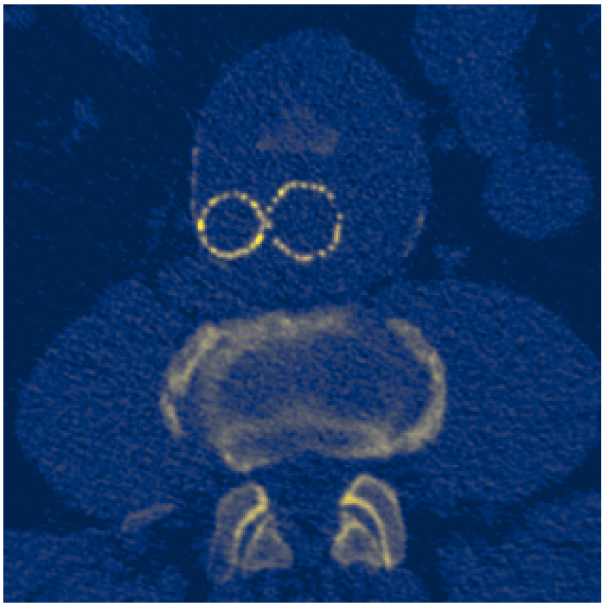


# A Multi-Stage Machine Learning Approach for Predicting Indicators of EVAR Stent Complications in CT Images

A Pilot Study on AI-Based Diagnosis of Post-EVAR Loss of Seal

SAGA FRISELL
KRISTOFFER GUSTAFSSON

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

# A Multi-Stage Machine Learning Approach for Predicting Indicators of EVAR Stent Complications in CT Images

A Pilot Study on AI-Based Diagnosis of Post-EVAR Loss of Seal

SAGA FRISELL
KRISTOFFER GUSTAFSSON


CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
*Division of Biomedical Engineering and Signal Processing*
Computer vision research group
Chalmers University of Technology
Gothenburg, Sweden 2025

A Multi-Stage Machine Learning Approach for Predicting Indicators of EVAR Stent Complications in CT Images
A Pilot Study on AI-Based Diagnosis of Post-EVAR Loss of Seal
SAGA FRISELL
KRISTOFFER GUSTAFSSON



Supervisor: Jennifer Alvén, Department of Electrical Engineering, Chalmers
Suvervisor: Håkan Roos, Department of Molecular and Clinical Medicine, Sahlgrenska Academy
Examiner: Ida Häggström, Department of Electrical Engineering, Chalmers

Master's Thesis 2025
Department of Electrical Engineering
Division of Biomedical Engineering and Signal Processing
Computer Vision Research Group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An axial CT slice from a patient with an EVAR stent.

Typeset in LaTeX
Gothenburg, Sweden 2025

# Abstract

Patients that have received endovascular aneurysm repair (EVAR) surgery regularly undergo computed tomography (CT) scans to monitor the placement of the inserted stent. The process of analyzing these 3D volumes is, however, difficult and requires specialized expertise. To improve the detection of complications, new clinical procedure has been developed to diagnose loss of seal, a common complication of EVAR surgery. The protocol consists of three steps: centerline definition, identification of stent ends, and measurement of the seal zone length. This project aims to employ a machine learning (ML) based approach to CT analysis in order to diagnose loss of seal, with annotations from the described protocol. The aim is achieved by predicting the seal zone length through a novel sequential approach, and a baseline. The sequential approach consists of two models in sequence, where the intermediate output is the stent endpoint locations. In order to use ML methods, a dataset of 143 patients and a total of 399 CTs was curated from unprocessed clinical data.

The results indicate that detecting the three stent endpoints with a transformer-based model is a promising first step towards diagnosis and could potentially automate one step in the clinical protocol. Regression of the seal zone length was, on the other hand, unsuccessful with the current model architecture and it was deemed a far more complex task. Several improvements could be made as future work, such as utilizing a transformer-based model for seal zone regression and predicting additional intermediate labels. The sequential approach has potential, but some steps could be tweaked to reach an accurate and generalizable method.



Keywords: Machine Learning, EVAR, Seal Zone, CT Analysis.

# Acknowledgements

We want to give special acknowledgment to our supervisors for their seemingly tireless dedication to this project. To begin with, we would like to thank Jennifer, for constantly being available on Slack to answer our technical questions. Furthermore, the project would have been dead in the water if not for her restarting the remote work computer whenever we accidentally ran out of memory. Håkan was a cornerstone of the project, providing expert-knowledge as the basis of this thesis. We would like to give additional appreciation to Håkan and his team of specialists for supplying annotations at a wrist-breaking pace. Finally, we are thankful for the collaboration between the supervisors at and outside our meetings, which has kept the project progressing smoothly. It has been truly inspiring working in such a dedicated, passionate team.

A small thanks to the Division for Computer Science at the University of Gothenburg for providing our makeshift office and tea.

Saga Frisell & Kristoffer Gustafsson, Gothenburg, June 2025

# List of Acronyms

Below is the list of acronyms that have been used throughout this thesis:

| | |
|---|---|
| AI | Artificial Intelligence |
| AUC | Area Under the Curve |
| CT | Computed Tomography |
| CNN | Convolutional Neural Network |
| DICOM | Digital Imaging and Communications |
| EVAR | Endovascular Aneurysm Repair |
| FCNN | Fully Convolutional Neural Network |
| ML | Machine Learning |
| MLP | Multi-layer Perceptron |
| ROI | Region of Interest |

# Contents

# 1

# Introduction

Abdominal aortic aneurysm is characterized by the aorta widening just below the kidneys and affects 2-3% of men aged 65-70 in Europe and the US [1]. When the disease progresses, the widening of the blood vessel walls can lead to a fatal rupture. A standard treatment, either preventative or immediately after rupture, is endovascular aneurysm repair (EVAR). The treatment is performed by delivering the components of a stent through openings in another blood vessel, expanding at the placement site. Once positioned, the stent is anchored in non-diseased blood vessels above and below the aneurysm through friction and small metal hooks [2], [3]. EVAR is preferred to open surgical repair as it is less invasive and has a lower 30-day mortality rate, but the rate of post-procedural complications is high, between 16% and 30% [2]–[4].

The stent graft redirects the blood flow by excluding it from the aneurysm, thereby reducing pressure and eliminating flow. Complications arise if flow is restored in the aneurysm, which may lead to new ruptures [3]. Due to the associated risk, patients who have received EVAR are surveilled post-operatively to detect stent anomalies, mainly by computed tomography (CT) imaging. While regular screening is standard practice, a study has shown that as many as 30% of treatment failures are undetected, and there is an overall lack of expertise at most hospitals [5].

The follow-up procedure is ineffective in preventing serious complications after EVAR since it focuses on the secondary and late effects of EVAR failure [6]. These secondary effects include aneurysm expansion and blood filling the aneurysm sac, called an endoleak, shown in Figure 1.1. Increasing evidence shows that focusing directly on stent graft sealing zones is more effective as a preventive follow-up method [6]. Loss of seal is a direct precursor of aneurysm rupture after EVAR. Additionally, patients with loss of seal have a rupture risk that approaches the risk of an untreated aneurysm. Identification of loss of seal therefore allows for targeted preventive treatment.

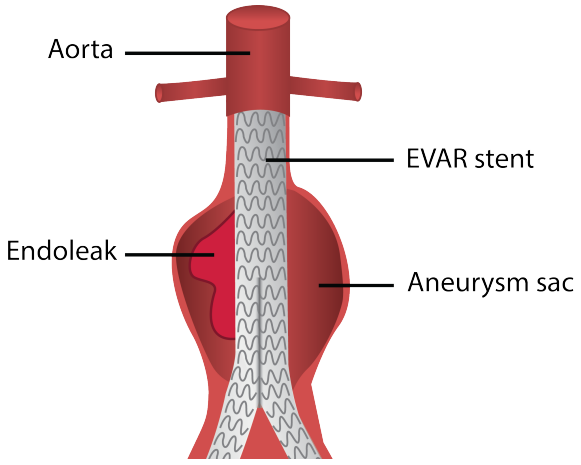



**Figure 1.1:** Illustration of the EVAR stent and the endoleak complication.

To standardize loss of seal diagnosis, a new clinical procedure has been developed to analyze CT images of post-EVAR patients, allowing earlier re-intervention [4]. The analysis consists of three steps: characterizing the stent centerline, finding the endings of the stent, and measuring various parameters of the stent, including the seal zone length. This method, developed by Sandström *et al.*, will henceforth be refered to as the protocol. Although the protocol offers valuable insights for diagnosis, it is time-consuming, requires specialized expertise to be followed appropriately, and is only available at a few hospitals.

## 1.1 Related Work

As an alternative to manual analysis of post-EVAR CT images, several studies have used machine learning (ML) to train models for detecting stent complications [5], [7]–[9]. A well-performing model by Talebi *et al.* focuses on endoleak detection with a U-net-based model [5]. The authors employ a model with a 2D image-based method where it segments the leaked blood in the aneurysm sac for each CT slice. Endoleak classification is then performed slice-wise, and all predictions for one volume are aggregated for the final classification. Their method performs well with the addition of an advanced data augmentation technique; it scores 95%, 90%, and 100% for accuracy, precision, and recall and has an area under the curve (AUC) of 99%. However, due to the limited dataset of 70 CT scans, the possibility that the model overfitted on the test data should be considered, especially since the authors did not raise any concerns about the issue.

Another model trained to detect aneurysm endoleak in EVAR patients used a similar method [7]. This method used a RetinaNet convolutional neural network (CNN) to detect bounding boxes of the aneurysm in each CT slice, and a ResNet-50 CNN to perform slice-based endoleak classification. The bounding boxes were aggregated into a 3-dimensional region of interest (ROI), and the aneurysm was segmented to calculate a few other measurements. Their model also performs well with an AUC of 94% for binary endoleak classification, indicating a stable and accurate approach. A

drawback to their method is that it relies on segmentation, which is time-consuming to create manual annotations for. While these two papers successfully employed machine learning for EVAR complication detection, they use 2D slices, which do not fully utilize the information stored in the three dimensions.

A recent study on AI-based post-EVAR analysis combined handcrafted and machine learning methods in a software that monitors patients over time for various complications [8]. While the exact methods are not described, the software has been implemented in a commercial medical device. Another successful approach predicted the risk of various complications, including endoleak, in pre-operative 3D CTs of patients [9]. While making a prognosis for the risk of endoleak is valuable, the paper fails to address how to analyze post-operative CTs.

The most prominent limitation in all these papers is the type of complication they focus on: endoleaks. Endoleak happens when blood flows outside the stent through the residual aneurysm sac. The leakage is usually a consequence of another fault with the stent, such as migration, breakage, or, most commonly, loss of seal. Focusing on endoleak is easier since contrast agents can make leaking blood visible in CT images, but detecting the complications might be delayed. Andersson *et al.* recently showed that it is possible to see loss of seal in CT images before endoleak occurs, allowing earlier re-intervention to prevent more fatal ruptures from occurring [6].

## 1.2 Aim

This thesis aims to build upon the procedure developed by Andersson *et al.*, including data analyzed and annotated by Håkan Roos, PhD, MD, and his team, to create a novel, machine learning-based approach for detecting loss of seal. Automating the process of post-EVAR analysis will eventually allow more hospitals to detect loss of seal early, even where specialized expertise is missing.

Loss of seal detection will be made by predicting the seal zone length of the EVAR stent attachment, an important metric in determining loss of seal. The length will be predicted using a sequential method where a model first predicts the seal zone locations in each CT. These locations will then define cropped patches used as inputs to the final model that predicts seal zone lengths, focusing on the relevant regions of the CT. A simple baseline CNN that takes the whole CT scan as input will also be used for comparison to justify the sequential approach. To use machine learning approaches, the provided clinical data will be curated into a dataset, ensuring further work can be done.

Finally, the aim is that the implemented ML approach should be interpretable by medical practitioners. Interpretability improves the efficiency of medical practitioners' workflow and enables insight into the decision-making process.

## 1.3 Limitations

Firstly, the data analyzed in this thesis has not previously been used for machine learning approaches. Therefore, the thesis is limited by the quality and quantity of the annotations made by the collaborating experts before the project starts. We expect substantial time to go into processing the data to formats that allow training the models.

Another limitation is made regarding the data. Each patient has one pre-operative scan and three consecutive post-operative scans. Exploring temporal forecasting is outside the scope of the thesis, so each scan from the same patient is treated independently, and the pre-operative CT is discarded. Additionally, the seal zone lengths are the only measurements from the protocol considered in this project. Furthermore, the thesis will only explore machine learning methods, neglecting any hand-crafted or statistical approaches to image analysis.

Hyperparameter tuning will be limited in scope, since it is a time and resource-intensive task. Therefore, hyperparameters will be chosen based on previous work and experience. Due to limited experimentation and parameter tuning, the thesis takes a proof-of-concept form, leaving room for further optimization.

Finally, the interpretability of the models will only be explored, given that they perform well. Realistically, poorly performing models are unlikely to be implemented in healthcare workflows, so human understanding of their inner workings is irrelevant.

# 2

# Background

This chapter will cover the necessary medical and technical background to the project. First, the medical images and the EVAR stent are described. Then, the essential parts of utilized machine learning methods are presented, including state-of-the-art models, data augmentation, and interpretability.

## 2.1 Medical Background

The medical background of the thesis are presented below, including the CT images and their data format, as well as details regarding the EVAR stent and the medical terminology used in this thesis.

### 2.1.1 Medical Image Data

Computed tomography (CT) is a reconstruction technique for X-ray images. The X-ray images are collected in a 180° field around the patient at multiple axial locations [10]. Afterwards, computation by filtered backprojection generates a high-contrast 3D image of the patient. The pixel intensities of the CT volume are given in Hounsfield units that are proportional to the X-ray attenuation coefficients of the materials they represent [10]. The lowest value of $-1000$ represents air, between $-1000$ and 200 represents various soft tissue and organs, and 300 to 2000 represents bone [11]. There is no upper limit to CT intensities, but the maximum value of common materials is around 3000. When visualizing a CT image, the lowest values are typically black, and the highest values are white. The difference in intensity between different tissues makes CT images useful for studying anatomy and is a common way to diagnose various diseases. A patient obtaining a CT scan can also have contrast agents injected that illuminate the blood, making it lighter than usual and thereby differentiating it from soft tissue.

CT images are usually stored as slices in the Digital Imaging and Communications in Medicine (DICOM) format, a medical image standard [12]. DICOM files contain the actual pixel values and additional information necessary to understand and modify the image, termed the image metadata. The information in the metadata includes the image orientation and spacing, the patient demographics, and the scanner model and settings.

In the DICOM format, slice positions are given in a real-world coordinate system, and if any annotations are made, their positions can be stored in the same coordinate system [13]. The medical coordinate system is always placed in the same way in relation to the patient, with each dimension as follows. The $x$-axis goes from the patient's left to right side, and the sagittal plane crosses it. The $y$-axis goes from the patient's front to their back, and the coronal plane is orthogonal. Finally, the $z$-axis goes from the patient's feet to their head, and the axial plane crosses it. The orientation of the planes and axes are further clarified in Figure 2.1 below.

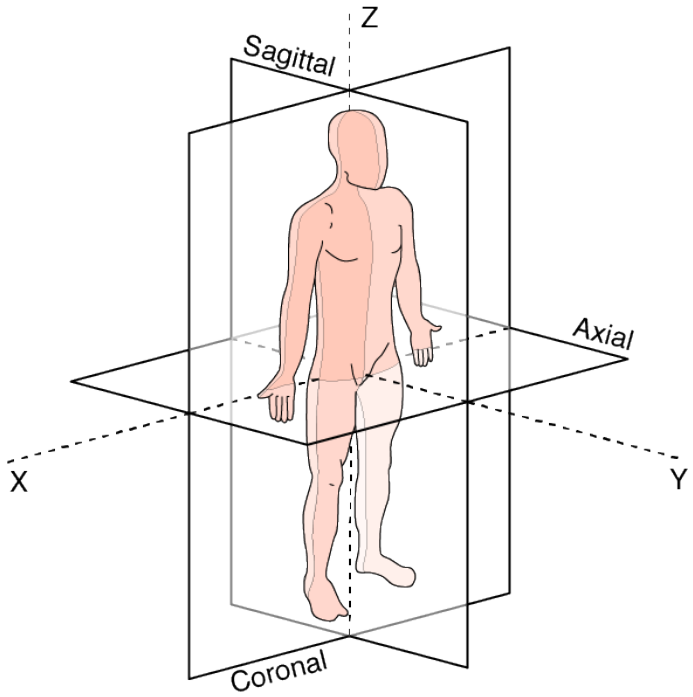



**Figure 2.1:** The anatomical planes and image axes. Source: modified Anatomical Planes by CFCF licensed under CC-BY-SA-3.0.

### 2.1.2 Endovascular Aneurysm Repair

The stent used for endovascular aneurysm repair is made of thin polyester fabric and a metal mesh [2]. The metal makes it highly visible in CT images as metal has a high Hounsfield value of around 3000, making it appear bright. Gold markings are also added to some points of the stent to give it higher Hounsfield values [14].

The shape of the stent graft is a tube that splits into two where the aorta divides into the iliac arteries. Since the stent has three openings, three seal points can fail, causing an endoleak and possible rupture. The upper seal point is called the proximal end, and the two lower ones are called the distal ends [6]. Figure 2.2 below shows the stent placement and its components.

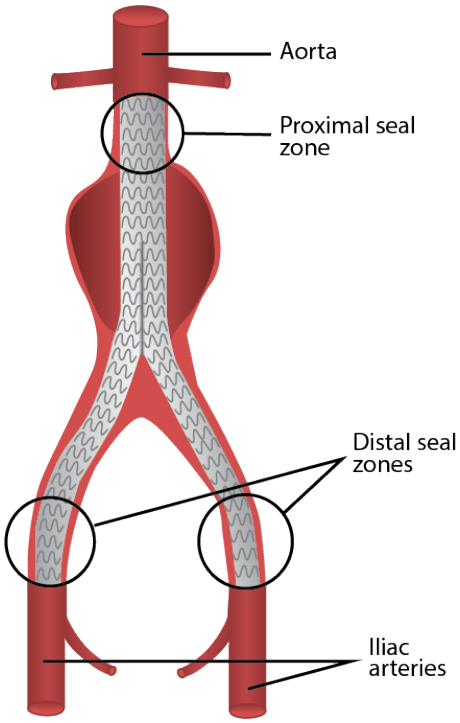



**Figure 2.2:** Abdominal aortic aneurysm blood vessels and stent zones. The proportions in the diagram are not medically correct.

The method developed by Håkan Roos and his team includes detailed measurements of different parts of the stent that form the basis of loss of seal diagnosis [6]. At each stent seal zone, multiple diameters of the blood vessel are recorded along with the seal zone length, which is the distance that the stent is in contact with the blood vessel wall. The seal zone length can decrease if the vessel where the stent is anchored expands, or if the stent moves. A patient is diagnosed with loss of seal if the measurements are no longer in line with the instructions-for-use of the EVAR stent manufacturer.

## 2.2 Deep Learning for Medical Images

Deep learning is a sub-field of ML where models have many layers conceptually thought of as their depth, making them deep. Each layer has parameters that can be trained towards better performance for a given task. Due to the models' depth, they have many parameters, sometimes ranging from a couple of million to several hundreds of billions[15]. The substantial number of parameters gives deep models great capacity for learning patterns and other relevant information from data. The downside is that tuning such a large amount of parameters requires a proportionally large amount of data to train on. Otherwise, such deep models could overfit the limited data by memorizing idiosyncratic patterns or noise in the training data rather than learning generalizable features.

One area particularly suited for the application of deep learning is computer vision.

As images are complex information compositions, they can benefit from higher-capacity models. By extension, deep learning is also a good approach to medical computer vision. However, many medical images are in 3D, increasing their informational complexity. The added complexity, in turn, necessitates higher capacity, which is achieved through more parameters. A large amount of data would be needed to avoid overfitting and train these parameters correctly. The difficulty with obtaining large datasets is that medical data is scarce due to the protection of private data [16]. To avoid overfitting, the capacity can be reduced. But if capacity is too low, it might cause underfitting when the model lacks the means to learn complex patterns.

### 2.2.1 Model Architectures

The early days of ML and neural networks started in the late 1950s and further expanded in the 1970s and 1980s with the invention of the multi-layered perceptron (MLP) and backpropagation[17]–[20]. The MLP quickly became a versatile and robust method in the ML field. It has seen a wide adaption ever since, with many state-of-the-art and breakthrough implementations. In essence, the MLP consists of several linear layers that take and transform a vector in size and element values. An activation function is applied between layers to enable non-linearity.

Convolutional neural networks (CNN) are deep neural networks that are especially well-adapted to analyzing images. Each convolutional layer consists of a learnable kernel that detects specific features in the image by convolution [21]. Using one kernel for the entire image reduces the number of parameters compared to MLPs while effectively analyzing the image. Convolutions are typically followed by max pooling and activation to reduce the image further. The output of a CNN is called a feature map, which holds the semantic representation of the original image.

One of the most recent architectures for machine learning is the transformer. It was first developed for text-reading tasks to address a problem of previous language models; they tend to forget words mentioned early in the text. The transformer employs a method called self-attention to look at the entire sequence and make connections between all words, no matter the distance between them. The meaning of the input words is taken into account through learnable embeddings, and it keeps track of their location in a sentence with positional embeddings. The transformer can also be used for computer vision tasks by dividing an image into a sequence of pixels or small patches. A multi-head mechanism can then be added that allows transformers to learn varying semantic features of the input image.

### 2.2.2 Transfer Learning

In computer vision and other machine learning tasks involving images, transfer learning can help achieve faster training and higher accuracy, especially when data is limited [22]. Transfer learning consists of pretraining a model on a large annotated dataset and reusing the model and its weights to finetune on a smaller dataset. To use transfer learning successfully, the small dataset must share some similarities with

the large dataset. Unfortunately, obtaining large pretraining datasets for medical image tasks is difficult. Popular datasets like ImageNet do not necessarily share enough similarities with medical images to be used for transfer learning. Instead, a large set of medical images would be preferable, but these datasets are difficult to obtain due to the protection of sensitive data [16].

When using transfer learning, it is vital that the model can transfer its previous knowledge to a new task. A specific pitfall that can occur, causing transfer learning to fail, is catastrophic forgetting. As the name implies, catastrophic forgetting occurs when the pretrained model's parameters are trained hastily, so the knowledge gained from the previous training is forgotten. Instead, it is replaced by knowledge from the current training. Ideally, the new data should adapt the old knowledge to the new domain, which can be achieved by a more careful approach [23].

#### 2.2.2.1 Model Backbone and Head

In the area of transfer learning in computer vision, the model's architecture is sometimes divided into two main parts: a backbone and a head [24]–[27]. The backbone works as an encoder to filter and reduce image information to a feature map. As is inherent to images, the information in each pixel is only relevant in the context of the neighboring values. Therefore, it is common practice to use CNNs as backbones [27]. A popular CNN backbone for medical image analysis is ResNet [27][25]. The architecture of ResNet is particularly advantageous due to the skip connection in their residual units, which allows information to flow more effectively in the model and, therefore, speeds up learning [28].

While it is possible to use convolutions down to a final prediction, they focus on local patterns and usually do not provide deeper reasoning. Therefore, a head is typically added. The head inputs the feature map and outputs the final prediction while recognizing complex global relationships [27]. The most standard head for CNN backbones is the MLP with linear layers. However, the transformer has also recently started being used as a head because of its ability to model global dependencies. The input for MLPs and transformers has to be flattened to one dimension, which means that the size of the flattened feature map is limited by the computational capacity available.

#### 2.2.2.2 Transfer Learning for Medical Images

Medical images of patients are typically not publicly available due to private data protection. As a result, the number of large datasets available for pretraining is limited but not non-existent. To address the lack of pretraining data in medical AI research, Chen *et al.* created a combined 3D segmentation dataset consisting of medical scans from 23 public segmentation challenges [25]. They designed a model consisting of a ResNet backbone and a segmentation head pretrained with the combined dataset, called MedNet. Pretraining was performed for multiple sizes of the ResNet backbone, such as ResNet-10, ResNet-18, etc. All ResNet backbone sizes have four main layers comprising several building blocks. The ResNet architecture

design was inspired by He *et al.*'s 2D version of the model with several modifications and improvements [29]. The pretrained models, along with all code, were made publicly available.

### 2.2.3 Landmark Detection

A common machine learning task for medical images is landmark detection, which is an object detection problem where the number and the class of objects in each image are fixed [30]–[32]. The goal of landmark detection algorithms is to identify specific anatomical structures in the image, and relies on the assumption that the structure the model searches for is always present in the image.

The earliest deep learning methods for landmark detection, published between 2017 and 2020, were based on CNNs. One successful approach by Zhang *et al.* employed a two-stage model that was first trained on small 3D patches sampled from the original data [30]. They reused the weights from the patch-based training and added a modified head to get an end-to-end detection model for the full medical volume. The model outperformed other state-of-the-art 3D detection models. Another approach also performed the detection in two steps, first globally with the entire image, and then, locally with identified sub-images, using a fully convolutional neural network (FCNN) [31]. The detection is done in small patches by classifying the presence of the landmark in the patch, and regressing the distance between the patch and the landmark. The final prediction is a combination of the regression and classification outputs.

While these early methods have been successful, they often rely on complex methods with no available code. Therefore, a recent transformer head for landmark detection with publically available code, Spine Transformer, could prove more useful [26].

#### 2.2.3.1 Spine Transformer

One recent approach to 3D object detection utilized a transformer head for detecting vertebrae in 3D imaging scans [26]. The authors implemented a ResNet-50 backbone with a transformer head and trained it to detect vertebrae with a varying field of view. The head comprises one encoder and one decoder, each with multi-head transformers, making it a relatively simple architecture, see Figure 2.3. The image retains its spatial information despite being flattened with learnable positional embeddings. The embeddings are added to the input before the encoder and the decoder. The addition necessitates that the hidden dimension of the position embedding is proportional to the number of input channels by a factor of three.

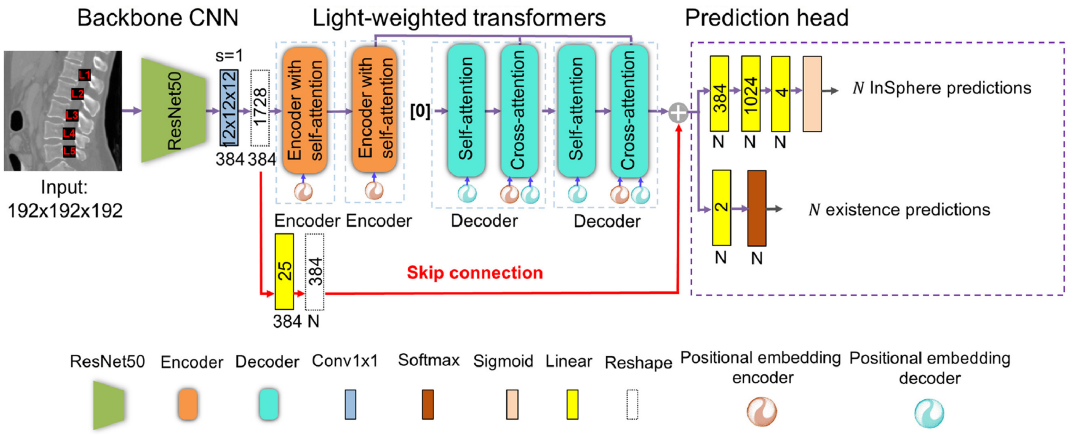



**Figure 2.3:** Architecture of the Spine Transformer, image retrieved from [26] and slightly modified.

The layers that make up the encoder and decoder consist of combinations of multi-head self-attention, adding and normalization, and feed-forward layers [33]. Since transformers converge slowly, Tao *et al.* also added a skip connection, which runs parallel with the transformer section and consists of a single linear layer. The output of both is combined before the last linear prediction section. The skip connection allows information and gradients from the backbone to be more prominent during the forward and backward pass, which speeds up training significantly.

Instead of using bounding boxes for each target, the authors transformed the annotations into spheres. Each sphere was defined by four values: $(x, y, z, r)$, where the first three values are the 3D coordinates of the sphere's center point, and $r$ is its radius. To make the detection invariant to the number of vertebrae visible in an image, a binary classification of the presence of each human vertebra was added. Detection loss is applied exclusively to spheres classified as present in the image.

The authors also implemented three detection loss functions, the first being the error between the predicted and true coordinates, and the second the intersection over union. The third loss function was the error between the true and predicted relative distances between the vertebrae, termed edge loss, given by $L_{edge}$ in equation 2.1 below:

$$\begin{aligned} \text{pair}_i &= ||v_i - v_{i+1}||_1, \quad \text{for } i = 0, 1, \ldots, n-1, v_i = [x_{v_i}, y_{v_i}, z_{v_i}] \in \mathbb{R}^3, \\ \hat{\text{pair}}_i &= ||\hat{v}_i - \hat{v}_{i+1}||_1, \quad \text{for } i = 0, 1, \ldots, n-1, \hat{v}_i = [x_{\hat{v}_i}, y_{\hat{v}_i}, z_{\hat{v}_i}] \in \mathbb{R}^3, \end{aligned} \tag{2.1}$$

$$L_{edge} = \frac{1}{n-2}(|\text{pair}_0 - \hat{\text{pair}}_1| + |\text{pair}_{n-2} - \hat{\text{pair}}_{n-1}| + 2\sum_{i=1}^{n-2} |\text{pair}_i - \hat{\text{pair}}_i|).$$

Above, $v_i$ is the sphere's center point for the i'th vertebra represented by its $x$, $y$, and $z$ coordinate, and $\hat{v}_i$ is the corresponding predicted center point, $\text{pair}_i$ is the

distance between the i'th vertebra sphere and its neighbor, and $\hat{\text{pair}}_i$ is the same distance but for the predicted spheres. A weighted sum of the coordinate absolute error losses, bounding box loss, and edge loss defines the final loss.

### 2.2.4 Data Augmentation

Data augmentation is a common strategy to increase data variability, help machine learning models generalize better, and reduce overfitting [34]. In medical applications, augmentations can also combat the problem of small datasets by simulating the addition of more data [35]. When the augmentations are added to the training data, variations are introduced to help the model focus on the right parts of the image by becoming more invariant to background noise.

The simplest and most common data augmentation techniques for medical images are transformations of the original data such as intensity transforms, scaling, and elastic deformations [34]. A subcategory is affine transforms, which includes all spatial transforms that preserve parallelism, including rotation, translation, scaling, and cropping of images. Any affine transform can be defined by

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where $x', y'$ and $z'$ are the positions of $x, y$ and $z$ after being transformed by the affine transform. In the affine matrix, $t_x$, $t_y$, and $t_z$ control the translation, and $a_{ij}$ controls the rotation and scaling [36]. The above equation can also be applied to the spatial coordinates of annotations when affine transforms are applied to augment images. However, since the affine transform can introduce changes to the coordinate grid of the image, an interpolation strategy is required. Garcea *et al.* performed a systematic review and found that 64% of studies that perform machine learning with medical images use affine transforms during training [34].

Another common data augmentation technique is noise addition, where noise such as Gaussian or salt and pepper is randomly added by changing pixel values [35]. An important consideration when using affine and noise-adding transforms is that they create highly correlated images; therefore, the model cannot generalize beyond the training population.

### 2.2.5 Interpretability

Using 3D medical images as input to diagnostic models, such as classifiers or regressors, can be difficult. Due to the large size and complexity of the images and the small datasets available, convolutional networks risk focusing on background noise instead of properly learning the relevant features [37]. This behavior can lead to

overfitting, which makes it essential to understand how the model makes its prediction. However, the black-box nature of deep learning models means they are challenging to interpret and understand in the decision-making process [38]. A way to increase awareness of the prediction process is to make the model reason more like an expert would. When radiologists analyze images for diagnostic purposes, the task is often broken down into subtasks. The strategy of creating smaller tasks can be imitated in machine learning through sequential learning.

#### 2.2.5.1 Sequential Learning

Sequential learning refers to using multiple machine learning models in a sequence, where each model in some way uses the output from the previous model. The models in the sequence can be chosen to perform different subtasks before making the final diagnosis. Sequential learning can also be referred to as cascaded learning or multi-stage learning.

A few papers have been found that employ sequential learning as a way to increase accuracy when diagnosing medical images [39]–[41]. Most of them utilize a segmentation model as a part of the sequence, since this isolates the anatomical region of interest and removes all irrelevant background. One successful approach to analyzing the aortic root chose a sequential method where the first model detects a region of interest on a low-resolution image [39]. The high-resolution ROI then becomes the input to succeeding models in the sequence.

#### 2.2.5.2 Methods of Interpretability

A potential secondary benefit to sequential machine learning methods is that they can increase understanding of the model. Within machine learning, the concept of understanding is called interpretability, which is "an attempt to explain the decision-making process of deep learning models in a way that is understandable for the end-users" [42]. Interpretability is especially important when implementing deep learning in clinical practice, as understanding the model can increase trust in its diagnostic capacity. The General Data Protection Regulation (GDPR) requires transparency of decision-making algorithms, which applies to artificial intelligence (AI) [43].

Salahuddin *et al.* describe nine interpretability methods, one of which is concept-learning models [42]. Here, the models predict a few high-level clinical concepts, and then the final diagnosis is predicted using these concepts as input. While concept-learning effectively creates transparency, specialists must identify and annotate concepts for initial prediction. It is more time-consuming than just annotating the diagnostic of interest. To address these concerns, other interpretability methods have been developed to explain black-box models after they have been trained, or post-hoc [42].

Post-hoc explanations are relatively easy to create, as they rely on the gradients of outputs with respect to different layers. Explaining post-hoc is especially useful for image-based models as gradients can be visualized in the input image to determine which regions are most relevant for the model's prediction [42]. The heatmaps

created by post-hoc explanation are generally given the umbrella term attribution maps. One of the simplest ways to calculate an attribution map is by saliency, where the gradient of one of the outputs is calculated with respect to each input pixel [44]. The saliency map thus represents what parts of the image the model focused on the most for a particular output, and is a way to localize essential features for the model's prediction [38]. It is important to note that the attribution maps do not offer any insights into how they contribute to the predicted output [42], so the explanation can only provide limited insights.

# 3

# Methods

In this chapter, the methods for data curation, processing, and augmentation are presented. Further, model architectures for the different learning tasks are introduced. Finally, the evaluation metrics used for assessing the performance of the models are described.

## 3.1 General Approach

Two overarching methods were chosen to perform a first proof-of-concept for predicting loss of seal in CT scans: a baseline and a sequential method, presented in Figure 3.1. The goal of both approaches was to diagnose an abdominal CT scan of a patient who has received EVAR surgery with loss of seal. The straightforward approach to diagnosing loss of seal is binary classification, but another predictor was chosen as it would be more interpretable from a medical perspective. The protocol developed by Håkan Roos and his team, which the thesis relies on, identified the seal zone length as most important in determining the loss of seal. Therefore, the seal zone length at the three stent endings was chosen as the final prediction target in both approaches. If at least one of the seal zones is shorter than a specified threshold, the patient can be diagnosed with loss of seal by rule-based classification according to the same protocol. However, the final rule-based classification will not be implemented in this thesis.

With seal zone length prediction as the goal, the most straightforward machine learning approach, the baseline, is to employ a regression model that predicts three seal zone lengths from each CT volume. However, the complexity of the problem likely exceeds the capabilities of such a simplistic model. A second approach is therefore proposed that employs well-tested machine learning methods, and reduces the input size for the regression model. This approach consists of a detection model that localizes the stent endpoints, which are used to define three smaller regions of interest surrounding the stent seal zone. The smaller region of interest (ROI) patches are then passed to a regression model, similar to the baseline, to predict each seal zone length. The sequential approach is inspired by the one presented by Krüger *et al.* [39]. The detection of the stent graft endings also imitates the same step performed by specialists when following the protocol by Håkan Roos.

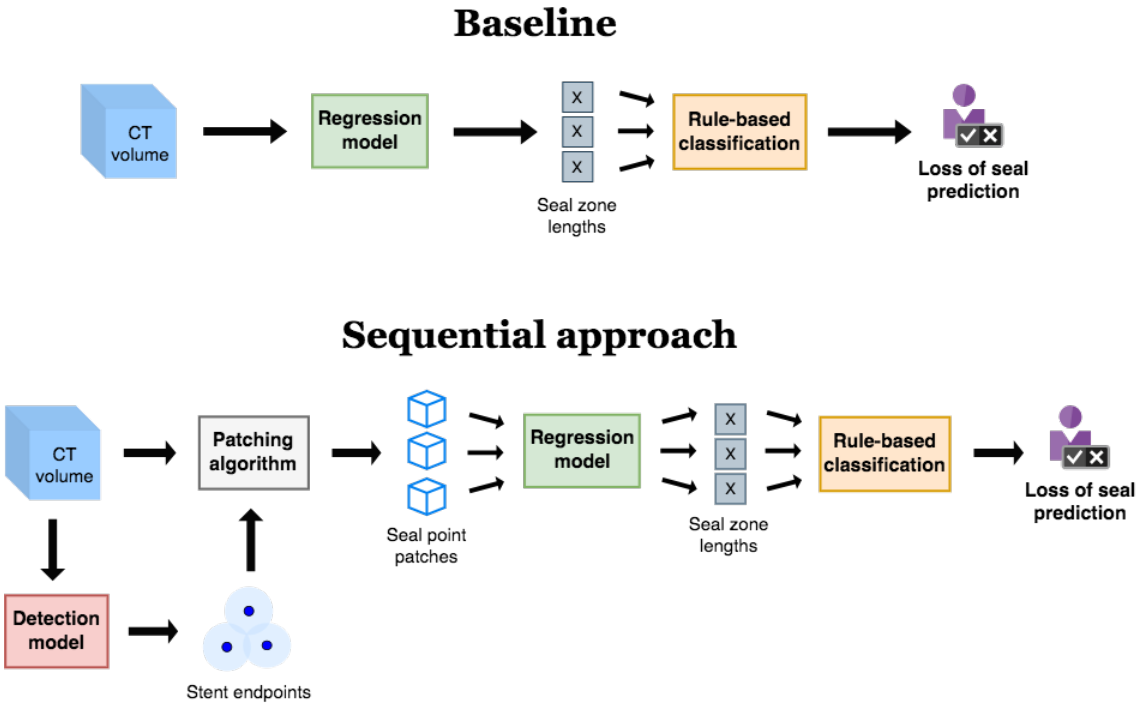



**Figure 3.1:** Overview of the machine learning methods implemented in the thesis.

## 3.2 Patient Data and Annotations

The data used for the thesis comes from the national registry of vascular disease, called Swedvasc. Ethical approval has been obtained through the Swedish Ethics Board of Approval with registration number 508-14, which means this project is also covered. The patient data could only be accessed on an authorized Chalmers computer to ensure the thesis follows ethical research.

The patients that receive EVAR surgery usually obtain a CT scan before and after surgery, then yearly for monitoring. Each CT used in this project is from one of three follow-up examinations after EVAR surgery: 1 month, 1 year, and the last follow-up several years later. In some cases, the patient passed away before their second yearly scan, in which case there are fewer than three examinations. The CT scans were annotated by specialized radiologists, including the stent's centerline and contours at each endpoint. Two centerlines were made for each CT that start in the aorta above the kidneys, and split into the left and right iliac arteries. There are four contours for each CT, one at each lower stent endpoint and two at the upper endpoint. An example of the annotations is shown in Figure 3.3. Figure 2.2 labels the different blood vessels and stent components.

Each CT scan has corresponding measurements of different parts of the stent, made by radiologists according to the protocol by Sandström *et al.* This thesis uses only the three seal zone attachment lengths from these measurements.

## 3.3 Data Processing

This project is the first to use the data described above for machine learning. Thus, some processing steps had to be performed to use the CT images and the annotations. The processing is described in the following sections, covering every step, from raw data to splitting the data for training.

### 3.3.1 Data Inspection and Quality Assessment

Before standardizing the data, all the annotations had to be validated by checking their location in the CT. The manual inspection ensured no faulty annotations could exacerbate the learning process or mislead the model. The software program 3D slicer [45] was the primary tool used to perform these inspections. The visual inspection confirmed that each CT had centerlines and contours with reasonable positions. Furthermore, a handful of CT volumes were excluded due to incompatibility with the given tools; see Section 3.4.

### 3.3.2 Data Preprocessing

The main purpose of the CT image preprocessing was to normalize the images to the same orientation, size, and spacing. The most common orientation was chosen as the standard, and those with different orientations were rotated. Additionally, all images were resampled using a common pixel spacing. A value of $2 \times 2 \times 2$ mm was chosen by considering the cumulative spacing distribution in each dimension, shown in Figure 3.2. Two-millimeter spacing in each dimension meant that most pixels were downsampled. The chosen spacing also gave a sufficiently high resolution to keep the stent details visible, according to supervisor Håkan Roos, an expert within the field. By resampling the images, most of them became smaller, which reduced the computational load during training. On average, the resampled images are 18 times smaller than the original images.

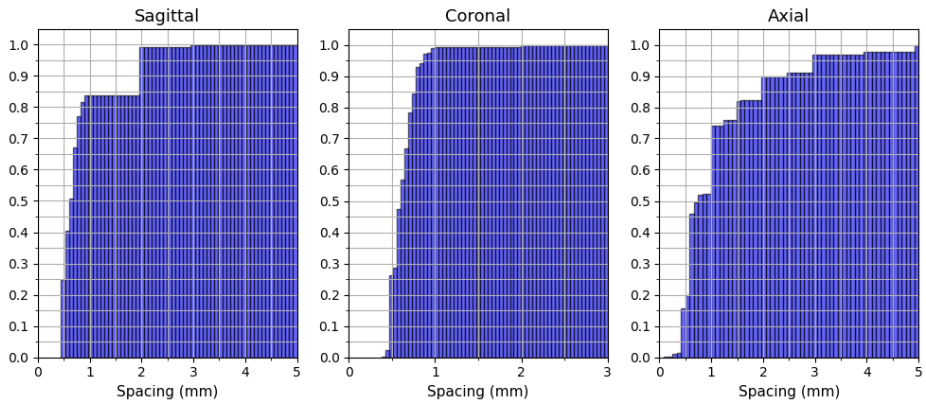



**Figure 3.2:** Cumulative distributions of pixel spacing along each dimension of the CT volumes.

Data processing was also done to get the necessary annotations. For the sequential approach, presented in Figure 3.1, there is an intermediate prediction of the stent endpoints. The true endpoints were not a part of the original annotations but could be calculated mathematically from the centerline and the four contours. The endpoint calculation consisted of finding the point of the centerline that was closest to the average of each contour. Since the proximal seal zone had two contour annotations, the average of the endpoints calculated from them was used. A visual representation of the annotations given and the endpoints calculated is shown below in Figure 3.3.

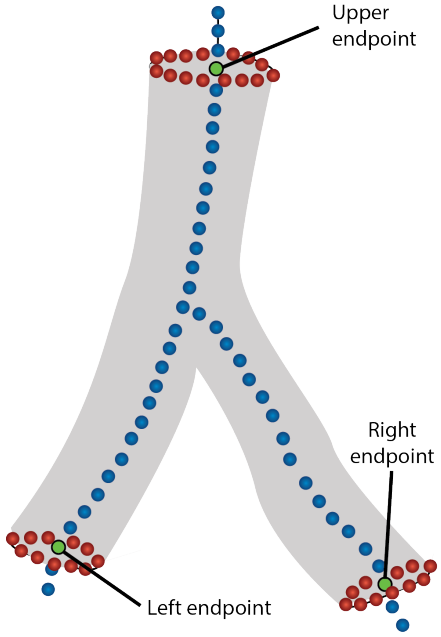



**Figure 3.3:** A visual representation of the annotations provided, where the centerline points are blue, the contour points are red, and the endpoints calculated from the two are green. The directions are given from the patient's perspective (i.e the patient's left and right). The proportions of the contours and centerlines are not medically accurate.

All seal zone lengths were collected from a large data sheet containing all the stent measurements according to the protocol. Depending on whether or not the stent extends beyond the inner iliac artery at the distal seal zones, the measurement would be written in one of two columns.

### 3.3.3 Recentering and Normalization

To further reduce and standardize the size of the images, the CTs were all cropped and centered around the stent using the positions of the endpoints. More specifically, the midpoint between the highest and lowest endpoints was calculated. Then, the $z$-coordinate of the midpoint became the image's center point after cropping, meaning that the same number of pixels were kept above and below the center point. The CT was only centered in the $z$-direction, as this direction is where most CTs varied in their field of view. After cropping, the size of the images was 220, 196, and 196

in the axial, coronal, and sagittal dimensions, respectively. Images smaller than $220 \times 196 \times 196$ before cropping were instead padded to the specified size. The height of 220 pixels was chosen by the length of the longest stent with a minimum padding added both above and below. By selecting the longest stent as the height for all volumes, some images had more padding around the stent than others. The recentering and cropping are shown by the first step in Figure 3.4, where the red points are the highest and lowest endpoints, and the blue point is the midpoint.

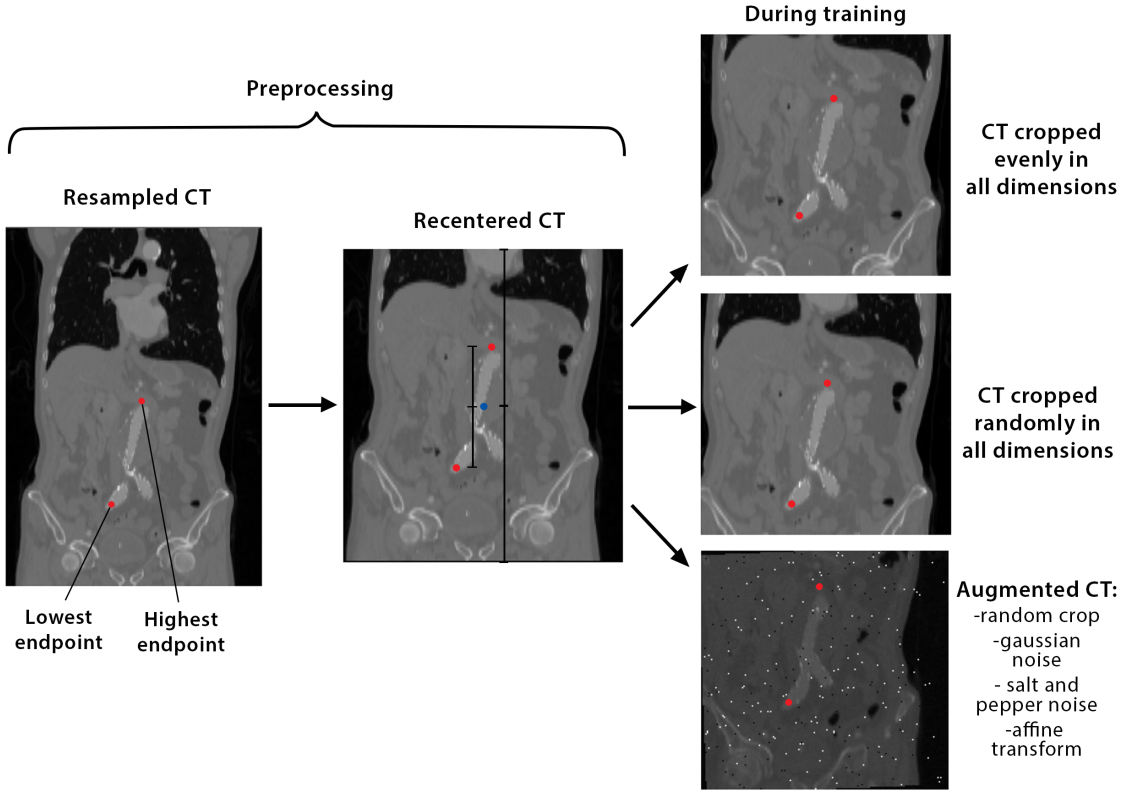



**Figure 3.4:** Processing pipeline example for one CT. The three final images represent the different levels of augmentations. Since it is difficult to show in 3D, only one coronal slice has been included. Therefore, the two endpoints have been fabricated for the visualization, in actuality they exist in different planes.

Before the images were fed through the models, several transforms were applied to standardize the CT volumes. To begin with, the voxel values were capped between -1000 and 3000 as these were the minimum and maximum Hounsfield unit values of materials to be expected in the CTs, and values outside this range could be considered noise. Afterward, the voxel values were normalized to min-max to get all intensities between 0 and 1 over the dataset.

### 3.3.4 Data Augmentation

To encourage the model to generalize, augmentation was applied to the data set during the training of all models. In the sections below, the volume transformations and

the spatial transformations of endpoints are described. Finally, the augmentation of the ROIs for seal zone length regression is presented.

#### 3.3.4.1 Volume Augmentation

The most crucial augmentation was a random crop applied to the images such that a total of 50, 20, and 20 pixels were randomly removed from both sides of the axial, coronal, and sagittal dimensions, respectively. Thus, the final size was 170, 176, and 176 in the $z$-, $y$-, and $x$-directions. Random cropping was performed as compensation for the centering in the axial dimension, described in section 3.3.2, by disbursing the distribution of the endpoint coordinates. The random cropping thereby promotes the model to generalize to varying field-of-views. To ensure that the validation and test inputs had the same size without performing augmentation, they were instead cropped evenly to remove the same amount of pixels. Cropping evenly meant that the stent was perfectly centered for validation and test images, while in the training set, it was translated in all dimensions.

Afterward, an affine transform was applied in the form of a random rotation in the ranges (-10°,10°), (-3°,3°), and (-3°,3°) for the axial, coronal, and sagittal dimensions respectively. Scaling was excluded in the transformation pipeline since it would distort positions and measurements. Scaling the images means that the models cannot reliably read the relative distances between features and pixels, complicating the task.

Finally, some intensity augmentations were added in the form of random noise. One type of noise added was salt and pepper, where a few pixels were randomly changed to the highest possible value (salt) and the lowest possible value (pepper). The probability that a pixel would become salt or pepper was 0.01. Gaussian noise was also added randomly with a probability of 0.5, where the distribution of the Gaussian noise had a mean of 0 and a standard deviation between 0 and 0.02. Figure 3.4 shows an example of one image's preprocessing, cropping, and augmentation.

#### 3.3.4.2 Label Augmentation

Two spatial transforms were applied: random cropping and a random affine transformation. Random cropping could be performed on the point indices by subtracting the number of lower-cropped pixels in each dimension. The affine transform could also be applied to the point coordinates by multiplying them by the affine matrix. Since the affine matrix was implemented independently from the spatial transforms, a validation algorithm was created to ensure that the transformed positions were accurate. After the affine transform, the point coordinates were no longer integers, meaning their location was not rounded to the nearest pixel.

#### 3.3.4.3 Region of Interest Augmentation

The regression model in the sequential approach was trained on individual ROIs. Therefore, separate instances of the augmentations were applied to the 3 ROIs from the same CT. The ROIs were augmented similarly to the entire volume, described

in section 3.3.4.1. Before augmentation, the regions of interest were created as $80 \times 80 \times 80$ pixel patches centered around each endpoint. Then, affine rotation was applied in the ranges (-10°,10°), (-3°,3°), and (-3°,3°) in the axial, coronal, and sagittal dimensions, respectively. After that, 16 pixels or 32 mm were cropped randomly from both sides in all dimensions to create $64 \times 64 \times 64$ pixel patches.

### 3.3.5 Dataset Split

To begin with, the dataset was shuffled to avoid any bias in the current order. Then, the data was split into three datasets: train, validation, and test, with the respective ratios of 78%, 12%, and 10%. If a patient had several examinations done, all of them were put into the same dataset. Keeping the CTs of one patient together ensured that each dataset was independently sampled, since exams from the same patient were highly correlated.

## 3.4 Implementation Details

The CT scans were processed using the SimpleITK Python library [46]. All DICOM files for one CT were loaded and aggregated into a volume. After the CTs had been process into a common standard, they were saved as tensors.

The code for creating and training the models was implemented in Python using the Pytorch and Lightning libraries [47], [48]. Pytorch was chosen for its versatility and broad adoption in the ML field, making it easy to implement publically available models. Lightning was selected as it is built on Pytorch but additionally facilitates, standardizes, and automates procedures for training. To begin with, a Lightning module was created where the model and optimizer were defined and configured. Then, the training and validation steps were specified where the loss, given inputs, and targets are defined. Once the lightning module has been created, it was paired with a lightning trainer that automatically trained and validated the models, logged predefined metrics, and saved checkpoints and hyperparameters in a version folder. Lightning also allowed customization through callbacks, such as early stopping, and supported training on one or several GPUs. All in all, Lightning simplified the maintenance and organization of the training, and reduced debugging significantly.

Image processing and augmentations were implemented partly through Torchio, a library dedicated to processing 3D images for machine learning [49]. The implementation of the augmentations is explained further in section 3.3.4. The attribution maps used for evaluation, described in section 3.7.2, were calculated with the Captum library [50].

## 3.5 Models

This section describes the architecture and other relevant details of each trained and implemented model. All models consist of a pretrained CNN backbone and an MLP or transformer head with slight modifications.

### 3.5.1 Baseline: Full Volume Seal Zone Length Regression

The baseline model consists of a backbone and a head. The backbone is a pretrained ResNet-18 model trained on 23 segmentation datasets of medical images [25]. The entire architecture is presented in Figure 3.5. The feature map that the backbone outputs is vast and, therefore, cannot be flattened and passed to the head directly. To reduce the size of the feature map, two convolutional layers were added after the backbone, the first one with kernel size (2, 2, 2), stride 2, and 64 groups. The second convolutional layer had a (1, 1, 1) kernel, only added to reduce the number of channels further.

After being passed through the backbone, the image size was 8 times smaller, but with 512 channels, and after reduction, the feature map was 16 times smaller and had five channels. The reduction allowed the feature map to be flattened to a vector that could fit in GPU memory. The head consisted of a multi-layer perceptron comprised of four layers with output sizes 256, 64, 16, and 3. The three final outputs correspond to the three seal zone lengths.

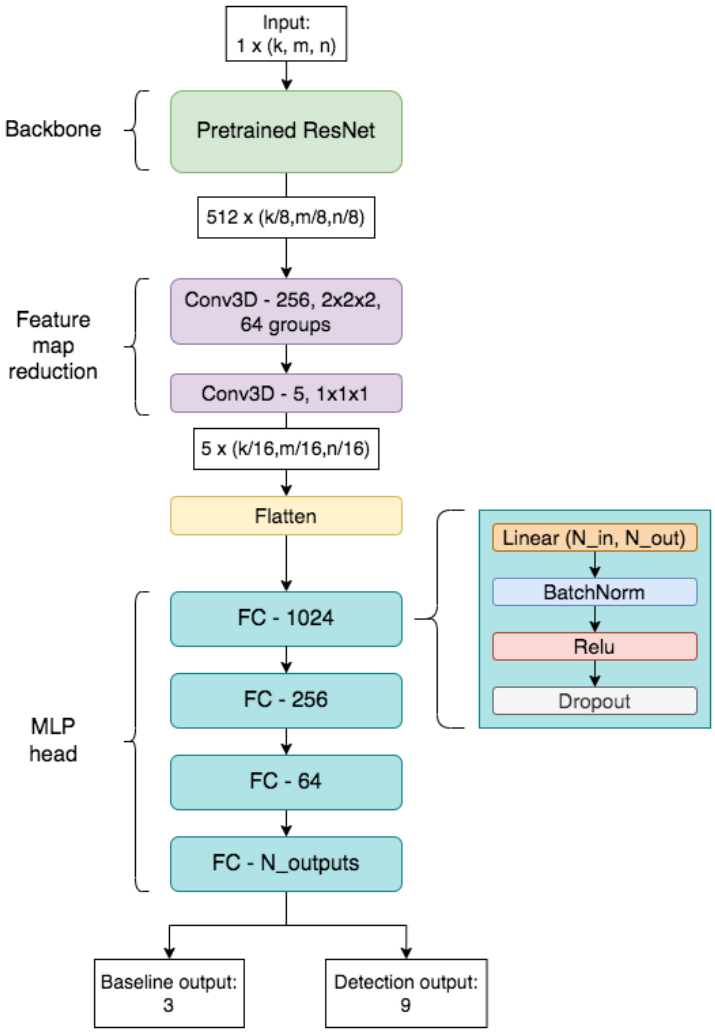



**Figure 3.5:** Baseline model architecture that takes a CT volume as input and performs either seal zone length regression or endpoint detection.

### 3.5.2 Full Volume Endpoint Detection

Two models were trained for the endpoint detection task; both had the ResNet backbone architecture with pretrained weights but different heads. The two model architectures are presented below.

#### 3.5.2.1 Multi-Layered Perceptron Head

The first model design was similar to the baseline approach, which consisted of an MLP head. The four layers of the MLP had output sizes 1024, 256, 64, and finally, 9 for each coordinate of the three endpoints. Batch normalization, dropout, and Relu activation were added between the linear layers. A reduction method was added between the backbone and the MLP with the same convolutional layers as the baseline, shown in Figure 3.5.

#### 3.5.2.2 Transformer head

The second model design combined a transformer head with the ResNet backbone. Since the goal of endpoint detection is similar to vertebra detection, SpineTransformer was used as a head with the pretrained ResNet-34 as backbone [26]. The model was implemented through the architecture available on GitHub with a few adjustments.

To begin with, instead of predicting a sphere for each vertebra, the position of the endpoints was predicted. Thus, the model prediction was modified by discarding the radius entirely. Furthermore, the binary prediction of the presence of each point (or sphere) was also removed, as it was assumed that the CT would always include all three endpoints. The number of queries was set to three for the three endpoints in each CT, 8 transformer heads were used and the feed-forward layer had 2048 outputs.

A reduction method had to be added as a 3D convolution with kernel size (2, 2, 2), stride 2, and 64 groups to pass the feature map to the transformer. Figure 3.6 shows the final architecture used for endpoint detection, where the transformer unit represents the encoder and decoder layers along with their associated positional encodings. Everything in the transformer unit (see Figure 2.3) is the same as the Spine Transformer, except for six encoder and six decoder layers instead of two of each in the original article.

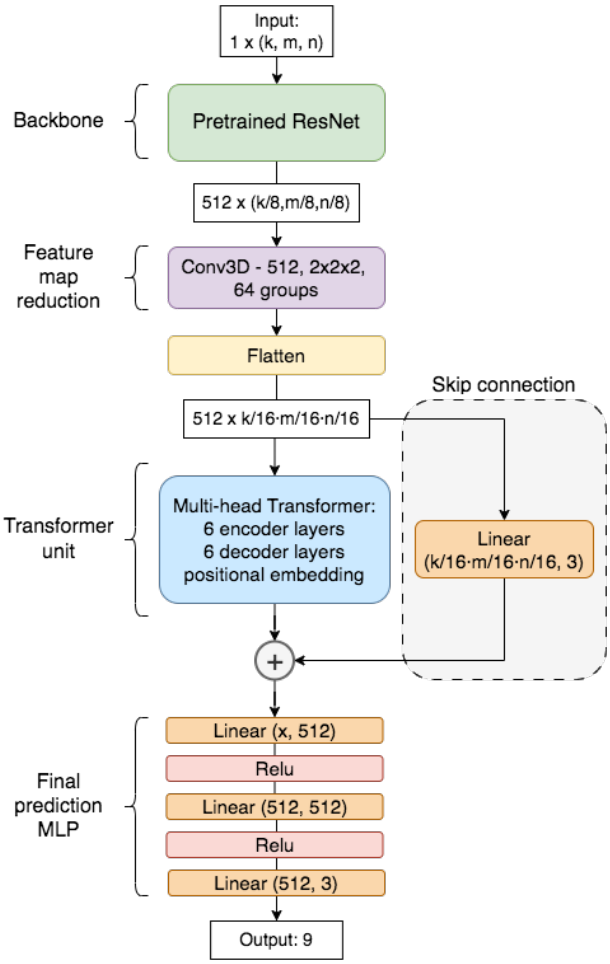



**Figure 3.6:** Architecture of the endpoint detection model with a transformer head.

Finally, a small change was made to the length of the three positional embeddings. The authors had defined their lengths as $l_1, l_2, l_3 = \frac{L_{\text{hidden}}}{3}$. The lengths were redefined in Equation 3.1 to enable values of $L_{\text{hidden}}$ non-divisible by three. The sum of the sizes had to equal $L_{\text{hidden}}$ due to the positional embeddings' connection to the hidden layer of the transformer. Therefore, the lengths were redefined as

$$l_1, l_2 = \left\lfloor \frac{L_{\text{hidden}}}{3} \right\rfloor, \quad \text{and} \quad l_3 = L_{\text{hidden}} - l_1 - l_2. \tag{3.1}$$

Four loss functions were used to train the original Spine-Transformer, two of which were used in this project: the L1-loss (also called the mean absolute error) of the endpoint positions, and the edge loss for the relative distances between the points. The intersection over union was discarded as no spheres were predicted, and the binary cross-entropy was excluded since no classification (sphere present or not) was done. Therefore, the final loss was the weighted sum of the two remaining loss functions. The original edge loss, presented in Equation 2.1 in Section 2.2.3.1,

was modified slightly for this implementation. The pairwise distances $\text{pair}_i$ were modified by defining them by Equation 3.2 as

$$\begin{aligned} \text{pair}_1 = ||\text{proxal} - \text{left}||_1, \quad \text{pair}_2 = ||\text{proxal} - \text{right}||_1, \quad \text{pair}_3 = ||\text{right} - \text{left}||_1, \\ \text{proxal} = [x_{proxal}, y_{proxal}, z_{proxal}] \in \mathbb{R}^3, \\ \text{right} = [x_{right}, y_{right}, z_{right}] \in \mathbb{R}^3, \\ \text{left} = [x_{left}, y_{left}, z_{left}] \in \mathbb{R}^3, \end{aligned} \tag{3.2}$$

where the distance between each pair of points is calculated.

### 3.5.3 Region of Interest Regression Model

The second step in the sequential approach, presented in Figure 3.1, consists of a regression model that predicts the seal zone lengths based on the reduced volume termed the ROI. The model was the same pretrained ResNet backbone and MLP head as the baseline architecture in Figure 3.5. The main difference is that only a $64 \times 64 \times 64$ subsection of the CT is used as input to predict the measurement it contains. The length of 64 pixels in each dimension was selected as a trade-off between informational retention and computational efficiency. The trade-off meant that the patch was large enough to contain relevant information but small enough to allow efficient computations during training. It was crucial to reduce the size sufficiently since the number of input volumes tripled by creating three patches from each original CT image, which could slow down training significantly. An added restriction was that the endpoints were sometimes close to the edges of the volume, and therefore, a larger size was impossible without adding unnecessary padding.

Since the ROI size is much smaller than the volumes used as inputs to the baseline model, the backbone's feature map was also smaller, decreasing the need for a reduction. Instead of reducing the feature map size, it was adequate to lower the number of channels from 512 to 16. As a result, the region of interest regression model had a higher resolution of its feature map relative to the input size. The channel reduction resulted in a flattened feature vector of size 8192 as an input to the MLP head, with 1024, 256, 64, and 1 as the rest of the layer sizes. The modified architecture for ROI regression is presented in figure 3.7. Each ROI was treated as an independent input during the training process such that the model only viewed one patch at a time. However, for efficiency reasons, all 3 ROIs from the same CT were loaded simultaneously and included in the same batch.

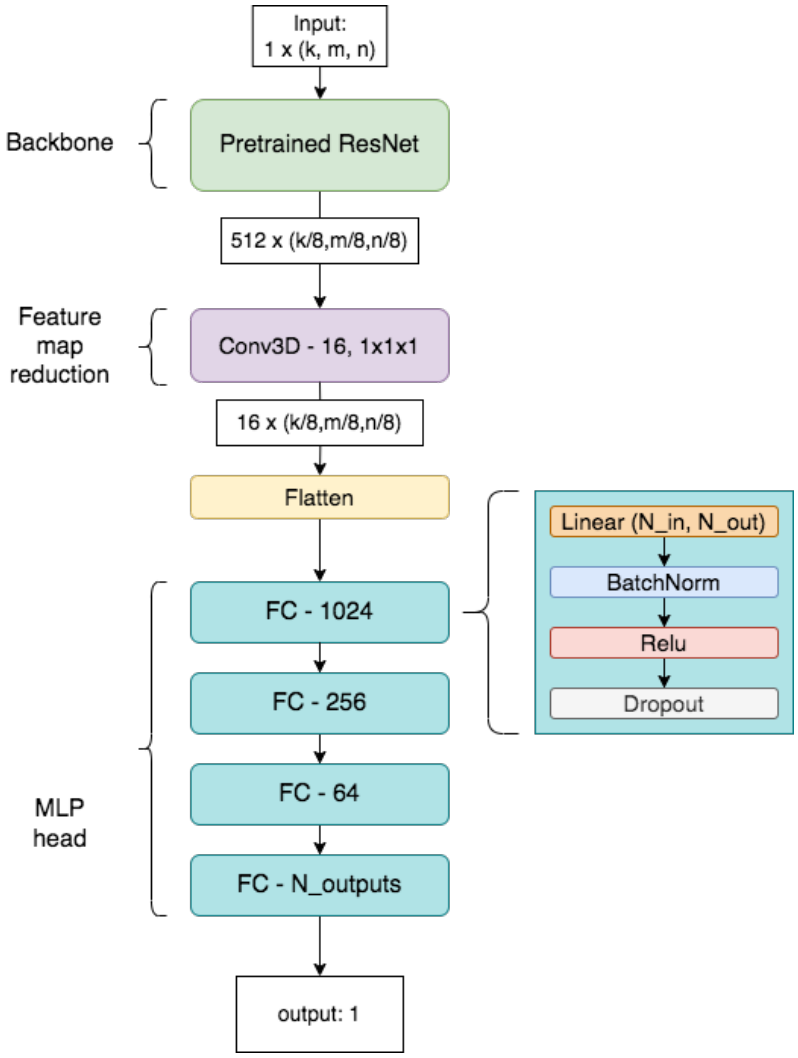



**Figure 3.7:** Architecture for the regression model that predicts seal zone lengths on ROI patches.

The ROI model was never trained with predicted endpoints; instead, the predefined annotations were used. Training with the target endpoint was done because the detection and measurement regression models were developed in parallel. Thus, accurate endpoint predictions were not available until the end of the project. By then, it had become evident that no further insight would be gained from using predicted endpoints due to the poor performance with the annotated endpoints. Therefore, the true endpoint locations were used as centerpoints of the regions of interest rather than the predictions made by the detection model.

## 3.6 Training

The models were all trained by first training the head and then gradually unfreezing the pretrained backbone for fine-tuning. The unfreezing of the backbone was divided into five sections: the first convolutional layer and the four subsequent main ResNet

layers. During training, five thresholds of the validation error were specified where each part of the backbone was unfrozen. Furthermore, AdamW was used for all models to apply the gradient step [51]. The backbone was also given a lower learning rate than the head to avoid catastrophic forgetting.

During training, the mean absolute error of the training and validation set was monitored to evaluate potential overfitting, underfitting, or stagnation. Instead of training for a set number of epochs, the training was halted when the validation error stagnated, triggering early stopping. Stagnation was defined as no decrease in the error over a threshold number of epochs.

## 3.7 Performance Metrics and Evaluation

### 3.7.1 Evaluation Metrics

The primary evaluation metric was the absolute error between the predicted and true values. In the sequential approach, the mean absolute error between the three true and predicted seal zone lengths was calculated for the baseline and the final model. For the endpoint detection, the absolute error between the predicted and true coordinates of the endpoints was calculated.

For the endpoint detection models, the Euclidian distance, or root square error, was also evaluated between the predicted and true endpoint coordinates. The Euclidean distance is defined as

$$L_2 = \sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{3.3}$$

where $y_i$ is the true value, and $\hat{y}_i$ is the predicted value. The Euclidean distance was calculated per endpoint with the $z$-, $y$-, and $x$-coordinates. Since all models of the project are regression models, the $R^2$ metric was also used for evaluation. $R^2$ is a value that ranges between negative infinity and one and indicates how well the model fits the data. Here, one represents a perfect fit, negative values represent overfitting, while values close to zero mean that the model underfits. The definition of $R^2$ is presented in Equation 3.4 below,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{3.4}$$

where $\bar{y}$ is the mean of all ground truth values.

### 3.7.2 Evaluation plots

A few evaluation plots were made using the presented metrics. One of these was a scatter plot of the predicted values versus the true values for each input in the test data. This kind of scatter plot was used to check for a strong correlation between the predicted and target values. For example, these plots can aid in assessing whether the model generally predicts values close to the target or if it tends to over- or underpredict.

Another type of plot was Bland-Altman error plots. They were used to plot the difference between the true and predicted value against the true value for each output in the test set. The Bland-Altman plots can similarly aid in finding patterns in the error.

Saliency maps were calculated as attribution maps to interpret the regions the model favors to make its predictions. The saliency map is the derivative of one output value with respect to each input pixel, calculated by backpropagation [44]. The saliency maps are three-dimensional, but to display them in 2D, a slice in each anatomical plane was chosen based on where the target endpoint is. Since saliency maps can only be calculated for one output, the saliency maps for predicting one endpoint's z, y, and x coordinates were aggregated by summation. To display the saliency map, a CT image from the test set was chosen arbitrarily for presentation in this report.

The saliency map was only included in the evaluation if the model performed sufficiently well on the test set. This decision was made because there is no point in checking which regions the model takes into account if it is clear that the model has not learned the task at hand.

## 3.8 Usage of Large Language Models

Large language models like ChatGPT were mainly used for troubleshooting and code suggestions. Additionally, Grammarly was used to correct and improve grammar. While such tools have aided the project's development, great care has been taken to ensure that they have been used in cooperation with critical thinking and proper academic practices. When smaller methods and code snippets have been generated and used, they have been validated and, if needed, cross-referenced with the relevant documentation.

# 4

# Results

This project has two primary outcomes. The first is a new, curated dataset of post-EVAR exams with corresponding CTs and manual annotations according to the protocol. Second, several models are trained on this dataset with different architectures and toward different prediction tasks.

The parameters and metrics used for training are presented for each model, along with the evaluation results on the test set. The models are presented in the following order: the baseline model for seal zone length regression, the model for endpoint detection with an MLP head, the endpoint detection model with a transformer head, and finally, the seal zone regression model with regions of interest as input.

## 4.1 Data Selection & Splits

Data transfer, conversion to compatible formats, quality check, and loading into Python eliminated quite a few patients from the data. Throughout the thesis, approximately 158 patients were provided with a median of 4 CT scans each, along with the corresponding annotations. Manual inspection eliminated 69 CT images because time and resource constraints made a corrective action infeasible. In this extensive process, two main causes of elimination were identified. The first cause was when the manual centerline was not inside the stent or blood vessel. The other cause was when the manual contours were incorrect, which was the case for a group of patients from a specific hospital. After this inspection process, 143 patients with a total of 399 CT scans remained in the dataset and could be used for training and evaluation. The training, validation and test split was set to 78/12/10 %, which led to 312, 48, and 39 CTs, and 111, 18, and 14 patients in the training, validation, and test sets, respectively.

## 4.2 Full-volume seal zone length regression

The first model trained was the baseline for seal zone length regression from full volumes. The model training parameters, the training metrics, and the evaluation results on the test set are presented below.

### 4.2.1 Training

The parameters and metrics in Table 4.1 were chosen after limited experimentation. This combination was selected for its training stability and performance. Furthermore, the unfreezing thresholds were regularly spaced within the range of the highest and the lowest validation errors seen during training.

**Table 4.1:** Baseline model training parameters

| Parameter | Value |
|---|---|
| Backbone learning rate | $5 \cdot 10^{-6}$ |
| Backbone weight decay | $2.5 \cdot 10^{-4}$ |
| Head learning rate | $3 \cdot 10^{-4}$ |
| Head weight decay | 0.0025 |
| Dropout | 0.05 |
| Unfreeze thresholds (val error) | 25, 19, 16, 13, 11 |
| Batch size | 20 |

In Figure 4.1, the model's training process, with augmentations applied, can be viewed until early stopping at epoch 442. The mean absolute error on the training and validation dataset is plotted for each epoch.

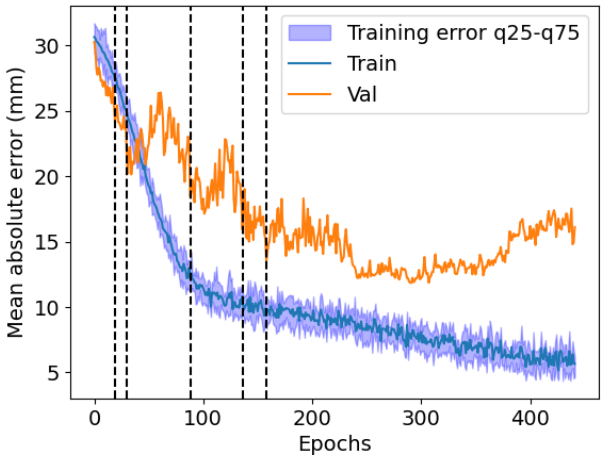



**Figure 4.1:** Training and validation mean absolute error during the baseline regression model's training, where the black vertical lines represent the unfreezing of each part of the backbone.

The figure shows that the training error rapidly decreases until about epoch 100 where it continues decreasing but at a lower rate. The validation error, on the other hand, has a lot more variation but has a decreasing trend until around epoch 300, where the variation from epoch to epoch slows down. However, the validation error increases slightly towards the end until early stopping sets in. This suggests that the model starts overfitting at epoch 300.

### 4.2.2 Evaluation

The baseline model was evaluated by the absolute error between the predicted and target seal zone length. The mean and standard deviation of the error are presented in Table 4.2 based on each seal zone and aggregated over all outputs. As demonstrated by these values, the errors are quite high suggesting that the model has not been able to learn how to characterize the seal zone length. The error for the left distal seal zone is higher than the other two.

| Seal zone | Mean absolute error (mm) |
|---|---|
| Proximal | $9.5 \pm 7.8$ |
| Left distal | $27.6 \pm 14.9$ |
| Right distal | $13.2 \pm 9.3$ |
| All | $16.8 \pm 13.5$ |

**Table 4.2:** Error between true and predicted seal zone lengths of the baseline model.

To further investigate the performance of the model, the predicted seal zone lengths were plotted against their corresponding true values, shown in Figure 4.2.

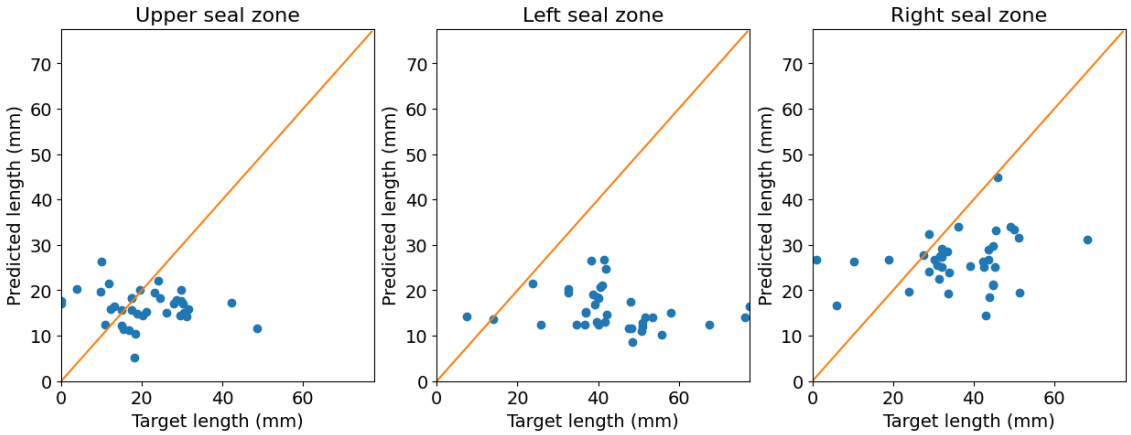



**Figure 4.2:** Scatter plot of predicted and target values of the baseline model for each seal zone, where the orange lines represent predictions equal to the target.

A few aspects can be observed in the above figure. Most importantly, in all three plots, the predicted value is often smaller than the target. It seems that the model is predicting a mean value with some variance rather than learning. This indicates that the model has high bias and variance, meaning it is under- and overfitting the data. No clear relation between predictions and targets can be found, further underlining the inability to generalize. The pattern is further exemplified by the Bland-Altman plots between predicted and true seal zone measurements, shown in Figure A.1 in Appendix A

## 4.3 Full-volume Stent Endpoint Detection - MLP Head

The first step in the sequential approach was to detect the EVAR stent's endpoints. To begin with, the most straightforward approach was chosen for detection by adopting the baseline architecture with a pretrained ResNet-18 backbone and an MLP head. Instead of predicting three seal zone lengths, the model predicted the 3D coordinates of three endpoints, corresponding to nine values. The training parameters, results, and the model's performance on the test set are presented below.

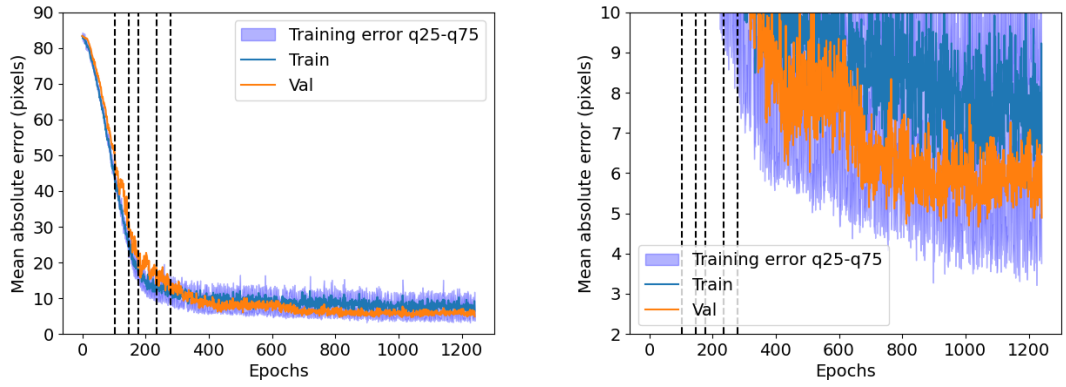
### 4.3.1 Training

The endpoint detection model with an MLP head was trained with the parameters presented in Table 4.3. The training parameters remained the same as those of the baseline model due to their similarities. Only the unfreeze thresholds were modified since this task's validation error was initially higher.

**Table 4.3:** Endpoint detection model with MLP head training parameters

| Parameter | Value |
|---|---|
| Backbone learning rate | $5 \cdot 10^{-6}$ |
| Backbone weight decay | $2.5 \cdot 10^{-4}$ |
| Head learning rate | $3 \cdot 10^{-4}$ |
| Head weight decay | $2.5 \cdot 10^{-3}$ |
| Dropout | 0.05 |
| Unfreeze thresholds (val error) | 50, 30, 20, 15, 12 |
| Batch size | 20 |

The model was trained for 1300 epochs with augmentations. Figure 4.3 presents the resulting train and validation mean absolute error.




**(a)** Full training plot

**(b)** Training plot with axis limits

**Figure 4.3:** Training and validation mean absolute error during training of the endpoint detection model with an MLP head. The black lines show the epochs where a layer of the ResNet backbone was unfrozen.

The figure demonstrates that both the training and validation error decrease rapidly at first and then slow down after about 200 epochs. After 300 epochs, the validation error becomes, on average, lower than the mean training error but remains higher than the 25th percentile of the training error. The final training error is around 8 pixels or 16 mm, and the final validation error is around 6 pixels or 12 mm. The high variance of the training and validation errors and the low validation errors are likely explained by the random augmentations applied to the training data. The images seen by the model differ significantly between epochs due to the random transforms. The validation set, on the other hand, has no augmentations and is cropped evenly, so it may be easier for the model to predict the endpoint locations in those images.

### 4.3.2 Evaluation

The Euclidean distance between the predicted and ground truth endpoints is presented in Table 4.4 below, where the mean was calculated for all CTs in the test set.

| **Endpoint** | **Euclidean distance (mm)** |
|---|---|
| Proxal | 21.58 ± 9.86 |
| Left distal | 20.99 ± 10.67 |
| Right distal | 23.98 ± 13.45 |
| All | 22.18 ± 11.40 |

**Table 4.4:** Euclidean distance between predicted and true endpoints of the detection model with an MLP head.

The test set's Euclidean distance has a total mean of 22.18 mm, and there are no significant differences between the three points. The mean absolute error per endpoint and coordinate is presented in Table 4.5 below.

| | **Mean absolute error (mm)** | | |
|---|---|---|---|
| **Endpoint** | Z-coordinate | Y-coordinate | X-coordinate |
| Proximal | 15.1 ± 4.0 | 10.8 ± 7.6 | 7.8 ± 4.5 |
| Left distal | 10.8 ± 8.8 | 12.0 ± 10.8 | 8.5 ± 5.5 |
| Right distal | 12.7 ± 11.8 | 8.4 ± 8.6 | 14.1 ± 10.8 |

**Table 4.5:** Mean absolute error of predicted and target endpoint coordinates for the detection model with an MLP head.

The errors are relatively similar between all coordinates except the z-coordinate of the proximal and x-coordinate of the right distal endpoints which are slightly higher. Most values have a relatively high standard deviation except the z-coordinate of the proximal endpoint. The high error and low standard deviation could suggest a systematic error of the model in predicting the proximal endpoint's z-coordinate.

The predicted endpoint coordinates are plotted against the true coordinates for all test data in Figure 4.4 below. The scatter plots show that the predictions somewhat

follow the line of perfect prediction, although a slight pattern of under estimation can be discerned. The relatively high correlation between prediction and ground truth suggests that the model learned the data to some degree, which is further underlined by the $R^2$ value of 0.911. The only aspect of the data that the model seems slightly unable to learn is the x-coordinate of the left distal endpoint, where there is almost no apparent correlation between true and predicted values.

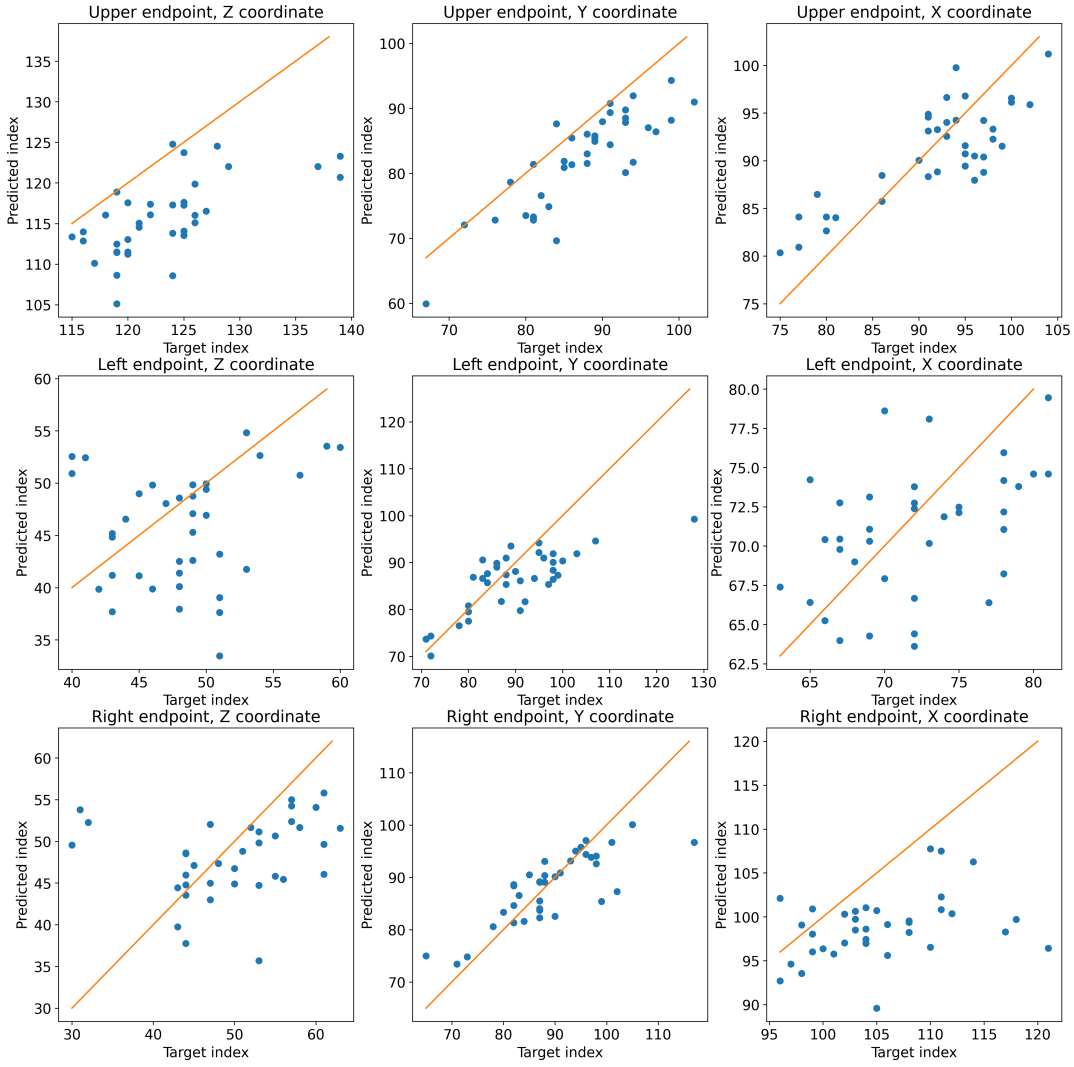



**Figure 4.4:** Scatter plot of predicted and target values of the detection model with MLP head for each point coordinate, where the orange lines represent when predicted values are equal to the target.

The Bland-Altman plots in Figure A.2 in appendix A further demonstrate the pattern of underfitting, where the mean error is always negative, especially so for the z-coordinate of the upper endpoint.

To be able to visualize how well the model performs on the test data, a predicted and target proximal endpoint was plotted inside its corresponding image for one test CT. This is shown in Figure 4.5.

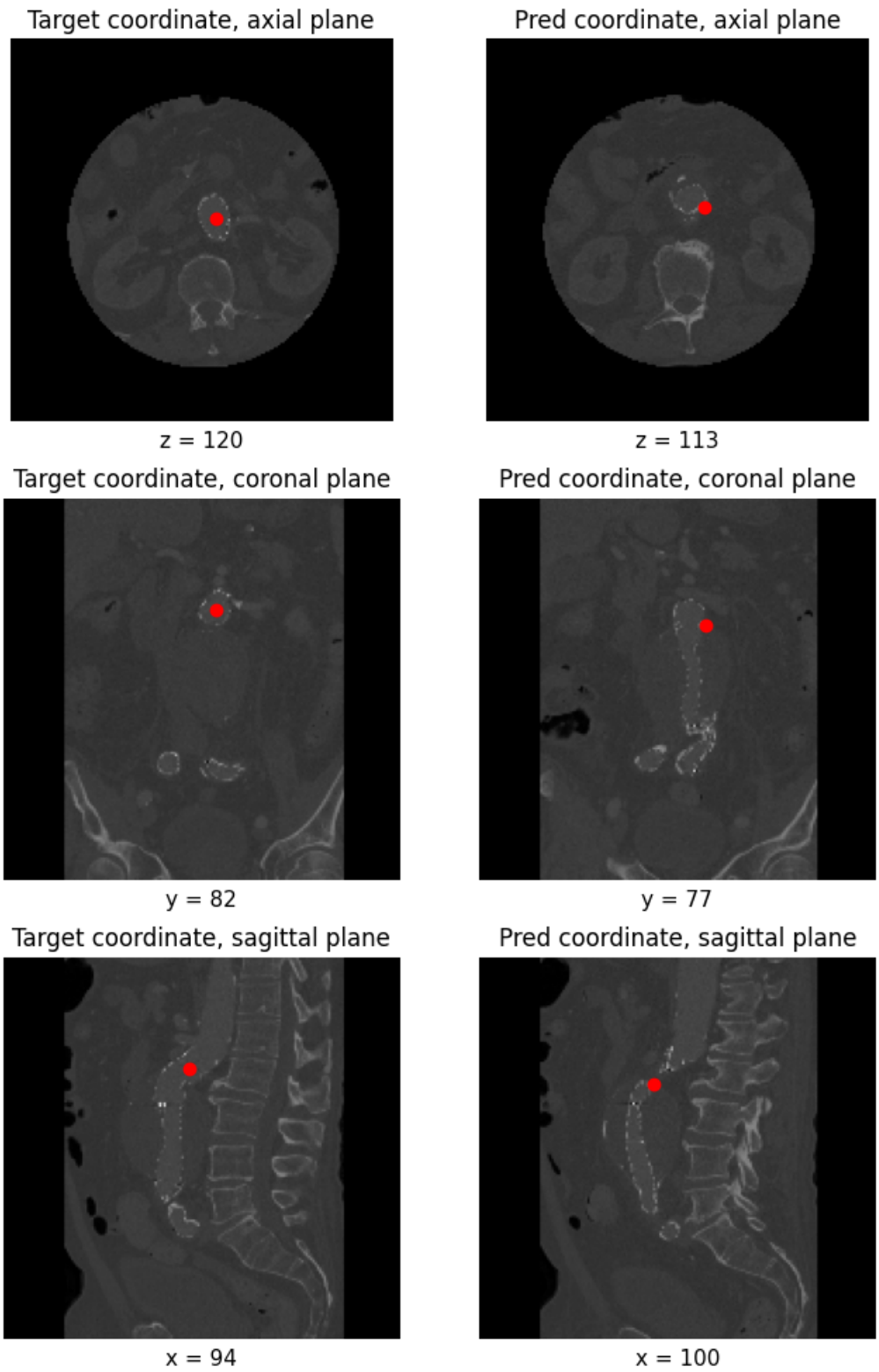



**Figure 4.5:** Target and predicted proximal endpoint for the detection model with MLP head, shown by their positions in each anatomical plane. The slice and endpoint positions are also written by their exact pixel index.

Above, one can see that the model prediction is in the vicinity of the target but not completely correct. The model seems to have failed to locate it inside the stent and possibly a bit too low in the z-direction.

A saliency map was made for the same CT and endpoint as the above figure to observe the regions that are important for the model's prediction. The saliency map is shown in Figure 4.6 along with the corresponding slices of the input CT. The predicted proximal endpoint is also marked in the images. Lighter pixel values in the saliency map represent higher saliency.

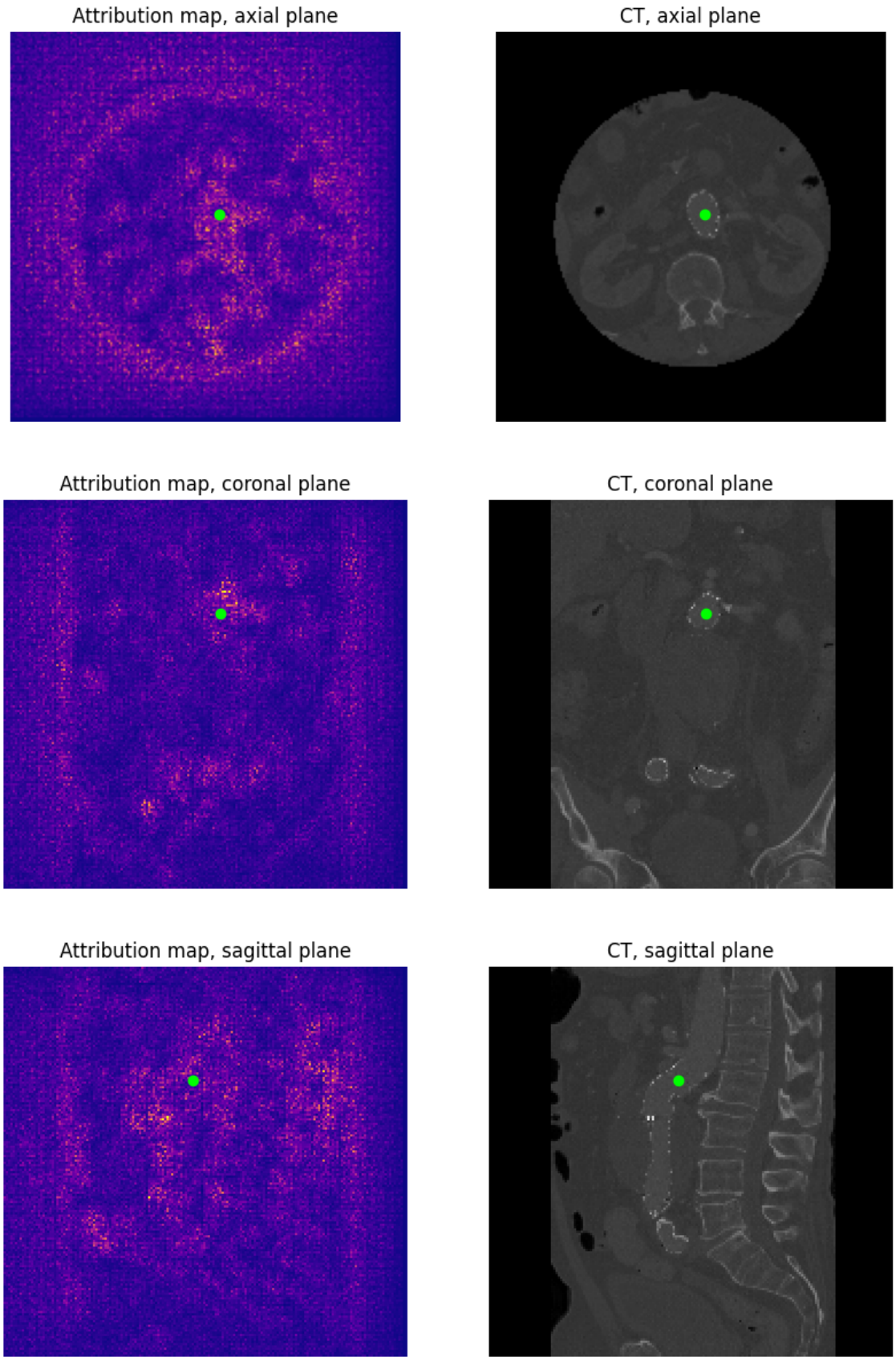



**Figure 4.6:** Attribution map of the detection model with MLP head predicting the three coordinates of the proximal endpoint, shown in each anatomical plane along with the corresponding slice of the input CT. The slices are given by the ground truth endpoint position, which is marked with the green point.

The saliency heat map suggests that the model looks at parts of the entire image rather than just the stent. It could be that it particularly notices high-contrast regions such as the edges of the patient, the spinal cord, and the air bubbles in the intestine since they have higher saliency. While the model looks at the region of and around the stent, the attribution maps could suggest that it also considers some irrelevant background noise.

The attribution maps have been placed on top of the CT image slices in Figure 4.7 to make the model's interest in certain anatomical features evident. Areas with high contrast and high saliency include the dark air pockets inside the patient and the lighter parts of the stent, the spinal cord, and some organs.

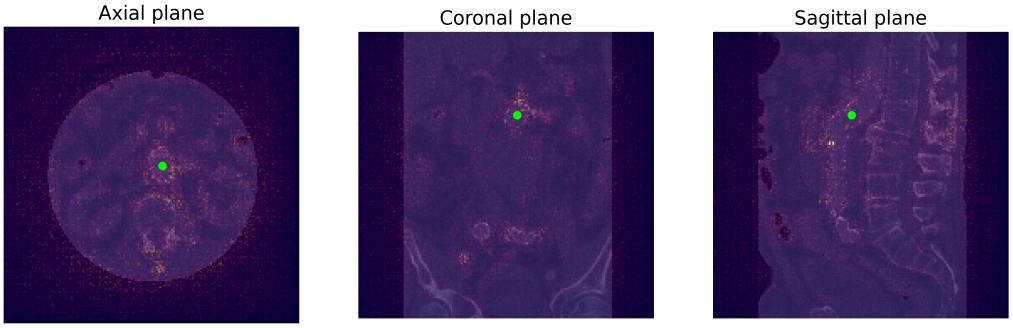



**Figure 4.7:** Attribution map of the detection model with MLP head predicting the three coordinates of the proximal endpoint, layed on top of the input CT images. The slices are given by the location of the ground truth endpoint, which is marked by the green point.

## 4.4 Full-volume Stent Endpoint Detection - Transformer Head

The endpoint detection model with an MLP head seemed to have a complicated loss landscape, as is evident from the training loss in figure 4.3. Therefore, another model was tested for the detection task with the same ResNet backbone but with a transformer head. The training of the detection model with a transformer head, as well as its performance on the test set, are presented in the sections below.

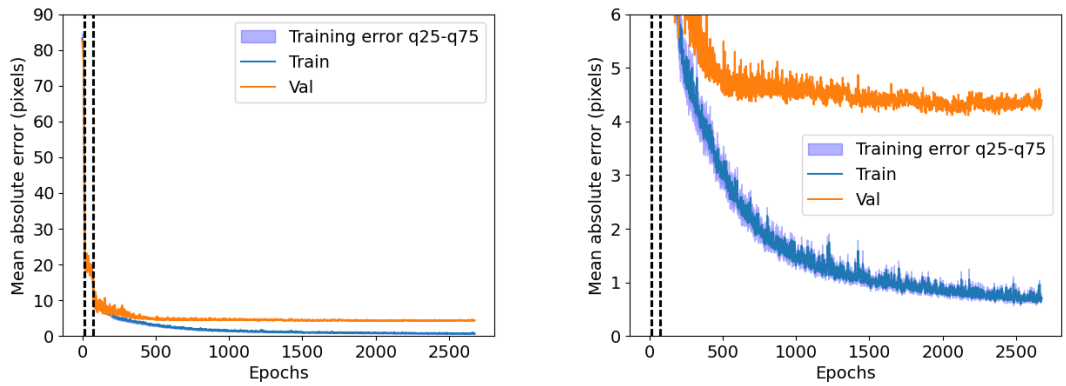
### 4.4.1 Training

The endpoint detection model with a Spine Transformer head was trained with the parameters presented in Table 4.6 below. The parameters were tested a few times, with the ones reported giving the best performance. The loss coefficients were chosen arbitrarily since the original paper did not disclose their ratio. The unfreeze thresholds were spaced evenly between the maximum and minimum validation loss. The batch size was chosen for optimal training time with the available capacity.

**Table 4.6:** Training parameters for the endpoint detection model with a transformer head

| Parameter | Value |
|---|---|
| Backbone learning rate | $5 \cdot 10^{-6}$ |
| Backbone weight decay | $5 \cdot 10^{-4}$ |
| Head learning rate | $1 \cdot 10^{-5}$ |
| Head weight decay | $5 \cdot 10^{-3}$ |
| Transformer dropout | 0.2 |
| Unfreeze thresholds (val error) | 50, 30, 20, 15, 12 |
| Batch size | 20 |
| L1 loss coefficient | 1.5 |
| Edge loss coefficient | 1 |

The training was performed until epoch 2674, when early stopping was triggered. Augmentations were added to the training set described in section 3.3.4. The training and validation error were monitored, shown in Figure 4.8.




**(a)** Full training plot.

**(b)** Training plot with y-axis limits.

**Figure 4.8:** Training and validation error of the endpoint detection model with a transformer head.

As can be seen in the full training plot in Figure 4.8a, both the training and validation error fall rapidly, then stagnate, and then decline rapidly again. Then, the error continues to decrease more slowly. The behavior is likely due to the skip connections in the transformer, which allows the model to learn quickly, causing the error to fall rapidly at first. Thereafter, the training of the transformer likely dominates, allowing a second rapid decrease. In Figure 4.8b, a zoomed in version is displayed, showing the training error decreases to about 1 pixel or 2 mm while the validation error converges earlier to around 4.5 pixels or 9 mm.

### 4.4.2 Evaluation

The model for endpoint detection with a transformer head was evaluated in the same way on the test set. The Euclidean distance for the endpoints is presented in Table 4.7 below.

| Point | Mean Euclidian distance (mm) |
|---|---|
| Proxal | 14.7 ± 6.1 |
| Left distal | 19.0 ± 12.1 |
| Right distal | 24.0 ± 14.9 |
| All | 19.3 ± 12.2 |

**Table 4.7:** Euclidean distance between predicted and true endpoints of the detection model with a transformer head.

As can be seen in Table 4.7, the mean Euclidean distance is 19.3, which is slightly lower than the errors of the detection model with an MLP head in Table 4.4. The lower errors of the transformer based model suggest that it is slightly better at detecting the stent endpoints. The MLP head model has a lower standard deviation in the Euclidian error of the distal endpoints but worse overall performance than the transformer head model.

Table 4.8 below shows the mean absolute error per endpoint coordinate.

| | Mean absolute error (mm | | |
|---|---|---|---|
| **Endpoint** | Z-coordinate | Y-coordinate | X-coordinate |
| Proximal | 8.3 ± 5.9 | 8.2 ± 5.0 | 6.1 ± 4.9 |
| Left distal | 7.8 ± 7.2 | 12.6 ± 12.0 | 7.5 ± 6.1 |
| Right distal | 11.6 ± 12.4 | 8.7 ± 10.4 | 14.0 ± 11.8 |

**Table 4.8:** Mean absolute error of predicted and target endpoint coordinates for the detection model with a transformer head.

The table demonstrates that the errors are similar overall for the different outputs. However, the x-coordinate of the right distal endpoint is very high, which is similar to the result of the endpoint detection model with an MLP head. The high error in both models could suggest that it is particularly difficult to determine the x-coordinate of the right distal endpoint in comparison to the other values. On the other hand, the transformer has a low error for the z-coordinate of the proximal endpoint, which differs from the MLP-based model.

When plotting the predicted coordinates against the true ones, as can be seen in Figure 4.9 below, the predictions are slightly more centered around the ideal prediction. The transformer does not seem to underestimate the coordinates of the MLP model; see the corresponding plots in Figure 4.4 for the MLP head model. This is also observed in the Bland-Altman error plots in figure A.3, where most coordinates' average error is close to zero. Another pattern observed here is that the model is more likely to underestimate when the true value is higher.

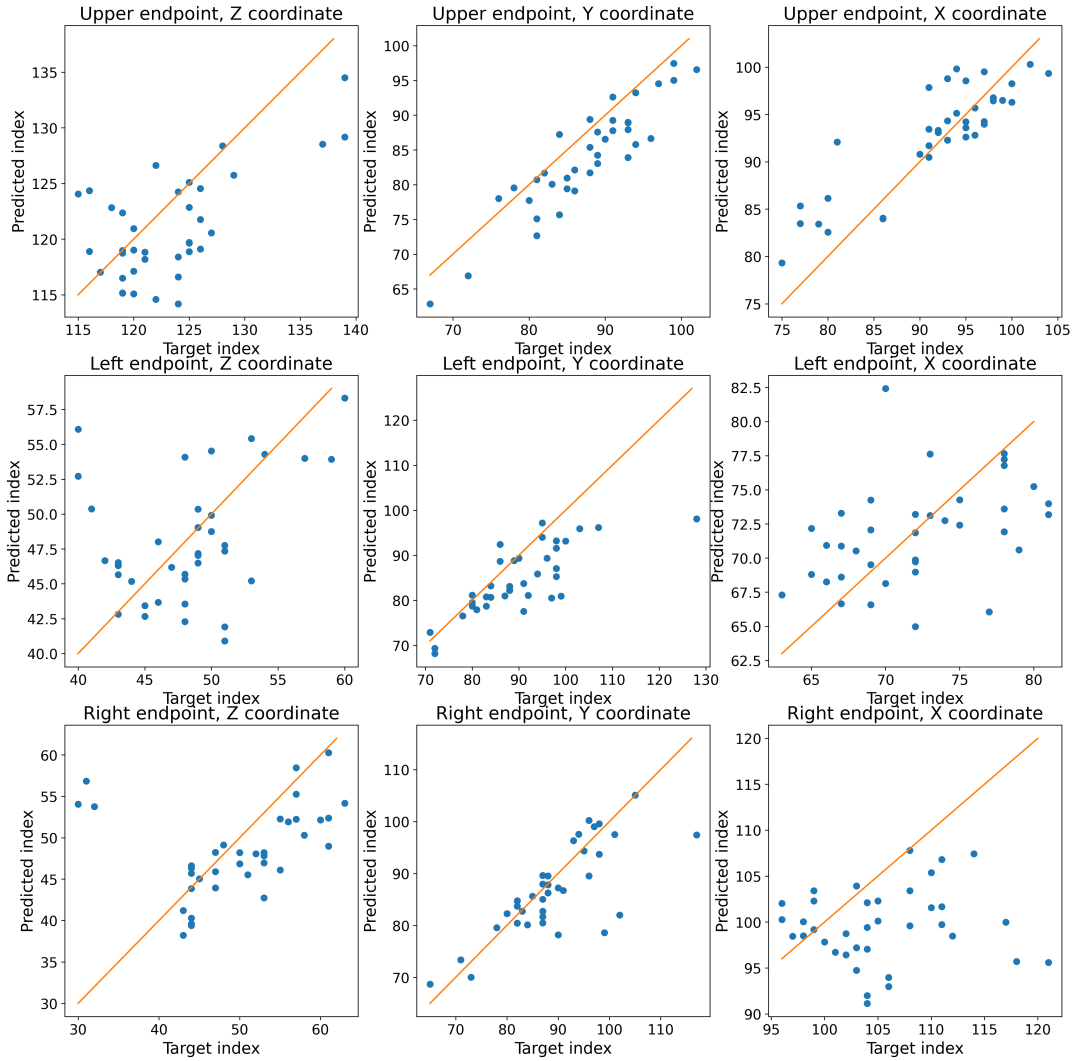



**Figure 4.9:** Scatter plot of predicted and target values of the detection model with transformer head for each point coordinate, where the orange lines represent when predicted values are equal to the target.

Finally, a $R^2$ score equal to 0.926 was calculated. This is slightly higher than the detection model with an MLP head.

An example of the predicted and target proximal endpoint is shown in Figure 4.10. For this CT, the predicted endpoint seems to be at the correct height but slightly to the right compared to the true value.

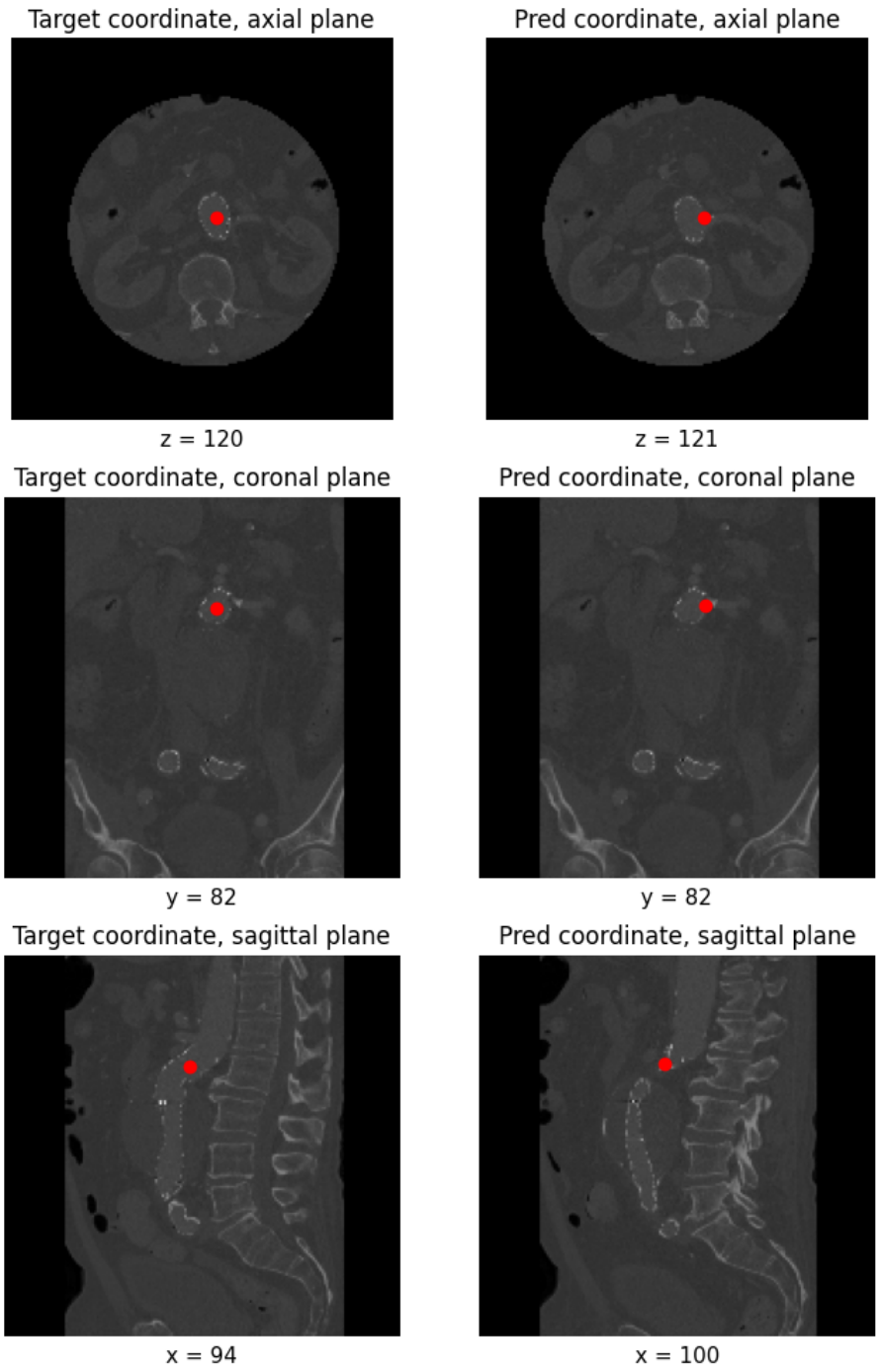



**Figure 4.10:** Target and predicted proximal endpoint for the detection model with a transformer head, shown by their positions in each anatomical plane. The slice and endpoint positions are also written by their exact pixel index.

The saliency map for predicting the proximal endpoint of the same CT volume is presented below in Figure 4.11

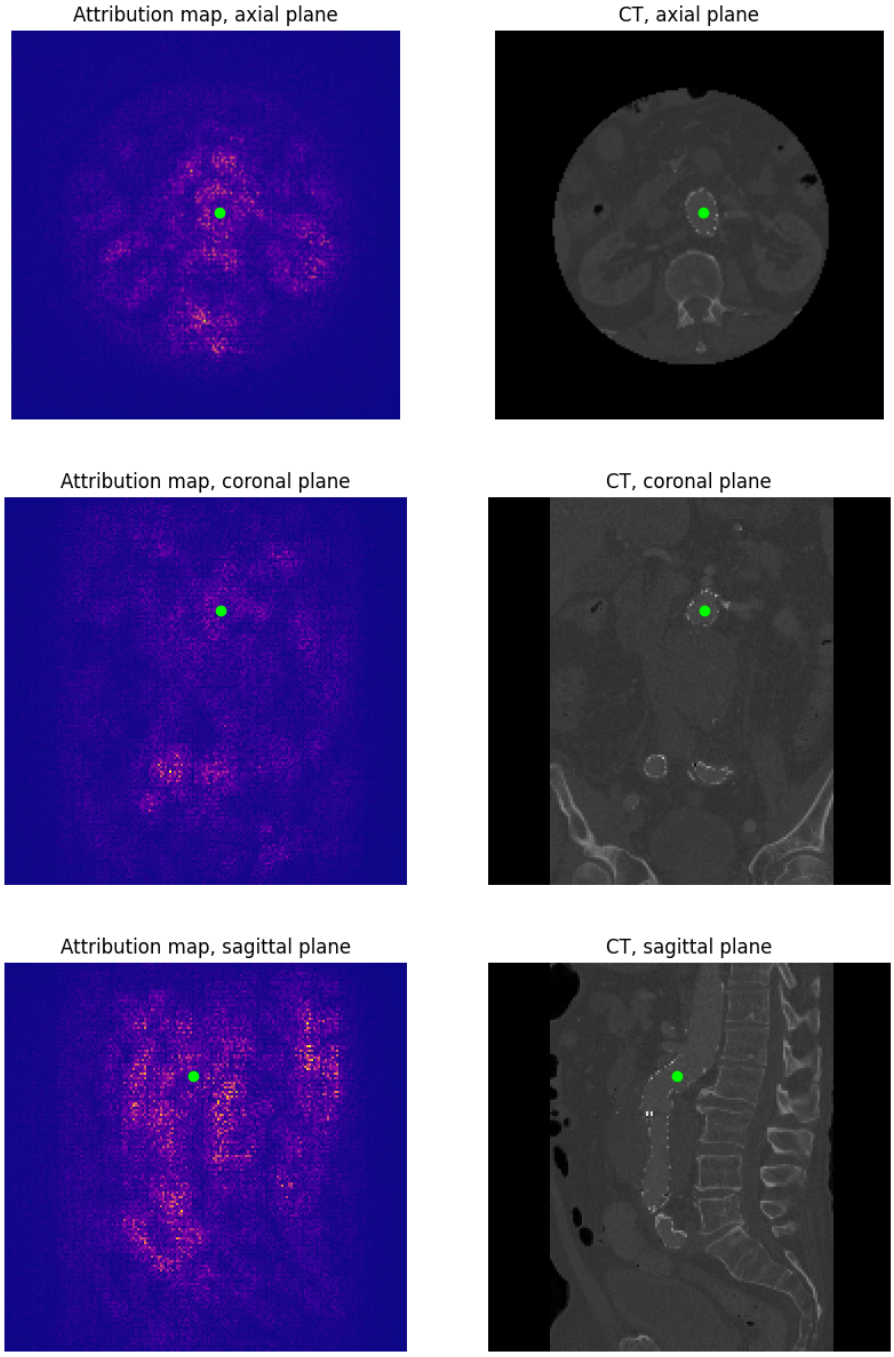



**Figure 4.11:** Attribution map of the detection model with a transformer head predicting the three coordinates of the proximal endpoint, shown in each anatomical plane along with the corresponding slice of the input CT. The slices are given by the ground truth endpoint position, which has been marked by the green point.

As can be seen in Figure 4.11 the model seems to have a high focus on relatively few spots in the image. For example, areas with kidneys, spine, and stents are particularly salient. This could suggest that the transformer based model disregards areas of high contrast in favor of relevant anatomical features, unlike the model with an MLP head.

The model additionally gives high attribution to the lower middle parts of the coronal plane, which seems odd considering that it is meant to predict the proximal endpoint. However, the model was trained with the edge loss, presented in Equation 2.1, that considers the relative distance between the endpoints. Therefore, the model has been trained to reason about the position of the proximal endpoint together with the other endpoints. Consequently, the distal zones have high attribution for predicting the proximal endpoint.

Below, the attribution map is shown layed on top of the input CT image in Figure 4.12. The sagittal plane of the attribution map shows that part of the spine and areas left of the stent have as high saliency as the stent itself, but in the axial plane, the region around the stent has high saliency.

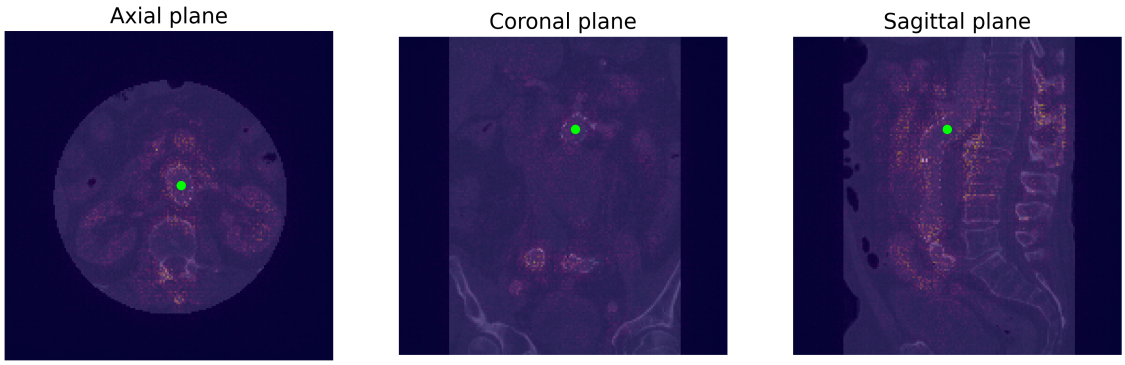



**Figure 4.12:** Attribution map overlay of the detection model with transformer head predicting the three coordinates of the proximal endpoint, shown in each anatomical plane along with the corresponding slice of the input CT. The slices are given by the ground truth endpoint position which has been marked with the green point.

### 4.4.3 Augmentation evaluation

A separate training was executed with minimum augmentations to evaluate its effect on the model's performance for the detection task. The model can not be fairly trained and assessed with no augmentation since the CTs were centered around the stent in the z-direction, as seen in Figure 3.4. Therefore, the minimum amount of augmentation was to keep the random axial cropping but remove the other spatial and intensity augmentations. Given only this minimum augmentation, the model performed as seen in the table below. An $R^2$ score of 0.826 was calculated.

| **Endpoint** | **Euclidian distance (mm)** |
|---|---|
| Proximal | 33.4 ± 6.8 |
| Left distal | 31.9 ± 9.3 |
| Right distal | 34.1 ± 15.4 |
| All | 33.1 ± 11.0 |

**Table 4.9:** Euclidian error between the true endpoints and the prediction by the detection model with a transformer head, trained with minimal augmentations; only random cropping in the z-direction.

All of the metrics in Table 4.9 and the $R^2$ score is worse than the transformer based detection model trained with full augmentation in Section 4.4.2. In fact, the transformer-based model with minimal augmentation performs even worse than the detection model with an MLP head trained with full augmentation in Section 4.3.2 on all metrics. By applying the full augmentation, the average Euclidean error was reduced by about 58% for the detection model with the transformer head. This shows that random affine and noise augmentations benefit the model's generalization, as expected.

## 4.5 ROI Seal Zone Length Regression

The final model in the sequential approach was trained to predict the seal zone lengths from smaller ROI patches. The regions of interest were created by cropping the full-size images with the ground truth endpoints as centerpoints. Individual predictions were made for each ROI, thus the model only had one output per input.

### 4.5.1 Training

The parameters used for training the regression model are presented in Table 4.10 below. These parameters are identical to the baseline and MLP detection models, except for a slightly larger batch size enabled by the smaller input size.

**Table 4.10:** Training parameters for the regression model with ROI input.

| **Parameter** | **Value** |
|---|---|
| Backbone learning rate | $5 \cdot 10^{-6}$ |
| Backbone weight decay | $2.5 \cdot 10^{-4}$ |
| Head learning rate | $10^{-4}$ |
| Head weight decay | 0.0025 |
| Dropout | 0.05 |
| Unfreeze thresholds (val error) | 25, 19, 16, 13, 11 |
| Batch size | $3 \cdot 12$ |

The model was trained until epoch 552, when early stopping set in. Furthermore, the training and validation mean absolute error were monitored, shown in Figure 4.13.

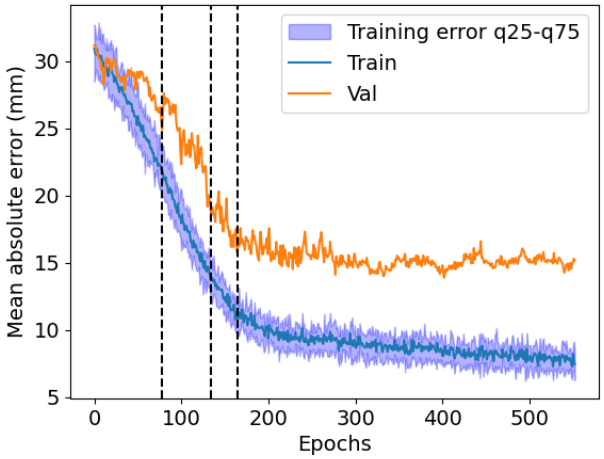



**Figure 4.13:** Training and validation mean absolute error during training of the ROI regression model, where the black vertical lines represent unfreezing of each part of the backbone.

In the figure above, it can be seen that the training and validation decrease at a high rate until around 150 epochs where they slow down. The validation error stagnates at around 14-17 mm while the training error steadily decreases.

### 4.5.2 Evaluation

The ROI model was also evaluated on the test set. Since each patch surrounding an endpoint was treated as an individual input, the absolute error is aggregated over all three endpoint locations.

| **Statistic** | **Mean absolute error (mm)** |
|---|---|
| Mean | 14.5 ± 11.3 |

**Table 4.11:** Error between true and predicted seal zone lengths of the ROI regression model.

The error is high in the table above but slightly lower than the baseline. Overall, it seems that the model is not learning properly, but to further investigate the performance, the predicted values were plotted against the true values in Figure 4.14.

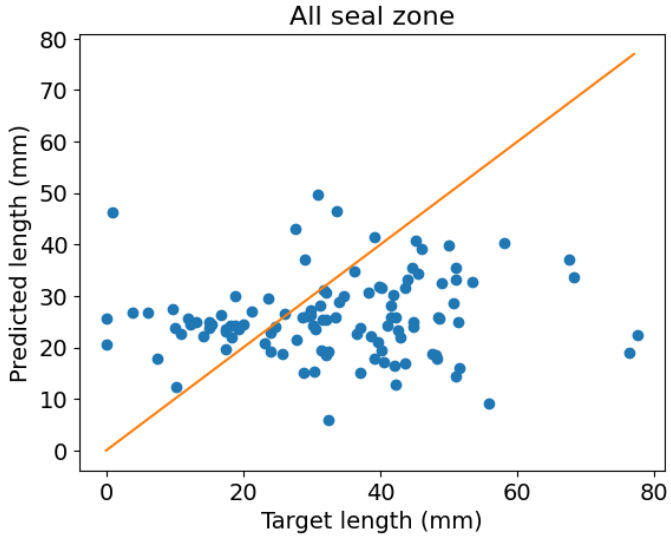



**Figure 4.14:** Scatter plot of target values and predicted values of the baseline model for all seal zones, where the orange line represents when predicted values are equal to the target

As can be seen in Figure 4.14 and Table 4.11, the model's performance is quite poor, as it fails to generalize. Instead of following the ideal orange line, predictions seem to be unrelated to the targets, possibly predicting an average value with some variance. This is further demonstrated by the Bland Altman plot in Figure A.4 in Appendix A, where there is a clear diagonal decreasing trend in the error, suggesting predictions around a constant value.

# 5

# Discussion

This chapter reviews and discusses the results from an analytical and critical point of view. The discussion helps to highlight the impact of the dataset, the chosen models, and data processing methods from Section 3, as well as the implications of the results presented in Section 4.

## 5.1 Dataset curation

A crucial step in the project was manually inspecting and processing the raw clinical data for machine learning. A few notable difficulties came up during these steps. First of all, the annotations and CT images came in formats that were unfit for processing in Python. To begin with, the CTs were converted from DICOM slices to 3D volumes. During the process, a few faulty CTs were discovered. Errors occurred when the slices for the CT could not be aggregated into a volume due to orientation variations.

The centerlines, on the other hand, were provided as XML files that were incompatible with the software for viewing them. Therefore, the centerlines' positions had to be recalculated into their corresponding image's coordinate system using the image metadata. However, in some peculiar cases, the axial coordinates were offset by the height of the image. In other cases, the centerline was visibly off-center by an unknown distance, and thus they had to be discarded. Consequently, to verify that the new centerline position was correct, the annotation was visually inspected inside its corresponding image. Additionally, the possibility that annotations were erroneous was ruled out by viewing the centerlines in the software they were made in. Finally, the presented pipeline successfully converted 92% of available exam data into usable formats.

Despite the complicated data-curating process, a relatively large dataset could be presented. Compared to the presented related work, it could be one of the largest datasets created for diagnosing post-EVAR complications with ML. Related papers have the following number of patients in their datasets: 273 [52], 191 [7], 70 [5], and 56 [8]. The dataset created in this project has 143 patients and 399 CTs, which could place it anywhere from the median to the largest yet.

Finally, the curated dataset was successful in machine learning, as two models could learn and generalize from it. It could, therefore, be helpful in upcoming projects and new learning tasks within the field of post-EVAR complication detection.

## 5.2 Data Selection and Split

As stated previously, each patient has been examined up to three times after EVAR surgery. Naturally, the three CTs from one patient have a high correlation since the anatomy does not change significantly over time. The correlation between CTs from one patient is likely higher than the correlation between CTs from different patients. Therefore, it was crucial to ensure that all exams from a single patient belonged to the same dataset, otherwise the situation would be akin to a data leak between datasets. The consequence of not keeping patient CTs together would be an unfair evaluation where superior performance on the test set would be attributed to generalization.

Thus, the exams from one patient were kept together in one dataset. Consequently, for every CT in a given dataset, there are about two other similar CTs in the same dataset. In the most extreme case, the effective sample size is 1/3 of what it appears to be. However, it is unclear how much informational overlap there is between the CT images for one patient. Nevertheless, claiming that the dataset was curated with data from 399 examinations could be misleading, as this implies that the examinations are independent.

## 5.3 Endpoint centering as a preprocessing step

Since the CTs came in differing sizes, they had to be standardized to a common size for training. In the $y$ and $x$ dimensions, the images were trivially cropped or padded equally on both sides to 176 pixels. However, this process was more complicated in the axial dimensions due to the inconsistent position of the stent in this dimension. As a consequence, the stent wasn't always centered and could be accidentally removed with even cropping. A larger field of view would suffice to resolve the issue, but smaller volumes are ideal for faster training. Therefore, the images were instead consciously cropped or padded in relation to the position of the stent endpoints. Centering around the stent caused a data leak from the endpoint positions to the volume they originate from, which could unrealistically simplify the task.

To remove this leak, random translation for a total of 100 mm was introduced as an augmentation of the training samples. The validation and test set had no random translation applied to them, which means that while the models could not learn from the data leak, it could still influence the evaluation results. More specifically, a biased model could achieve higher validation and evaluation scores by simply gravitating toward the mean of the random translation. This might be what happened to the MLP head detector since figure 4.3b shows that the validation error is lower than the

training error. Fortunately, this is unlikely to be a problem since it is, if anything, worse at predicting the z-coordinate than the other coordinates. Thus, it is doubtful that the model has been able to take advantage of the data leak for the validation scores.

## 5.4 Comparison of the two models for endpoint detection

The two models trained for the endpoint detection tasks both performed well on the test set such that a clear relation between target positions and predicted positions could be seen. When viewing the models' training plots, a few conclusions can be drawn. The MLP model is more biased as it fails to understand the augmentations of the training set. The transformer-based model, on the other hand, performed much better on the training set. Unfortunately, this performance improvement did not carry over to the validation set, indicating higher variance. However, the model with a transformer head performed better when comparing their $R^2$ values. Additionally, the transformer based model had a more focused attribution map for predicting the proximal endpoint, which suggests a superior anatomical understanding. This is likely due to the specialized nature of the Spine-Transformer head which was specifically made to detect landmarks in medical volume data [26]. The MLP head, on the other hand, is not adapted to work better on medical volumes as it was instead chosen for being the most common type of prediction head for convolutional networks.

A few aspects of the Spine Transformer architecture make it perform well in the endpoint detection task. The transformer head's prominent advantage is the voxel and positional encoding. The encoding allows the transformer to receive and maintain a more coherent representation of the feature map from the backbone. It also incorporates the position from where the representation originated. Furthermore, Spine-Transformer has a residual connection that bypasses the majority of the head and allows information and gradients to have a stronger connection to the output. This allows for faster learning and simple reasoning for simple features. Lastly, the transformer head model was trained with edge loss, which the MLP head model was not. This in itself could be advantageous since it encourages the model to reason about the endpoints in relation to each other.

Despite all of these advantages, the transformer head model only performs 13% better on Euclidean distance and has a slightly better $R^2$ score than its MLP counterpart. This highlights the viability of such a general and "out of the box" model. It is also worth mentioning that the MLP head model had its hyperparameters manually tuned to a larger extent than the transformer head. Despite the transformer head receiving less hyperparameter tuning, it performed better.

## 5.5 Baseline model architecture for regression compared to detection

In this project, the architecture consisting of a ResNet backbone, feature map reduction, and an MLP head was used for various tasks with varying performance. As previously mentioned, the model trained on the detection task performed relatively well on the test set. On the other hand, the baseline model applied to full-volume seal zone length regression and ROI seal zone length regression performed poorly. Architecturally, the three MLP-based models are almost identical, with slight differences in feature map reduction and output layers, which can partially be seen in figure 3.5. The largest difference is the feature map reduction layer, where the ROI model is the least restrictive. However, since the endpoint detection model generalized better than the ROI regression model, one can conclude that the issue is unlikely to be caused by this reduction.

The next difference is the input. The ROI seal zone length regression model was trained on smaller patches while the two others were trained on the full-size recentered CTs. This might have caused an issue for the ROI model since the small size may exclude relevant information for determining the seal zone length. The ROIs were $128 \times 128 \times 128$ mm in size and centered around the endpoints. Since the endpoint marks one end of the seal zone, at most, 64 mm of the seal zone would be visible in the patch. The limited size would not be a problem for most patients since the average seal zone length for all patients is 30 mm. However, a few patients have much longer seal zones, up to 80 mm, which means their seal zones do not fit into the ROI. The implication is that the beginning of the seal zone is no longer visible, which complicates the regression of the seal zone lengths.

However, the most significant difference between the three models was not the architecture or the preprocessing; it was the task for which they were trained. From an intuitive perspective, it would be relatively easy for a human to find the stent and subsequent endpoints in a CT volume. This could even be done by someone who isn't a radiologist, as limited knowledge is sufficient for the task. On the other hand, measuring the seal zone length is a much more complex task. More specifically, the seal zone length is characterized by how far the stent edge is in contact with the artery wall before the aneurysm begins and, therefore, requires a higher degree of expertise and precision. Consequently, better methods are needed to predict the seal zone lengths.

## 5.6 Endpoints as Indices in the Volume

When preprocessing was done, the CTs were recentered, and the endpoint coordinates were converted into indices in the volume. Since the detection models are trained against these coordinate values, some inaccuracy could occur as the spacing is 2 mm. Although this introduces some potential model error, the effect is likely minimal and would only become a problem if the goal is for the absolute error to be under 2 mm or one pixel.

## 5.7 Effectiveness of the Sequential Approach

The sequential approach was proposed in this thesis as a way to improve the accuracy and interpretability of seal zone failure diagnosis. The first step in the approach, detecting the stent endpoints, was relatively successful. Through training the detection model with a Spine Transformer head, it could be shown that endpoint detection can be learned with a relatively low error. The step afterward, seal zone length regression based on ROIs around the endpoints, could neither be proven nor disproven to be better than the baseline. The results of the two models trained for seal zone length regression are relatively similar, showing that the models could not, or did not have the ability to, learn the task. Thus, it is unclear whether the seal zone regression improved with the sequential approach. Therefore, a conclusion cannot be reached as to whether the sequential approach improved the accuracy of diagnosis.

Interpretability could also not be assessed since the regression models performed so poorly. A model's interpretability only becomes relevant when it performs competently enough to be implemented into workflows and used by healthcare workers; a point that the implemented approach did not reach. For the same reason, the final classification step was not included, so the classification accuracy of the two approaches could not be compared.

On the bright side, attempting the sequential approach compared to the baseline led to a different insight. Since the MLP model could detect endpoint positions but not regress seal zone measurements, something can be said about the task. That is, seal zone length regression seems harder than endpoint detection or, at the very least, needs a drastically different approach. Therefore, the sequential approach brought a valuable understanding of how to continue development and further improve models for the automatic detection of loss of seal.

## 5.8 Future work

Some improvements could be made to the work in this thesis, both to address the concerns raised during the discussion and to expand the method. The future work is presented below, categorized into smaller sections.

### 5.8.1 Improve the ROIs

As mentioned in section 5.5, it could have been beneficial to have larger ROIs, allowing the entire seal zone to be in the ROI. To begin with, the ROIs should be large enough to fit even the largest of seal zones. A size should be chosen that balances the trade-off between including relevant information and excluding background noise. Naturally, the optimal size could be found through experimentation. However, consulting an expert would likely prove time-efficient while providing a size sufficiently close to the optimal.

An improvement could also be to increase the resolution of the ROIs compared to the entire volumes, as was done by Krüger *et al.*[39]. This would allow for a more detailed feature map from the backbone, which might be needed to make such detailed measurements. The characterization of the seal zones is quite complex as it involves finding the point where the artery wall starts separating from the stent wall. Greater detail benefits radiology experts analyzing CT images and could likely help the model as well.

### 5.8.2 Extending the Sequential Approach

Since predicting the seal zone length of the three stent endings turned out to be a complex task, it might be necessary to introduce additional steps in the sequential method before performing regression. One potential improvement is to predict both ends of the seal zone. This could be helpful as the most difficult task in estimating the seal zone length is finding the exact point where the artery wall lets go of the stent edge.

By predicting the seal zone starting point in addition to the endpoint, the ROI could be cropped exactly around the seal zone. Alternatively, the seal zone length regression could be skipped by estimating the distance between the start and endpoint as the seal zone length. Annotations for the seal zone start could be made with a contour similar to the one marking the end of the stent.

Another possible addition to the sequential approach is segmentation. Some cascaded methods utilized to analyze medical images have segmentation as a part of the sequence [39]–[41]. Segmentation is a task that has been shown to work well with medical images and becomes an effective approach in a cascade as it essentially removes all the irrelevant parts of the image. For example, instead of predicting the endpoints as the first step, the stent could be segmented. The only limitation to using segmentation is that it would require new, time-consuming annotations.

An additional way to extend the sequential approach would be to include centerline prediction as a previous step to endpoint detection. Since the centerline is already a required annotation in the protocol by Andersson *et al.*, no further annotations are needed. According to the hypothesis laid out in section 1.2, the endpoints should be easier to find, given the location of the centerline. Just as ROIs could be created from the endpoints, the larger ROI could be formed around the centerline. The practical aspect of implementing a centerline prediction model is further discussed in section 5.8.3.

### 5.8.3 Detecting the Centerline

Currently, the centerline has to be manually annotated since some medical software is confused by the high intensity of the stent. In this project, it has been shown that a model can learn to locate three of the centerline points, the endpoints. Therefore, assuming that the rest could be found similarly is reasonable. Furthermore, the original architecture of the Spine-Transformer was created to localize the position

of several vertebrae [26]. The sequence of positions in the spine is similar to the centerline of the stent. Therefore, applying Spine-Transformer[26] to construct the presented dataset's centerlines could be worthwhile. Automating the centerline generation could be helpful by reducing the time needed for radiologists to perform the task, thereby improving one step in the post-EVAR analysis method.

However, there are major differences between the centerline and the endpoints. Each endpoint represents the position of a seal zone, while the centerline coordinates can only be meaningfully viewed as a collection. Furthermore, the centerline annotations are not equally spaced or have a consistent start and end. This makes each point in the annotated centerline an unreliable target. One possible solution would be to use the Chamfer Distance as a loss function [53]. The Chamfer Distance is a distance metric between two sets of points, defined as $A$ and $B$ in the equations

$$\begin{aligned} A &= \{a_1, a_2, \ldots, a_n\} \subset \mathbb{R}^d, \\ B &= \{b_1, b_2, \ldots, b_m\} \subset \mathbb{R}^d, \\ \mathrm{CD}(A,B) &= \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}\|a-b\|_2 + \frac{1}{|B|}\sum_{b\in B}\min_{a\in A}\|b-a\|_2. \end{aligned} \tag{5.1}$$

For machine learning purposes, $A$ should be viewed as a set of predicted points and $B$ as a set of corresponding ground truth points. Then, by allowing $A$ to be variable, the best fit for $B$ can be found through

$$A^* = \arg\min_A \ \mathrm{CD}(A,B). \tag{5.2}$$

The optimal fit $A^*$ covers the same shape as $B$ to the degree allowed by $n$. Furthermore, it can be generally said that each point in $A^*$ has a close point in $B$ and each point in $B$ has a close point in $A^*$ [53].

### 5.8.4 Include Cross Validation

An effective way to verify the generalization of a model would be to use cross validation. Cross validation is a well-established method for increasing the numerical stability of evaluation metrics. Several train, validation, and test splits of the dataset are created to cross-validate the models. The entire dataset can be evaluated by training the same model on multiple splits without overlapping the seen and unseen data. Due to time and resource constraints, cross validation was not performed for this project. Nevertheless, this is something that should be done so that the variance in the evaluation metrics can be reduced as far as possible. Currently, the evaluation is only based on 14 patients for a total of 39 CTs.

### 5.8.5 Full Implementation of the Spine-Transformer Approach

In this project, several deviations were made from the approach by Tao *et al.* To begin with, the original Spine-Transformer [26] was trained on patches of CTs to avoid resizing their dataset to a common size before training. Consequently, each patch contained a subset of the spine, eliminating the need for cropping around the entire spine.

If such an approach had also been implemented in this project, the need to center the CTs in the axial dimension around the stent would have been eliminated. This would further address the concerns raised regarding the recentering introducing a small data leak, described in Section 5.3. Furthermore, the augmentations described in 3.3.4 would not be restricted by the endpoints remaining in the volume. This would open up for experimentation with more extreme spatial augmentation and cropping. It would also, for similar reasons, address the problem raised in Section 5.5 concerning the limitations placed on the ROI volumes by restrictive preprocessing.

### 5.8.6 Improved Regression Model

The results showed that the performance of the regression model chosen for predicting the seal zone length was poor. This suggests a more advanced model could be necessary to perform this task. In the last couple of years, there has been a shift towards using and experimenting with vision-transformers (ViT). The model has been seen excelling at many tasks in medical computer vision [54]. Furthermore, the same observation has been made in this thesis—the transformer based model substantially improved detection with minimal hyperparameter tuning. Therefore, employing a ViT for the regression tasks could prove beneficial. Additionally, it can be observed that the training process is more stable for the transformer-based detection model when comparing figure 4.3b and 4.8b.

### 5.8.7 Calculating the Effective Sample Size

To fairly compare this project's results with another, it would be essential to know details about the respective datasets. One of, if not the most important, improvements would be to estimate the effective sample size. As was called into question in Section 5.2, the effective sample size of the presented dataset could theoretically be between 143 and 399. Knowing the effective sample size would also help assess if resources should be focused on adding new patients or fixing the faulty CTs of patients with some correct CTs.

Moreover, there is a straightforward way to gauge whether the effective sample size is close to 143 or not. To begin with, the existing training, validation, and test could be reduced to one exam per patient. Then, the transformer based detection model would be trained on the reduced training set. Finally, the original and new models would be evaluated on the new, reduced test set. By comparing the results, an assessment could be made whether adding CTs from the same patients improves performance.

### 5.8.8 Sequential Training

Once a better performing model for seal zone length regression of the patches has been found, a cascade training step could be added. First, the detection and regression models could be trained and optimized separately. After that, the parameters of both models would be trained towards the final output, the seal zone length. The regression model would then receive ROI patches cropped based on the predicted endpoints of the detection model. Currently, the assumption is that the endpoints make ideal centerpoints of the ROI patch for seal zone length regression. Combined training could allow this assumption to be slightly adjusted by shifting the predicted positions away from the correct endpoints. By doing so, more relevant information could be captured in the ROI.

However, all intermediate operations must be differentiable to train the models in a cascade. The process of centering ROIs around the endpoints is not differentiable, thus requiring a surrogate gradient.

# 6

# Conclusion

The professed aim of this thesis was to produce an ML approach to predict loss of seal. For this purpose, a dataset of 143 patients totaling 399 CT images was curated, on which two model configurations were trained (see figure 3.1). Neither produced any significant result in the pursuit of seal zone length regression. However, a combination of the MedNet backbone and the Spine-Transformer head was able to locate the seal zones with substantially low error. Therefore, significant advancements have been made towards loss of seal prediction. A conclusion cannot be drawn regarding the interpretability of the sequential approach due to poor regression performance. After eliminating potential causes of the poor generalization, current methods were identified as insufficient for the complex task of seal zone regression. In conclusion, given the presented dataset, the project has demonstrated the potential of the sequential approach through the promising endpoint localization results.

# Bibliography


[1] "Screening för bukaortaaneurysm, socialsyrelsen vetenskapligt underlag." (), [Online]. Available: https://www.socialstyrelsen.se/globalassets/sharepoint-dokument/artikelkatalog/nationella-screeningprogram/2016-6-5-bilaga-1-vetenskapligt-underlag.pdf (visited on 01/21/2025).

[2] M. Prinssen, E. L. G. Verhoeven, J. Buth, *et al.*, "A randomized trial comparing conventional and endovascular repair of abdominal aortic aneurysms," *New England Journal of Medicine*, vol. 351, no. 16, pp. 1607–1618, Oct. 14, 2004, Publisher: Massachusetts Medical Society _eprint: https://www.nejm.org/doi/pdf/10.1056/NEJMoa042002. DOI: 10.1056/NEJMoa042002.

[3] D. Daye and T. G. Walker, "Complications of endovascular aneurysm repair of the thoracic and abdominal aorta: Evaluation and management," *Cardiovascular Diagnosis and Therapy*, vol. 8, S138–S13S156, Suppl 1 Apr. 2018. DOI: 10.21037/cdt.2017.09.17. (visited on 01/21/2025).

[4] C. Sandström, M. B. Andersson, M. Bogdanovic, *et al.*, "Sealing zone failure decreases the long term durability of endovascular aneurysm repair," *European Journal of Vascular and Endovascular Surgery*, Sep. 7, 2024. DOI: 10.1016/j.ejvs.2024.09.007. (visited on 01/21/2025).

[5] S. Talebi, M. H. Madani, A. Madani, *et al.*, "Machine learning for endoleak detection after endovascular aortic repair," *Scientific reports*, vol. 10, no. 1, p. 18 343, 2020.

[6] M. Andersson, C. Sandström, O. Stackelberg, *et al.*, "Editor's choice – structured computed tomography analysis can identify the majority of patients at risk of post-endovascular aortic repair rupture," *European Journal of Vascular and Endovascular Surgery*, vol. 64, no. 2, pp. 166–174, 2022. DOI: https://doi.org/10.1016/j.ejvs.2022.04.042.

[7] S. Hahn, M. Perry, C. S. Morris, S. Wshah, and D. J. Bertges, "Machine deep learning accurately detects endoleak after endovascular abdominal aortic aneurysm repair," *JVS-Vascular Science*, vol. 1, pp. 5–12, Jan. 1, 2020. DOI: 10.1016/j.jvssci.2019.12.003.

[8] Q. Coatsaliou, F. Lareyre, J. Raffort, *et al.*, "Use of artificial intelligence with deep learning approaches for the follow-up of infrarenal endovascular aortic repair," *Journal of Endovascular Therapy*, p. 15 266 028 241 252 097, May 9, 2024. DOI: 10.1177/15266028241252097.

[9] B. Long, D. L. Cremat, E. Serpa, S. Qian, and J. Blebea, "Applying artificial intelligence to predict complications after endovascular aneurysm repair," *Vascular and Endovascular Surgery*, vol. 58, no. 1, pp. 65–75, Jan. 1, 2024. DOI: 10.1177/15385744231189024.

[10] T. M. Buzug, "Computed tomography," in *Springer handbook of medical technology*, Springer, 2011, pp. 311–342.

[11] E. S. Pretorius, "Introduction to ultrasound, ct, and mri," in *Radiology secrets plus*, Elsevier, 2011, pp. 13–18.

[12] D. R. Varma, "Managing DICOM images: Tips and tricks for the radiologist," *The Indian Journal of Radiology & Imaging*, vol. 22, no. 1, pp. 4–13, 2012. DOI: 10.4103/0971-3026.95396. (visited on 05/05/2025).

[13] N. Adaloglou, *Understanding coordinate systems and DICOM for deep learning medical image analysis*, en, Jul. 2020. [Online]. Available: https://theaisummer.com/medical-image-coordinates/ (visited on 05/05/2025).

[14] J. H. Kim, Y. K. Cho, T.-S. Seo, *et al.*, "Clinical outcomes for endovascular repair of abdominal aortic aneurysm with the seal stent graft," *Journal of Vascular Surgery*, vol. 64, no. 5, pp. 1270–1277, 2016.

[15] N. Shazeer, A. Mirhoseini, K. Maziarz, *et al.*, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[16] M. J. Willemink, W. A. Koszek, C. Hardell, *et al.*, "Preparing medical imaging data for machine learning," *Radiology*, vol. 295, no. 1, pp. 4–15, 2020. DOI: 10.1148/radiol.2020192224.

[17] S. Linnainmaa, "Taylor expansion of the accumulated rounding error," *BIT Numerical Mathematics*, vol. 16, no. 2, pp. 146–160, 1976.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[19] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[20] J. Schmidhuber, "Annotated history of modern ai and deep learning," *arXiv preprint arXiv:2212.11279*, 2022.

[21] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.

[22] R. Ribani and M. Marengoni, "A survey of transfer learning for convolutional neural networks," in *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, 2019, pp. 47–57. DOI: 10.1109/SIBGRAPI-T.2019.00010.

[23] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[24] O. Elharrouss, Y. Akbari, N. Almadeed, and S. Al-Maadeed, "Backbones-review: Feature extractor networks for deep learning and deep reinforcement learning approaches in computer vision," *Computer Science Review*, vol. 53, p. 100 645, 2024.

[25] S. Chen, K. Ma, and Y. Zheng, "Med3d: Transfer learning for 3d medical image analysis," *arXiv preprint arXiv:1904.00625*, 2019.

[26] R. Tao, W. Liu, and G. Zheng, "Spine-transformers: Vertebra labeling and segmentation in arbitrary field-of-view spine cts via 3d transformers," *Medical Image Analysis*, vol. 75, p. 102 258, 2022. DOI: https://doi.org/10.1016/j.media.2021.102258.

[27] A. Sethi *et al.*, "Which backbone to use: A resource-efficient domain specific comparison for computer vision," *Transactions on Machine Learning Research*,

[28] W. Xu, Y.-L. Fu, and D. Zhu, "Resnet and its application to medical image processing: Research progress and challenges," *Computer Methods and Programs in Biomedicine*, vol. 240, p. 107 660, 2023. DOI: https://doi.org/10.1016/j.cmpb.2023.107660.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, Springer, 2016, pp. 630–645.

[30] J. Zhang, M. Liu, and D. Shen, "Detecting anatomical landmarks from limited medical imaging data using two-stage task-oriented deep neural networks," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4753–4764, 2017. DOI: 10.1109/TIP.2017.2721106.

[31] J. M. H. Noothout, B. D. De Vos, J. M. Wolterink, *et al.*, "Deep learning-based regression and classification for automatic landmark localization in medical images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 4011–4022, Dec. 2020, Conference Name: IEEE Transactions on Medical Imaging. DOI: 10.1109/TMI.2020.3009002. (visited on 02/11/2025).

[32] P. Xi, C. Shu, and R. Goubran, "Localizing 3-D Anatomical Landmarks Using Deep Convolutional Neural Networks," in *2017 14th Conference on Computer and Robot Vision (CRV)*, May 2017, pp. 197–204. DOI: 10.1109/CRV.2017.11.

[33] D. Zhu and D. Wang, "Transformers and their application to medical image processing: A review," *Journal of Radiation Research and Applied Sciences*, vol. 16, no. 4, p. 100 680, 2023. DOI: https://doi.org/10.1016/j.jrras.2023.100680.

[34] F. Garcea, A. Serra, F. Lamberti, and L. Morra, "Data augmentation for medical imaging: A systematic literature review," *Computers in Biology and Medicine*, vol. 152, p. 106 391, 2023. DOI: https://doi.org/10.1016/j.compbiomed.2022.106391.

[35] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 545–563, 2021.

[36] *Geometric Transformations — pages.mtu.edu*, https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/geometry/geo-tran.html, [Accessed 12-05-2025].

[37] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, "3d deep learning on medical images: A review," *Sensors*, vol. 20, no. 18,

2020, ISSN: 1424-8220. DOI: 10.3390/s20185097. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5097.

[38] A. Q. Wang, B. K. Karaman, H. Kim, *et al.*, "A framework for interpretability in machine learning for medical imaging," *IEEE Access*, vol. 12, pp. 53 277–53 292, 2024. DOI: 10.1109/ACCESS.2024.3387702.

[39] N. Krüger, A. Meyer, L. Tautz, *et al.*, "Cascaded neural network-based CT image processing for aortic root analysis," *International Journal of Computer Assisted Radiology and Surgery*, vol. 17, no. 3, pp. 507–519, Mar. 1, 2022. DOI: 10.1007/s11548-021-02554-3. (visited on 01/21/2025).

[40] E. Ul Haq, Q. Yong, Z. Yuan, H. Jianjun, R. Ul Haq, and X. Qin, "Accurate multiclassification and segmentation of gastric cancer based on a hybrid cascaded deep learning model with a vision transformer from endoscopic images," *Information Sciences*, vol. 670, p. 120 568, Jun. 2024. DOI: 10.1016/j.ins.2024.120568.

[41] B. Asadi and Q. Memon, "Efficient breast cancer detection via cascade deep learning network," *International Journal of Intelligent Networks*, vol. 4, pp. 46–52, Jan. 2023. DOI: 10.1016/j.ijin.2023.02.001.

[42] Z. Salahuddin, H. C. Woodruff, A. Chatterjee, and P. Lambin, "Transparency of deep neural networks for medical image analysis: A review of interpretability methods," *Computers in Biology and Medicine*, vol. 140, p. 105 111, 2022. DOI: https://doi.org/10.1016/j.compbiomed.2021.105111.

[43] M. Temme, "Algorithms and Transparency in View of the New General Data Protection Regulation," *European Data Protection Law Review (EDPL)*, vol. 3, p. 473, 2017. [Online]. Available: https://heinonline.org/HOL/Page?handle=hein.journals/edpl3&id=512&div=&collection=.

[44] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[45] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, *et al.*, "3d slicer as an image computing platform for the quantitative imaging network," *Magnetic resonance imaging*, vol. 30, no. 9, pp. 1323–1341, 2012.

[46] Z. Yaniv, B. C. Lowekamp, H. J. Johnson, and R. Beare, "Simpleitk image-analysis notebooks: A collaborative environment for education and reproducible research," *Journal of digital imaging*, vol. 31, no. 3, pp. 290–303, 2018.

[47] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[48] W. Falcon and The PyTorch Lightning team, *PyTorch Lightning*, version 1.4, Mar. 2019. DOI: 10.5281/zenodo.3828935. [Online]. Available: https://github.com/Lightning-AI/lightning.

[49] F. Pérez-García, R. Sparks, and S. Ourselin, "Torchio: A python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning," *Computer methods and programs in biomedicine*, vol. 208, p. 106 236, 2021.

[50] N. Kokhlikyan, V. Miglani, M. Martin, *et al.*, *Captum: A unified and generic model interpretability library for pytorch*, 2020. arXiv: 2009.07896 [cs.LG].

[51] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[52] B. Long, D. L. Cremat, E. Serpa, S. Qian, and J. Blebea, "Applying artificial intelligence to predict complications after endovascular aneurysm repair," *Vascular and Endovascular Surgery*, vol. 58, no. 1, pp. 65–75, Jan. 1, 2024, Publisher: SAGE Publications Inc. DOI: 10.1177/15385744231189024.

[53] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Balanced chamfer distance as a comprehensive metric for point cloud completion," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 088–29 100, 2021.

[54] A. Parvaiz, M. A. Khalid, R. Zafar, H. Ameer, M. Ali, and M. M. Fraz, "Vision transformers in medical computer vision—a contemplative retrospection," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106 126, 2023.

# A

# Appendix 1: Evaluation plots

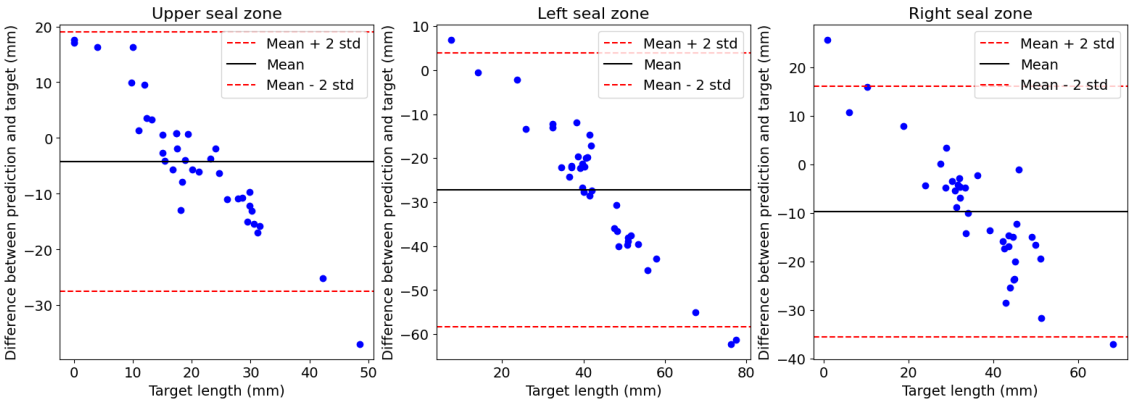



**Figure A.1:** Bland-Altman plot of the predicted and target seal zone lengths by the baseline model.

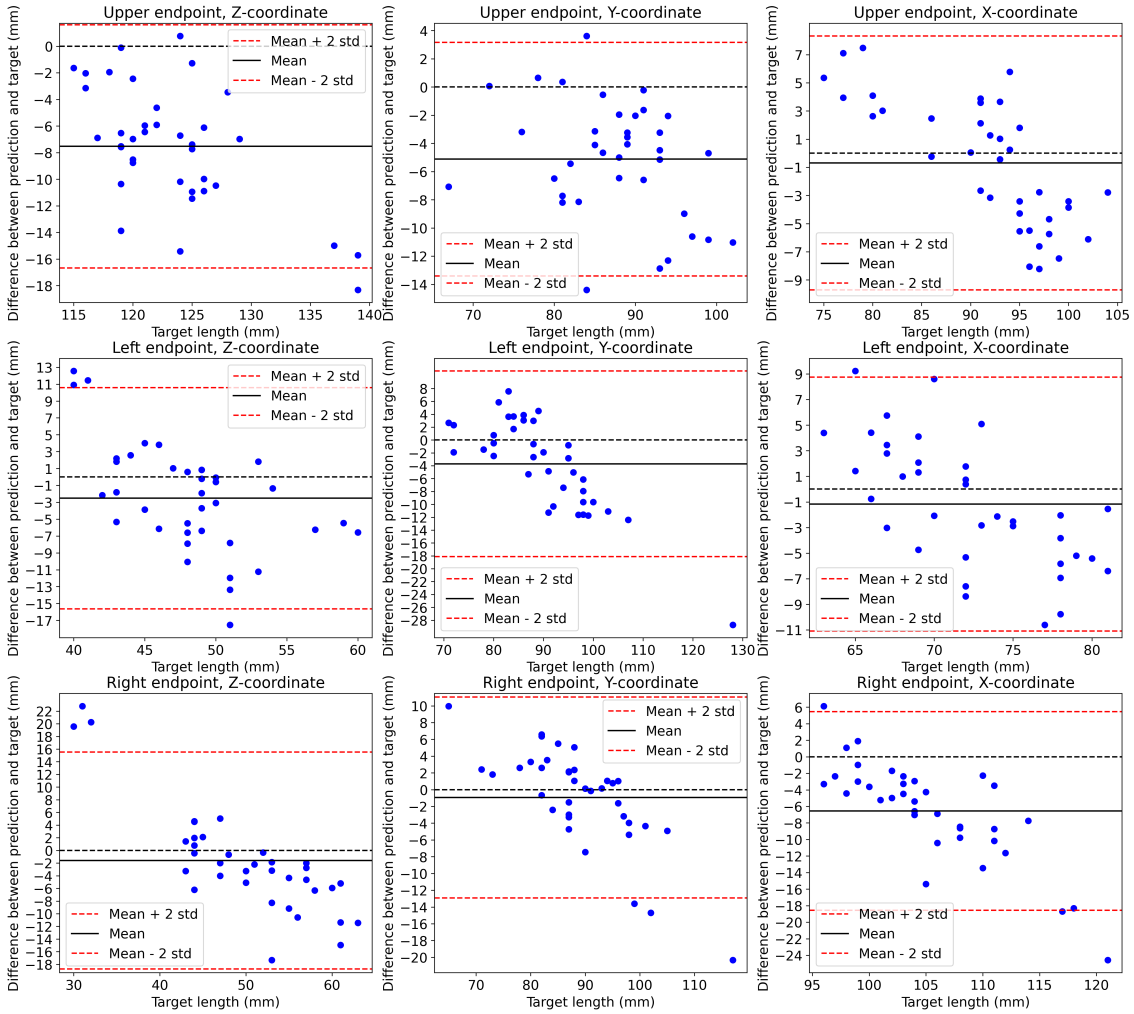



**Figure A.2:** Bland-Altman plot of the predicted and target endpoint coordinates by the detection model with an MLP head.

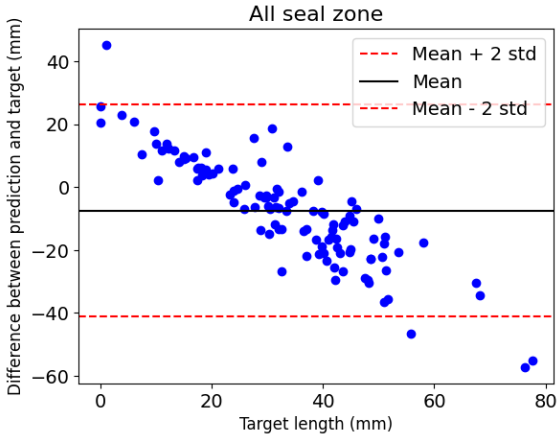



**Figure A.4:** Bland-Altman error plot of the regression model for ROI images. The horizontal black dashed line shows zero error.

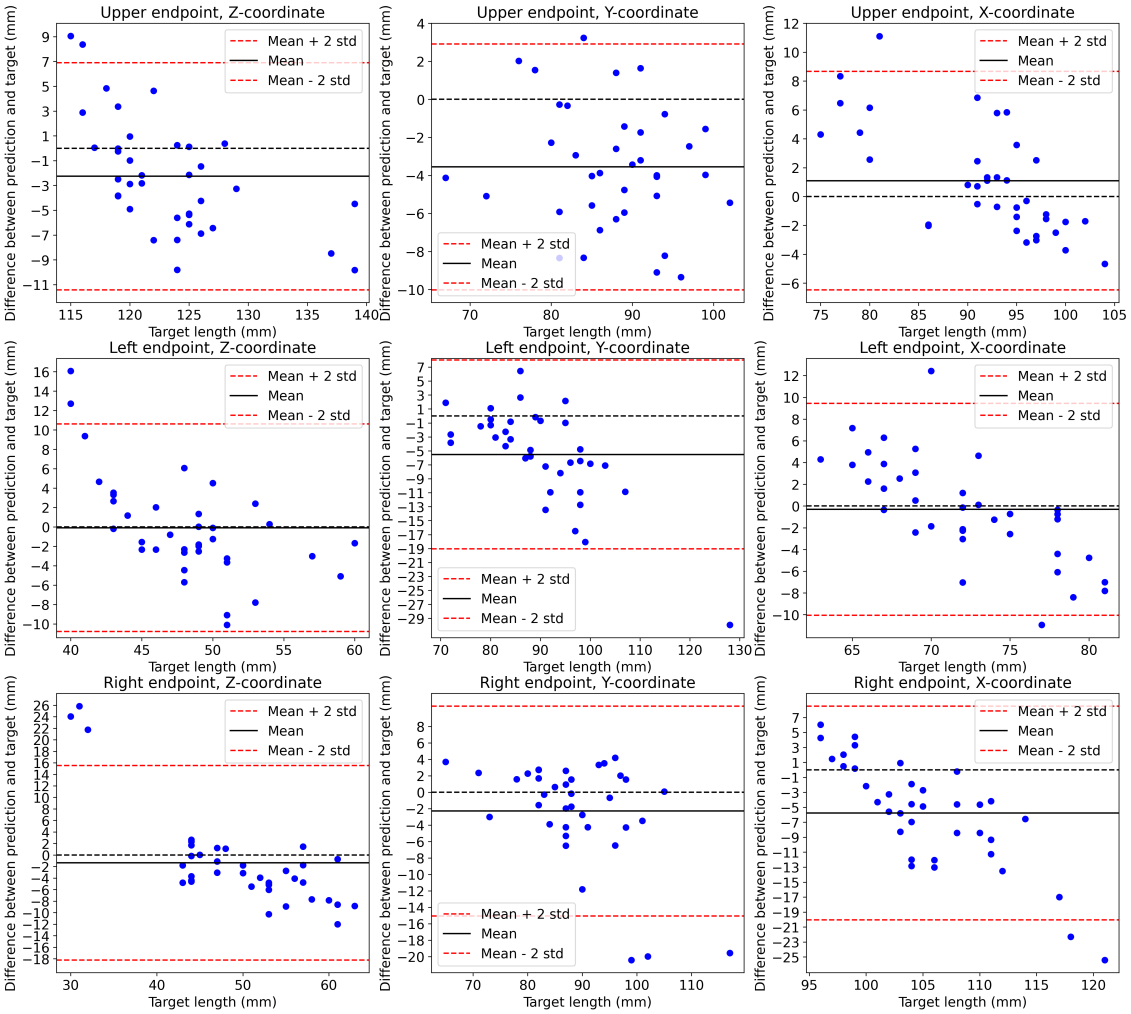



**Figure A.3:** Bland-Altman plot of the predicted and target endpoint coordinates by the detection model with a transformer head.

**DEPARTMENT OF ELECTRICAL ENGINEERING**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden
www.chalmers.se


CHALMERS
UNIVERSITY OF TECHNOLOGY