Universiteit Leiden

Master Computer Science

Aorta segmentation and its geometric analysis

Name:	Priya Prabhakar
Student ID:	S3292983
Date:	13/03/2024
Specialisation:	Artificial Intelligence
Daily supervisor:	Dr. Samy Abo Seada
1st supervisor:	Mike de Boer
2nd supervisor:	Dr. Lu Cao

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The planning of thoracic endovascular aortic repair (TEVAR) can be enhanced with thorough computed tomography (CT)-based image analysis for finding the geometric features of the aorta, to avoid hostile proximal landing zones (PLZs) for endograft deployment. Saitta et al. [1] have proposed an automated analysis approach using aortic image segmentation and zone-based geometric analysis. The aim of our work was to reproduce their work using different modified methods. We compared two CNN-based architectures, namely a 3D U-Net and UNETR for Aortic segmentation from CT images. The 3D U-Net model exhibited the highest dice score of 0.931, while UNETR achieved a score of 0.897. Subsequently, we tested the models on 10 additional subjects, successfully mapping PLZs on three subjects, including two with standard arches and one with a CILCA arch. The segmentation with PLZs mapping was validated on three subjects against manual measurements. We calculated geometric parameters for the arch and individual zones, achieving an average error of less than 10% for all the parameters except the tortuosity index. For the tortuosity index, the average error was 18.78%. Our automated tool helps clinicians obtain reproducible PLZs mapping along with the geometric parameters from the automatic segmented aorta to avoid hostile zones.

Contents

1 Introduction

Thoracic endovascular aortic repair (TEVAR) is a minimally invasive procedure used to treat aneurysms and various other pathologies in the thoracic aorta. This procedure employs a stent graft, a fabric-covered metal tube, which serves to prevent an aneurysm from rupturing. In contrast to traditional open surgery, TEVAR offers the advantages of reduced recovery times and potentially improved survival rates [2]. Successful TEVAR implementation relies on accurate measurements of Ishimaru's proximal landing zones (PLZ) [3].

Though TEVAR is typically preferred for treating descending thoracic aortic pathologies, placing the endograft within the aortic arch poses surgical challenges due to the inherent risks of medium and long-term complications, such as endoleak, endograft migration, and collapse because of its complex structure [1]. A 90% incidence of endoleak was linked to the highly tortuous PLZs. Therefore, the computation of geometric parameters like tortuosity index and tortuosity angle is crucial along the criteria of minimum length of 20 mm and diameter of 40 mm as it quantifies the twisting and bending along the path. This information can help to avoid hostile regions and reduce the risk for endoleak after thoracic endovascular aortic repair [4].

For the computation of these geometric parameters, external software is typically used, and prior knowledge is also necessary. To streamline the whole process, Saitta et al. [1] implemented an automated pipeline for aorta segmentation and geometric analysis. Saitta et al. [1] used U-Net for the segmentation, mapped proximal landing zones, and computed various geometric parameters for the thoracic arch and PLZs. This automated approach eliminates the requirement for prior anatomical knowledge while ensuring accessibility across a wide range of user backgrounds. In this thesis, our goal is to improve this pipeline through the addition of comparison analysis and the use of various implementation strategies for further process.

To achieve this, we explored and compared various U-Net based convolutional neural network architectures for the aorta segmentation. U-Net based architectures were chosen due to their widespread use and success in both computer vision and medical imaging domains. Subsequently, our method computes the geometric parameters for both the aortic arch and landing zones, providing a comprehensive analysis of the anatomical structure.

The remainder of this thesis is organized as follows: Chapter 2 delves into the essential background information. Chapter 3 explores related work in the field. In Chapter 4, we discussed the methods employed for aorta segmentation and geometric analysis. Chapter 5 outlines the experiments conducted during the course of this research. Finally, Chapter 6 provides a comprehensive summary of the findings and suggests potential directions for future research.

2 Background/Preliminaries

This section lays down the foundation for the methodologies used in our research. Initially, we introduce Thoracic Endovascular Aortic Repair (TEVAR), highlighting its importance and requirements. Subsequently, we discuss different Convolutional Neural Network (CNN) architectures utilized for aorta segmentation from CT images. Additionally, we outline the geometric parameters crucial to our analysis.

2.1 Clinical background

In this subsection, we discuss the aorta and Thoracic Endovascular Aortic Repair (TEVAR), which is a substitute for open surgery. We also highlight the importance of mapping the proximal landing zones (PLZs) and geometric analysis of each zone for the successful TEVAR procedure.

2.1.1 Aorta and Thoracic endovascular aortic repair (TEVAR)

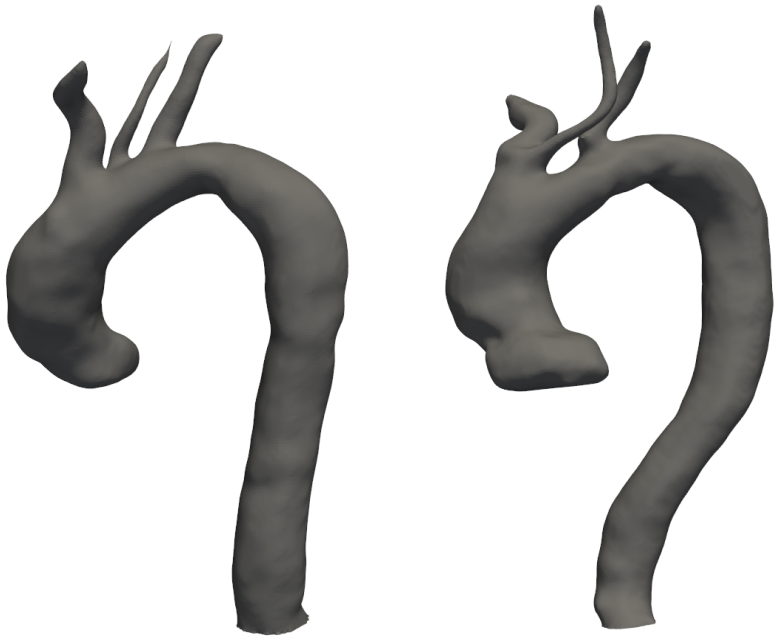
The aorta is the largest artery in the human body and it is involved in the circulation of oxygen and distribution of nutrients throughout the body. It originates from the left ventricle of the heart [5]. The aorta is split into 2 segments: abdominal and thoracic aorta. The thoracic aorta is further divided into the following three sections[6]:

- Ascending aorta: The ascending aorta is the first part of aorta. It begins at the sinotubular junction, where it meets the root, and continues towards the arch after the pericardium.
- Aortic arch: The standard aortic arch curves over the pulmonary trunk's bifurcation and the left pulmonary artery. It has three branches: the brachiocephalic artery, the left common carotid artery, and the left subclavian artery. The brachiocephalic artery supplies blood to head, neck, arm and chest wall while the left common carotid and left arteries supply on the left side. Compared to standard arch, CILCA arch exhibits a unique origin where both brachiocephalic and left carotid arteries are combined as shown in Figure 1. This thesis will center around Aortic arch.
- Descending aorta: The descending aorta originates at the branching point of the left subclavian artery and then it extends into abdominal region.

Aortic aneurysm is a bulging in the wall of the aorta. Aortic aneurysm increases the risk of developing a tear in the inner layer of the aorta's wall, known as an aortic dissection. Other risks include the potential rupture of the aneurysm, and surgical intervention is critical for addressing both conditions.
Thoracic endovascular aortic/aneurysm repair (TEVAR) is the non-invasive procedure for aortic repair and involves the deployment of a stent graft within the aorta relieving pressure on the damaged vessel wall by redirecting blood flow, providing treatment without direct operation [7].
While TEVAR has shown reduced early morbidity and mortality compared to open surgical repair, manipulating a diseased aorta with endovascular devices still poses significant risks [8]. Complications such as endoleak, endograft migration, and collapse are notable concerns, particularly in the aortic arch, which has unique geometric features. For TEVAR in the aortic arch, careful consideration of additional geometric parameters such as tortuosity, angulation, and specific length and diameter requirements are crucial to identify viable sites for endograft [9].

2.1.2 Ishimaru's proximal landing zones

A crucial prerequisite for the deployment of endograft is the selection of a sufficiently long segment of healthy aorta to serve as the landing site. Therefore, a standardized anatomical landing zone map, illustrated in Figure 2, is used. This map delineates zones Z0 to Z3 in the aortic arch, employing lines drawn distally to each arterial branch from the arch. This standardized approach provides a



(a) Standard arch (b) CILCA arch

Figure 1: Thoracic Aortic arch

clear framework, aiding in the accurate and consistent identification of landing zones across different cases [3].

2.2 Aorta segmentation

The foundational elements of aorta segmentation from CT images are covered in this section.

2.2.1 CT images

CT images are acquired using a computerized x-ray imaging technique and the focused x-ray beam rapidly rotates around a patient's body. The cross-sectional images, or slices, that are created by this procedure are stacked together to present the 3D view. Compared to conventional X-rays, these tomographic images provide physicians with more detail and aid in the identification of important structures and possible problems.
The resulting CT image serves as a map of tissue attenuation, with quantitative values established using the Hounsfield unit (HU) scale. In this scale, the air is assigned -1000 HU, water is 0 HU, and dense bone is +1000 HU [10].

2.2.2 CNN architectures

For the proposed automated approach, these CT scans must be segmented in order to obtain the binary map of the aorta for the geometric analysis. Automatic segmentation has made great advances in improving CNN architectures. The building blocks of CNNs are the following:

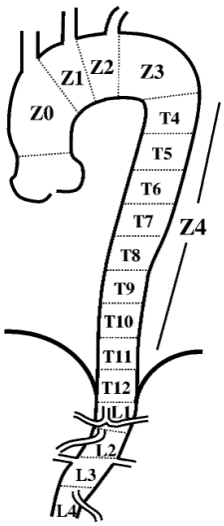


Figure 2: Anatomical landing zone map of aorta produced from [3]

- Convolution layer: A convolutional layer helps in detecting local features from the previous layer by convolving various filters or kernels over the input image. Each filter captures distinct local features over the image which helps the network to recognize intricate patterns and representations within the input data. The weights of these kernels are updated during training to minimize the training loss. However, there is a trade-off between the number of filters and the image size.

- Activation layer: The activation function plays a crucial role in determining whether a neuron should be activated and enabling backpropagation. Its primary purpose is to introduce non-linearity into the output of a neuron which facilitates backpropagation of errors and allows for learning complex relations between convolution layers.
  One widely used activation function is the Rectified Linear Unit (ReLU). Implemented primarily in the hidden layers of neural networks, its equation is denoted as f(x) = max(0, x), where it outputs x if x is positive and 0 otherwise. Its ability to become 0 is important as useless hidden layers stop contributing toward prediction.

- Pooling layer: The max pooling operation identifies the maximum output within a rectangular neighborhood. This layer serves a dual purpose: it reduces the spatial size of the feature map and eliminates redundant spatial information. This reduction is particularly advantageous for enhancing translation and scaling invariance, making the network robust to small shifts and distortions. By discarding less crucial spatial details, the max pooling layer allows for an increase in the number of filters in deeper convolutional layers without a proportional rise in the computational load per layer. This feature contributes to the overall efficiency and effectiveness of the convolutional neural network.

- Batch and instance Normalization: In the training process, the distribution of feature maps undergoes changes due to the parameter updates, posing challenges for the filter weights to effectively learn and adapt. This phenomenon is commonly referred to as covariate shift. To address this issue, batch normalization or instance normalization is used in normalizing

both activation output and gradients. This normalization process enhances the stability and efficiency of the learning process.
The mean and standard deviation for each batch are computed to determine batch normalization. In instance normalization, each batch sample is normalized independently.

The CNN architectures that we have used in this thesis are:

3D U-Net: Within biomedical imaging segmentation, the U-Net has become very popular due to its simple encoder-decoder structure that analyzes the entire image and progressively generates a full-resolution segmentation [11]. The 3D U-Net is designed to process 3D volumes as input, employing specialized 3D operations such as 3D convolutions, 3D max pooling, and 3D up-convolutional layers. However, this ability to take advantage of interslice context may improve performance at the expense of increased processing costs due to more parameters employed by these 3D convolutional neural networks (CNNs).

No New U-net (nnU-Net): The nnU-Net is a segmentation method based on the U-Net architecture that configures itself for each specific task including preprocessing, network architecture design, training, and post-processing. Specifically developed for medical image segmentation, nnU-Net addresses the variability in medical image properties resulting from factors like imaging techniques, scanner variations, and anatomical differences. nnU-Net adapts itself to a specific task by using a set of fixed parameters, interdependent rules, and empirical decisions. By consolidating these design elements into a unified framework, nnU-Net enhances the segmentation workflow, improving efficiency and accessibility across a wide range of applications and datasets [12].

UNEt TRansformers (UNETR): Transformer-based models consistently achieved state-of-the-art benchmarks across diverse Natural Language Processing (NLP) tasks using a self-attention mechanism. It emphasizes important aspects within sequences of words, empowering the model to learn intricate relationships and dependencies. Transformers have also impacted the field of computer vision. It does so by substituting the interrelations of words with relations of image patches. The U-Net Transformers (UNETR) [13] incorporates a transformer as the encoder and a conventional U-net decoder. The transformer enables the learning of sequence representations for the input volume, capturing global multi-scale information effectively. Subsequently, the U-net decoder maps these features to the full resolution. The direct connection between the transformer encoder and the decoder, facilitated by skip connections at different resolutions, computes the final semantic segmentation output. The transformer blocks within the architecture encompass multi-head self-attention (MSA) and multilayer perceptron (MLP) sublayers [14][15].

- Multihead attention network: An attention function is defined as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is the weighted sum of the values of dimension d_v, with the weights determined by the query and the corresponding key. The weights are described as the softmax function on the dot product of query and keys of dimension d_k, divided by $\sqrt{d_k}$ as given in the below equation [14].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The concept of Multi-head Attention involves the parallel execution of an attention mechanism multiple times. The individual outputs from these independent attention processes are subsequently concatenated and linearly transformed to achieve the desired dimension. This multi-head approach enables the model to collectively attend to information from various representation subspaces at different positions.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O$$

where each head is computed as $\text{head}_i = \text{Attention}(QW_{Qi}, KW_{Ki}, VW_{Vi})$

The input volume is divided into uniform patches, flattened, and then projected embedding space of K dimension using a linear layer. Learnable positional embeddings are introduced to retain crucial spatial information.

- Multi-Layer Perceptron (MLP): The MLP constitutes the second stage in the transformer block. This consists of two linear transformations with a ReLU activation function in between.

2.3 Hyperparameters

Hyperparameters are parameters set by the user before training that are not learned during the training process. The selection of these hyperparameters is crucial as they significantly impact the overall learning progress and performance. The following hyperparameters were experimented with within this thesis:

2.3.1 Learning rate

Learning rate is an important hyperparameter that guides the adjustment of weights based on the loss gradient during the gradient descent process. It essentially determines the size of each step taken in the direction of minimizing the loss, thereby influencing the convergence of the neural network toward optimal weights. Choosing optimal value of learning rate is important as a very large value may cause the model to overshoot the optimal solution. Conversely, if the learning rate is too low, it may take a long time to converge or become stuck in a local minimum.

2.3.2 Loss function

The loss function is a differentiable function that quantifies the difference between the target values and predicted values. It is used to compute gradients for weight updates during backpropagation. The choice of loss function is determined by the problem to be solved and input data. The choice of an appropriate loss function is important, as the network tries to minimize its value to achieve optimal performance.

- Dice Loss: The Dice Loss is the Dice Similarity Coefficient subtracted from 1 and is generally used for segmentation purposes. The Dice Similarity Coefficient (Dice score) is a metric used to find the overlap between prediction and target image. It is represented by the following equation:

$$\text{Dice Loss} = 1 - \frac{2 \times (\text{predicted\_values} \cap \text{target\_values})}{(\text{predicted\_values} \cup \text{target\_values}) + \epsilon}$$

- Binary Cross Entropy: It quantifies the disparity between ground truth values and predicted probabilities, penalizing incorrect predictions. Logarithmic values are used as they provide a lesser penalty for small differences, while larger differences incur a higher penalty. This approach ensures that the severity of the penalty aligns with the magnitude of the prediction error.

  It is calculated as:

$$BCE(x, y) = -\frac{1}{N}\sum_{i=1}^{N}[x_i \cdot \log(y_i) + (1 - x_i) \cdot \log(1 - y_i)]$$

  where x is the ground truth, y is the predicted output and N is the number of elements in the vectors.

- Focal Loss: Focal Loss deals with class imbalance during training by introducing a modulating term to the cross-entropy loss. The scaling factor in focal loss as shown in the below equation, diminishes the impact of easily classified examples during training and swifts the model's attention towards challenging instances[16].

$$\text{Focal Loss} = -(1 - y)^{\gamma} \cdot \log(y)$$

  where y is the predicted probability for the correct class, γ is the scaling factor.

2.4 Geometric analysis

2.4.1 Morphological operation: skeleton

Skeletonization is a method that reduces binary images to 1-pixel-wide representations. The algorithm iteratively sweeps over the image, removing pixels at each iteration until no further changes occur. This iterative technique requires two steps: first, a list of candidates for removal, and then progressively rechecking pixels from this list to ensure that the connectivity of the image is maintained [14].

2.4.2 Marching cube

The Marching Cubes algorithm generates a triangular mesh that represents an isosurface based on the volumetric data. The method entails "marching" through a subdivided 3D region, using each cube generated by the voxels as a vertex. It determines if a triangular surface intersects a particular cube, focusing on cases in which some voxels have a value of 1 and others have a value of 0, indicating the cube's interior. To achieve this, the algorithm relies on a predefined lookup table, facilitating efficient computation and surface triangulation. In this thesis, marching cube was used to generate a surface mesh from segmented aorta.

2.4.3 Geometric parameters

In addition to length and diameter, geometric parameters such as angulation and tortuosity values are critical for avoiding locations with sharp twists and excessive angulation in order to avoid post-operative problems. These parameters play an important role in preventing a hostile environment

and associated problems such as leakage and endograft failure.
Tortuosity is a measure of vascular curvature that is evaluated using factors such as the tortuosity index and angle of tortuosity. The tortuosity index is the ratio of actual path length to shortest path length and the angle of tortuosity is the extent of vessel curvature. Similarly, angulation is assessed using the beta angle, which measures the degree of vessel curvature. Furthermore, the ratio between the radius of curvature of the centerline and the outer surface aids in identifying maximum bulges on the vessel surface.
By considering these parameters alongside length and diameter measurements, clinicians can better assess the suitability of zones for the TEVAR procedure, mitigating the risk of complications.

2.4.4 The Vascular Modeling Toolkit (VMTK)

In our implementation, we utilized the VMTK library for computing geometric parameters. VMTK is a comprehensive set of libraries and tools focused on 3D reconstruction, geometric analysis, and surface data analysis. We specifically use VMTK to generate the centerline and tube, which are critical for mapping PLZs. The centerline computation in VMTK entails determining the center of the maximum inscribed spheres from the source to the target point. Similarly, VMTK renders the surface using maximum inscribed spheres for the given centerline for tube generation.

3 Related work

In the assessment and planning of thoracic endovascular aortic repair (TEVAR), precise identification of landing zones (LZs) through computed tomography (CT) images based geometric analysis is crucial. Saitta et al. [1] employed a fully automated pipeline utilizing a convolutional neural network (CNN) with a 3D U-Net architecture. This CNN automatically segments the thoracic aorta, detects proximal landing zones (PLZs), and quantifies relevant geometric features for TEVAR planning. The pipeline was trained on 395 CT scans with manual segmentations and tested on 70 scans. The resulting metrics, including aortic arch centerline radius of curvature, PLZs maximum diameters, angulation, and tortuosity, were statistically analyzed. The trained CNN demonstrated a mean Dice score of 0.95 and successfully generalized to nine pathological cases of thoracic aortic aneurysm, providing accurate segmentations. Similarly, other studies have implemented aorta segmentation using CNN.
Noothout et al. [17] introduced an automated approach for segmenting the ascending aorta, aortic arch, and thoracic descending aorta in low-dose chest CT scans without contrast enhancement. They used a dilated convolutional neural network (CNN) on axial, coronal, and sagittal 2D image slices. Their method achieved a Dice score of 0.83 ± 0.07, 0.86 ± 0.06, and 0.88 ± 0.05 for the ascending aorta, aortic arch, and descending aorta, respectively.
Chen et al. [18] proposed a segmentation method for cardiac aorta in CT images and MRI using the XR-MSF-U-Net model. The XR-MSF-U-Net integrates an X ResNet (XR) convolution module and a Multi-scale features fusion module (MSF), enhancing feature extraction efficiency and capturing global, local, and spatial features. In comparison to the benchmark U-Net model, the XR-MSF-U-Net demonstrates a significant improvement of 7.99% in the dice similarity coefficient.
Lin et al. [19] introduced a Geometry-Constrained Deformable Attention Network (GDAN) to learn common aortic features by interacting with anatomical context information. The deformable

attention extractor adjusts patches adaptively, accommodating various aortic shapes, while the self-attention mechanism explores long-range dependencies in CT sequences. The geometry-constrained guider simplifies morphological representation, imposing strong constraints on geometric boundaries and aiding the correct extraction of semantic features. Testing on a dataset of 204 aortic CT cases, including normal aortas, and aortic dissections, the GDAN method achieves a mean dice similarity coefficient of 0.943.

4 Method

4.1 Pipeline implementation

We have developed an automated pipeline that integrates a convolutional neural network (CNN) for aorta segmentation, followed by the identification of landing zones and automatic computation of geometric parameters for both the aortic arch and its individual zones. The pipeline as in Figure 3 comprises several key processing steps:

- **Aorta segmentation from CT scan images:**
  Utilize a U-Net based deep learning model for aorta segmentation, fine-tuning hyperparameters, and comparing its performance with other models.
- **Triangulated surface mesh generation and centerline extraction:**
  Generate triangulated surface meshes for the segmented aortic regions and compute centerline representing the course of the aorta in 3D.
- **Landmark detection and landing zone mapping:**
  Develop an algorithm for identifying Ishimaru's proximal landing zones Z0, Z1, Z2, and Z3. Proper selection of the landing zone is crucial for the TEVAR procedure.
- **Parameter computation of each zone and aortic arch:**
  Calculate the arc length, angulation, maximum diameter, and tortuosity angle for each landing zone, and determine the outer radius of curvature, centerline radius of curvature, and centerline tortuosity index for the aortic arch.

4.2 Aorta segmentation

This subsection section delves into the dataset used for training, its pre-processing methodologies, and the subsequent training procedure. Furthermore, the training procedure, encompassing the choice of algorithms and hyperparameter tuning, and validation strategies are also discussed to provide a comprehensive understanding of the model training.

4.2.1 Dataset, pre-processing and augmentation

Dataset: We obtained 3D CT scans of 100 subjects from Erasmus MC for MedicalVR's viewer. The same CT scans were utilized to train the neural network for segmentation. In addition, 75 CT scans from the Medical Segmentation Decathlon dataset [20] were also used. All CT images used for the training process adhered to the NIfTI (Neuroimaging Informatics Technology Initiative) format [21].

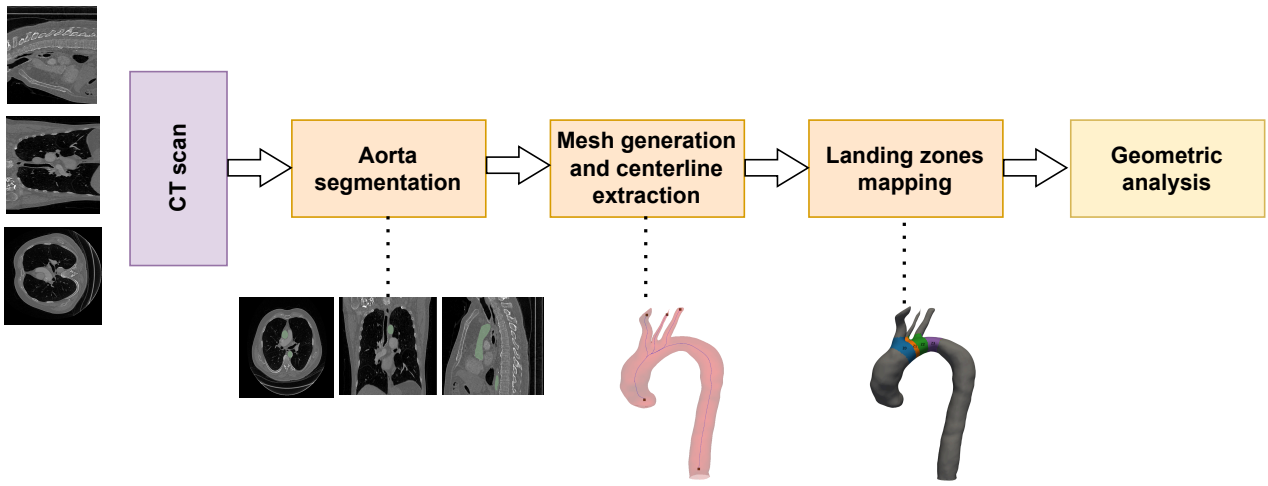


Figure 3: Schematic illustration of the pipeline.

Pre-processing: The pixel spacing in our acquired CT scans demonstrated variability, ranging from 0.29×0.29 to 0.976×0.976 mm^2, with slice thickness spanning from 0.29 to 1.5 mm. To establish a uniform spatial resolution across all scans, facilitating effective learning of spatial semantics by the neural network, we resampled CT images to a standardized pixel spacing of 0.75 mm and a thickness of 1 mm. These values were determined as the median pixel size and thickness within the dataset, ensuring a consistent and representative basis for the resampling process.

Augmentation: Augmentations on images are performed as shown in Figure 4 to expand the dataset and prevent overfitting during training, as models are prone to overfitting when trained on a very limited number of samples . With augmentation, the dataset was expanded to 1225 images. [22]. In our case, the following augmentations were used using TorchIO library [23]:

- Random Anisotropy: This operation involves downsampling an image along an axis and subsequently upsampling it to its initial space. This mimics the effect of an image acquired with anisotropic spacing. It is downsampled using nearest neighbor interpolation for downsampling and upsampled using 'linear' or 'nearest' interpolation.
- Random Flip: This operation introduces a random reversal of element order within an image along the specified axes, namely 'AP' (Anterior-Posterior) and 'IS' (Inferior-Superior) in this case.
- Random Affine: The Random Affine transformation is a specialized operation that upholds collinearity among points. It is the composition of a linear transformation and translation of the input image of X. The output image Y is represented as:

$$Y = Ax + B$$

where A is an invertible matrix and Ax represents linear transformation for scaling and translation is achieved by adding matrix B.
In this case, the scale range was 0.9-1.3, and a maximum rotation of 10 degrees.

- Random Elastic deformation: It introduces random displacements to a coarse grid of control points both around and inside the image. The displacements are interpolated using cubic B-splines, efficiently simulating realistic deformations. This transformation enables the network to learn invariance to deformations, a crucial aspect in biomedical segmentation. Maximum displacement values are set to 15, 10, and 5 in our implementation.

- Randum Blur: Utilizing a random-sized Gaussian filter, this operation introduces blur to the image.

- Random Noise: Gaussian noise is added randomly to the image with random parameters sampled from a normal distribution, with mean 0 and standard deviation 0.05 in our implementation.

- Random Gamma: This transformation randomly adjusts the contrast of an image by raising its values to the power of γ. The gamma correction formula is expressed as

$$I_{\text{corrected}} = I_{\text{original}}^{\gamma}$$

  where $I_{\text{corrected}}$ represents the corrected image intensity, I_{original} represents the original image intensity and γ represents the gamma value.

4.3 Manual segmentation

To create a dataset's ground truth for training a neural network in aorta segmentation, we manually segmented 100 CT volumes using 3D Slicer [24]. Subsequently, these 100 CT images, along with their acquired ground truth were used to train the nnU-Net [12]. The trained model was then applied to predict segmentation for an additional set of 75 images. Each prediction underwent manual refinement to achieve accurate segmentation results.

4.4 Different model architecture

4.4.1 3D U-Net Architecture

Similar to the conventional U-Net architecture, the described model follows a structure comprising an encoder and a decoder path, each with four resolution steps. The network takes a 128x128x128 voxel tile of the image as input. In the encoder path, each layer incorporates two $3 \times 3 \times 3$ convolutions, succeeded by a rectified linear unit (ReLU), and then a $2 \times 2 \times 2$ max pooling with strides of two in each dimension. Notably, batch normalization precedes each ReLU activation. Conversely, in the decoder path, each layer involves an upconvolution of $2 \times 2 \times 2$ with strides of two in each dimension, followed by two $3 \times 3 \times 3$ convolutions, each succeeded by a ReLU. Decoder path also includes skip connections from layers of equal resolution in the encoder path. These skip connections play a key role in recovering spatial information lost during downsampling. In the final layer, a $1 \times 1 \times 1$ convolution is employed to reduce the number of output channels to match the number of labels. This architectural design allows the model to effectively capture and reconstruct complex spatial features in the input data [11].

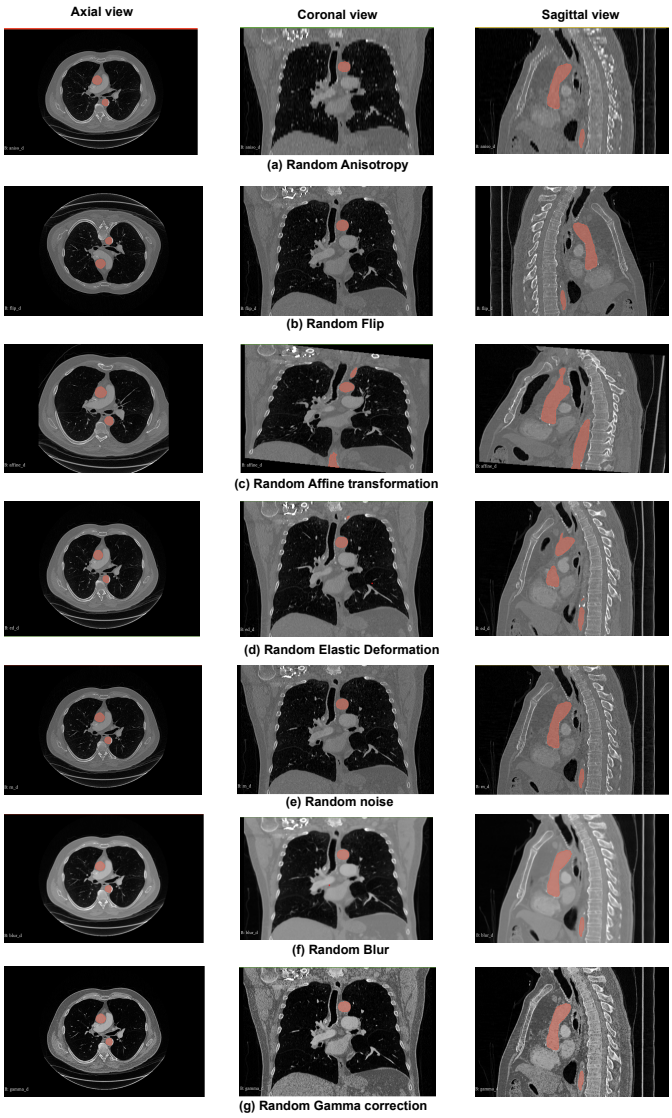


Figure 4: Different augmentation techniques

4.4.2 UNETR Architecture

The input image size is $H \times W \times D$ and the patch size for the transformer is denoted as $P \times P \times P$. In the UNETR architecture, a series of 12 transformer blocks are employed, and the outputs from the 3rd, 6th, 9th, and 12th transformers are reshaped to $\frac{H \times W \times D}{P^3} \times K$, where K represents the embedding size of the transformer. At each resolution step, the reshaped tensors from the embedding space are translated back into the input space through consecutive $3 \times 3 \times 3$ convolutional layers followed by normalization layers. Subsequently, the resized feature map is concatenated with the previous transformer output, and these are fed into two $3 \times 3 \times 3$ convolutional layers. The output is then upsampled using a deconvolutional layer. This is similar to the decoder path of 3D U-Net. In the final layer, $1 \times 1 \times 1$ convolutional layer with a softmax activation function is used to generate function to generate segmented predictions [13].

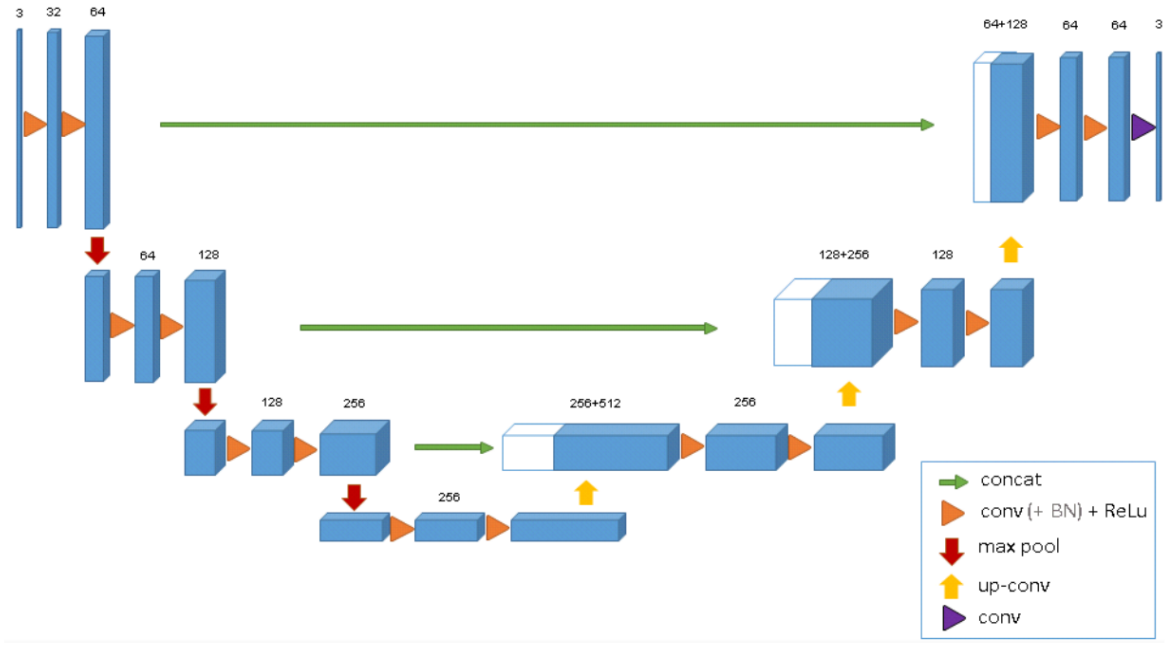


Figure 5: 3D U-Net architecture[11]

4.5 Training and evaluation

The dataset with 175 images was augmented and split into training and validation sets with a splitting ratio of 0.2. The patch size was set at 128 × 128 × 128 pixels, with a corresponding batch size of 2. The training was implemented using PyTorch and trained on VIDIA A6000s GPU for up to 10 epochs. Each epoch consisted of one iteration across 60 patches for each CT image. Subsequently, 10 scans were reserved for testing purposes. The ADAM optimizer was used for the training process. A Dice score is used for the evaluation. Several experiments were conducted to determine the optimal hyperparameters:

1. Learning rate: In our implementation, we conducted model training experiments with learning rate values of 0.01, 0.0001, and 0.00001, with the aim of identifying the most suitable learning rate for optimal performance. Binary cross-entropy was used in this experiment as loss function. This empirical approach allowed us to fine-tune the learning rate, ensuring effective convergence and enhancing the overall efficiency of the training process.

2. Loss: The function targeted for minimization or maximization is referred to as the objective function or criterion. In the case of minimization, it is alternatively termed the cost function, loss function, or error function. In our case, we will be dealing with different loss functions. The selection of an appropriate loss function is crucial, as attaining its minimum value directly corresponds to optimal performance for the specified objective. Learning rate of 0.00001 is used in this experiment.

3. Normalization method: We replaced the batch normalization layer with an instance normalization layer in the U-Net architecture. We conducted a thorough comparison between the

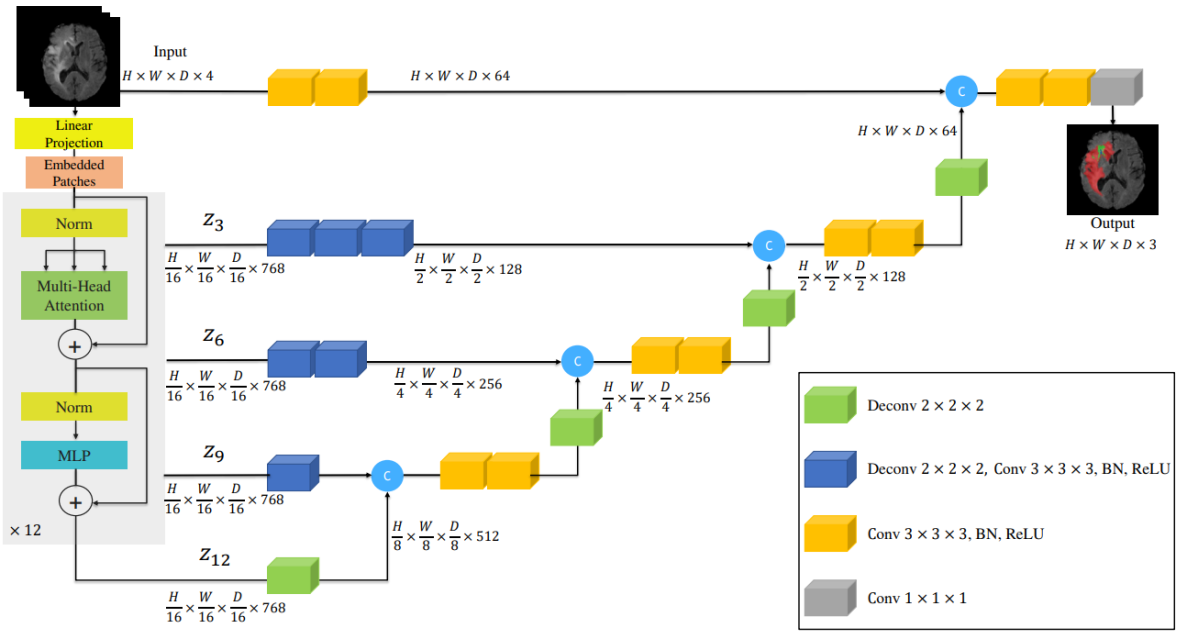


Figure 6: UNETR architecture with patch size as $16 \times 16 \times 16$ and embedding size as 768 [13]

original 3D U-Net and the 3D U-Net incorporating instance normalization. A learning rate of 0.00001 and Binary cross-entropy as a loss function were used in this experiment.

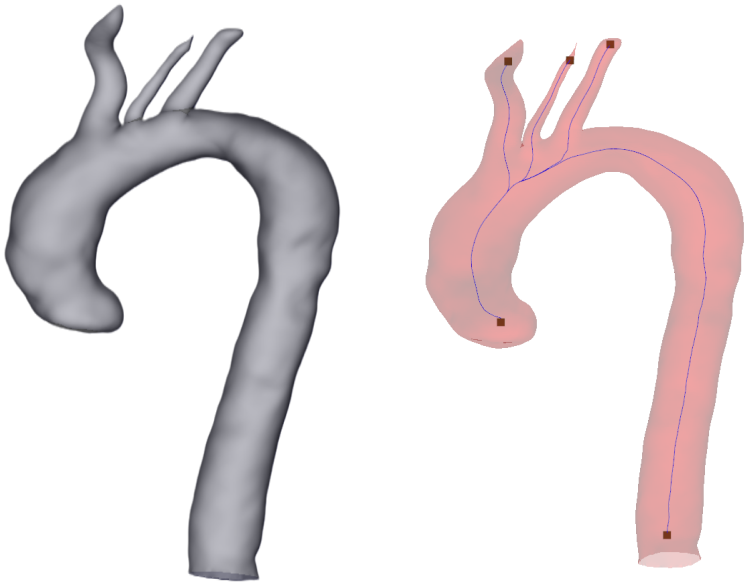
4. Model architecture: A comparison between the performance of 3D U-Net and UNETR for aorta segmentation was conducted. A learning rate of 0.00001 and Binary cross-entropy as a loss function were used in this experiment.

4.6 Geometric analysis

Geometric parameters are computed for the main aortic arch and for the identified PLZs.

4.6.1 Centerline implementation and mesh generation

The surface mesh, as shown in Figure 7a, is obtained using the marching cubes algorithm [25], which used the aortic segmentation obtained using 3D U-Net. We used the VMTK library to compute the centerline. However, source and target points are required for centerline computation with VMTK. Source and target points together are referred to as endpoints. To get endpoints, we found the centerline using the morphological skeleton operation first. The centerline is then transformed into a graph to determine the endpoints. A refinement phase is used since the skeleton process may discover more endpoints than necessary, resulting in the detection of undesired endpoints. During the refinement phase, the length of each identified node's edge is calculated. Nodes with edge lengths less than 12 are systematically deleted, and the surviving nodes are the final endpoints as shown in Figure 7b. These endpoints served as source points and target points for centerline computation using VMTK [26].



(a) Mesh (b) Centerline

Figure 7: Representation of segmented aorta (a) Aortic mesh generated using the marching cubes algorithm from the obtained segmentation (b) Aortic mesh and centerline obtained using VMTK library, highlighting its end points

4.6.2 Proximal landing zones (PLZs) mapping

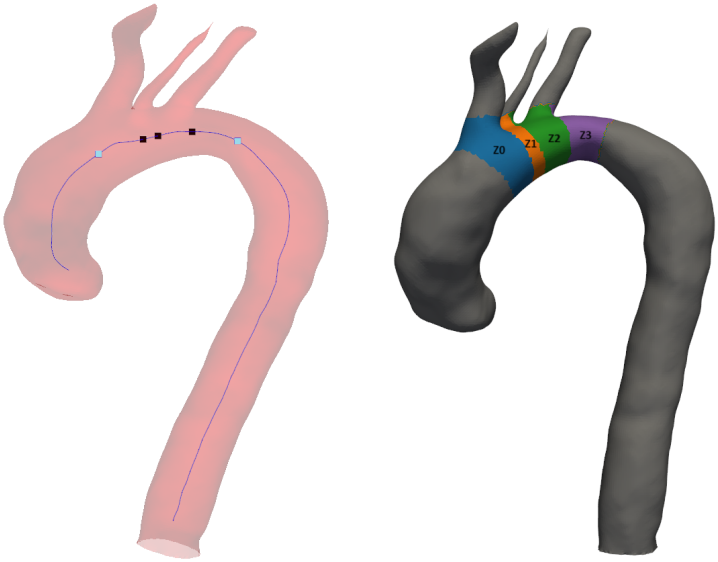
Landmark points are initially identified for the mapping of landing zones by locating the intersections between the tubes generated for the aortic branches and the centerline of the aortic arch. For every centerline connected to the aortic arch branches, a tube surface is generated using the VMTK library, as illustrated in Figure 8. A tube is a centered envelope whose cross sections are made up of circles whose local radii match the radius of the largest sphere that can be inscribed inside the vessel. The algorithm determines the locations of intersection between tubes that correspond to the aortic branches and the aortic centerline. In Figure 9a, these intersection points are represented by black points and the aortic centerline by blue line. The other two landmark points are also automatically positioned at a distance of 20 mm along the first landmark point proximally and the last intersection point distally, as indicated by the blue points in Figure 9a. The selection of a 20mm separation ensures that each zone maintains a minimum length of 20mm which is required for TEVAR procedure[27]. These points are then used to identify the PLZs (ref Figure 9b).

(a) Tube surface for the brachiocephalic trunk (blue)

(b) Tube surface for the left carotid artery (yellow)

(c) Tube surface for the subclavian artery (green)

Figure 8: Tube surface representations for the aorta



(a) Landmark points

(b) Landing zones

Figure 9: Landing zones mapping (a) Landmark points represented by black and blue points (b) Landing zones: Z0 in blue, Z1 in yellow, Z2 in green, and Z3 in purple

4.6.3 Geometric parameters

Whole-Arch Metrics: The centerline of the entire aortic arch is defined as the centerline stretching from the initial point (Point A) to the terminus at Point B, situated on the same axial plane as Point A (refer to Figure 10). The radius of curvature represents the radius of the circle that best encapsulates the centerline's trajectory. This process of calculating the radius of curvature involves centering all data points around the origin through the subtraction of their mean. Subsequently, Singular Value Decomposition (SVD) is used to find the normal vector of the fitting plane. These points are then projected onto a 2D plane using Rodrigues rotation. The radius of the circle fitting these transformed 2D coordinates yields the radius of curvature for the centerline. Concurrently, the outer curvature path is identified by first finding out the point in the Dijkstra path from Point A to Point B. The outer curvature path is then defined by mirroring these Dijkstra path points with respect to the centerline. In Figure 10, R_c signifies the radius of curvature of the yellow circle encapsulating the centerline, while R_o signifies the radius of curvature of the brown circle encapsulating the outer curvature path.
The ratio R_o/R_c describes potential irregularities in the outer boundary of the arch. The tortuosity index is computed using the VMTK library and it quantifies these sharp bends [28].

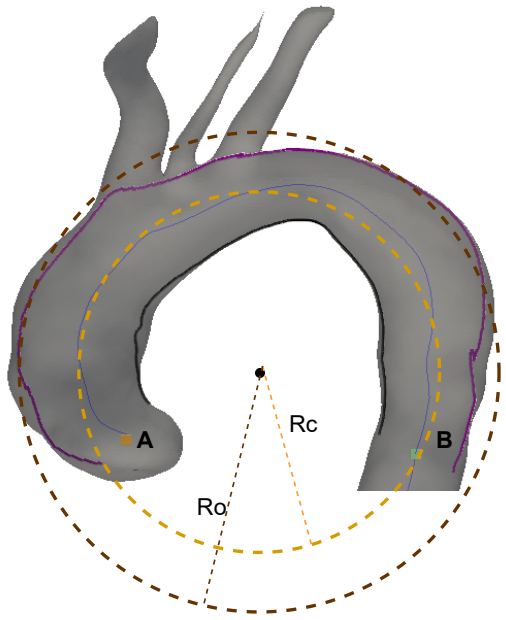


Figure 10: Illustration of the aortic arch anatomy for radius of curvature calculation

Zone-based metrics: Zone-based metrics involve the computation of arc length along the centerline, maximum diameter, β angle, and tortuosity angle. The determination of maximum diameters within each zone relies on identifying the diameters of the largest inscribed spheres.
To calculate the β angle, the most proximal point (p0) of the zone and another point (p40), positioned 40mm from p0, are identified. The β angle is then defined as the angle formed by the tangent vectors to p0 and p40 as shown in Fig.11. It is calculated as the inverse cos of the ratio of the dot product of the two tangent vectors and its magnitudes

Similarly, for the tortuosity angle, two points are selected: point (p20), located 20mm from p0 along the centerline, and point (p40), situated 40mm from p0 along the centerline. The tortuosity angle is precisely characterized as the angle formed by the vectors connecting p0 to p20 and p20 to p40 as shown in Fig.11. The tortuosity angle is calculated as the inverse cos of the ratio of dot products of the two vectors and its magnitude. This approach ensures a comprehensive geometric evaluation of zone-specific metrics, offering insights into arc length, maximum diameter, β angle, and tortuosity angle for each zone.

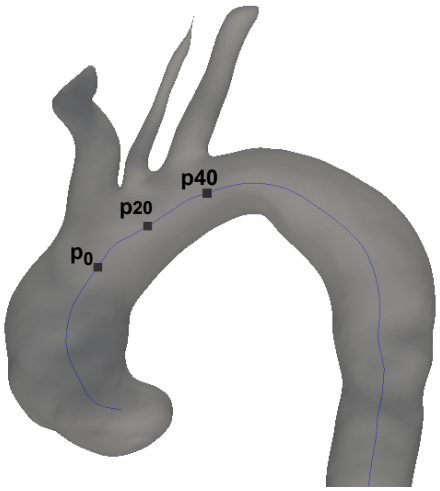


Figure 11: Illustration of the aortic arch anatomy for tortuosity angle and β angle calculation

4.6.4 Validation of geometric parameters

We used a 3D slicer tool to validate the results obtained from our method for calculating geometric parameters for the aortic arch and each PLZs. Since the 3D slicer lacks built-in direct functions, all parameters were computed approximately using line, circle, and angle tools, which took a long time to compute. Saitta et al. [1] used 3Mensio Vascular 8.0 software instead of a 3D slicer for validation purposes. Because it is not free software, we utilized a 3D slicer to validate the results.

5 Results

In this section, we report the results of all of the aorta segmentation experiments, as well as the geometric analysis output for three segmented aortas.

5.1 Aorta segmentation

In this subsection, we present the results of all the experiments conducted for aorta segmentation.

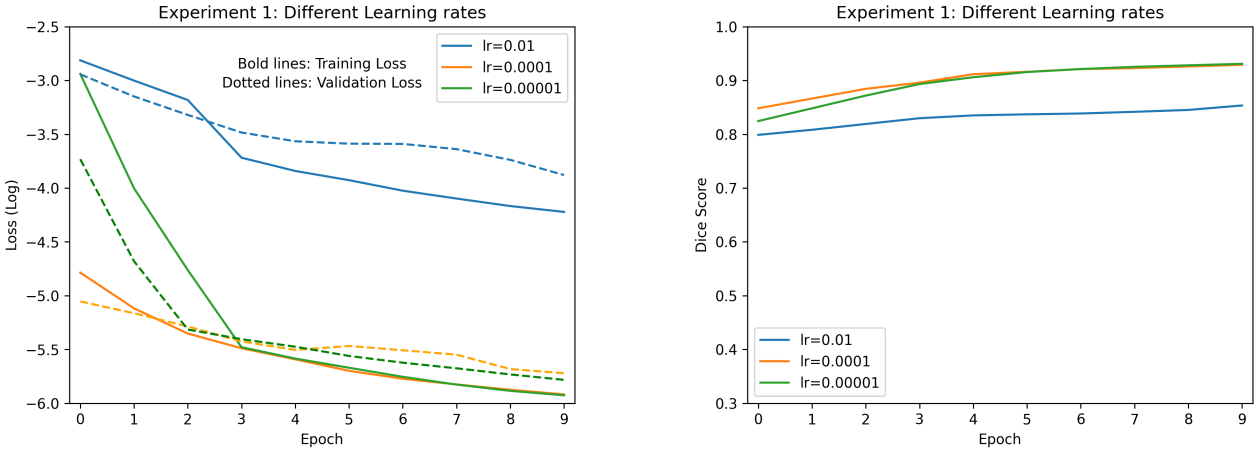
5.1.1 Different learning rate

Table 1 shows that the maximum Dice score of 0.931 was observed for 0.00001 and lowest for 0.01, respectively. Figure 12a demonstrates that for learning rate 0.01, both training and validation loss

were highest throughout 10 epochs. In case of learning rates, at epoch 0 both losses were lowest for 0.0001. However, for learning rate 0.00001, a significant dip was seen in training and validation loss at epoch 3. By 5 epochs, both training and validation losses surpassed those observed for 0.0001 marginally. Consequently, the difference in Dice scores was also marginal by the end of 10 epochs, with 0.931 achieved for 0.0001 and 0.928 for 0.00001 as in Figure 12b.

S.No	Leaning rate	Dice score
1	0.01	0.856
2	0.0001	0.928
3	0.00001	0.931

Table 1: Dice score for different learning rate



(a) Training loss

(b) Validation accuracy

Figure 12: Performance of 3D-Unet model with varying learning rates for aortic segmentation

5.1.2 Different loss functions

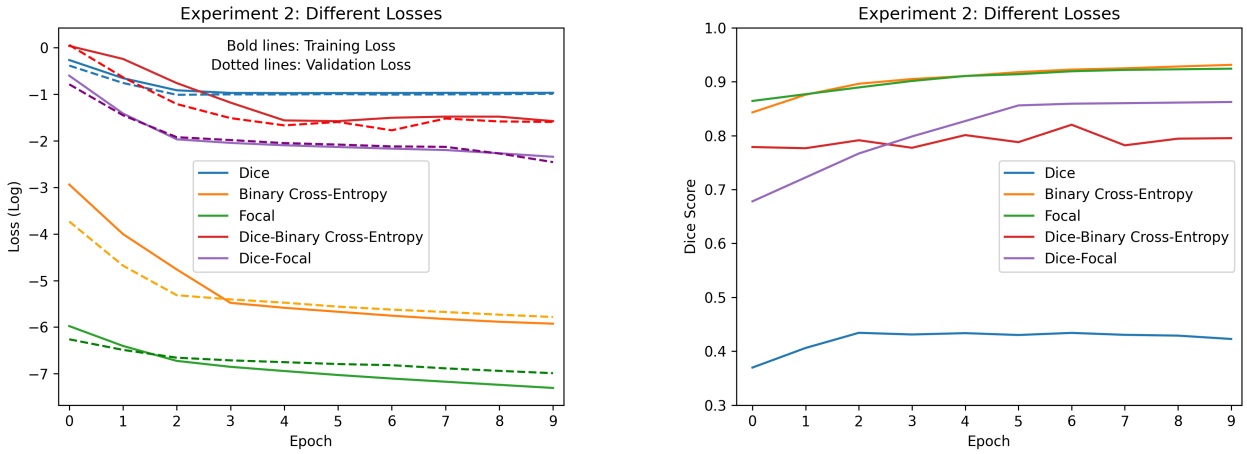
Table 2 shows that cross-entropy achieved the highest Dice score of 0.931 followed by a Dice score of 0.92 achieved with Focal loss, whereas Dice loss resulted in the lowest score of 0.462. Furthermore, the combined Dice-Binary cross-entropy and the Dice-Focal loss had Dice scores of 0.813 and 0.847, respectively.

From Figure 13a, it can also be observed that the training and validation losses plateaued at a constant level after only 3 epochs with the Dice loss, indicating premature cessation of learning. In contrast, for other loss functions, a gradual decrease in both training and validation loss was observed which indicates the learning process without any over-fitting.

Figure 13b further demonstrates that Binary cross-entropy attains a Dice score of 0.931 within the initial 10 epochs, surpassing the peak values achieved by other loss functions. No significant improvement in Dice score was observed for Dice loss compared to other losses which was reflected in Figure 13a also.

S.No	Loss	Dice score
1	Dice	0.462
2	Binary Cross entropy	0.931
2	Focal	0.92
2	Dice-Binary Cross entropy	0.813
3	Dice-Focal	0.847

Table 2: Dice score for different losses



(a) Training loss

(b) Validation Dice Score

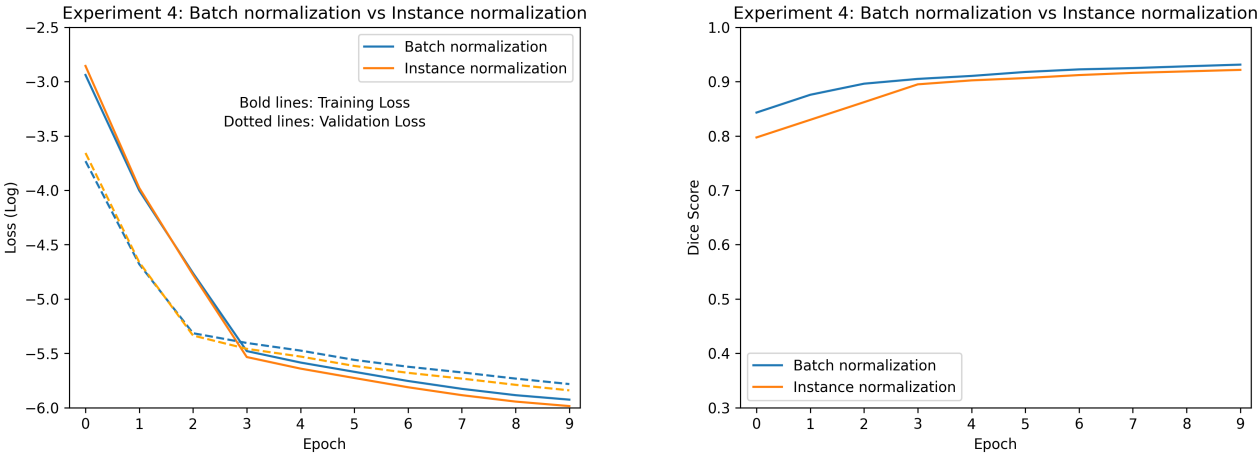
Figure 13: Performance of 3D U-Net model with varying Loss functions for aortic segmentation

5.1.3 Batch normalization vs instance normalization

Table 3 and Figure 14b indicate that when batch normalization is substituted by instance normalization, there is a marginal decline in validation Dice score after 10 epochs of training. The same observation is reflected in the training and validation loss. However, during the initial 3 epochs, instance normalization exhibits higher losses compared to batch normalization, indicating a slower learning pace for instance normalization. Despite this initial disparity, the subsequent performance demonstrates comparable performance between the two normalization techniques.

S.No	Normalization type	Dice score
1	Batch normalization	0.931
2	Instance normalization	0.926

Table 3: Dice score for Batch normalization and instance normalization



(a) Training loss

(b) Validation Dice Score

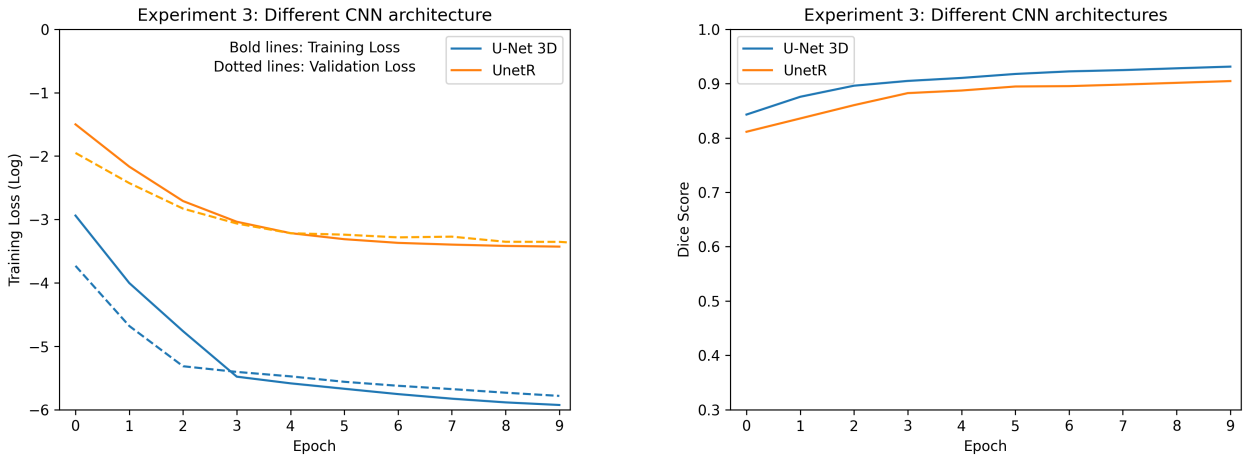
Figure 14: Performance of 3D U-Net model with different normalization for aortic segmentation

5.1.4 Different CNN architectures

Table 4 illustrates the performance metrics of both 3D U-Net and UNETR, with a Dice score of 0.931 for 3D U-Net and 0.89 for UNETR. From Figure 15a, it can be observed that over the span of 10 epochs, both 3D U-Net and UNETR exhibit a decline in training and validation losses with a similar trend, indicating the absence of over-fitting. However, throughout all 10 epochs, 3D U-Net consistently demonstrates lower loss values compared to UNETR, implying potentially more pronounced learning dynamics.
Consequently, Figure 15b shows that the Dice score for 3D U-Net consistently surpasses that of UNETR, suggesting a greater degree of overlap between segmented and ground truth data in the case of 3D U-Net. This underscores the efficacy of 3D U-Net in achieving superior segmentation performance compared to UNETR.

S.No	Model	Dice score
1	3D U-Net	0.931
2	UNETR	0.897

Table 4: Dice score for different CNN architectures



(a) Training loss

(b) Validation Dice Score

Figure 15: Comparison of different CNN architectures

5.2 Geometric analysis

The trained model was then used to segment 10 test cases. Out of these 10 cases, we present the results for geometric analysis of 3 CT images.

5.2.1 Aortic arch

Table 5 and Table 6 represent the results of the geometric analysis of 3 cases, out of these 3 cases 1 is of the CILCA arch and the other 2 cases are of the standard arch. For the aortic arch, the outer curvature radius and centerline curvature radius ratio are computed. Its ratio and tortuosity index were then computed to characterize the geometric parameters of the arch. The results are comparable to the values obtained using a 3D slicer, with an average error of 8.9% for the ratio of outer and centerline curvature radius and 18.78% for the tortuosity index.

S.No	Arch Metric	Our method	Reference	Error
CT image 1 (Standard arch)				
1	Centerine curvature radius (mm)	48.26	46.30	1.96
2	Outer curvature radius (mm)	60.34	63.29	2.95
3	Outer and centerline curvature radius ratio	1.42	1.36	0.06
4	Tortuosity index	1.25	1.63	0.48
CT image 2 (Standard arch)				
1	Centerine curvature radius (mm)	51.42	49.23	2.19

Continued on next page

Table 5 – continued from previous page

S.No	Arch Metric	Our method	Reference	Error
2	Outer curvature radius (mm)	62.71	65.3	2.59
3	Outer and centerline curvature radius ratio	1.22	1.32	0.1
4	Tortuosity index	1.99	1.75	0.34
CT image 3 (CILCA arch)				
1	Centerine curvature radius (mm)	70.36	66.8	3.56
2	Outer curvature radius (mm)	87.45	89.7	2.25
3	Outer and centerline curvature radius ratio	1.54	1.34	0.2
4	Tortuosity index	1.24	1.57	0.33

Table 5: Aortic arch metric (For 3 CT images)

S.No	Arch Metric	Average error (%)
1	Centerine curvature radius (mm)	4.74
2	Outer curvature radius (mm)	3.56
3	Outer and centerline curvature radius ratio	8.9
4	Tortuosity index	18.78

Table 6: Aortic arch metric (Average error (%))

5.2.2 PLZs

For the standard arch, PLZs (zone 0-3) are identified and for the CILCA arch, due to common origin of the innominate and left carotid artery, there is no zone 1 and therefore, zone 0 and zones 2-3 are computed. For each zone, arc length, maximum diameter, tortuosity angle, and β angle are computed as shown in Table 7 and Table 8. The results are comparable to the values obtained using 3D slicer, with maximum average error of 5.87% for arc length, 5.74% for max diameter, 8.41% for tortuosity angle and 6.57% for β angle.

S.No	Zone Metric	Zone	Our method	Reference	Error
CT image 1 (Standard arch)					
1	Arc length (mm)	Zone 0	20	-	-
		Zone 1	6.59	6.325	0.265
		Zone 2	15.06	15.77	0.71
		Zone 3	20		-
2	Maximum diameter (mm)	Zone 0	30.59	31.31	0.72
Continued on next page					

Table 7 – continued from previous page

S.No	Zone Metric	Zone	Our method	Reference	—Error—
		Zone 1	24.01	25.07	1.06
		Zone 2	22.5	21.87	0.63
		Zone 3	19.93	20.12	0.19
3	Tortuosity angle (°)	Zone 0	12.84	13.1	0.26
		Zone 1	23.71	25.7	1.99
		Zone 2	29.03	32.1	3.07
		Zone 3	34.18	37.9	3.72
4	β angle (°)	Zone 0	34.17	34.61	0.44
		Zone 1	43.96	45.8	1.84
		Zone 2	58.48	55.3	0.82
		Zone 3	57.29	59.3	2.01
CT image 2 (Standard arch)					
1	Arc length (mm)	Zone 0	20	-	-
		Zone 1	5.55	6.01	0.46
		Zone 2	28.65	29.13	0.48
		Zone 3	20	-	-
2	Maximum diameter (mm)	Zone 0	31.83	30.91	0.82
		Zone 1	23.99	22.32	1.67
		Zone 2	22.59	23.87	1.28
		Zone 3	19.02	20.91	1.89
3	Tortuosity angle (°)	Zone 0	9.81	10.12	0.31
		Zone 1	24.28	25.77	1.49
		Zone 2	25.58	24.39	1.19
		Zone 3	26.92	28.81	1.89
4	β angle (°)	Zone 0	20.36	21.54	1.18
		Zone 1	16.88	19.32	2.44
		Zone 2	19.19	21.78	2.59
		Zone 3	28.14	29.13	0.99
CT image 3 (CILCA arch)					
1	Arc length (mm)	Zone 0	20	-	-
		Zone 1	-	-	-
		Zone 2	41.32	42.34	1.02
		Zone 3	20	-	-
2	Maximum diameter (mm)	Zone 0	46.2	47.31	1.11
		Zone 1	-	-	-
		Zone 2	32.39	33.27	0.88
		Zone 3	27.73	29.21	1.48
3	Tortuosity angle (°)	Zone 0	5.54	6.23	0.69
		Zone 1	-	-	-
		Zone 2	1.33	2.01	0.68

Continued on next page

Table 7 – continued from previous page

S.No	Zone Metric	Zone	Our method	Reference	—Error—
		Zone 3	24.18	25.11	0.93
3	β angle (°)	Zone 0	3.96	5.34	1.38
		Zone 1	-	-	-
		Zone 2	13.69	16.34	2.65
		Zone 3	37.70	38.92	1.22

Table 7: Zone metric (For 3 CT images

S.No	Zone Metric	Zone	Average error (%)
1	Arc length (mm)	Zone 0	-
		Zone 1	5.87
		Zone 2	2.51
		Zone 3	-
2	Maximum diameter (mm)	Zone 0	2.27
		Zone 1	5.74
		Zone 2	3.53
		Zone 3	5.04
3	Tortuosity angle (°)	Zone 0	4.28
		Zone 1	6.76
		Zone 2	8.41
		Zone 3	7.12
3	β angle (°)	Zone 0	4.88
		Zone 1	6.57
		Zone 2	6.48
		Zone 3	3.32

Table 8: Zone metric (Average error (%))

6 Discussion

In this section, we discuss the results from the previous section for aorta segmentation and geometric analysis.

6.1 Aorta segmentation

Aorta segmentation constitutes a critical component within our proposed automated pipeline. Prior to training, all 175 CT images underwent preprocessing and augmentation to expedite training duration. We attained the highest Dice score of 0.931 within 10 epochs, with approximately 2 hours spent per epoch as each epoch involved training 60 patches for every one of the 952 augmented images, resulting in training on 57,120 samples per epoch. In the paper [1], a superior Dice score of 0.95 was reported. This can be attributed to the dataset used as the 175 CT images in our dataset

are primarily intended for lung anatomy, thus possessing inherently low contrast. Conversely, the majority of images employed in the study by Saitta et al. [1] featured high contrast levels.
To achieve the Dice score of 0.931, we conducted various experiments to fine-tune the hyperparameters of the 3D U-Net model. We experimented with learning rates of 0.01, 0.0001 and 0.00001. It was observed that better performance was achieved with low learning rates (0.0001 and 0.00001) compared to higher learning rate value of 0.01. This may be due to the reason that there are large gradient updates with high values of learning rate which led to overshooting the optimal solution due to large gradient updates. However, when the learning rate was set to 0.00001, it required more time to converge to a lower training and validation loss value compared to a learning rate of 0.0001. As the learning rate decreases, it takes a longer time to converge due to smaller gradient updates. However, after 10 epochs, it achieved a higher Dice score than that achieved by the learning rate of 0.0001.
After determining the optimal learning rate of 0.00001, we conducted experiments with different loss functions. Focal loss and Binary cross-entropy loss demonstrated superior performance, yielding Dice scores of 0.92 and 0.931, respectively, outperforming Dice loss and their combination. Dice loss exhibited the lowest Dice score, which can be attributed to its inherent volatility, particularly in scenarios with a small positive class in either the prediction or ground truth, potentially resulting in issues like exploding gradients. However, upon combining Dice loss with focal loss and Binary cross-entropy leading to a substantial improvement in the Dice score. This might be due to more stable and differentiable function.
Additionally, we explored the impact of using instance normalization instead of batch normalization in the 3D U-Net architecture. While instance normalization is commonly used in many CNN architectures, we found that batch normalization showed marginally better results for aorta segmentation. This can be due to the reason that batch normalization helps to maintain the consistency of the activation function across batches, mitigates the influence of outliers, and helps prevent overfitting.
With a learning rate of 0.00001 and binary cross-entropy loss, we conducted a comparison between the 3D U-Net and UNETR CNN architectures. Ideally, hyperparameters should be optimized separately for each architecture; however, for the sake of comparison, we maintained the same hyperparameters. The results indicate that within 10 epochs, UNETR achieved a Dice score of 0.89, while the 3D U-Net achieved a higher Dice score of 0.931. This can be attributed to the complex structure and incorporation of transformers in the UNETR architecture, which may require more training data and epochs to achieve optimal performance compared to the 3D U-Net.

6.2 Geometric analysis

The reliability of our automated tool was validated through comparison with geometric measurements obtained using 3D Slicer, yielding comparable results. Analyzing the geometric parameters of the aortic arch across all three CT images, the outer and centerline curvature radius ratios and tortuosity indices were found to be comparable. However, for the CILCA arch, both centerline curvature and outer curvature values were higher, indicating differences in the trajectory compared to the standard arch.
Regarding the geometric parameters of the proximal landing zones (PLZs) across the three images, it is evident that zone 3 can be considered the most tortuous zone due to its high tortuosity angle and β angle. The high tortuosity angle and β angle in zone 3 suggest a high degree of bending in the aorta, which is often associated with post-surgery TEVAR failures. This finding aligns with

previous studies [29], which have consistently associated zone 3 as hostile zone. It can also be observed that zone 1 in both images with a standard arch exhibited the shortest length, measuring less than 20mm. This insufficient length may not provide the required space for deploying the stent effectively.

7 Limitations

The CNN training for aorta segmentation was implemented using 175 CT images obtained from patients with no aorta-related issues. However, the performance of the model on pathological CT images remains to be evaluated. Given that the dataset consisted mostly of images with low contrast, it's likely that the Dice score can be enhanced by training the model on a larger and more diverse set of CT images.
Furthermore, the approach involving aorta segmentation followed by geometric analysis was tested on a few Ct images and its applicability in a clinical scenario needs to be thoroughly assessed by medical professionals. Validating the significance of this approach in clinical practice requires testing on pathological CT images and careful evaluation by experts to ensure its accuracy and reliability in real-world settings. Only through such rigorous testing and validation can the effectiveness and relevance of the proposed method be established for clinical use.

8 Future work

The following improvements can be done as part of future work:

1. **Extended Training**: Increasing the number of training epochs beyond 10 could potentially enhance the performance of the segmentation model. Additional epochs allow the model to learn more intricate features and patterns within the data, leading to improved segmentation accuracy.

2. **Expansion of Dataset**: Expanding the dataset to include pathological cases as well can make the segmentation model more robust and applicable to a wider range of clinical scenarios. Pathological cases often present unique challenges and variations that can further refine the model's segmentation capabilities.

3. **Exploration of Different Architectures**: Experimenting with various CNN architectures, such as UNET++, can provide insights into which architecture best suits the segmentation task. Each architecture may capture different aspects of the anatomy and exhibit varying levels of performance.

4. **Transfer Learning**: Leveraging pre-trained models through transfer learning can accelerate the training process and potentially improve segmentation accuracy, especially when dealing with limited training data. Fine-tuning a pre-trained model on the specific segmentation task can help exploit features learned from a large dataset.

5. **Hyperparameter Optimization**: Employing automated machine learning-based methods like grid search, random search, or Bayesian optimization for hyperparameter tuning can

streamline the process of finding optimal hyperparameters. This approach saves time and computational resources while systematically exploring the hyperparameter space.

6. **Geometric Analysis Enhancement**: Including the calculation of the location of the maximum diameter as part of the geometric analysis can provide valuable insights into the morphology of the aorta and aid in clinical decision-making.

9 Conclusion

The main objective of this research was to develop an automated approach for aorta segmentation and its geometric analysis to aid in the pre-planning of the TEVAR procedure. The primary focus was on aorta segmentation, for which we employed the 3D U-Net architecture and fine-tuned the hyperparameters. Additionally, we compared its performance with the UNETR architecture. The 3D U-Net model, trained with 175 CT images, was then evaluated on 10 test images. Subsequently, centerlines were computed from the segmented data, and different Proximal Landing Zones (PLZs) were identified. Various geometric parameters were computed for the aortic arch and each zone. These geometric parameters were computed for three CT images, two featuring a standard arch and one with a CILCA arch. From the results, it can be observed that our proposed approach provides comparable geometric parameters to those obtained using 3D Slicer.

References

[1] S Saitta, F Sturla, A Caimi, A Riva, MC Palumbo, E Votta, A Redaelli, and MM Marrocco-Trischitta. A deep learning-based and fully automated pipeline for thoracic aorta geometric analysis and tevar planning from computed tomography. *European Heart Journal-Cardiovascular Imaging*, 22(Supplement_1):jeaa356–251, 2021.

[2] Davy Cheng, Janet Martin, Hani Shennib, Joel Dunning, Claudio Muneretto, Stephan Schueler, Ludwig Von Segesser, Paul Sergeant, and Marko Turina. Endovascular aortic repair versus open surgical repair for descending thoracic aortic disease: a systematic review and meta-analysis of comparative studies. *Journal of the American College of Cardiology*, 55(10):986–1001, 2010.

[3] Shin Ishimaru. Endografting of the aortic arch. *Journal of endovascular therapy*, 11(6_suppl): II–62, 2004.

[4] Takuya Ueda, Hiroyuki Takaoka, Bhargav Raman, Jarrett Rosenberg, and Geoffrey D Rubin. Impact of quantitatively determined native thoracic aortic tortuosity on endoleak development after thoracic endovascular aortic repair. *AJR-American Journal of Roentgenology*, 197(6): W1140, 2011.

[5] Thomas C Gasser. Aorta. *Biomechanics of Living Organs*, pages 169–191, 2017.

[6] Bartosz Rylski, Camila Muñoz, Friedhelm Beyersdorf, Matthias Siepe, Diana Reser, Thierry Carrel, Florian Schoenhoff, Christian Schlensak, Mario Lescan, Hans-Henning Eckstein, et al. How does descending aorta geometry change when it dissects? *European journal of cardio-thoracic surgery*, 53(4):815–821, 2018.

[7] I Akin, S Kische, TC Rehders, H Schneider, H Ince, and MD Ca Nienaber. Tevar. *Herz*, 36(6): 539, 2011.

[8] Sheena W Chen, Kyongjune B Lee, Michael A Napolitano, Alejandro E Murillo-Berlioz, Anna P Sattah, Shawn Sarin, and Gregory Trachiotis. Complications and management of the thoracic endovascular aortic repair. *Aorta*, 8(03):049–058, 2020.

[9] David A Nation and Grace J Wang. Tevar: endovascular repair of the thoracic aorta. In *Seminars in interventional radiology*, volume 32, pages 265–271. Thieme Medical Publishers, 2015.

[10] *Principles of computed tomography*. Elsevier eBooks, 2022. doi: 10.1016/b978-0-12-822960-6.00144-7.

[11] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016.

[12] Fabian Isensee, Paul F Jaeger, Simon AA 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):203–211, 2021.

[13] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Richard E Turner. An introduction to transformers. *arXiv preprint arXiv:2304.10557*, 2023.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[17] Julia MH Noothout, Bob D De Vos, Jelmer M Wolterink, and Ivana Išgum. Automatic segmentation of thoracic aorta segments in low-dose chest ct. In *Medical Imaging 2018: Image Processing*, volume 10574, pages 446–451. SPIE, 2018.

[18] Weimin Chen, Hongyuan Huang, Jing Huang, Ke Wang, Hua Qin, and Kelvin KL Wong. Deep learning-based medical image segmentation of the aorta using xr-msf-u-net. *Computer Methods and Programs in Biomedicine*, 225:107073, 2022.

[19] Weiyuan Lin, Hui Liu, Lin Gu, and Zhifan Gao. A geometry-constrained deformable attention network for aortic segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 287–296. Springer, 2022.

[20] 'medical segmentation decathlon'. URL https://drive.google.com/drive/folders/1HqEgzS8BV2c7xYNrZdEAnrHk7osJJ--2.

[21] Sara Samuel, Michael Moore, Brandon Patterson, Helenmary Sheridan, and Chris Sorensen. Neuroimaging data primer: A resource for curating digital imaging and communications in medicine (dicom) and neuroimaging informatics technology initiative (nifti) files. 2020.

[22] Karen López-Linares, Maialen Stephens, Inmaculada García, Iván Macía, Miguel Ángel González Ballester, and Raúl San José Estepar. Abdominal aortic aneurysm segmentation using convolutional neural networks trained with images generated with a synthetic shape model. In *Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting: First International Workshop, MLMECH 2019, and 8th Joint International Workshop, CVII-STENT 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1*, pages 167–174. Springer, 2019.

[23] Augmentation. https://torchio.readthedocs.io/transforms/transforms/augmentation.html. URL https://torchio.readthedocs.io/transforms/transforms/augmentation.html. Accessed on January 14, 2024.

[24] Ron Kikinis, Steve D Pieper, and Kirby G Vosburgh. 3d slicer: a platform for subject-specific image analysis, visualization, and clinical support. In *Intraoperative imaging and image-guided therapy*, pages 277–289. Springer, 2013.

[25] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

[26] Lu Cao. Biological model representation and analysis. In *Ph. D. dissertation, Section imaging and bioinformatics*, 2014.

[27] Massimiliano M Marrocco-Trischitta, Hector W de Beaufort, Francesco Secchi, Theodorus M van Bakel, Marco Ranucci, Joost A van Herwaarden, Frans L Moll, and Santi Trimarchi. A geometric reappraisal of proximal landing zones for thoracic endovascular aortic repair according to aortic arch types. *Journal of vascular surgery*, 65(6):1584–1590, 2017.

[28] Yehuda G Wolf, Manfred Tillich, W Anthony Lee, Geoffrey D Rubin, Thomas J Fogarty, and Christopher K Zarins. Impact of aortoiliac tortuosity on endovascular repair of abdominal aortic aneurysms: evaluation of 3d computer-based assessment. *Journal of vascular surgery*, 34(4):594–599, 2001.

[29] Massimiliano M Marrocco-Trischitta, Hector W de Beaufort, Gabriele Piffaretti, Stefano Bonardelli, Mauro Gargiulo, Michele Antonello, Joost A van Herwaarden, Sara Boveri, Raffaello Bellosta, Santi Trimarchi, et al. The modified arch landing areas nomenclature predicts proximal endograft failure after thoracic endovascular aortic repair. *European Journal of Cardio-Thoracic Surgery*, 58(2):309–318, 2020.