity Search

In addition to bounding box interpolation, we also tried bounding box similarity search based on a variety of image similarity metrics as a prompt engineering strategy. However, the results are generally worse than both bilinear and nearest neighbor interpolation, likely because image similarity metrics are usually developed for natural images and seem to struggle with medical images. Nevertheless, we report our findings in this appendix chapter.

B.1 Similarity Search Implementation

To expand upon both the nearest neighbor and bilinear bounding box interpolation discussed in Chapter 4, we apply an additional similarity search. For this we use the interpolated bounding box as a starting point and compare its image content as well as the image content of bounding box variations around it to the image content of the closest user-provided bounding box from a different slice. We use one of the image similarity metrics presented in B.2. As input for the similarity search, we use the preprocessed images which are also used as input for MedSAM (see Section 5.4).

To facilitate an efficient similarity search which allows both for translation and shape transformation of the bounding box, we split up the search in to two steps. First, the bounding box size and shape is kept constant, but translation variations are tested. In particular, we try all variations for translation along the x- and y-axes with possible translation distances between -10 pixels and 10 pixels (in steps of 2). In a second step, we start from the best translated bounding box and try size transformations, keeping the bounding box center constant. Here, we try all variations for adjusting the height and width of the bounding box by -16 pixels to 16 pixels (in steps of 4). Because all similarity metrics expect the compared images to have the same size, we resample the content of the transformed bounding box to the size of the original bounding box before calculating the similarity score. The bounding box with the best similarity score in the second step is used for prompting MedSAM on that particular slice.

While this two step process is less exhaustive than optimizing over all combinations of translations and transformation, it is necessary in order to keep the

run time of the search reasonable. With the two step process, we still test 202 different bounding boxes, compared to 9801 boxes which would result from testing all combinations.

Before we move on to the results of our experiments, we now present the similarity metrics we experimented with.

B.2 Image Similarity Metrics

In this section we discuss the different image similarity metrics we experimented with and highlight some metric specific implementation details.

B.2.1 Weighted Pixel Similarity

The simplest image similarity metric is pixel similarity (PS). Given two gray-scale images A and B where $a_{i,j}$ and $b_{i,j}$ are the pixels of the images, we calculate PS as follows:

$$PS = -1 \cdot \sum_i \sum_j |a_{i,j} - b_{i,j}| \tag{B.1}$$

However, as we are more interested in the similarity of the region which is part of the segmentation in the reference image, we actually use weighted pixel similarity (WPS). With S being the segmentation in the reference image and $s = i, j \in \{0, 1\}$ with 1 representing $s_{i,j}$ being part of the segmentation and 0 representing not part of the segmentation, we calculate WPS as follows:

$$WPS = -1 \cdot \sum_i \sum_j (s_{i,j} + 0.5) \cdot |a_{i,j} - b_{i,j}| \tag{B.2}$$

This means that pixels which are part of the segmentation in the reference image get a weight of 1.5 and all other pixels get a weight of 0.5.

B.2.2 Structural Similarity

A more advanced image similarity metric is structural similarity (SSIM) [Wan+04]. This metric combines measures for luminance, contrast, and structure of the image. However, these characteristics are not calculated globally for the whole image. Instead, they are calculated locally, on a per pixel basis, taking a 2D sliding window around each pixel into account.

The luminance comparison measures the closeness of the image brightness. The average brightness of each image is calculated and compared. The idea is that similar images should have similar average brightness.

The contrast comparison involves comparing the contrast, or the spread of pixel values, in the images. This is done by calculating and comparing the standard deviation of the pixel values in each image. Images with similar contrast will have similar standard deviations.

The structure comparison is about comparing the patterns or textures in the images. This is done by first normalizing the pixel values (subtracting the average brightness and dividing by the standard deviation), which leaves you with values that represent the structure. The correlation between these normalized values in the two images is then calculated.

Finally, the three measures are pixel-wise multiplied and then averaged across the whole image to calculate the SSIM value.

B.2.3 Multi-scale Structural Similarity

An expansion upon SSIM is multi-scale structural similarity (MS-SSIM) [WSB03]. Here, the SSIM metric is calculated for the original image and down-sampled, lower resolution versions of the image. The MS-SSIM is then a weighted combination of these multiple SSIM values. The result is a metric which takes both large and small structures of the image into account.

B.2.4 Visual Information Fidelity

An image similarity metric which more closely tries to model the human visual system is visual information fidelity (VIF) [SB06]. VIF is targeted at natural images and relies on the fact that these natural images possess statistical properties which are altered when the image undergoes distortion. It uses these statistics to model the information content of images.

For this, VIF models the information present in the reference image considering the human visual system and then quantifies how much of this information is present in the test image.

B.2.5 Learned Perceptual Image Patch Similarity

While all the metrics presented above are manually defined mathematical models, learned perceptual image patch similarity (LPIPS) [Zha+18] uses deep features from

deep neural networks to represent images. LPIPS builds on networks learning features which correlate with perceptual judgements when they are trained for visual tasks. In particular, features are extracted for the two images which are compared from the different layers of the network. These features are then compared and a weighted combination of the feature similarity is the resulting LPIPS score.

While LPIPS works well with various different network architectures, we use AlexNet [KSH17] as the backbone.

B.3 Results

For evaluating the similarity search, we use the same experimental setting presented in Section 5.4. The only difference is, that both nearest neighbor and bilinear interpolation are enhanced with the similarity search we outline in Appendix B.1. We test separately for the different similarity metrics presented in Appendix B.2. In Table B.1, we present the results of plain bounding box similarity search (compared to nearest neighbor interpolation), and in Table B.2, we present the results of combining bilinear bounding box interpolation with similarity search (compared to plain bilinear interpolation).

The most significant finding is that the plain similarity search results in lower segmentation quality than nearest neighbor interpolation for all of the similarity metrics. The same is true for the combination of bilinear interpolation with similarity search compared to plain bilinear interpolation.

Among the similarity metric, SSIM results in the highest segmentation quality, closely followed by MS-SSIM. Both SSIM-based metrics come close to the DSC of the baselines. LPIPS and VIF follow with only a slight decrease in segmentation quality. WPS, however, seems to be not well suited at all, resulting on average in a DSC 0.1 to 0.15 lower than the baselines.

Overall, with the approaches we tried, we do not find an improvement due to the added similarity search. Additionally, we note that the similarity search adds to the run-time of the prompt engineering approaches significantly. Thus, we currently do not recommend using similarity search. Nevertheless, we believe there might be potential in the approach with more purpose-designed similarity metrics and possibly a more flexible search algorithm.

Table B.1: Average DSC of nearest neighbor (NN) box interpolation and similarity search with different similarity metrics used with MedSAM compared to applying MedSAM slice-by-slice, all applied to the Medical Segmentation Decathlon datasets. User input was simulated on a varying percentage of slices.

	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.	**Average**
100% user prompts	0.830	0.846	0.955	0.822	0.903	0.792	0.822	0.782	0.949	0.783	**0.848**
NN Interpolation											
33% user prompts	0.828	0.843	0.953	0.783	0.874	0.769	0.800	0.745	0.929	0.724	**0.825**
20% user prompts	0.824	0.839	0.949	0.728	0.840	0.759	0.771	0.727	0.900	0.701	**0.804**
14% user prompts	0.817	0.833	0.945	0.676	0.814	0.750	0.745	0.716	0.871	0.692	**0.786**
11% user prompts	0.812	0.826	0.941	0.618	0.793	0.743	0.719	0.711	0.854	0.689	**0.770**
9% user prompts	0.804	0.822	0.936	0.586	0.785	0.741	0.698	0.706	0.830	0.688	**0.759**
NN Interpolation + WPS											
33% user prompts	0.743	0.712	0.939	0.508	0.798	0.662	0.702	0.641	0.887	0.623	**0.721**
20% user prompts	0.717	0.656	0.932	0.422	0.764	0.622	0.660	0.613	0.848	0.595	**0.683**
14% user prompts	0.701	0.634	0.927	0.385	0.738	0.603	0.629	0.598	0.808	0.580	**0.660**
11% user prompts	0.688	0.614	0.923	0.339	0.718	0.589	0.609	0.593	0.793	0.577	**0.644**
9% user prompts	0.679	0.594	0.918	0.305	0.708	0.580	0.584	0.588	0.762	0.576	**0.629**
NN Interpolation + LPIPS [Zha+18]											
33% user prompts	0.826	0.840	0.953	0.776	0.858	0.758	0.788	0.721	0.926	0.702	**0.815**
20% user prompts	0.819	0.830	0.949	0.736	0.814	0.736	0.754	0.696	0.906	0.671	**0.791**
14% user prompts	0.809	0.819	0.945	0.695	0.789	0.725	0.726	0.684	0.884	0.664	**0.774**
11% user prompts	0.801	0.807	0.941	0.642	0.769	0.716	0.699	0.679	0.871	0.662	**0.759**
9% user prompts	0.791	0.800	0.937	0.625	0.764	0.711	0.657	0.674	0.855	0.662	**0.747**
NN Interpolation + SSIM [Wan+04]											
33% user prompts	0.828	0.843	0.953	0.781	0.872	0.764	0.799	0.740	0.930	0.722	**0.823**
20% user prompts	0.824	0.838	0.949	0.734	0.830	0.750	0.765	0.717	0.907	0.694	**0.801**
14% user prompts	0.816	0.829	0.945	0.691	0.808	0.740	0.736	0.705	0.884	0.682	**0.784**
11% user prompts	0.811	0.819	0.941	0.628	0.777	0.729	0.706	0.698	0.872	0.678	**0.766**
9% user prompts	0.803	0.812	0.936	0.612	0.770	0.724	0.684	0.692	0.849	0.677	**0.756**
NN Interpolation + MS-SSIM [WSB03]											
33% user prompts	0.827	0.843	0.953	0.774	0.872	0.763	0.798	0.735	0.930	0.714	**0.821**
20% user prompts	0.823	0.837	0.950	0.725	0.825	0.747	0.760	0.708	0.909	0.678	**0.796**
14% user prompts	0.815	0.825	0.945	0.680	0.804	0.736	0.728	0.695	0.888	0.667	**0.778**
11% user prompts	0.809	0.811	0.942	0.619	0.776	0.727	0.697	0.689	0.877	0.663	**0.761**
9% user prompts	0.800	0.803	0.936	0.609	0.769	0.721	0.674	0.682	0.856	0.662	**0.751**
NN Interpolation + VIF [SB06]											
33% user prompts	0.827	0.843	0.952	0.681	0.870	0.766	0.797	0.730	0.927	0.713	**0.811**
20% user prompts	0.823	0.837	0.947	0.602	0.822	0.745	0.759	0.702	0.903	0.679	**0.782**
14% user prompts	0.815	0.824	0.942	0.550	0.792	0.734	0.729	0.690	0.879	0.668	**0.762**
11% user prompts	0.809	0.807	0.937	0.468	0.767	0.723	0.699	0.682	0.866	0.664	**0.742**
9% user prompts	0.800	0.801	0.932	0.448	0.761	0.723	0.675	0.677	0.841	0.663	**0.732**

Table B.2: Average DSC of bilinear (BL) bounding box interpolation and similarity search with different similarity metrics used with MedSAM compared to applying MedSAM slice-by-slice, all applied to the Medical Segmentation Decathlon datasets. User input was simulated on a varying percentage of slices.

	Brain T.	Heart	Liver	Hippoca.	Prostate	Lung T.	Pancreas	Hepatic T.	Spleen	Colon T.	**Average**
100% user prompts	0.830	0.846	0.955	0.822	0.903	0.792	0.822	0.782	0.949	0.783	**0.848**
BL Interpolation											
33% user prompts	0.829	0.842	0.954	0.808	0.870	0.780	0.804	0.751	0.938	0.729	**0.830**
20% user prompts	0.828	0.842	0.953	0.790	0.836	0.776	0.774	0.732	0.917	0.707	**0.815**
14% user prompts	0.827	0.841	0.951	0.770	0.783	0.767	0.741	0.716	0.886	0.684	**0.797**
11% user prompts	0.823	0.840	0.948	0.749	0.747	0.769	0.708	0.704	0.860	0.678	**0.783**
9% user prompts	0.819	0.838	0.944	0.716	0.738	0.763	0.670	0.696	0.827	0.677	**0.769**
BL Interpolation + WPS											
33% user prompts	0.727	0.698	0.935	0.501	0.791	0.655	0.706	0.648	0.883	0.625	**0.717**
20% user prompts	0.701	0.649	0.929	0.407	0.743	0.606	0.663	0.618	0.847	0.596	**0.676**
14% user prompts	0.686	0.615	0.926	0.359	0.704	0.581	0.634	0.600	0.809	0.569	**0.648**
11% user prompts	0.675	0.603	0.921	0.331	0.674	0.574	0.602	0.587	0.771	0.561	**0.630**
9% user prompts	0.667	0.573	0.917	0.320	0.667	0.559	0.571	0.579	0.732	0.560	**0.615**
BL Interpolation + LPIPS [Zha+18]											
33% user prompts	0.824	0.835	0.953	0.777	0.851	0.761	0.786	0.716	0.932	0.702	**0.814**
20% user prompts	0.818	0.826	0.951	0.726	0.815	0.742	0.754	0.693	0.907	0.670	**0.790**
14% user prompts	0.812	0.819	0.948	0.690	0.766	0.725	0.718	0.676	0.882	0.644	**0.768**
11% user prompts	0.805	0.813	0.944	0.661	0.739	0.718	0.689	0.664	0.862	0.637	**0.753**
9% user prompts	0.798	0.808	0.940	0.610	0.732	0.709	0.657	0.656	0.834	0.637	**0.738**
BL Interpolation + SSIM [Wan+04]											
33% user prompts	0.827	0.841	0.954	0.790	0.869	0.770	0.794	0.732	0.934	0.717	**0.823**
20% user prompts	0.824	0.835	0.951	0.735	0.834	0.756	0.757	0.709	0.912	0.693	**0.801**
14% user prompts	0.821	0.831	0.949	0.690	0.775	0.733	0.723	0.689	0.877	0.670	**0.776**
11% user prompts	0.816	0.825	0.945	0.651	0.736	0.728	0.690	0.678	0.855	0.664	**0.759**
9% user prompts	0.810	0.818	0.940	0.593	0.725	0.719	0.654	0.670	0.824	0.663	**0.742**
BL Interpolation + MS-SSIM [WSB03]											
33% user prompts	0.826	0.840	0.953	0.779	0.866	0.770	0.793	0.727	0.934	0.710	**0.820**
20% user prompts	0.823	0.835	0.951	0.722	0.829	0.753	0.754	0.700	0.911	0.685	**0.796**
14% user prompts	0.819	0.823	0.948	0.679	0.777	0.731	0.717	0.680	0.879	0.662	**0.772**
11% user prompts	0.813	0.814	0.944	0.643	0.733	0.725	0.684	0.667	0.858	0.656	**0.754**
9% user prompts	0.807	0.804	0.940	0.587	0.723	0.713	0.648	0.660	0.827	0.655	**0.736**
BL Interpolation + VIF [SB06]											
33% user prompts	0.826	0.839	0.953	0.695	0.862	0.767	0.790	0.723	0.932	0.707	**0.809**
20% user prompts	0.823	0.833	0.950	0.594	0.826	0.754	0.750	0.694	0.909	0.678	**0.781**
14% user prompts	0.819	0.821	0.948	0.529	0.768	0.734	0.712	0.674	0.874	0.653	**0.753**
11% user prompts	0.813	0.814	0.944	0.508	0.729	0.729	0.675	0.661	0.852	0.649	**0.737**
9% user prompts	0.807	0.801	0.937	0.449	0.718	0.716	0.636	0.653	0.819	0.648	**0.719**