applied sciences

MDPI

Article

Adapting the Segment Anything Model for Volumetric X-ray Data-Sets of Arbitrary Sizes

Roland Gruber [1,2,*], Steffen Rüger [1] and Thomas Wittenberg [1,2]

[1] Fraunhofer IIS, Fraunhofer Institute for Integrated Circuits IIS, Division Development Center X-ray Technology, 90768 Fürth, Germany
[2] Chair for Visual Computing, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany
* Correspondence: roland.gruber@iis.fraunhofer.de

Abstract: We propose a new approach for volumetric instance segmentation in X-ray Computed Tomography (CT) data for Non-Destructive Testing (NDT) by combining the Segment Anything Model (SAM) with tile-based Flood Filling Networks (FFN). Our work evaluates the performance of SAM on volumetric NDT data-sets and demonstrates its effectiveness to segment instances in challenging imaging scenarios. We implemented and evaluated techniques to extend the image-based SAM algorithm for the use with volumetric data-sets, enabling the segmentation of three-dimensional objects using FFN's spatial adaptability. The tile-based approach for SAM leverages FFN's capabilities to segment objects of any size. We also explore the use of dense prompts to guide SAM in combining segmented tiles for improved segmentation accuracy. Our research indicates the potential of combining SAM with FFN for volumetric instance segmentation tasks, particularly in NDT scenarios and segmenting large entities and objects. While acknowledging remaining limitations, our study provides insights and establishes a foundation for advancements in instance segmentation in NDT scenarios.

Keywords: instance segmentation; Segment Anything Model; computed tomography; non-destructive testing; neural networks; machine learning

check for updates

Citation: Gruber, R.; Rüger, S.; Wittenberg, T. Adapting the Segment Anything Model for Volumetric X-ray Data-Sets of Arbitrary Sizes. *Appl. Sci.* **2024**, *14*, 3391. https://doi.org/10.3390/app14083391

Academic Editors: Zahid Mehmood Jehangiri, Mohsin Shahzad and Uzair Khan

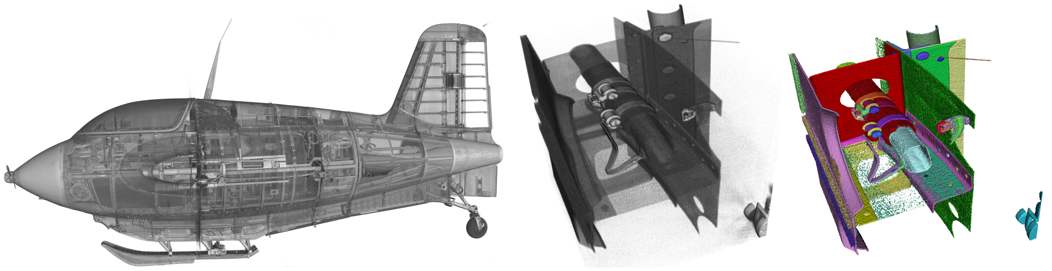
Received: 6 March 2024
Revised: 5 April 2024
Accepted: 8 April 2024
Published: 17 April 2024

1. Introduction

In the field of Non-Destructive Testing (NDT) of large-scale components and assemblies, cars [1], shipping containers [2,3], or even airplanes [4,5] are often captured using large-scale 3D X-ray computed tomography (CT) and are subsequently subjected to automated analysis and evaluation. In this context, an important step of the analysis process consists of instance segmentation, where an attempt is made to assign a unique semantic identifier or label to each entity in a data-set. For example, all voxels belonging to a specific screw are hereby assigned the same unique identifier, while voxels belonging to another component are assigned a different unique identifier.

The complexity of computing accurate instance segmentation varies significantly across different problem domains and data-sets. While simple threshold- or flood-filling-based methods from classical image processing suffice for data-sets from many fields, it remains uncertain as to whether an adequate solution for segmentation is feasible for others. Recent efforts, such as those in a challenge [6], tested multiple techniques to segment the data-set of a Me 163 [7], a historic German airplane with a rocket engine during the Second World War, with mixed success. This contribution aims to evaluate the suitability of an approach based on the currently highly appraised Segment Anything Model (SAM) [8], a foundational model for instance segmentation of such complex data-sets.

The task of instance segmentation shown in Figure 1 exemplifies this attempt using the XXL-CT data-set of the historic airplane. It begins with acquisition of data from the specimen, in this case the airplane, and proceeds with the reconstruction of a volumetric voxel data-set (Figure 1a). Figure 1b,c shows a sub-volume of size 512 × 512 × 512 voxels of the reconstruction and the instance segmentation. In Figure 1c, each semantic entity within the sub-volume is assigned a unique identifier. The classes of these entities (primarily screws and metal plates) are not considered, as the classification of the entities is not performed and is the focus of future work.



(**a**) Reconstruction (**b**) Input sub-volume (**c**) Reference sub-volume

Figure 1. Rendered example of instance segmentation (**c**), of a sub-volume of size 512 × 512 × 512 voxels (**b**), from the XXL-CT Me 163 data-set with a data resolution of 10,000 × 10,000 × 8000 voxels (**a**). The objective of instance segmentation is to generate a plausible segmentation of individual objects or instances, as depicted in (**c**), from an input sub-volume such as that shown in (**b**), applicable to data-sets of any size, akin to the one demonstrated in (**a**).

Instance segmentation is essential for automated image processing and data exploration in NDT and medical [9] applications. By segmenting a large-scale volumetric image data-set into its semantic instances, it becomes easier to extract valuable information and to analyse complex component geometries. This is particularly important in cases where the data-set contains various acquisition and reconstruction that can make interpretation difficult for both experts and non-experts.

Instance segmentation is a critical task in computer vision, leading to the proposal and development of numerous methods that leverage both classical image processing and neural networks. These approaches, however, are not without their limitations. Some methods necessitate manual intervention and corrections [10,11]; others are specifically tailored to predefined component classes [12]. Challenges associated with data quality, particularly in data-sets with a high incidence of artefacts, can significantly hinder the effectiveness of segmentation algorithms.

1.1. Segment Anything Model

The Segment Anything Model (SAM) [8] is an instance segmentation model based on the vision transformer architecture [13]. It is an advanced model for segmenting arbitrary entities out of photographs. It stands out primarily for its high quality, robustness, and minimal required user input. One of its notable features is the ability to be queried using a variety of prompts, allowing it to segment a RGB input image with a spatial resolution up to 1024 × 1024 pixels into multiple segments in one inference call. SAM supports prompts in various forms such as seed points (point prompts), bounding boxes, brush masks (dense prompts), and text prompts.

Furthermore, SAM allows the generation of multiple output masks for each input prompt, hence enabling image segmentations at varying hierarchical levels of granularity. Another advancement presented by the SAM is the extensive training data-set SA-1B, which has been iteratively collected and refined through prior versions of SAM during its own training process.

A multitude of studies and publications are currently emerging, which aim to apply SAM as a foundation model across a diverse range of fields, testing its segmentation quality. The application domains are varied. For instance, Li et al. [14] assess SAM for GeoAI vision tasks particularly in permafrost mapping. Alternatively, Noe et al. [15] utilise SAM to introduce a new approach for tracking black cattle on photographs. Another application within the domain of so-called "Precision Agriculture" is investigated by Carraro et al. [16], where mapping of crop features by automated mechanisms is conducted. In the field of NDT, the work by Weinberger et al. [17] examines how SAM can distinguish various segments in CT volume slices through unsupervised learning techniques. However, the direct application of SAM for instance segmentation is not the only focus of resent research. For example, Xu et al. [18] explored how an expanded data-set computed via SAM can be used to train an object detection network to improve license plate detection under severe weather conditions. Similarly, Liu [19] employed SAM to optimise road sign detection by using the model for background pixel exclusion in the data-set. In all these named studies, SAM exhibits a performance ranging from high quality to mixed results, which are strongly influenced by the data-set and specific problem domain under investigation.

1.2. Combination with Tile-Based FFN

This work aims to evaluate the applicability of SAM for segmenting volumetric NDT data-sets and to examine its potential enhancement through the integration of Flood Filling Networks (FFN), initially proposed by Januszewski et al. [20]. FFNs are instance segmentation methods originally based on convolutional networks [21,22], which are able to segment arbitrarily large data-sets based on tiles. Originally, FFN was developed for the segmentation of organic objects but in the past, was extended to other applications, including the delineation of large-scale XXL-CT data [4].

The FFN approach maintains the current state of segmentation within an accumulator volume, which is sized to match the dimensions of the input volume. During each segmentation step, a sub-volume or tile of the input volume and the corresponding partially computed tile of the accumulator is passed to the model (in our case, a volumetric variant of SAM). The segmentation proposal of the tile is then updated and written back to the corresponding tile position within the accumulator.

Candidates for neighbouring tile positions with significant overlap, which could extend the current segment, are determined using the updated accumulator state and added to a queue of tiles pending processing. In the subsequent iteration, the next unprocessed tile is removed from the front of the queue for processing. Starting from a seed point, the FFN then processes all of the tiles that potentially belong to the current segment. The processing of the current segment is completed when the queue of potentially belonging tiles is depleted. The algorithm then proceeds with the next segment starting from another seed point.

The seed points of the segments can be manually specified or computed automatically by a reasonable algorithm.

1.3. Contributions

In this work, we propose a novel approach for volumetric instance segmentation in NDT by combining SAM with FFN. Our contributions include the following:

1. Evaluation of SAM on NDT data-sets

We assess the performance of SAM on data-sets from the field of non-destructive testing and demonstrate its effectiveness in accurately segmenting instances in challenging CT imaging scenarios.

2. Implementation and evaluation of various methods to combine image-based SAM for the application with volumetric data-sets

We implement and evaluate different techniques to integrate and fuse the output of the image-based SAM approach for the application of volumetric data-sets, hence enabling the segmentation of three-dimensional objects using FFN's spatially adaptive capabilities.

3. Extending SAM for objects of arbitrary size through tile-based approaches

We propose a tile-based approach that leverages FFN's capabilities to segment objects of arbitrary size. By initially dividing the input volumes into tiles and then applying SAM on each tile individually, we achieve accurate and efficient segmentation results for objects of any size.

4. Utilizing dense prompts for SAM to combine tiles in an accumulator
To further improve the accuracy of the proposed tiled-based approach of SAM, we use dense prompts to guide SAM in combining the segmented tiles into a cohesive instance segmentation result. By leveraging the accumulated information from neighbouring tiles, we try to achieve more robust and accurate instance segmentation results.

2. Materials and Methods

This section presents the methodology and the experimental setup used, including the introduction of the data-sets (Section 2.1) used for the evaluation of the proposed methods. Furthermore, we describe a technique to improve the image segmentation performance of SAM with respect to the Me 163 airplane XXL-CT data-set by fine-tuning it specifically for this task (Section 2.2). Additionally, we detail our inference workflow in Section 2.3, which adapts the top-performing SAM model for volumetric data-sets. This process includes tile-based segmentation, accumulator-based dense prompts, and post-processing. The workflow aims to integrate the best model into a cohesive volumetric inference approach.

2.1. Data-Sets and Data Processing

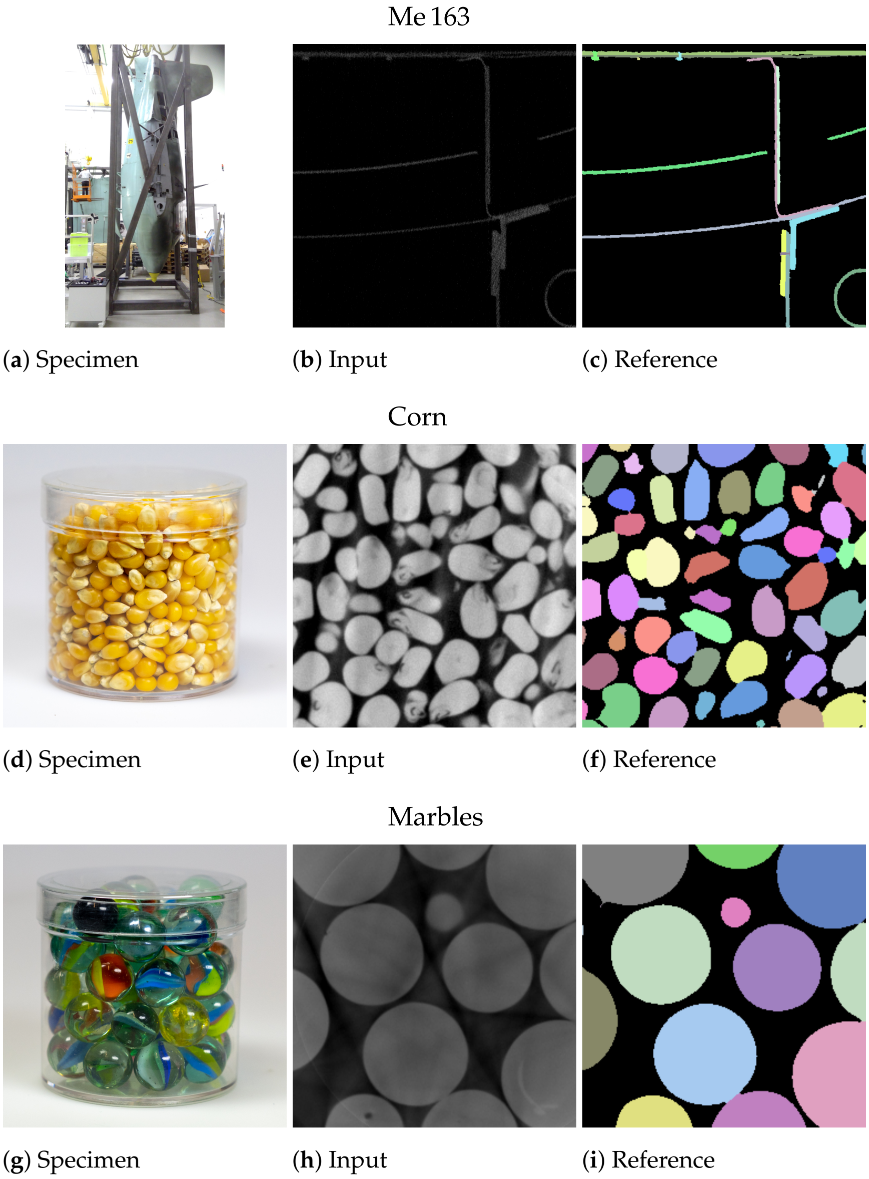
To demonstrate, exemplify, and evaluate our achievements, we make use of three distinct data-sets. A specific sub-volume of the Me 163 data-set of a Second World War fighter airplane [7] as well as two bulk material data-sets depicting entities of glass marbles and corn kernels [4]. Figure 2 shows a photograph of each specimen, along with one typical slice from the reconstructed volume and a corresponding reference segmentation.

The Me 163 data-set utilized in this study consists of a volumetric subset and a manually obtained reference segmentation XXL-CT data-set from a historic airplane [5], which itself was extracted from an XXL-CT reconstruction. The reference segmentation sub-volumes of the Me 163 data-set were manually annotated and underwent morphological post-processing to clean up the edges. The acquisition process involved addressing challenging aspects such as noisy data, low contrast, and limited spatial resolution. A detailed description of the data-set creation, including the annotation and post-processing process, can be found in [7].

The data-set consists of eight sets of sub-volume pairs, each sub-volume having the spatial dimensions of 512 × 512 × 512 voxels. For training, six sub-volume pairs of the data-set are used, while one sub-volume pair is used for validation and one for testing, respectively. Each sub-volume pair consists of a reconstructed sub-volume (see Figure 2b) and its corresponding reference segmentation sub-volume (see Figure 2c).

The reconstruction sub-volume is a small volumetric region that is extracted from the reconstructed Me 163 XXL-CT data. To ensure compatibility with SAM, both the reconstruction or input sub-volumes and the corresponding reference segmentation sub-volumes are extended with zero-padded 512 voxels in every direction. This results in an embedded version of the sub-volumes with working dimensions of 1536 × 1536 × 1536 voxels. This arrangement allows for the extraction of a slice, centred on any arbitrary voxel within the original sub-volume, with the resolution of 1024 × 1024 × 1 voxels, matching the native input dimensions required by SAM.

The first row of Figure 3 illustrates the described enframing process for the Me 163 data-set. The green rectangles in the first two columns indicate the unembedded region with 512 × 512 × 512 voxels and their manually annotated references. Due to the fact that the input sub-volumes of this data-set are located directly at the edge of the XXL-CT volume, it was not possible to fill the border of the sub-volumes with actual reconstruction values. Instead, we decided to use a border with a constant value of zero in all directions. The last two columns of Figure 3 display the prepared input and reference slices used in the subsequent processing.



Me 163

(**a**) Specimen (**b**) Input (**c**) Reference

Corn

(**d**) Specimen (**e**) Input (**f**) Reference

Marbles

(**g**) Specimen (**h**) Input (**i**) Reference

Figure 2. Photographs, exemplary CT slices, and reference segmentation of the Me 163 (**a**–**c**), corn (**d**–**f**), and marbles (**g**–**i**) data-sets, respectively.

The other two data-sets, which consist of CT scans of jars filled with marbles and corn, also contain two sub-volumes each: one for the input CT reconstruction sub-volume and one for its reference segmentation sub-volume. The segmentation process to yield the reference volumes of the bulk material data-set involved semi-automatic segmentation using threshold binarization with a threshold obtained from Otsu's method [23], followed by a distance transform, watershed transform, and label-wise morphological closing, as described in more detail in [4]. As this traditional computer-vision process resulted in some erroneous segmentations in the contact regions between the jar and the bulk material, we only used a correctly segmented sub-volume in the centre of the jar, having a spatial dimension of 256 × 256 × 256 voxels (denoted by the green rectangle in Figure 3). Also, the sub-volumes of the bulk material were enframed by a border of 512 voxels thickness with a constant value of zero.

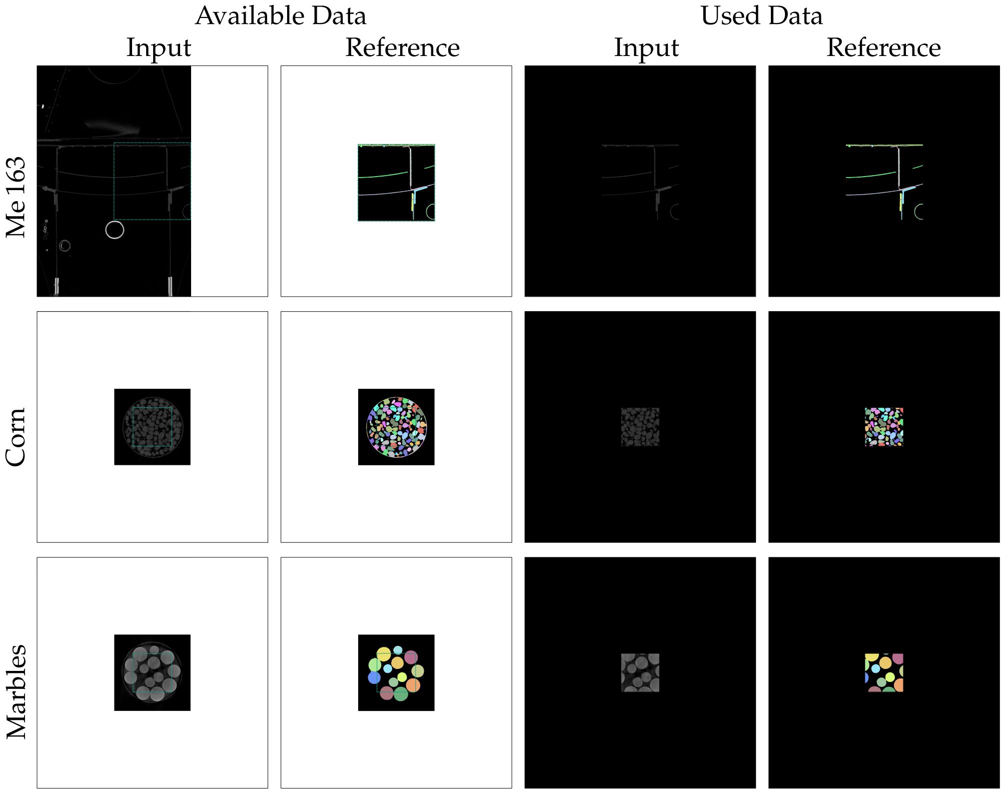


Figure 3. Zero-padding preparation steps were performed on the input and reference slices of the different data-sets to create slices of size 1024 × 1024 pixels centred around each possible seed point. The white border regions in the available input and reference slices were filled with constant values of zero.

2.2. Fine-Tuning on the NDT Data-Set

The SA-1B training data-set published by the authors of the SAM [8] contains predominantly coloured natural photographs, such as street scenes or still life compositions of semantically well-known objects from daily life. In contrast, volumetric data-sets obtained from the NDT field and particularly the slices extracted from the volumes are frequently of a rather abstract nature and do not depict recognizable objects. Hence, these NDT images deviate from the familiar photographic data-set used by SAM and this deviation poses several challenges in achieving sufficient segmentation quality (see Section 3.1). This, within the CT imaging domain, means that even familiar objects can be difficult to recognize for non-experts, as they exhibit unusual structures or non-orthogonal sections due to the specimen's imaging geometry; or, they may contain strong imaging and reconstruction artefacts.

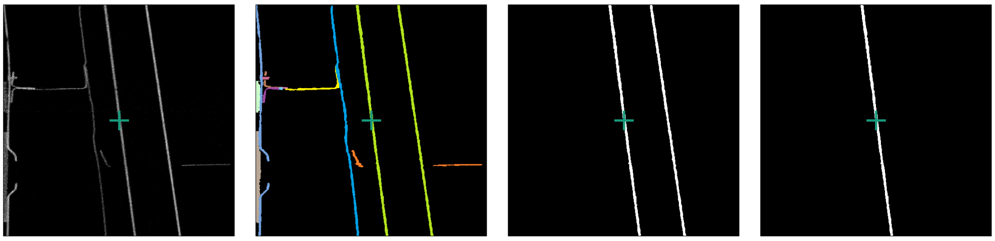
Ma et al. [24] showcased a potential improvement in segmentation quality by fine-tuning SAM on the problem domain, which inspired us to adopt a similar fine-tuning approach.

In this study, we opted to perform fine-tuning on a certain part of the SAM, specifically the Mask Decoder. For this purpose, we utilized, extracted, and pre-processed slices from the Me 163 training data-set. Our approach adhered to the guidelines outlined in [24], which have previously been employed for fine-tuning on medical volume CT data-sets.

The Me 163 data-set was chosen due to its distinct level of complexity, setting it apart from the bulk material data-sets also being investigated. In contrast, the marble and corn data-sets can be segmented relatively easily using conventional image processing techniques.

For the fine-tuning process, we randomly selected voxel positions from the Me 163 training data-set. If the chosen voxel was a foreground voxel belonging to a known labeled entity, three orthogonal slices centred around its position were extracted. These slices were used as training examples, with the data range of the input slice normalised to $[0.0, 255.0]$. For the target slice, all voxels of the entities belonging to the centre voxel were one-hot encoded.

The original SAM operates on images, while our attempted input is a single slice from a volumetric data-set. To ensure that a three-dimensional connected object was represented by a single segment in the two-dimensional slices, a connected component analysis (CCA) was performed on the one-hot encoded target slice. This issue is depicted in Figure 4. Specifically, in the one-hot encoded a foreground target after the CCA (Figure 4d), where only the central component is visible, as we isolated the segment connected to the centre of the target slice, marked by a green cross. This central segment was then selected as the training target. The surrounding image does not provide sufficient information to distinguish if neighbouring non-touching segments belong to the same segment. Thus, we performed a CCA and treated the parts of segments not connected in the current slice as separate segments.



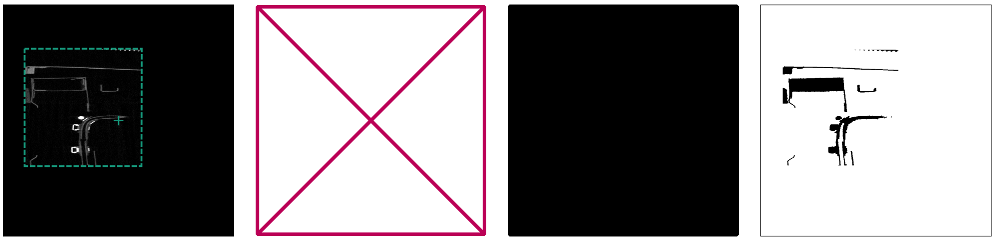
(**a**) Input (**b**) Reference (**c**) One-hot encoded foreground target (**d**) Connected component target

Figure 4. Processing of an example foreground slice used for fine-tuning SAM. Consisting of reconstruction slice (**a**), reference slice (**b**), one-hot encoded slice (**c**), and connected component training target slice (**d**). The green cross marks the centre of the slice.

If the voxel at the centre of a slice represented the background, we generated three orthogonal background examples, each containing a normalised input slice and a target slice. We evaluated three versions: *ForegroundOnly*, which included only foreground input slices; *ConstantValueBackground*, where we provided both background and foreground input and target slices for training but expected SAM to produce a completely empty response for background slices; and *ConnectedComponentBackground*, where we identified all background voxels connected to the centre voxel of the slice as the target segment. This was achieved through CCA on the data-set's background, formed by also enframing the reference segmentation with a zero-padded boundary. Consequently, the network was prompted to consider all voxels connected to the air space in the slice's centre as part of that segment. Figure 5 provides an illustrative example of the different target versions.

Due to the significantly lower count of foreground voxels (0.1–9.4%) compared to background voxels in the Me 163 data-set, we included all foreground examples while randomly selecting a subset of background examples of the same size. This approach ensured a balanced representation of both classes. To prevent batches from containing closely located examples, the selected examples were shuffled and grouped into batches, with each batch containing 16 foreground examples and 16 background examples. Additionally, to further diversify the examples within each batch, we employed a relatively large stride during the example extraction process. This ensured that the examples originated from different sub-volumes within the data-set. In each iteration over the data-sets, a new random initial position offset was chosen, employing a non-repetitive selection process to extract different examples.

We chose a single point prompt in the exact centre of each slice as the input for SAM during training. This choice aligns with the input for our validation application as well as the tile-based SAM integration for volume data-sets (see Section 2.3).



(**a**) Input (**b**) No background (**c**) Constant value background (**d**) Connected component background

Figure 5. Processing of an example background slice used for fine tuning SAM. The green cross marks the centre of the slice, which is located in the background of the reconstruction. The green border around the reconstruction slice in (**a**) depicts the original volume size, which was then enframed with a constant value border. The other sub-figures show the tested possibilities for target slices for the fine-tuning: *ForegroundOnly* (**b**), *ConstantValueBackground* (**c**), and *ConnectedComponentBackground* (**d**).

The batch size was set to 64. We initiated the training with a learning rate of 8×10^{-4}, which was linearly increased over the first 250 iterations. For optimization, we utilized the AdamW optimizer [25] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, along with a weight decay of 0.1. Our loss function consisted of a combination of dice loss (sigmoid = true, squared-pred = true, and mean reduction) and binary cross-entropy loss (mean reduction). We let the training run until overfitting for 10 to 25 days. We selected the model with the lowest validation loss, determined at moving window intervals of 128 iterations.

2.3. Inference Workflow for Volumetric Data-Sets

Since SAM works only on RGB image data-sets but we wanted to segment volumetric data-sets, we had to incorporate an adequate workflow to translate between these two spatial domains. Since our goal was to evaluate SAM for volumetric data-sets and not necessarily to implement a complete new volumetric version, we referred to simple operators. Figure 6 shows an overview of the approximate workflow for a volumetric data inference of SAM. In short, we extract a sub-volume tile from the input volume and pass it to the volumetric SAM adaption, which transforms it into three orthogonal slice stacks.

For each slice stack, we perform slice preparations (such as normalization and zero-padding), a forward pass through SAM, selection of the corresponding outputs, and slice post-processing. The output slice stacks are then merged and undergo further volumetric post-processing to generate segmentation proposals, which are returned from the volumetric SAM adaption into the inference algorithm. The evaluated algorithms are listed and compared in Table 1.

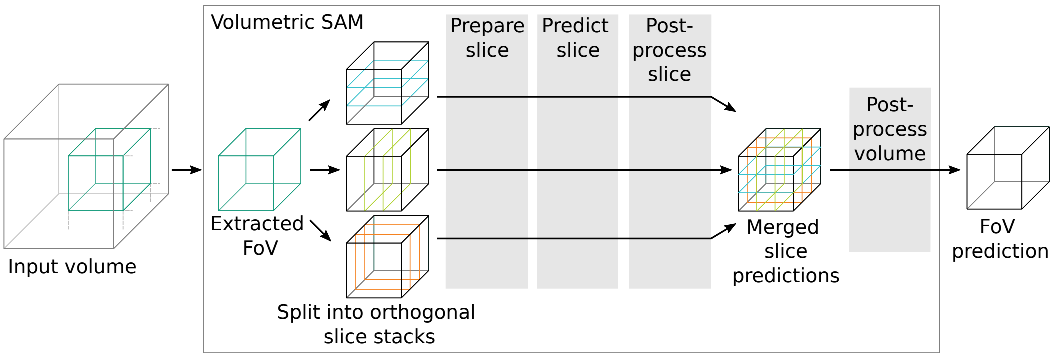


Figure 6. Schematic workflow of the volumetric data inference segmentation using SAM. Algorithm options and steps for the configurable stages (grey boxes ▨) are listed in Table 1.

Table 1. Overview of algorithm choices and options for different stages of the volumetric SAM adaption seen in Figure 6.

Stage	Algorithm	Description (*Options*)
Preprocess Slice Algorithms		
	Slice Normalization	Normalization of pixel values in each slice to the minimum and maximum range of the slice.
	Outlier and Empty Slice Detection	Identification and handling of outlier and empty slices.
	RGB Conversion	Conversion of grey values to RGB colour in order to comply with SAM interface requirements.
	Enframing	Adds a zero-padded border to each slice to centre the seed point to comply with SAM interface requirements.
	Estimated Foreground Volume	Utilizes different *binarization strategies* and *thresholds* to estimate the foreground volume.
Predict Slice Algorithms		
	Prompt Type	Type of prompt is used for invoking SAM: point prompt for tile centre and dense prompt from accumulator.
	Multimask Output Selector	Select mask from multiple disambiguating instance output channels predicted by SAM: maximum predicted IoU, fixed index of channel; maximum IoU with estimated foreground to avoid segmenting background; and minimum count of voxels to reduce under segmentation.
	Mask Output Selector	Selected output format of SAM: binary full resolution mask and quarter resolution logits with subsequent *threshold* and *upscaling algorithm*.
Postprocess Slice Algorithms		
	Seed Point Filter	Aborts or continues prediction based on the seed point's classification as background or foreground (*count of slices*).
	Merge Slice Rule	Rule that should be used to decide if and how to merge slices to stacks and when to abort an computation stack: *Always*; *BreakOnEmptySlice*; *MinimumIOUToLastSlice* (*threshold*); *MinimumIOUToForeground* (*threshold*).
	Slice Median Connected Component Analysis and analyse connected components and keep only segment connected to seed point (*enabled or disabled*).	Apply median filter to each slice (*enabled or disabled*).
Postprocess Volume Algorithms		
	Merge Slice Predictions	Merge orthogonal slice stack predictions based on *count* of foreground voxels.
	Volume Median	Apply median filter to merged volume (*enabled or disabled*).

2.3.1. Adapting SAM for Volumetric Data-Sets

Adapting SAM, which was originally designed for segmenting image data-sets, to our volumetric CT data-sets required certain modifications and the implementation of appropriate post-processing steps. In this section, we explore various possibilities for this transition and subsequently outline the approach we finally selected.

Several 2D to 3D techniques can be utilized to facilitate this transformation [26]. For example, in [27], a Volumetric Fusion Net (VFN) was employed to merge multiple 2D segmentation predictions into a comprehensive 3D prediction volume. In a related work, Ref. [28] adopted a similar methodology for pancreas segmentation, albeit utilizing a different VFN. According to [26], other approaches involve incorporating neighbouring 2D slices as additional channel information or utilizing specialized topologies to extract and merge features in both the 2D and 3D domains. However, the effectiveness of these methods for improving segmentation results heavily depends on the specific data-sets at hand.

Due to reports on the segmentation performance of SAM on volumetric medical data-sets, such as those in [29] and our own preliminary experiments, which suggested that the segmentation quality of SAM was likely to be mixed, we opted for a simple majority voting approach to merge the 2D predictions into 3D volumes.

During the slice merging process, we experimented with different rules to determine when to terminate the slice-wise merging. We either combined all slice within the current field of view regardless of their content or stopped at the first empty slice, i.e., a slice without foreground voxels. We also tested various rules based on different thresholds of overlap or Intersection over Union (IoU) between the proposed segmentation of the current slice and the preceding slice or a foreground volume obtained through global Otsu thresholding followed by a morphological closing step.

As an optimization strategy, slice-wise prediction was performed in an alternating manner, starting from the centre of the current sub-volume and moving outward slice-wise in both directions. This approach was implemented to save computational time and prevent the segmentation of unconnected segments, ensuring that only cohesive regions were accurately identified.

In situations where the segmentation results in an identification of unconnected segments, the algorithm may inadvertently continue segmenting entire regions composed of non-cohesive segments. This phenomenon occurs when the segmentation quality is significantly compromised. During the subsequent hyperparameter search, we also permitted segmentations without applying these rules. However, it appears that these deviations have only minimal impact on the output quality.

Subsequently, a new target volume is constructed. Voxels are included in the output volume if they are segmented as the foreground in at least one and depending on the configuration, up to three slice-wise predictions.

Additionally, we employed post-processing techniques such as slice-wise and volume-based median filtering and CCA prior to and after merging the slices into volumes to smooth scattered and miss segmented voxels.

We also conducted experiments with different variants of SAM's outputs. Since SAM has the ability to generate multiple outputs per prompt, such as separating a backpack from a person wearing it, we investigated whether selecting any of these outputs could improve the segmentation quality. Specifically, we examined whether it is better for volumetric segmentation to use the segmentation proposal provided by SAM with the highest probable IoU or the one with the maximum IoU of the approximated foreground volume. Additionally, as SAM often tends to under-segment and include background or neighbouring segments as part of the foreground, we investigated whether selecting the output with the smallest count of voxels among the multiple outputs would improve the segmentation quality.

In this context, experiments were conducted using both the binarized output of SAM and the raw probability values, which are available at a lower resolution than the binary mask. After upscaling, different threshold values can be applied to the probability outputs for further processing and experimentation.

2.3.2. Tile-Based Segmentation for Data-Sets of Arbitrary Size

Due to SAM's image-based nature, we encounter segmentation challenges when dealing with topologically complex objects depicted by volumetric CT NDT data-sets. These volumes may contain holes or inclusions; complex folds are spatially sparse or may extend beyond the boundaries of the currently processed tile.

To clarify this, Figure 7 offers a visual exposition of several schematically depicted objects of varying complexity. The figure serves to illustrate how, in a volumetric context, such complex segments are easier to understand but when segmenting them slice by slice there is a risk of mistakenly delineating them as multiple segments. This effect also occurs when the tile is smaller than the entity's size.

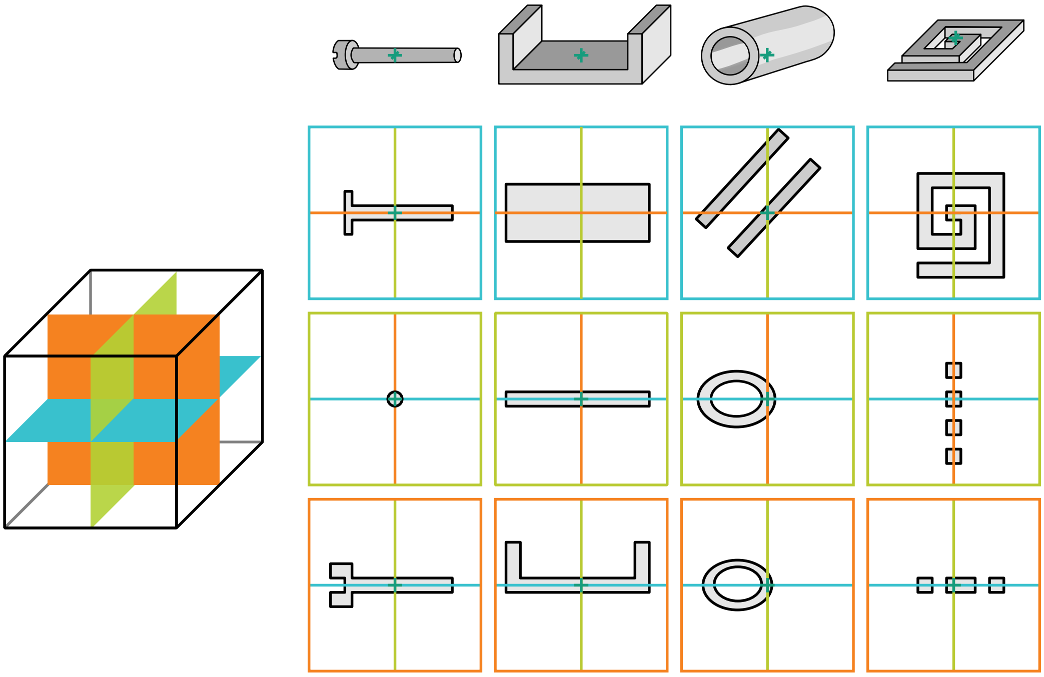


Figure 7. Schematic views of multiple simple volumetric objects (bolt, U-profile, pipe, and spiral spring) and cross-sectional slices along their central axes in three orthogonal directions marked by three respective colours (🟦, 🟩, 🟧). The disjunction of simple objects into multiple components if processed slice-wise poses a challenge as there are no straightforward rules for merging them without a step-by-step traversal of the object.

To overcome these challenges, we utilize volume-based SAM inference (see Section 2.3.1) within the FFN framework (see Section 1.2). The inference process starts with a single seed point and is applied to a small sub-volume tile. The resulting segmentation proposal is then stored in a result buffer, the accumulator volume. If a segment intersects the outer boundaries of a tile, the intersection position is added to a queue. In subsequent iterations, corresponding slightly shifted tiles aimed at these intersection points are processed by the volume-based SAM inference. This iterative process generates segmentation proposals, which are incorporated into the accumulator. This process repeats until the intersection points queue is empty and the segmentation proposal in the accumulator is no longer constrained by the boundary of the processed tiles.

As an optimization step, the proposed additional intersection positions are filtered based on the approximated foreground volume. They are added to the intersection points queue only if the corresponding voxels have a high probability to be foreground voxels.

The proposed combination of SAM and FFN allows us to compute segments and input volumes of arbitrary size by combining multiple overlapping tiles using a temporary accumulator volume. Nevertheless, this approach also increases the runtime due to the recomputation of the overlapping tiles.

The choice of using 48 voxels per tile side was made heuristically based on the original FFN algorithm, which also uses this tile size. However, the algorithm can be adjusted by changing the tile size up to 1024 voxels in each dimension; the maximum dimension SAM can handle without resizing the input. When the tile size is below this threshold, no resizing of tiles is required as we add a constant value border around the tile. Additionally, the step width between tiles and the overlap of the tiles can be adjusted to mitigate artefacts caused by the tile-based algorithm. Tile-based algorithms are capable of assembling entities with complex topologies. These algorithms can follow or trace the segment itself over multiple tiles and steps, even if it forms highly complex shapes. But tile-based algorithms may introduce additional artefacts. The segmentation result of the combined algorithms is heavily dependent on the performance of the SAM segmentation.

2.3.3. Prompt Selection and Accumulator Integration

As mentioned above, SAM allows queries using various prompts such as point prompts (seed points and bounding boxes) and dense prompts (masks and brushes). Multiple studies [24,30] have shown that, depending on the input data, higher segmentation quality can be achieved by using multiple prompts, such as point prompts distributed evenly over the segment region or negative point prompts, which are not considered part of the segment. Additionally, the use of rectangular prompts consisting of two anchor points often leads to adequate segmentation results.

Given that the main objective of this study is to evaluate the applicability of SAM in the automated NDT domain, we have opted to solely assess single point prompts and dense prompts as they can be easily automated.

We placed a single point prompt at the exact centre of the tile. The centre point of a tile was either chosen by a seed point or deemed highly likely to belong to the current segment, due to the iterative processing of the tiles.

For dense prompts, we utilized the SAM output stored in the accumulator, which was shifted by the relative position of the current point prompt. This requires SAM to complete the segmentation proposal at the edge of the current tile. Since our tile step size was $[1, 20]$ voxels, the overlap between the tiles and the dense prompt with the expected segmentation proposal was high, allowing SAM to only predict a relative slim border of new voxels. Figure 8 illustrates an idealized schematic of such an operation. In the case of dense prompts, we also include a corresponding point prompt at the centre of the tile as more prompts tend to increase the segmentation performance [24].

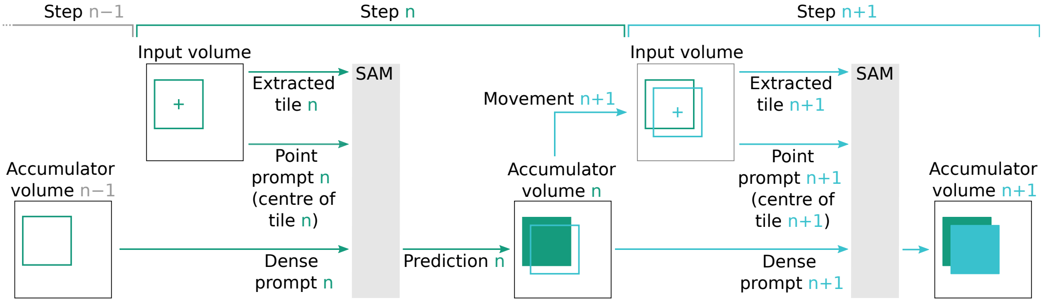


Figure 8. Schematic view of two subsequent inference steps, denoted as n (represented by 🟩) and $n+1$ (represented by 🟦), which use the modified accumulator volume from the previous step to create a dense SAM prompt. In step n, the content of the accumulator volume of the previous step $n-1$ is used to generate a dense SAM prompt $n+1$. This prompt, along with the point prompt n and the extracted input volume tile n, is used by SAM to compute prediction n. Subsequently, the accumulator volume is updated to the state n based on this prediction. In the subsequent step $n+1$, the accumulator volume n is used to determine the movement $n+1$ to the tile $n+1$. Tile $n+1$ significantly overlaps with tile n. SAM is parametrized with the extracted input volume tile $n+1$, point prompt $n+1$, and dense prompt $n+1$ to compute prediction $n+1$, which is used to update the accumulator volume $n+1$.

3. Results

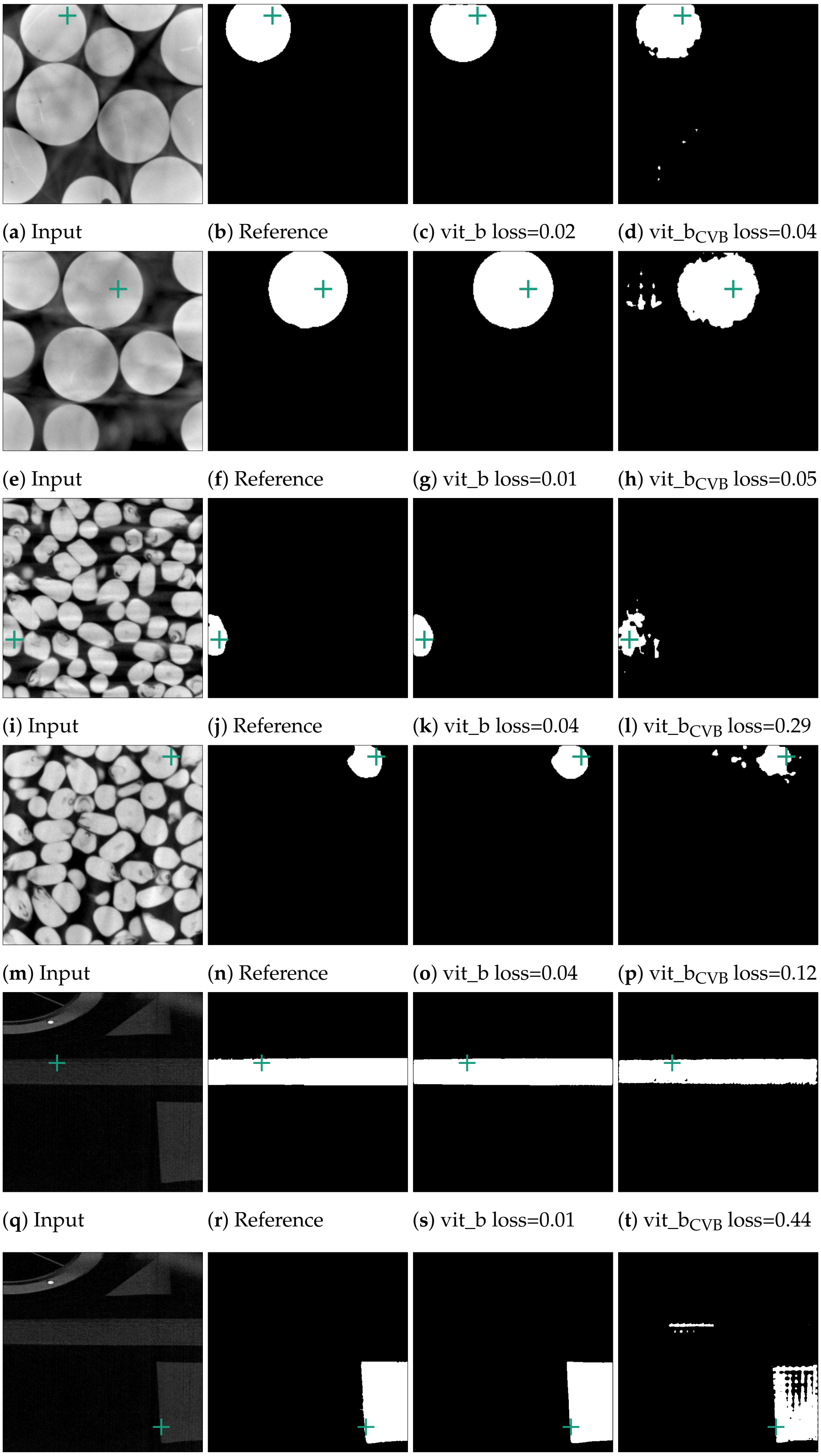
3.1. Evaluation of SAM Segmentation Quality in NDT Slice Data-Sets

In an initial test of SAM's segmentation quality for CT NDT data, we applied SAM to segment individual slices from NDT volumetric data-sets. We used three pre-trained SAM models, vit_h, vit_l, and vit_b, based on Vision Transformers (ViT) arranged in descending order of size. Additionally, we tested three fine-tuned versions of the vit_b model, each adapted to the Me 163 data-set with unique target configurations. For each of the three data-sets introduced in Section 2.1, randomly selected slices were selected and segmented, which accounted for approximately 0.5% of all available validation data-sets. Each example underwent the preparation steps outlined in Section 2.2 before being processed by SAM. SAM then tried to segment the entity located at the exact centre of each slice using point prompts. Examples of typical segments can be seen in Figure 9. Notably, SAM demonstrated good segmentation performance for the marbles and corn kernels data-sets, while the segmentation quality was significantly inferior for the individual segments of the Me 163 data-set. To quantify the segmentation performance across data-sets and models, Table 2 presents the mean loss values and standard deviations for slice-wise predictions made by multiple SAM model configurations. The statistics in this table show that while the vit_b model yields the lowest loss for the corn kernels data-set, with a mean loss of 0.10, the application of vit_b with a ConstantValueBackground modification achieved the best performance on the Me 163 data-set, reducing the mean loss to 0.36.

Table 2. Mean loss value (and standard deviation) over all slice-wise predictions on the validation data-sets by multiple models for the graphs in Figure 10. Models yielding the optimum performance for each data-set are denoted in bold. Models vit_h, vit_l, and vit_b denote pre-trained SAM models that utilise Vision Transformers (ViT) as their foundation, ordered from largest to smallest. The remaining models represent fine-tuned versions of vit_b applied to the Me 163 data-set, each employing distinct target configurations.

	Marbles	**Corn**	**Me 163**
vit_h	**0.03** (0.06)	0.11 (0.10)	0.49 (0.34)
vit_l	**0.03** (0.06)	0.11 (0.10)	0.46 (0.34)
vit_b	**0.03** (0.07)	**0.10** (0.10)	0.44 (0.32)
vit_b ForegroundOnly	0.41 (0.27)	0.66 (0.24)	0.49 (0.26)
vit_b ConstantValueBackground	0.15 (0.10)	0.44 (0.16)	**0.36** (0.25)
vit_b ConnectedComponentBackground	0.51 (0.24)	0.51 (0.19)	0.57 (0.23)

Figure 10 demonstrates the segmentation dynamics of the individual models on the different data-sets. These plots represent the loss of the segmentation proposals generated by SAM for the entities at the centre of each layer of the corresponding validation data-set. The loss values are determined with respect to the reference data-set. From left to right, the loss values are sorted in ascending order, so that the nearly correctly segmented segments are on the left side of the graph, while the difficult and often incorrectly segmented segments are on the right side. The seed points of the segments were chosen in such a way that each of them corresponds to a foreground voxel, so the networks are not tasked with segmenting the background. The different colours in the plots correspond to different networks.



(**a**) Input (**b**) Reference (**c**) vit_b loss=0.02 (**d**) $\text{vit\_b}_{\text{CVB}}$ loss=0.04

(**e**) Input (**f**) Reference (**g**) vit_b loss=0.01 (**h**) $\text{vit\_b}_{\text{CVB}}$ loss=0.05

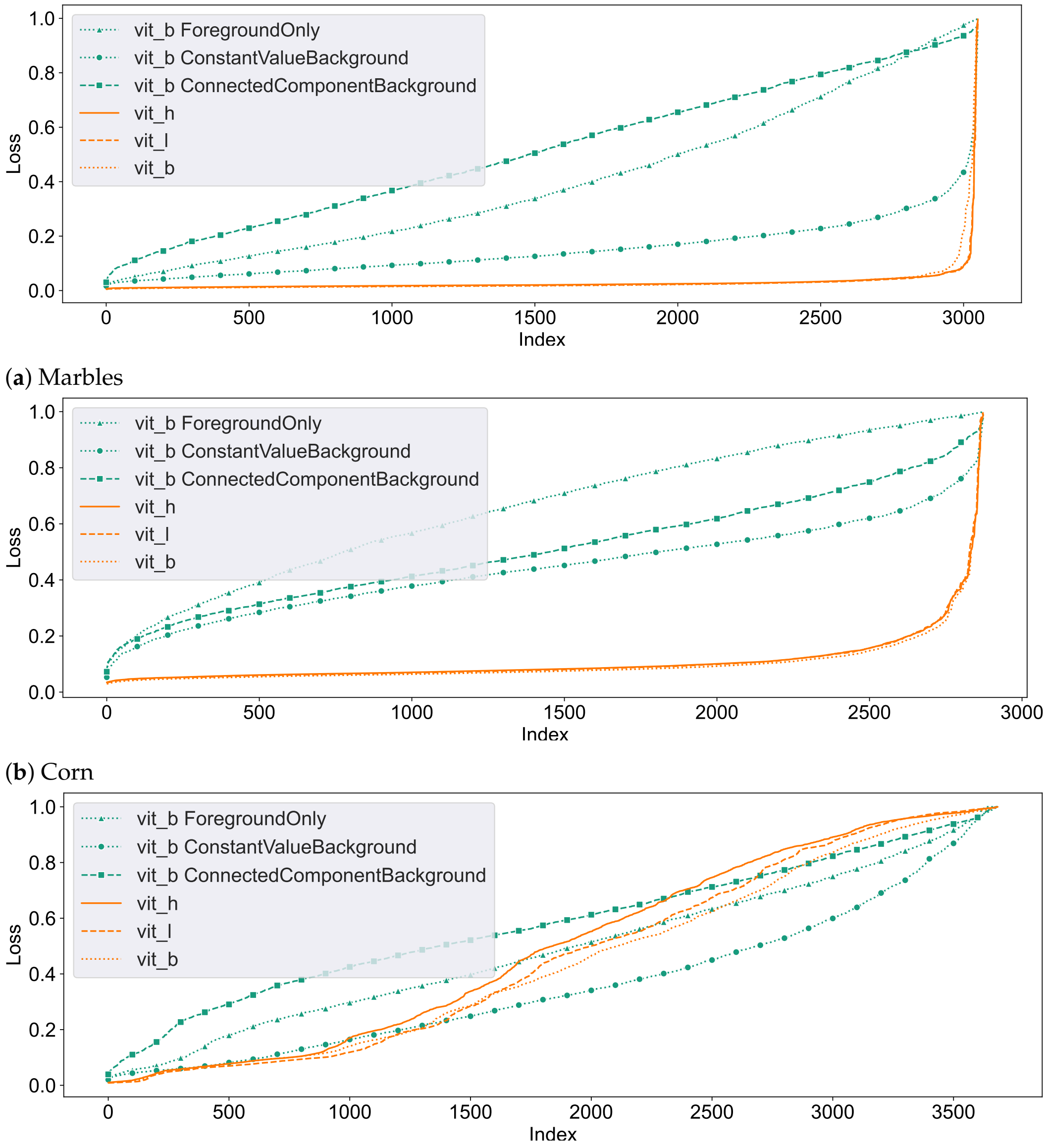
(**i**) Input (**j**) Reference (**k**) vit_b loss=0.04 (**l**) $\text{vit\_b}_{\text{CVB}}$ loss=0.29

(**m**) Input (**n**) Reference (**o**) vit_b loss=0.04 (**p**) $\text{vit\_b}_{\text{CVB}}$ loss=0.12

(**q**) Input (**r**) Reference (**s**) vit_b loss=0.01 (**t**) $\text{vit\_b}_{\text{CVB}}$ loss=0.44

(**u**) Input (**v**) Reference (**w**) vit_b loss=0.01 (**x**) $\text{vit\_b}_{\text{CVB}}$ loss=0.13

Figure 9. Segmented examples of the corn and marbles data-set. The green crosses mark the position of the currently used point prompt. The last column depicts the result of the vit_b model, which was fine-tuned on the Me 163 data-set.



(**a**) Marbles

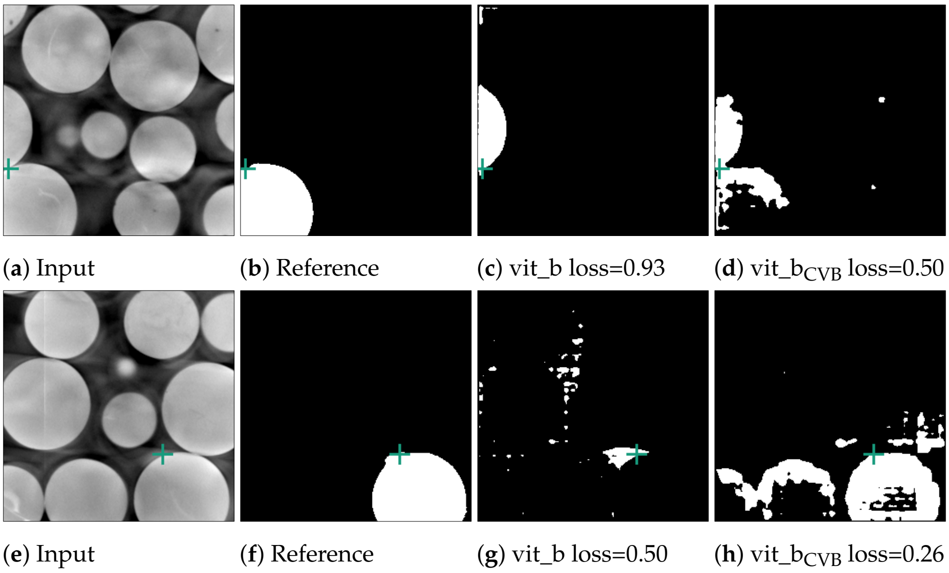
(**b**) Corn

(**c**) Me 163

Figure 10. Graphs depicting the slice segmentation performance of the six evaluated SAM models on the three different testing data-sets. From left to right, the index of each segmented slice sorted by their loss value. In an ideal case, only a horizontal line close to the loss value of 0 would be visible.

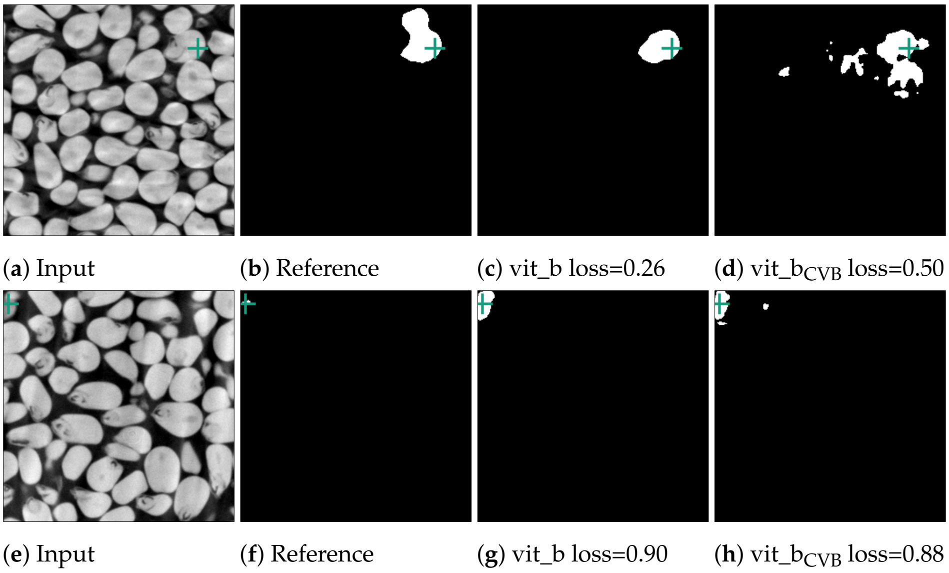
It can be observed that the unchanged SAM networks perform very well in segmenting the marble and corn data-sets. The few entities which exhibit lower segmentation quality in these data-sets and are located on the right edge are often due to insufficient quality in the reference segmentation data-set, as illustrated in Figures 11 and 12. A slightly lower segmentation quality can be observed for the corn data-set, which consists of a higher count of entities that are also not as homogeneous in colour compared to the marble data-set.

Figure 10c demonstrates that the segmentation quality for the Me 163 data-set is notably lower compared to the previously mentioned data-sets. Figure 13 displays some typical error patterns in the original trained SAM images. Both under-segmentation and over-segmentation occur and segments are sometimes partially or not recognized at all.


(**a**) Input (**b**) Reference (**c**) vit_b loss=0.93 (**d**) vit_b$_{\mathrm{CVB}}$ loss=0.50

(**e**) Input (**f**) Reference (**g**) vit_b loss=0.50 (**h**) vit_b$_{\mathrm{CVB}}$ loss=0.26

Figure 11. Error cases for the marble data-set. Here, the reference segmentation, which was generated by a connected component analysis, is erroneous. In (**b**), the point prompt (marked with a green cross) lies on the boundary of two marbles and vit_b segments the upper marble instead of the lower marble. In (**f**), the point prompt lies inside an artefact region.


(**a**) Input (**b**) Reference (**c**) vit_b loss=0.26 (**d**) vit_b$_{\mathrm{CVB}}$ loss=0.50

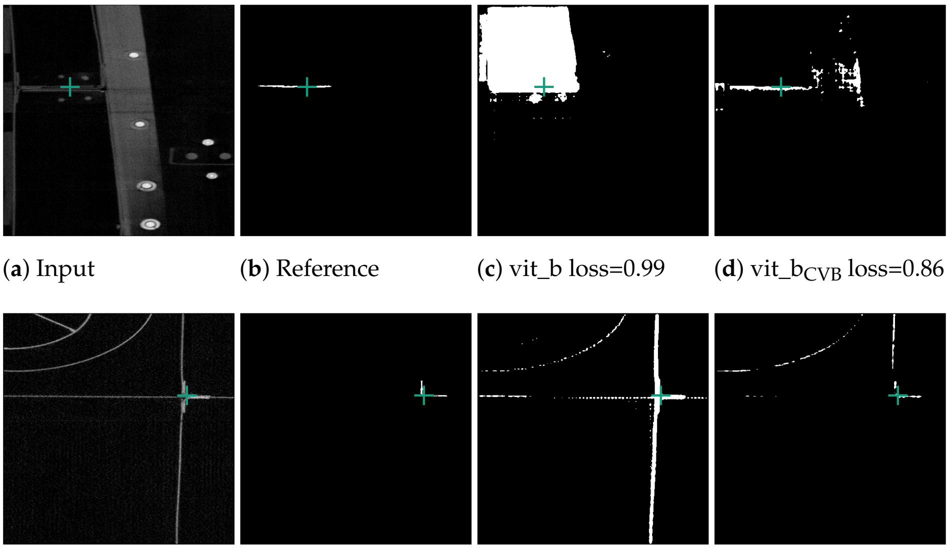
(**e**) Input (**f**) Reference (**g**) vit_b loss=0.90 (**h**) vit_b$_{\mathrm{CVB}}$ loss=0.88

Figure 12. Error cases of the corn data-set. In the first case in (**b**), two kernels were erroneously segmented together in the reference segmentation. In contrast, in (**f**), the reference segmentation only appears erroneous as the current slice only depicts one voxel. The next slice in the input volume contains the kernel this voxel belongs to. The green crosses mark the position of the currently used point prompt.

Among the different not fine-tuned SAM models, the smallest model vit_b showed the most promising results. While it was sometimes outperformed by the other two original SAM models, vit_l and vit_h, in the well-segmented slices, it still had a higher segmentation quality in the moderately segmented slices. Therefore, we decided to use vit_b as the base model for fine-tuning and volumetric segmentation experiments.

Among the subsequently trained networks, vit_b$_{\mathrm{CVB}}$ exhibits the highest quality in Figure 10c. It is based on vit_b and uses *ConstantValueBackground* (*CVB*) (see Section 2.2) for background examples. In simple cases, it matches the segmentation quality of non fine-tuned SAM variants. A considerable improvement in segmentation quality on the challenging entities could be achieved through training, although not to a satisfactory level.

This model was chosen as the representative of our fine-tuned model for further tests on our data.



(**a**) Input (**b**) Reference (**c**) vit_b loss=0.99 (**d**) vit_b$_{CVB}$ loss=0.86

(**e**) Input (**f**) Reference (**g**) vit_b loss=0.99 (**h**) vit_b$_{CVB}$ loss=0.76

Figure 13. Poorly performing cases for SAM vit_b segmenting thin metal sheets in the Me 163 data-set as well as the better but still not optimal segmentation results achieved by the model fine-tuned on the Me 163 data-set.

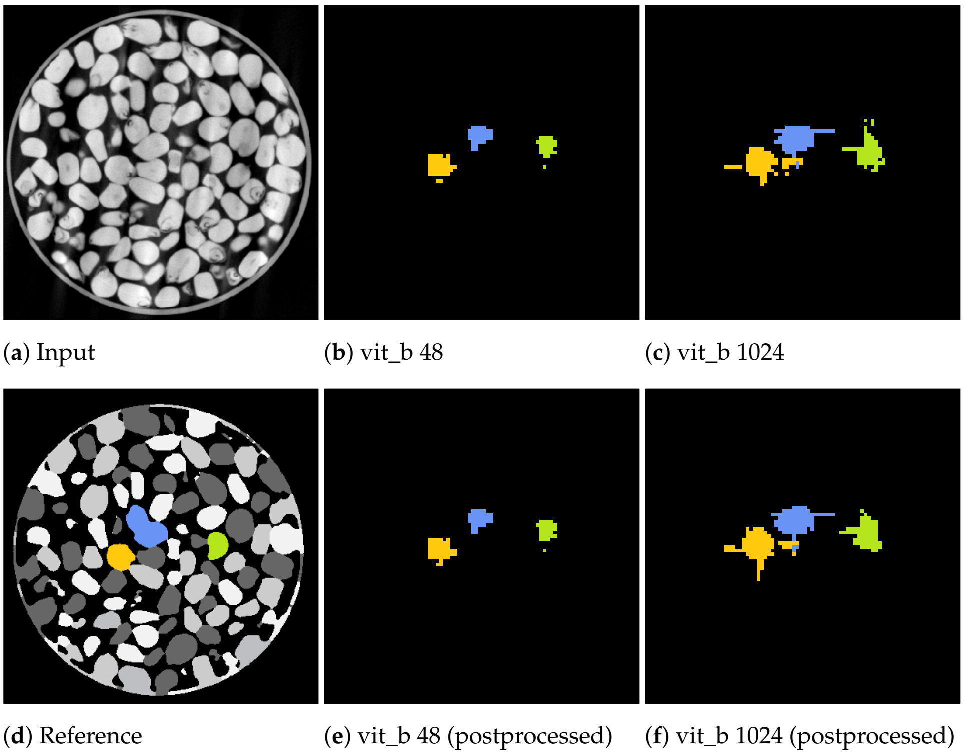
3.2. Tile-Based Algorithms and Artefact Mitigation

Figures 14 and 15 showcase the segmentation results of a volumetric inference run using the proposed SAM algorithm on a small subset of the marble and corn data-sets for the two tile sizes 48 × 48 × 48 voxels and 1024 × 1024 × 1024 voxels. These results exhibit segmentation errors in the form of erroneous segmented edges as well as tiling artefacts, resulting in a textured appearance of the segment with noticeable gaps.

Notably, for a tile size of 48 × 48 × 48 voxels, the marble example in Figure 15b demonstrates tiling artefacts. Since the volumetric inference algorithm with the small tile size cannot segment the entire marble in a single step, it must combine multiple steps, which can introduce and propagate errors. These artefacts can be cleaned up using a morphological closing operation as a post-processing step.

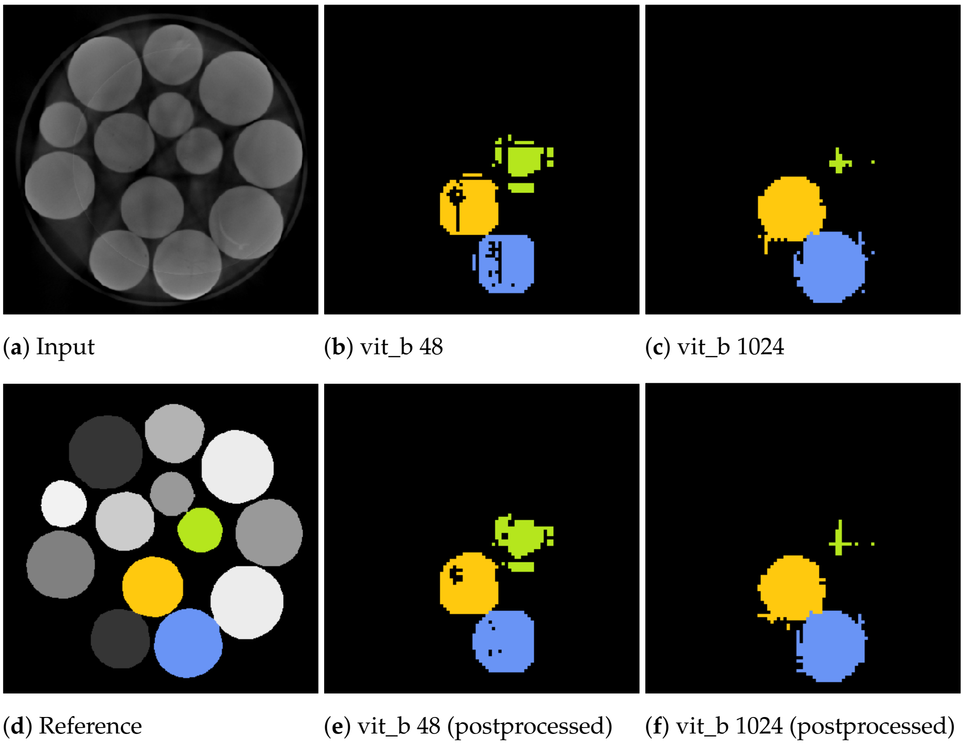
In contrast, segmentations using a larger tile size of 1024 × 1024 × 1024 voxels exhibit fewer of these textured artefacts. However, segmentations may extend beyond the actual segment due to segmentation errors, as illustrated in Figure 14c, where thin segments protrude vertically and horizontally beyond the intended boundaries. These protrusions often occur within the initially segmented slices that include the seed point of the current segment. In the green upper right marble of the example in Figure 15c, the adjacent slices directly connected to the seed point were misclassified as not belonging to the marble, resulting in an early termination of the slice-wise segmentation process.

The inference algorithm with a tile size of 1024 × 1024 × 1024 voxels can only attempt to segment the segment once as, due to its high field of view, it performs a single volumetric step per seed point. In contrast, the inference algorithm with a tile size of 48 × 48 × 48 voxels iterates over the volume in multiple steps, providing the ability to compensate for weak and erroneous segmentations in subsequent steps. However, this approach tends to under-segment when a neighbouring segment has already been partially segmented in a previous step.


(**a**) Input (**b**) vit_b 48 (**c**) vit_b 1024

(**d**) Reference (**e**) vit_b 48 (postprocessed) (**f**) vit_b 1024 (postprocessed)

Figure 14. Slices from a volumetric inference run on three corn kernels of the corn data-set. The input volume (**a**), reference volume (**d**), and the proposed segmentations generated by the proposed algorithm using the two tile sizes: 48 × 48 × 48 voxels (**b**) and 1024 × 1024 × 1024 voxels (**c**). Additionally, the postprocessed volumes are depicted in (**e**,**f**).


(**a**) Input (**b**) vit_b 48 (**c**) vit_b 1024

(**d**) Reference (**e**) vit_b 48 (postprocessed) (**f**) vit_b 1024 (postprocessed)

Figure 15. Slices from a volumetric inference run on three marbles of the marbles data-set. The input volume (**a**), reference volume (**d**), and the proposed segmentations generated by the proposed algorithm using the two tile sizes: 48 × 48 × 48 voxels (**b**) and 1024 × 1024 × 1024 voxels (**c**). Additionally, the postprocessed volumes are depicted in (**e**,**f**).

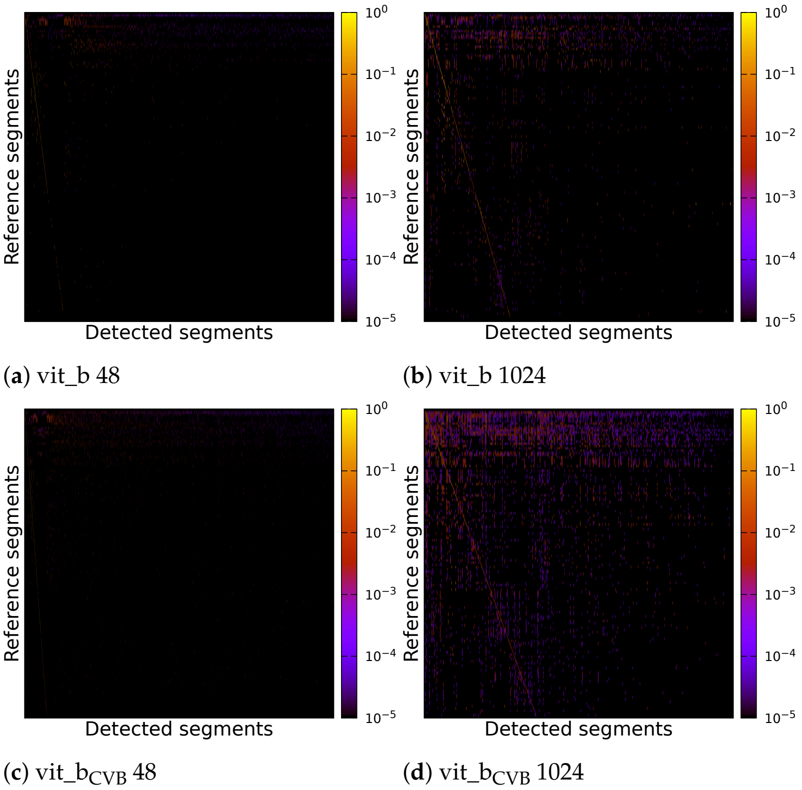
Due to the suboptimal quality of the segmentation, it proves problematic to compute a definitive overall numerical assessment of the complete segmentation. This difficulty arises from the ambiguity in assigning each segment unambiguously to a reference segment, a result of widespread under-segmentation or over-segmentation, which gives rise to various possible interpretations. Figure 16 shows the correlation matrices for the result of four inference runs on the Me 163 testing data-sets. Two of the inference runs were performed using the default SAM model vit_b, while the other two were performed using the fine-tuned model vit_b$_{\mathrm{CVB}}$. Two of the four experiments used a tile size of 48 × 48 × 48 voxels and the other two used a tile size of 1024 × 1024 × 1024 voxels. Each experiment was fine-tuned on the validation data-set using [31].

The correlation matrices show the IoU of each reference segment in relation to each detected segment. The reference segments are sorted from top to bottom based on their voxel count, with the segment having the largest voxel count at the top. Similarly, the columns representing the detected segments are sorted so that the segment with the highest IoU, if compared with the largest reference segment, is on the left side. The segment with the highest IoU if compared with the second largest reference segment is then placed in the second column and so on. Each detected segment can only be linked to one reference segment once. In an ideal case, we would see a bright diagonal line from the upper left corner to the lower right corner of the matrix, indicating a perfect match between the reference and detected segments. Segments outside this diagonal indicate segmentation errors. Vertical lines indicate under-segmentation, where reference segments extend over multiple detected segments. Horizontal lines indicate over-segmentation, where reference segments are falsely split into multiple detected segments.

The individual parameters of the four inference runs can be found in Table 3. Figure 17 displays correlation matrices from Figure 16 but constrained to the detected segments with the highest IoU.

Table 3. Parameters optimized on the Me 163 validation data-set for the default vit_b and fine-tuned vit_b$_{\mathrm{CVB}}$ SAM model for the tile sizes of 48 × 48 × 48 voxels and 1024 × 1024 × 1024 voxels. (FG = foreground; – = not applicable; Options marked with * indicate volumetric SAM parameters as seen in Table 1; Options marked with × indicate FFN related parameters).

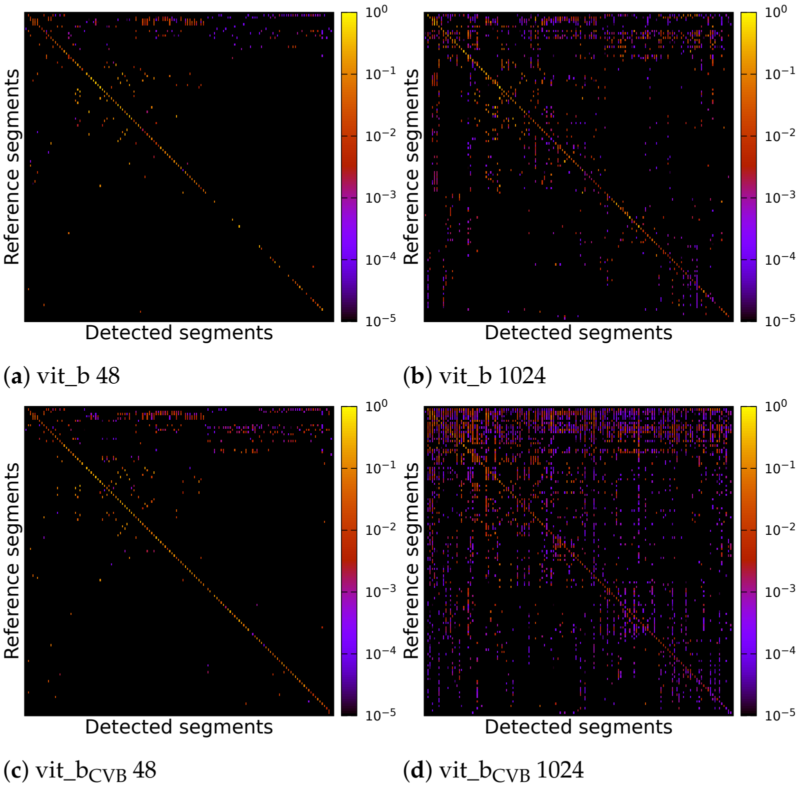
	vit_b 48	**vit_b 1024**	**vit_b$_{\mathrm{CVB}}$ 48**	**vit_b$_{\mathrm{CVB}}$ 1024**
best IoU	0.15	0.17	0.07	0.09
movement step *	1	–	1	–
seed FG count *	2	2	1	1
slice FG count *	3	1	1	1
FG threshold *	0.3	0.2	0.2	0.5
prompt type *	centre and dense	centre	centre and dense	centre and dense
SAM output channel *	index 1	max IoU	max IoU with FG	max IoU with FG
slice merge rule *	IoU to previous slice > 0.5	IoU to previous slice > 0.25	IoU to previous slice > 0.5	always
slice median *	✓	×	×	×
CCA *	✓	✓	×	×
volume median *	×	✓	×	✓
check step width ×	13	–	19	–
accumulator update ×	FG	FG	always	always
restrict movement ×	FG (128 steps)	eroded FG	eroded FG (128 steps)	FG



(**a**) vit_b 48 (**b**) vit_b 1024

(**c**) vit_b$_{\mathrm{CVB}}$ 48 (**d**) vit_b$_{\mathrm{CVB}}$ 1024

Figure 16. Correlation matrix of default and fine-tuned volumetric SAM with multiple tiles of size 48 × 48 × 48 voxels or a single tile of size 1024 × 1024 × 1024 voxels of the Me 163 testing data-set.

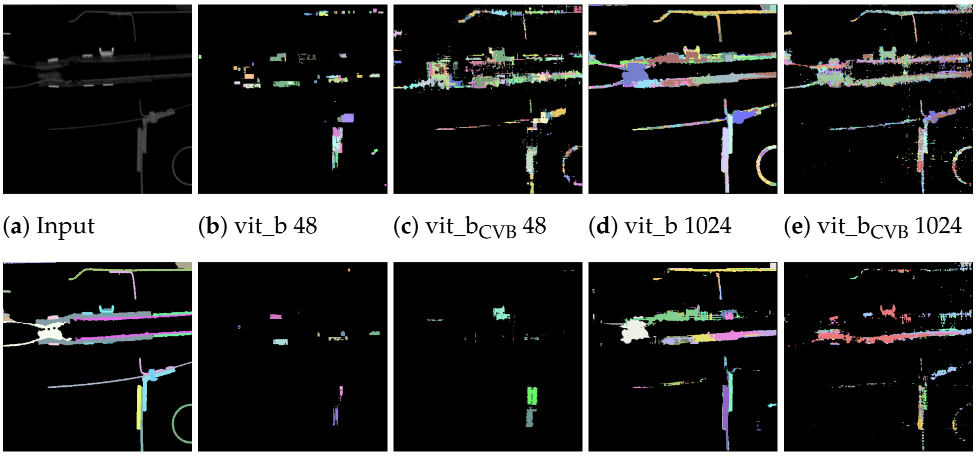


(**a**) vit_b 48 (**b**) vit_b 1024

(**c**) vit_b$_{\mathrm{CVB}}$ 48 (**d**) vit_b$_{\mathrm{CVB}}$ 1024

Figure 17. Correlation matrix of default and find-tuned volumetric SAM with multiple tiles of size 48 × 48 × 48 voxels or a single tile of size 1024 × 1024 × 1024 voxels of the Me 163 testing data-set. Detected segments have been limited to the best matches for each reference segment.

As can be seen, the vit_b$_{\text{CVB}}$ models tends to generate more noise outside the main diagonal. Figure 17d especially depicts many over- and under-segmented segments. This can also be observed in the corresponding segmentation volume slice shown in Figure 18e,j. The correlation matrix of the fine-tuned vit_b$_{\text{CVB}}$ model with tile size 48 × 48 × 48 voxels in Figure 17c seems to perform best with respect to diagonal segments. But comparing the corresponding segmentation volume slice in Figure 18h shows that this model, tile, and parameter combination tends to miss most of the foreground segments. It seems that the default vit_b model with tile size 1024 × 1024 × 1024 voxels produces the visually best results, followed by the fine-tuned vit_b$_{\text{CVB}}$ model with tile size 48 × 48 × 48 shown in Figure 18c.



(**a**) Input (**b**) vit_b 48 (**c**) vit_b$_{\text{CVB}}$ 48 (**d**) vit_b 1024 (**e**) vit_b$_{\text{CVB}}$ 1024

(**f**) Reference (**g**) vit_b 48 (main diagonal) (**h**) vit_b$_{\text{CVB}}$ 48 (main diagonal) (**i**) vit_b 1024 (main diagonal) (**j**) vit_b$_{\text{CVB}}$ 1024 (main diagonal)

Figure 18. Exemplary slices of the proposed volumetric inference output performed by default and fine-tuned SAM models on the Me 163 reference data-set (**a**,**f**). For the remaining figures, the top row (**b**–**e**) shows all segments depicted in Figure 16 while the bottom row (**g**–**j**) only shows the segments corresponding to the main diagonal in Figure 17.

Figure 19 presents multiple renderings of the seven largest reference segments in the Me 163 testing data-set, alongside their corresponding segment predictions generated by different SAM snapshots using the volumetric algorithm and fine-tuned parameters. The *true positive* voxels are coloured green, the reference segments are coloured blue, and the *false positive* voxels are coloured orange. It is evident that the volumetric segmentation of the data-sets using tiles of size 1024 × 1024 × 1024 voxels yields visually more appealing segments compared to using a tile size of 48 × 48 × 48 voxels.

The predicted segmentation using the tile size of 48 × 48 × 48 voxels often appears *empty*, as only a small count of voxels has been segmented correctly. This is because the segmentation quality of the algorithm is too poor to generate connected tiles and so often, only a limited amount of steps (see Section 2.3.2) will be iterated for each segment. The segments are interrupted and only found in pieces. However, using a tile size of 48 × 48 × 48 voxels also often leads to under-segmentation. Figure 20 exemplifies this, showcasing two orthogonal slices from the fine-tuned vit_b$_{\text{CVB}}$ model's segmentation output. On the left, we present the reference segments and on the right, the corresponding predictions. Here, three adjacent segments were mistakenly connected by a single predicted segment.

But even the segmentation with a tile size of 1024 × 1024 × 1024 voxels is often insufficient, as both large-scale under-segmentations and over-segmentations occur, as can be seen from the correlation matrices in Figure 17 and the cross-sectional images in Figure 18j.

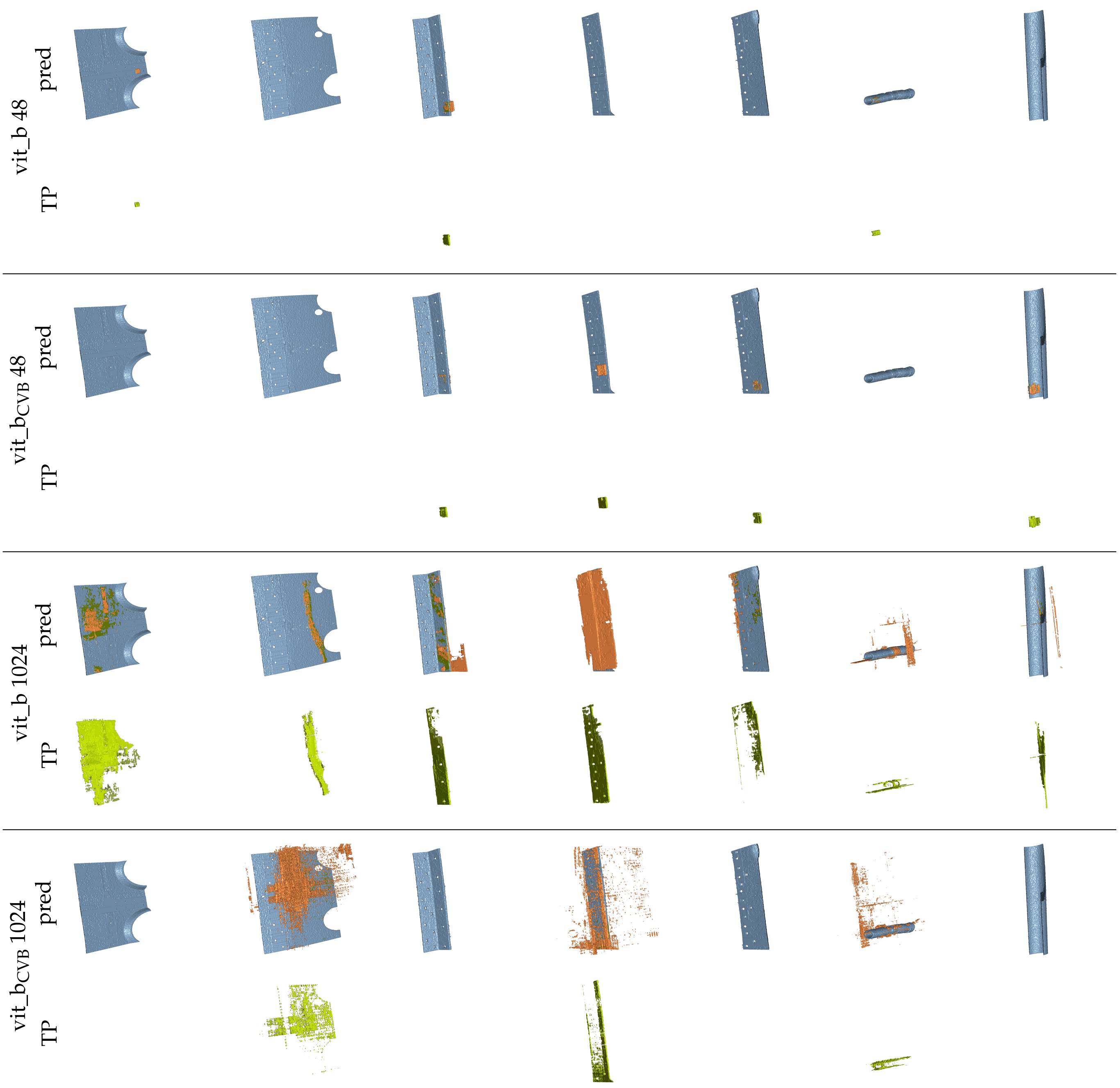


Figure 19. Renderings of the seven largest segments of the reference data-set and their corresponding predictions (pred) created with different snapshots of SAM and the volumetric algorithm. The colour coding is as follows: blue 🟦 reference segment, green 🟩 true positives (TP), and orange 🟧 false positives.

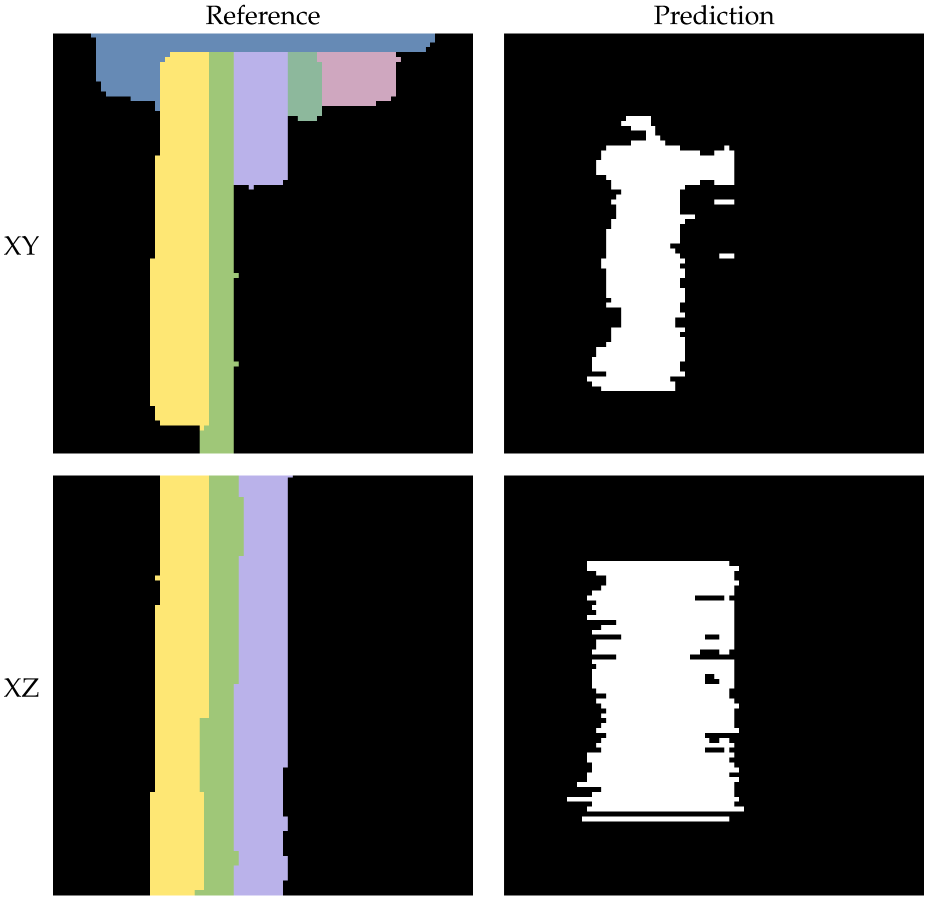


Figure 20. Slices obtained using the fine-tuned vit_b$_{\mathrm{CVB}}$ model and tile size of 48 × 48 × 48 voxels. Due to under-segmentation, the predicted segment erroneously intersects and merges multiple reference segments.

4. Discussion

The transferability of the SAM model to instance segmentation of volumetric XXL-CT data-sets requires careful consideration. The presented results indicate that its two-dimensional image-based segmentation quality is insufficient for this specific problem domain. This limitation becomes particularly evident when dealing with the concatenation of numerous intertwined cross-sectional images in the volumetric case. The low contrast and high noise in these images pose challenges in accurately delineating individual segments. Additionally, using domain specific fine-tuning and improving slice-wise predictions did not yield substantial improvements for volumetric predictions.

One potential source of error in the presented method might be the limited computational resources allocated for both fine-tuning and subsequent hyperparameter search. A more thorough optimization process could potentially improve the results. Furthermore, the availability of labelled training data-sets of sufficient quality in this problem domain was relatively limited for training the vision transformers included in SAM. Specifically, the absence of neighbouring voxels when adding the 512 voxel wide border around the data-set for the Me 163 data-set may have possibly contributed to a decrease in segmentation quality.

Additionally, considering improved algorithms for merging the slice-wise predictions could be an initial step in the further development process. Previous studies [26–28] have demonstrated ample opportunities for the development of more sophisticated algorithms in this area. Implementing and embedding such algorithms into the processing pipeline has the potential to significantly enhance the segmentation quality.

5. Conclusions

The primary objective of this study was the exploration and possible applicability of the SAM algorithm for general image delineation to instance segmentation in XXL-CT volumetric data-sets.

In conclusion, our study highlights the potential of SAM for instance segmentation in XXL-CT volumetric data-sets, while acknowledging that there is still significant room for improvement. Furthermore, our research contributes to the following areas: (1) the evaluation of SAM on data-sets from the field of non-destructive testing based on CT image data, (2) the exploration of various methods for integrating and fusing the output from image-based SAM with volumetric data-sets, (3) the introduction of a tile-based approach for segmenting objects of arbitrary size, and (4) the utilization of dense prompts for tile combination using an accumulator. Separately and in combination, these contributions provide novel insights to the community and hence establish a foundation for further advancements in this field.

Author Contributions: Conceptualization, R.G.; Data curation, R.G.; Investigation, R.G. and S.R.; Methodology, R.G.; Resources, R.G.; Software, R.G. and S.R.; Supervision, T.W.; Validation, R.G.; Visualization, R.G.; Writing—original draft, R.G.; Writing—review and editing, S.R. and T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the centre for Analytics—Data—Applications (ADA-Center) within the framework of "BAYERN DIGITAL II" (20-3410-2-9-8).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Me 163 data-sets presented in this study are openly available in [7]. The bulk material data-sets will be made available by the authors on request.

Acknowledgments: The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High-Performance Computing Center (NHR@FAU) of the Friedrich—Alexander Universität Erlangen—Nürnberg (FAU) under the NHR project b179dc. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG)—440719683. Additionally, the authors also extend their appreciation to Moritz Ottenweller for his valuable assistance during the manuscript revision process.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Salamon, M.; Reims, N.; Böhnel, M.; Zerbe, K.; Schmitt, M.; Uhlmann, N.; Hanke, R. XXL-CT capabilities for the inspection of modern Electric Vehicles. In Proceedings of the International Symposium on Digital Industrial Radiology and Computed Tomography, Fürth, Germany, 2–4 July 2019.
2. Kolkoori, S.; Wrobel, N.; Hohendorf, S.; Redmer, B.; Ewert, U. Mobile High-energy X-ray Radiography for Nondestructive Testing of Cargo Containers. *Mater. Eval.* **2015**, *73*, 175–185.
3. Kolkoori, S.; Wrobel, N.; Hohendorf, S.; Ewert, U. High energy X-ray imaging technology for the detection of dangerous materials in air freight containers. In Proceedings of the 2015 IEEE International Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 14–16 April 2015; pp. 1–6. [CrossRef]
4. Gruber, R.; Gerth, S.; Claußen, J.; Wörlein, N.; Uhlmann, N.; Wittenberg, T. Exploring Flood Filling Networks for Instance Segmentation of XXL-Volumetric and Bulk Material CT Data. *J. Nondestruct. Eval.* **2021**, *40*, 1. [CrossRef]
5. Gruber, R.; Reims, N.; Hempfer, A.; Gerth, S.; Wittenberg, T.; Salamon, M. *Fraunhofer EZRT XXL-CT Instance Segmentation Me163*; Zenodo: Geneva, Switzerland, 2024. [CrossRef]
6. Gruber, R.; Engster, J.C.; Michen, M.; Blum, N.; Stille, M.; Gerth, S.; Wittenberg, T. Instance Segmentation XXL-CT Challenge of a Historic Airplane. *arXiv* **2024**, arXiv:cs.CV/2402.02928.
7. Gruber, R.; Reims, N.; Hempfer, A.; Gerth, S.; Salamon, M.; Wittenberg, T. An annotated instance segmentation XXL-CT data-set from a historic airplane. *arXiv* **2022**, arXiv:cs.CV/2212.08639.

8. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. *arXiv* **2023**, arXiv:cs.CV/2304.02643.
9. Hafiz, A.M.; Bhat, G.M. A survey on instance segmentation: State of the art. *Int. J. Multimed. Inf. Retr.* **2020**, *9*, 171–189. [CrossRef]
10. Wen, C.; Matsumoto, M.; Sawada, M.; Sawamoto, K.; Kimura, K.D. Seg2Link: An efficient and versatile solution for semi-automatic cell segmentation in 3D image stacks. *Sci. Rep.* **2023**, *13*, 7109. [CrossRef] [PubMed]
11. Zhao, T.; Olbris, D.J.; Yu, Y.; Plaza, S.M. NeuTu: Software for Collaborative, Large-Scale, Segmentation-Based Connectome Reconstruction. *Front. Neural Circuits* **2018**, *12*, 00101. [CrossRef]
12. Ohtake, Y.; Yatagawa, T.; Suzuki, H.; Kubo, S.; Suzuki, S. Thickness-Driven Sheet Metal Segmentation of CT-Scanned Body-in-White. *e-J. Nondestruct. Test.* **2023**, *28*, 27743. [CrossRef]
13. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:cs.CV/2010.11929.
14. Li, W.; Hsu, C.Y.; Wang, S.; Yang, Y.; Lee, H.; Liljedahl, A.; Witharana, C.; Yang, Y.; Rogers, B.M.; Arundel, S.T.; et al. Segment Anything Model Can Not Segment Anything: Assessing AI Foundation Model's Generalizability in Permafrost Mapping. *Remote. Sens.* **2024**, *16*, 797. [CrossRef]
15. Noe, S.M.; Zin, T.T.; Tin, P.; Kobyashi, I. Efficient Segment-Anything Model for Automatic Mask Region Extraction in Livestock Monitoring. In Proceedings of the 13th IEEE International Conference on Consumer Electronics—Berlin, ICCE-Berlin 2023, Berlin, Germany, 3–5 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 167–171. [CrossRef]
16. Carraro, A.; Sozzi, M.; Marinello, F. The Segment Anything Model (SAM) for accelerating the smart farming revolution. *Smart Agric. Technol.* **2023**, *6*, 100367. [CrossRef]
17. Weinberger, P.; Schwarz, L.; Fröhler, B.; Gall, A.; Heim, A.; Yosifov, M.; Bodenhofer, U.; Kastner, J.; Senck, S. Unsupervised Segmentation of Industrial X-ray Computed Tomography Data with the Segment Anything Model. *Res. Sq.* 2024, *preprint*. [CrossRef]
18. Xu, B.; Yu, S. Improving Data Augmentation for YOLOv5 Using Enhanced Segment Anything Model. *Appl. Sci.* **2024**, *14*, 1819. [CrossRef]
19. Liu, Z. Optimizing road sign detection using the segment anything model for background pixel exclusion. *Appl. Comput. Eng.* **2024**, *31*, 150–156. [CrossRef]
20. Januszewski, M.; Kornfeld, J.; Li, P.H.; Pope, A.; Blakely, T.; Lindsey, L.; Maitin-Shepard, J.; Tyka, M.; Denk, W.; Jain, V. High-precision automated reconstruction of neurons with flood-filling networks. *Nat. Methods* **2018**, *15*, 605–610. [CrossRef] [PubMed]
21. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
22. LeCun, Y. Generalization and network design strategies. *Connect. Perspect.* **1989**, *19*, 143–155.
23. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
24. Ma, J.; He, Y.; Li, F.; Han, L.; You, C.; Wang, B. Segment Anything in Medical Images. *arXiv* **2023**, arXiv:2304.12306.
25. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101. [CrossRef].
26. Zhang, Y.; Liao, Q.; Ding, L.; Zhang, J. Bridging 2D and 3D segmentation networks for computation-efficient volumetric medical image segmentation: An empirical study of 2.5D solutions. *Comput. Med. Imaging Graph.* **2022**, *99*, 102088. [CrossRef]
27. Xia, Y.; Xie, L.; Liu, F.; Zhu, Z.; Fishman, E.K.; Yuille, A.L. Bridging the Gap between 2D and 3D Organ Segmentation with Volumetric Fusion Net. In Proceedings of the Medical Image Computing and Computer Assisted Intervention—MICCAI 2018, Granada, Spain, 16–20 September 2018; Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G., Eds.; Springer: Cham, Switzerland, 2018; pp. 445–453.
28. Zheng, H.; Qian, L.; Qin, Y.; Gu, Y.; Yang, J. Improving the slice interaction of 2.5D CNN for automatic pancreas segmentation. *Med. Phys.* **2020**, *47*, 5543–5554. [CrossRef]
29. Huang, Y.; Yang, X.; Liu, L.; Zhou, H.; Chang, A.; Zhou, X.; Chen, R.; Yu, J.; Chen, J.; Chen, C.; et al. Segment Anything Model for Medical Images? *arXiv* **2023**, arXiv:2304.14660.
30. Mazurowski, M.A.; Dong, H.; Gu, H.; Yang, J.; Konz, N.; Zhang, Y. Segment anything model for medical image analysis: An experimental study. *Med. Image Anal.* **2023**, *89*, 102918. [CrossRef]
31. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.