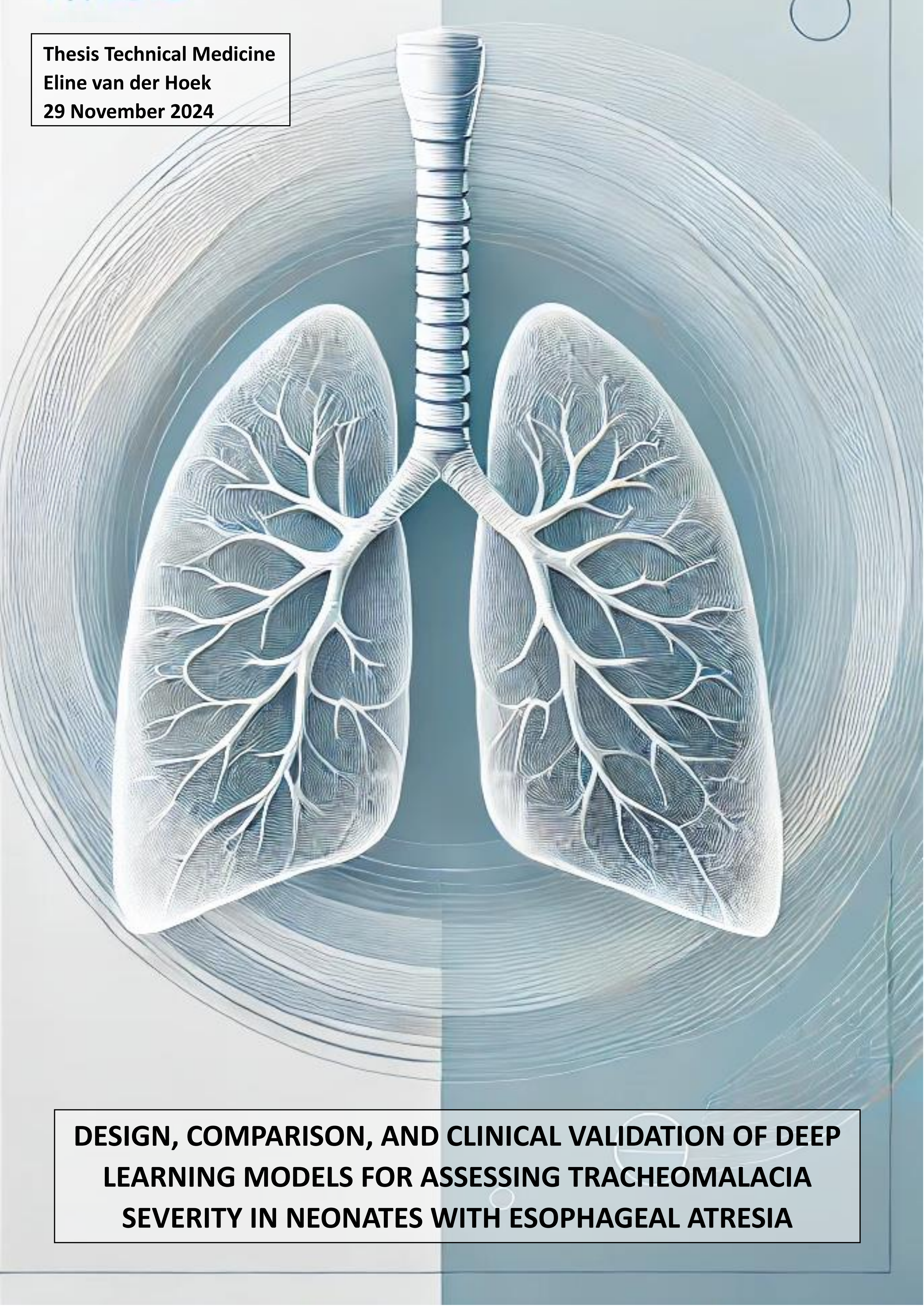
Thesis Technical Medicine
Eline van der Hoek
29 November 2024



DESIGN, COMPARISON, AND CLINICAL VALIDATION OF DEEP LEARNING MODELS FOR ASSESSING TRACHEOMALACIA SEVERITY IN NEONATES WITH ESOPHAGEAL ATRESIA

Design, comparison, and clinical Validation of deep learning models for assessing tracheomalacia severity in neonates with esophageal atresia

Eline van der Hoek

Student number : 4838890

November 2024

Thesis in partial fulfilment of the requirements for the joint degree of Master of Science in

Technical Medicine

Leiden University ; Delft University of Technology ; Erasmus University Rotterdam

Master thesis project (TM30004 ; 35 ECTS)

Dept. of Pediatric Surgery, UMC Utrecht
02th April 2024 – 29th November 2024

Supervisor(s):

Dr. B.C. (Berend) Stoel
Dr. S.H.A.J. (Stefaan) Tytgat
Dr.ir. K.L. (Koen) Vincken

Thesis committee members:

Dr. B.C. Stoel (PhD), Leiden UMC (Chair)
Dr. S.H.A.J. Tytgat (MD, PhD), UMC Utrecht
Dr.ir. K.L. Vincken (PhD), UMC Utrecht
Dr. J. Vlot (MD, PhD), Erasmus MC

Contents

Summary

This thesis investigates and compares the use of deep learning models for automated assessment of tracheomalacia (TM) severity in neonates with esophageal atresia (EA). TM is a condition characterized by weakening of tracheal cartilage, leading to airway collapse. It is especially common in neonates with EA and requires accurate severity evaluation for diagnosis and treatment. Currently, the severity classification is reliant on the bronchoscopist's interpretation, which makes it susceptible to inter-observer variability. The first aim of this thesis was to develop automated image segmentation techniques and validate the results. The second objective was quantifying airway dimensions based on the segmentations.

The study included data from 14 neonates who underwent bronchoscopy, resulting in 127 bronchoscopy images used for analysis. Various pre-processing techniques were applied to improve the quality of input data, including normalization and histogram equalization. The dataset was expanded using data augmentation techniques.

Four deep learning models were evaluated: 1) the standard U-Net model, 2) the Depth-Anything model with processing steps to create segmentations, 3) the U-Net model with Depth-Anything images as input and 4) the U-Net model using both the Depth-Anything and the original images as input (see Figure 1). Performance was primarily evaluated using the dice score, which measures the overlap between predicted segmentations and ground truth. The U-Net model achieved the best results, with a mean dice score of 0.79 on training data and 0.75 on test data, indicating effective segmentation performance. A study comparing clinician assessments of ground truth and model-predicted airway segmentations found no significant difference in accuracy (63% vs 56%, p = 0.616), though notably, 37% of ground truth segmentations were deemed incorrect by clinicians. Linear prediction models for the parameter 'roundness' and the principal component resulted in a correlation of 0.81 and 0.84, and a mean absolute error of 14.40% and 12.73%, respectively, for the standard U-Net model.

In conclusion, this research presents multiple deep learning-based models for the segmentation of airway collapse in bronchoscopy images, with a focus on improving accuracy and clinical relevance. Through the application of U-Net and Depth-Anything models, significant advancements were made in automatic segmentation, providing a useful tool for clinicians in evaluating the extent of airway collapse.

From a future perspective, there is significant potential to integrate the standard U-Net model into clinical settings, where it could play a crucial role in supporting early diagnosis and enabling personalized treatment planning. Several approaches could enhance the model's clinical applicability, including dataset expansion, improved model generalizability, and advanced post-processing techniques. Ultimately, these improvements can be used to create a model that not only quantifies the airway collapse accurately but also serves as a reliable tool for predicting clinical outcomes and guiding treatment planning.

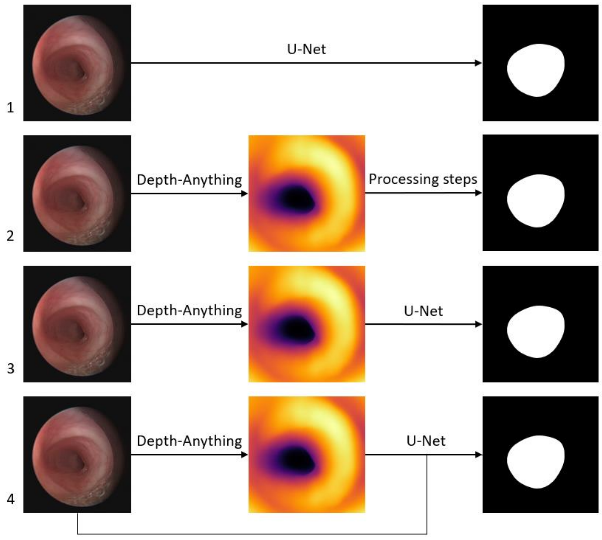


Figure 1. Diagram showing how Depth-Anything and U-Net are used in the 4 different deep learning models.

Abbreviations

2D = two-dimensional
3D = three-dimensional
4D = four-dimensional
AHE = Adaptive Histogram Equalization
CNN = Convolutional Neural Network
CT = Computed Tomography
dA/dB = diameter A (the longest) and diameter B (the shortest)
EA = Esophageal Atresia
ERS = European Respiratory Society
FCN = Fully Convolutional Neural Network
IoU = Intersection over Union
MiDaS = Mixed Data Sampling
MSE = Mean Squared Error
PCA = Principle Component Analysis
ReLU = Rectified Linear Unit
RGB = Red – Green – Blue
ROI = Region of Interest
TEF = Tracheal-Esophageal Fistula
TM = Tracheal Malacia
WKZ = Wilhelmina Children’s Hospital

1. Medical introduction

1.1. Clinical relevance

Tracheomalacia (TM) is the most common congenital tracheal abnormality, with an incidence of 1 in every 2.100 children [1]. However, due to the wide spectrum of nonspecific symptoms and thus many initially misdiagnosed cases, this number is probably underestimated [2]. Tracheomalacia is characterized by an increase in tracheal compliance due to excessive softness of cartilage and flaccidity of the posterior membrane [3]. This softness causes collapse of the central airway, see Figure 2, which results in clinical symptoms. These clinical symptoms can include barking cough, expiratory rhonchi, inspiratory stridor, ineffective cough, blue spells, and reduced clearance of secretions [4]. As a result, TM patients have an increased risk of frequent upper respiratory infections, such as bacterial bronchitis or pneumonia [3, 5]. Moreover, patients can have difficulties with activities that increase the intrathoracic pressure, such as coughing, crying, feeding, forced expiration or lying supine [6, 7]. Tracheal collapse can even lead to life-threatening blue spells.

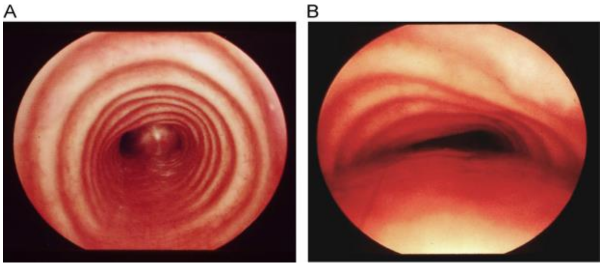


Figure 2. Tracheoscopy of (A) a healthy trachea and (B) severe tracheomalacia with almost complete tracheal collapse during expiration. Image reproduced from Fraga et al (2016) [6]

Congenital tracheomalacia is associated with a wide range of etiologies, but is most commonly present in children with esophageal atresia (EA) and tracheal-esophageal fistula (TEF) [3, 6, 8]. Normally, the esophagus is a direct connection from the mouth to the stomach, see Figure 3A. In patients with EA, the esophagus is interrupted, resulting in a blind upper pouch, see Figure 3B. There are different types of EA, but around 90% of the neonates born with EA have a tracheo-esophageal fistula: an abnormal connection between the trachea and the distal esophagus, see Figure 3C [9]. This could be explained embryologically because both the trachea and esophagus develop from the foregut. Around 87% of the neonates born with an EA suffer from TM, especially those with a tracheo-esophageal fistula [8]. During EA surgery, a suture is placed in this fistula, followed by transecting the fistula. The distal esophagus is now disconnected from the posterior wall of the trachea. The tension of this posterior wall is also removed, which may result in an increased floppiness of this posterior wall [10].

In addition to congenital tracheomalacia (primary TM), tracheomalacia can also develop later in life (secondary TM) as a result of factors such as extended intubation, external pressure, injury, or inflammation. However, secondary TM will be outside the scope of this research project.

There are multiple methods for diagnosing and quantifying the degree of tracheal collapsibility. Radiographs (including comparative inspiratory and expiratory views), computed tomograms, tracheo- or bronchograms, and fluoroscopy have all been used in the diagnosis of TM [4, 7, 11]. Computed tomography (CT) scanning provides high anatomic details but exposes the patient to ionizing radiation. Another downside of using CT is that it is difficult to assess the dynamics of the airways in neonates or very young children because of noncompliance with breathing instructions [7].

Bronchoscopy is currently the gold standard for diagnosing TM. A bronchoscopy is performed under general anesthesia while the neonate is spontaneously breathing, without the use of positive end-expiratory pressure [12]. This procedure provides a direct view of the trachea, see Figure 2. Collapsibility can be visualized directly and the extent and severity can be estimated by the user [11]. Following the European Respiratory Society (ERS) guidelines, collapse less than 50% is accepted to be within normal limits, while 50-75% is classified as mild, 76-90% is considered moderate, and 91-100% as severe malacia [13]. However, several studies have shown that the classification is subjected to large inter-observer variabilities, since severity classification is dependent on the interpretation of the endoscopist [6, 14–20].

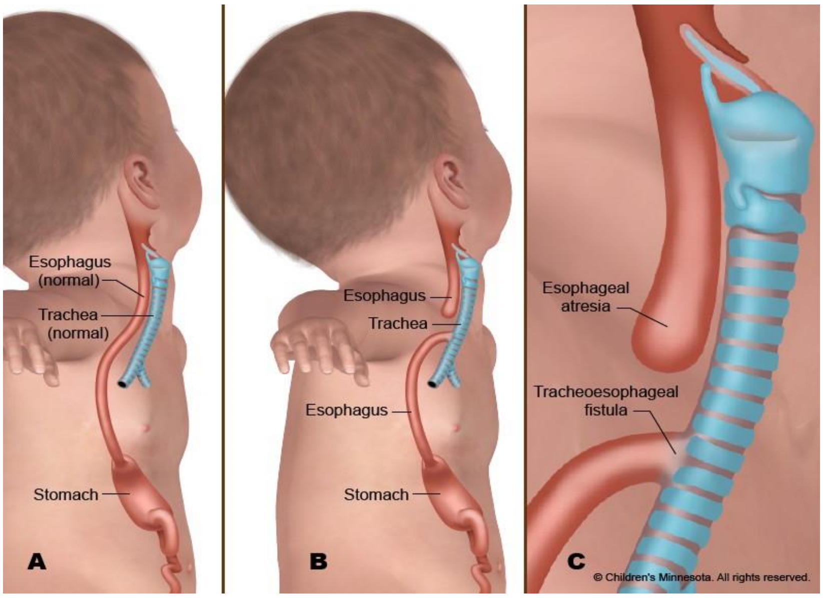


Figure 3. Esophageal atresia and esophageal fistula. A: Normal esophagus and trachea; B: Esophageal atresia in combination with a tracheo-esophageal fistula; C: Closer view of tracheo-esophageal fistula. Image reproduced from Children's Minnesota (n.d.) [21]

Around 16-33% of the neonates with EA and a TEF suffer from symptomatic TM [22]. The gold standard for treatment of TM is in most hospitals an aortopexy [4]. During this surgery, the aorta is sutured to the sternum, resulting in decompression on the anterior side of the trachea, see Figure 4. In the Wilhelmina Children Hospital (WKZ), a thoracoscopic posterior tracheopexy is the gold standard [8, 23]. During this surgery, the posterior wall of the trachea is sutured with non-absorbable sutures at one to three places to the spinal ligament, see Figure 5. By placing these sutures, the posterior wall is stabilized under traction, which prevents collapse of the tracheal lumen. In the WKZ, the classification of tracheomalacia severity in patients with EA is as follows: less than 33% collapse is graded as mild TM, between 33% and 66% collapse as moderate TM and more than 66% as severe TM. When graded as moderate or severe TM, posterior tracheopexy is performed during thoracoscopic esophageal atresia repair [24].

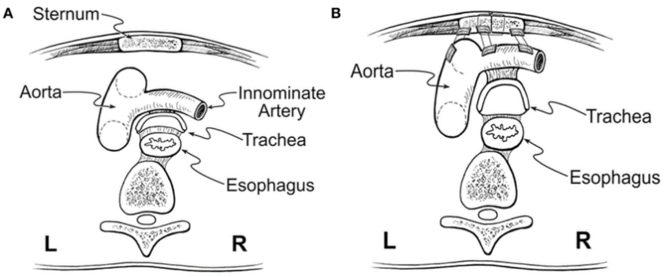


Figure 4. Aortopexy procedure. (A) Anatomical representation before anterior tracheopexy. (B) Anatomical representation after anterior tracheopexy. Both the ascending aorta and the innominate artery are sutured to the posterior surface of the sternum. Image reproduced from Kamran et al [4].

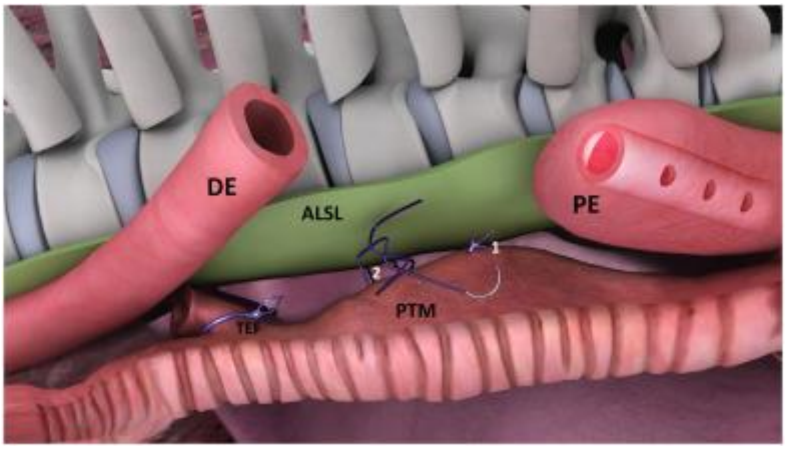


Figure 5. Illustration of the thoracoscopic posterior tracheopexy during primary esophageal atresia correction. After closure and transection of the tracheo-esophageal fistula (TEF), the proximal esophageal (PE) pouch is mobilized from the posterior tracheal membrane (PTM). A nasal tube lifts the PE away from its position between the trachea and the spinal column. The posterior tracheopexy is performed by placing non-absorbable sutures (1,2) that pull and fixate the PTM to the anterior longitudinal spinal ligament (ALSL). Then, the PE and distal esophagus (DE) are approximated and joined by an end-to-end anastomosis. Image reproduced from Tytgat et al (2018) [25].

1.2. Research question and goals

Assessing the percentage of tracheal lumen collapse depends on a subjective evaluation by the endoscopist and pediatric surgeon, leading to variability between different observers and even within the same observer [14, 26]. Pediatric patients with tracheomalacia would benefit from a precise and unbiased evaluation to help determine the need for a primary posterior tracheopexy during EA repair. This is especially crucial for newborns with EA, as integrating this assessment during initial surgery for EA repair can prevent a complex second operation with significant morbidity (due to post-surgical adhesions) [27]. Moreover, it may help prevent or minimize the more severe respiratory symptoms [27].

In this thesis, the main research question is: *'How can the percentage of tracheal lumen collapse in neonates born with esophageal atresia objectively be assessed, using bronchoscopy videos?'* The goal is to develop a fully automatic algorithm that has bronchoscopy frames (images of the video) as input and gives a percentage of remaining lumen as output. An important subgoal is to develop a model that is able to segment the trachea automatically from a single input image.

2. Technical introduction

2.1. Previous work

To identify what has yet been done in the field of automatic airway dimensions quantification based on bronchoscopy data, a systematic research was conducted in PubMed and arXiv. The systematic review focused on studies that utilized image analysis techniques. The following exclusion criteria were used: 1) The airway dimensions were not quantified by an image analysis technique, 2) The analysis was not based on bronchoscopy video's or frames, 3) The image analysis technique was not described and 4) The outcome parameters were not described. Articles were also excluded if they were not written in English

The search retrieved 932 studies, of which 20 remained after applying the exclusion criteria. These included 13 studies with human subjects, 6 with animals (one using deceased animals), and 1 using an airway training mannequin. ImageJ, or its predecessor, was the most commonly used software (employed in 10 studies) [28]. Besides ImageJ, other software used was SENSA, Bersoft Image Measurement, IPLEX, Carnoy, Global Lab Image and C++ openGL kit [29–33].

The primary outcome measures were airway diameter, cross-sectional area, cricoid to trachea ratio and degree of airway collapse. The need for standardized methods in image analysis to ensure consistency and reproducibility in quantifying airway dimensions during bronchoscopy was underscored. Articles that used a stereoscopic bronchoscopes were the only ones that were able to fully automatically quantify the airway dimensions. All other methods required some sort of manual input from the user.

While stereoscopic bronchoscopy demonstrates significant potential, its widespread adoption faces practical challenges. Replacing all currently used bronchoscopes with stereoscopic versions is impractical, and even more important, the existing stereoscopic bronchoscopes are too large for use in neonates.

Emerging technologies, such as monocular depth estimation algorithms, offer a promising alternative [34, 35]. Monocular depth estimation is especially important for autonomous systems, such as self-driving cars, drones or robots, but can also be valuable for augmented reality and virtual reality, image and video processing, and more [36].

Moreover, these monocular depth estimation algorithms have the potential to significantly improve diameter estimations without the need for extensive hardware modifications. Gil et al suggest three-dimensional reconstruction as potential solution to account for distance errors [37]. Understanding the varying depths of the tracheal rings in bronchoscopy images can help to segment them correctly, aiding in more precise diagnosis and treatment. A monocular depth estimation algorithm might therefore be of great value.

The potential of a new, recently developed monocular depth estimation algorithm called 'Depth-Anything', as well as a standard U-Net model, were explored as part of this thesis.

2.2. This thesis

This thesis describes four models, all using certain pre-processing steps, model characteristics and post-processing steps. The output of all models is a segmented image. Image segmentation is an important process in calculating the collapse percentage of airways where precise measurements are essential for accurate diagnosis and treatment planning. Segmentation enables the isolation of specific regions of interest (ROI) from the rest of the image, facilitating detailed analysis and quantification. This process is essential for determining the degree of airway collapse, as it allows for the clear delineation of airway boundaries and the assessment of changes in tracheal dimensions. The two main deep learning models that are used are the U-Net model and the Depth-Anything model [38, 39]. Where the U-Net model is able to directly produce a predicted segmentation of the trachea, the Depth-Anything model produces a depth estimation image. This image will thereafter need further processing steps to create a segmented image. The other 2 deep learning models consist of a U-Net model but (also) use Depth-Anything output images as input.

Prior to segmentation, image pre-processing is necessary to enhance the quality and consistency of the input images. The goal of pre-processing is to improve the clarity and uniformity of the images, making segmentation algorithms more effective and reliable.

In post-processing, the primary task is to convert the grayscale output of the model into a binary image to accurately define the segmented region. The grayscale image generated by the model indicates the probability of each pixel belonging to the segmentation output (mask), with brighter pixels representing higher probabilities. All post-processing steps ensure that the final binary mask accurately represents the segmented region, allowing for precise measurement and analysis of airway collapse. By refining the mask, post-processing enhances the clarity and reliability of the segmentation results.

The segmentation mask made by the deep learning model will be compared to the segmentation mask delineated manually by the researcher. The segmentation of this user will be used as gold standard (ground truth). When evaluating a binary segmentation model, several metrics can be used to assess its performance, each offering unique insights. In this thesis, the dice score will be used as primary metric to evaluate performance and identify the most suitable model for further analysis and development. The dice score measures the overlap between two sets, namely the predicted and ground truth segmentations. It ranges from 0 to 1, where a score of 1 indicates perfect overlap (i.e., the predicted and actual segmentations are identical), and a score of 0 signifies no overlap at all. The formula for calculating the dice score is as follows: 2 * (number of common pixels) / (number of pixels in set A + number of pixels in set B). See Figure 6 for an illustration of how the dice score is calculated. Which dice score is acceptable depends on the context in which it is used, but a dice score of 0.8 is generally seen as a high score and therefore as acceptable [40–42].

$$\text{Dice} = \frac{2\ X\ Area\ of\ overlap}{Total\ area} =$$

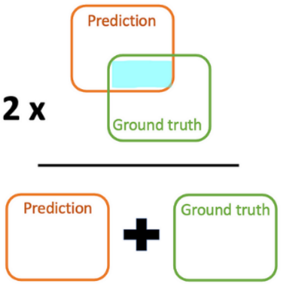


Figure 6. Calculation of the dice score. Image reproduced from Huynh (2013) [43].

In addition to evaluating the performance of different deep learning models based on standard metrics such as the dice score, this thesis included a substudy to assess the clinical acceptability of the segmentations. In this substudy, a physician reviewed both the ground truth segmentation and the predicted segmentation from the best-performing model to determine whether each segmentation was acceptable. This approach is necessary to validate the ground truth segmentations that were segmented by the (inexperienced) researcher. Moreover, it is possible that multiple correct segmentations exist for a given image and in such cases, a low dice score may not accurately reflect a poor segmentation, as the prediction could still be clinically acceptable. The other way around, a high dice score might indicate that the segmentation closely matches the ground truth, but it doesn't guarantee that the segmentation is clinically meaningful or useful. By incorporating expert evaluation, we aimed to account for this variability and provide a more comprehensive assessment of the model's performance in real-world applications.

Moreover, the segmentation data and corresponding metrics was transformed into a meaningful percentage of airway collapse, as this is the desired output for the physicians rather than the raw segmentation outputs themselves. The correlation between the available metrics and the remaining lumen percentages as predicted by a physician was evaluated to identify the most relevant metric and evaluate if a linear relationship existed. Next, a Principal Component Analysis (PCA) was conducted. PCA is a statistical technique that simplifies complex data by reducing its dimensionality, while retaining as much relevant information as possible. It does this by transforming the original metrics into new variables, known as principal components, which capture the largest sources of variation in the data. Each principal component is a linear combination of the original metrics and is designed to represent unique, independent patterns within the data. By selecting only the top components, PCA reduces complexity, allowing for easier analysis and modeling. The correlation between these principal components and the percentage of remaining lumen was investigated, followed by the development of a model to assess its accuracy in predicting remaining lumen percentages.

3. Dataset construction

3.1. Patient characteristics

In this study, 14 neonates diagnosed with EA and TM were included. All patients underwent at least one bronchoscopy at the WKZ between 2022 and 2024. These patients were evaluated to assess TM severity, and a total of 20 bronchoscopy procedures were conducted, from which a total of 127 frames (individual images) were extracted.

The cohort consisted of 9 male and 5 female neonates, with a gestational age at birth ranging from 29 weeks and 3 days to 41 weeks and 3 days. The patients' ages at the time of their first bronchoscopy varied significantly, from as early as 1 day to as late as 11 days postpartum. Correspondingly, their weights at the time of the procedure ranged from 1.2 kg to 4.1 kg, highlighting the diversity in the physical development of these neonates.

Some neonates required multiple bronchoscopic evaluations for reasons that were not assessed. For example, patient 6, born at 40 weeks gestation, underwent three bronchoscopy procedures at 5, 208, and 233 days old, with weights of 2.9 kg, 6.4 kg, and 6.6 kg, respectively. Similarly, patient 11 had three bronchoscopies, the last of which occurred at 254 days old, when the neonate weighed 7.5 kg. Patient characteristics can be found in Table 1.

Table 1. Patient characteristics. The age is the age in days at time of the bronchoscopy and the weight is the weight in kilograms at time of the bronchoscopy.

PATIENT	GENDER	GESTATION	BRONCHOSCOPY	AGE (DAYS)	WEIGHT (KG)
1	M	36+5	1	4	2.3
2	M	36+4	1	1	2.8
3	F	29+3	1	11	1.2
			2	63	2.9
4	M	34+2	1	11	2.9
5	F	40+3	1	3	3.0
6	F	40+0	1	5	2.9
			2	208	6.4
			3	233	6.6
7	M	33+0	1	6	2.0
8	F	39+6	1	4	3.3
9	M	41+3	1	4	3.3
10	M	39+2	1	4	3.5
11	M	37+2	1	4	2.0
			2	65	3.6
			3	254	7.5
12	M	41+2	1	3	4.1
			2	71	6.5
13	M	38+1	1	5	2.4
14	F	33+6	1	4	1.7

M = male, F = female

3.2. Frame selection

All 20 videos were individually reviewed in Clinical Assistant version 2019.2.1.7930, and specific frames were carefully extracted. These included one frame where the cricoid cartilage was clearly visible, at least one frame where tracheomalacia was evident (captured during both inspiration and expiration), and at least one frame showing a normal trachea (also captured during both inspiration and expiration). So for each video, at least 5 frames were captured. Additional criteria for the selected images included the visibility of at least one tracheal ring and no air/fluid bubbles within the ROI. When multiple frames were suitable, the one that appeared the sharpest was chosen.

3.3. Ground truth segmentation

Frames were loaded into the program ITK snap version 4.2.0. ITK-SNAP is a free, open-source, multi-platform software application used to segment structures in biomedical images [44]. In this thesis, the segmentation was drawn manually using the paintbrush tool. When possible, a trachea ring was followed and segmented. After segmentation, it was exported as mask in the nii.gz file. A NIfTI (.nii.gz) file is a widely used format for storing medical and scientific imaging data [45]. It is an acronym for Neuroimaging Informatics Technology Initiative and is often used to store 3D or 4D volumetric data. The NIfTI format provides an efficient way to store medical imaging data, including segmentation, and is important for extracting valuable information from these images, supporting both clinical and research applications. The process followed in ITK-SNAP is visualized in Appendix A.

3.4. Splitting data for model development and assessment

The decision was made to use the first 13 patients, with their corresponding 117 frames, to train the algorithms. The last patient (patient 14, with 10 frames) was reserved for testing the algorithms. By using only one patient for testing, the goal was to simulate real-world clinical scenarios where the model might encounter new patients with different characteristics. Although this approach may result in results that are not as generalizable as testing on two or more patients, we aimed to closely replicate the practical use of the model in clinical settings. Additionally, it was crucial to maximize the size of the training dataset to improve model performance, which is why this approach was preferred.

An alternative option was to randomly select frames from all available images for both training and testing. However, this approach was avoided because it would result in the train and test images being obtained under similar circumstances, potentially introducing bias. For example, lighting conditions may vary between different patients, but frames from a single patient typically share more consistent imaging conditions.

For the algorithms employing a U-Net model, cross-validation was conducted within the training dataset. Cross-validation is a technique used to assess the performance of machine learning models by training multiple models on different subsets of the input data and evaluating them on the remaining subsets (see Figure 7). This approach helps minimize the risk of overfitting. In this thesis, a 5-fold cross-validation was applied, where the training data was divided into five subsets (folds). In each fold, one subset was used as validation data, while the remaining subsets were used for training. When the cross-validation results were considered acceptable, one final model was made using all training data. The test data was kept separate and only used for final evaluation.

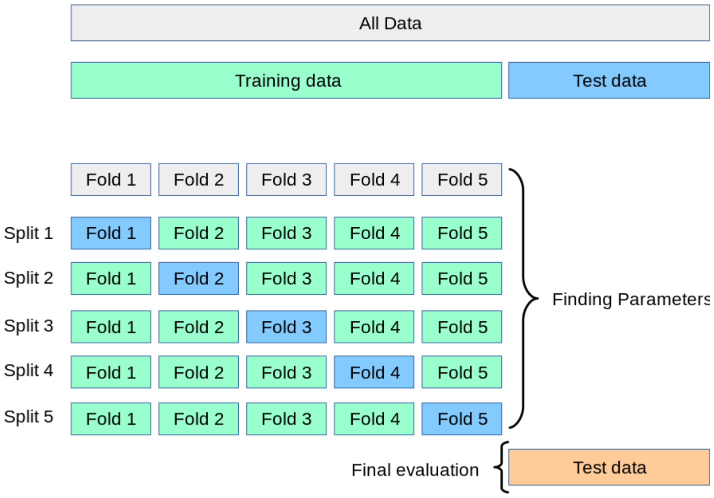


Figure 7. Diagram illustrating the process of 5-fold cross-validation, where the training data is divided into five subsets (folds). In each fold, one subset is used as validation data, while the remaining subsets are used for training. The test data is kept separate and only used for final evaluation. Image reproduced from scikit-learn.org [46]

For the Depth-Anything model, the pre-trained model is used. The training data was used for development of the processing steps following the Depth-Anything model while the test data was kept separate and only used for final evaluation. This approach was chosen to maintain consistency with the original Depth-Anything algorithm, avoiding modifications to the established methodology. All training images were processed through the model and following processing steps, and the mean dice score was calculated. In cases where no contour could be detected for an image, this was recorded and taken into account for the overall performance evaluation.

Prediction masks with dice scores close to zero (below 0.1) were excluded from the mean dice score calculation, as these low scores indicate minimal to no overlap between the predicted mask and the ground truth, marking the predictions as inaccurate. In clinical contexts, such outputs would offer no practical value for physicians. However, a dice scores between 0.1 and 0.6 may still provide value, especially if the model segments a different tracheal ring than the ground truth. Masks with scores between 0.1 and 0.6 were therefore not excluded in performance evaluation. Additionally, the number of prediction masks with near-zero dice scores was recorded for both training and test datasets, providing further insights into the model's performance and reliability.

4. Pre-processing

4.1. Introduction

Image segmentation is an important process in calculating the collapse percentage of airways where precise measurements are essential for accurate diagnosis and treatment planning. Segmentation enables the isolation of specific regions of interest, such as airway structures, from the rest of the image, facilitating detailed analysis and quantification. Prior to segmentation, image pre-processing is necessary to enhance the quality and consistency of the input images. Common pre-processing steps include noise reduction, contrast enhancement, and normalization of pixel intensity values, which improve the clarity and uniformity of the images, making segmentation algorithms more effective and reliable. These steps ensure that the segmentation process can accurately identify and quantify the relevant anatomical structures, leading to more precise and clinically useful measurements.

The training process began with three basic pre-processing steps: loading and resizing data, cropping images, and normalization. The model's performance was evaluated after applying these basic steps. Next, each of the four additional pre-processing steps—extrapolation, histogram equalization, anisotropic blurring, and data augmentation—was tested in sequence. The process followed these steps:

1. The dataset was trained using only the basic pre-processing steps.
1. The first additional pre-processing step (extrapolation) was added to the basic pipeline. If this addition improved the model's dice score, the step was preserved in the pipeline; if not, the pipeline reverted to the previous version.
2. The second additional pre-processing step (histogram equalization) was then tested. It was incorporated only if it further improved the dice score. Otherwise, the pipeline reverted to the version that excluded this step.

This iterative process continued for the remaining steps (anisotropic blurring and data augmentation), each being added to the pipeline only if it resulted in a performance increase.

By the end of this process, only those steps that demonstrably enhanced the model's performance were preserved in the final pre-processing pipeline. For a detailed overview of all pre-processing steps, see Figure 8, and for the results of the testing of each step, see Chapter 7.2.

For the Depth-Anything model, the decision to include a pre-processing step was not based solely on the dice score. In addition to the dice score, the number of times a mask could be successfully generated and the frequency with which the dice score exceeded 0.1 were also recorded and taken into account. Dice scores below 0.1 indicate that that the segmentation results were poor and unreliable, suggesting that the pre-processing step might not be effective. These factors were carefully considered before determining whether to incorporate a particular pre-processing step into the final pipeline.

The following subsections provide a brief discussion of each pre-processing step. After pre-processing the frames, they will be used to train the deep learning models, as described in Chapter 0. The output of the deep learning models will then undergo post-processing, as described in Chapter 6.

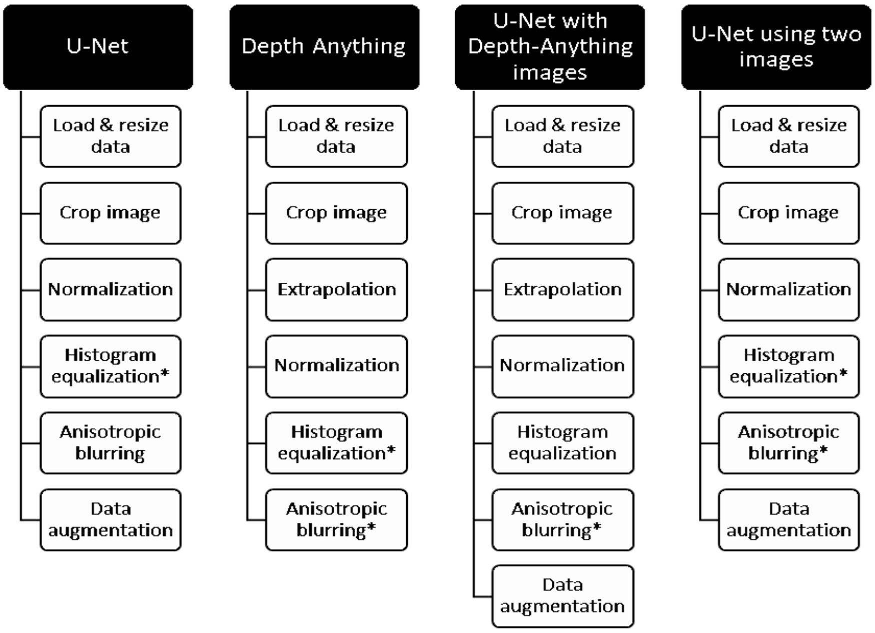


Figure 8. Preprocessing steps for the four models: U-Net, Depth Anything, U-Net with Depth-Anything images, and U-Net using two images. Steps marked with an asterisk were evaluated but not included in the final model due to poorer performance see the results section.

4.2. Data import and resize

Images and their corresponding ground truth masks were stored in separate folders, with each image and its matching mask sharing the same filename. This naming system allowed the model to correctly associate each image with its respective mask. To load the data in Python, all files in the storage folders were iterated over, and the images were stored in a list. Each image was then resized to a target size of 128x128 pixels. A similar process was applied to the masks: they were loaded into a list, resized to the same 128x128 pixel dimensions, and converted to binary format by thresholding the grayscale image at an intensity of 0.5. Resizing to 128x128 pixels allows for faster experimentation with different model architectures, hyperparameters, and training strategies and reduces the computational load on hardware (GPU/CPU). Deep learning models, especially those based on convolutional neural networks (CNNs), require significant processing power. By using smaller images, the memory usage is decreased and the time needed to train the model will be smaller. Larger images, like 512x512 pixels, would require more memory and longer training times, which might be unnecessary if the smaller images still capture the essential features needed for the task. A 128x128 image has 16,384 pixels (128 * 128), while a 512x512 image has 262,144 pixels (512 * 512). This is an increase by a factor of 16 in the number of pixels. A rough estimation suggests that the training time for the model would increase by a factor of 16 when switching from 128x128 to 512x512 resolution. Since training times exceeding 12 hours were not desired for this thesis due to time constraints, a higher resolution would likely exceed the preferred training duration. Therefore, an initial attempt was made using 128x128 images. When no acceptable model could be developed with 128x128 pixels, the model would then be retrained using 512x512 pixel images.

4.3. Image crop

To crop the image and retain only the true bronchoscopy image, removing the large black regions surrounding it, each image was converted to a binary format using thresholding. Hereafter, any holes within white structures were filled, and small unwanted structures were removed using erosion and dilation techniques. The minimum and maximum non-zero pixel values along the x and y axes were identified and used to crop the original images to the area containing the bronchoscopy view, which was defined as the ROI. The masks, initially having the same dimensions as the images, were also cropped using the same pixel boundaries, ensuring that each image and its corresponding mask reserved identical dimensions. See Figure 9 for an illustration of the image cropping process.

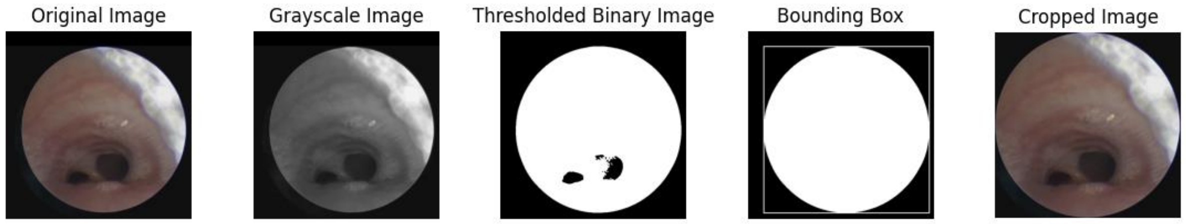


Figure 9. Illustration of the image cropping process. In this image, there are no visible unwanted structures (outside the ROI) so the effect of erosion and dilation is not visible.

4.4. Extrapolation

Image extrapolation was used to extend the field of view of an input image beyond its originally captured area. By using extrapolation, there is no need to crop the image to remove the black border. This way, the full image data will be reserved, preventing any loss of useful information. Moreover, the black border represents an artificial boundary that doesn't correspond to real-world objects, so it may confuse the Depth-Anything model. Extrapolation allows for extending the depth information into the black border region, ensuring that the depth model only focuses on the relevant parts of the image and avoids considering the black border as part of the scene.

Several extrapolation methods are available, including closest point, mirrored point, and wrapping. For this work, the closest point extrapolation method was selected, see Figure 10 for an example of the resulting image. This technique was applied only in the Depth-Anything model, as the other models could be trained to ignore boundary regions.

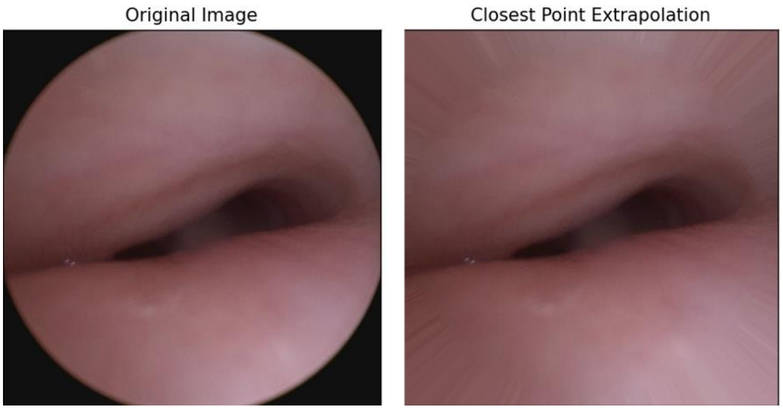


Figure 10. Example of an image that is extrapolated by using the closest point method

4.5. Normalization

Color images typically have intensity values ranging from 0 to 255 for each color channel. During normalization, these intensities are rescaled to a range of 0 to 1. Neural networks, particularly deep learning models, work better when input values are within a smaller, standardized range. Normalization helps the network converge faster during training. When the input values are in a consistent range (such as 0-1), it allows the optimization algorithm to find the optimal weights more efficiently. This leads to shorter training times and often better overall performance. Many activation functions commonly used in deep learning work best when the input values are small and centered around 0.

4.6. Histogram equalization

Histogram equalization is a technique which redistributes pixel intensities across the entire range to improve contrast. In this thesis, adaptive histogram equalization (AHE) was implemented. Unlike traditional histogram equalization, AHE calculates multiple histograms for different sections of the image and uses them to redistribute the image's lightness values. This approach enhances local contrast and sharpens edge definitions in each region of the image. Other histogram equalization techniques, like contrast stretching and the traditional histogram equalization, were also evaluated but resulted in a lower model performance.

4.7. Anisotropic blurring

Anisotropic gaussian blurring is a preprocessing technique that applies directional smoothing, preserving edges and significant structural details in an image while selectively blurring less important regions. Unlike isotropic blurring, which applies uniform smoothing across the entire image, anisotropic blurring uses the directionality of gradients or edges to adapt the amount of blur. By reducing noise without significantly distorting important features, anisotropic blurring can help improve the model's robustness and enhance feature extraction during training.

4.8. Data augmentation

To increase the size of the training dataset, data augmentation was performed. This involved applying various transformations to the images, including horizontal flipping, 20% rotation in either direction, and zooming by 5-20%. The augmentation factor, which determines how many times the dataset was artificially expanded, was set to 5. An augmentation factor of 5 resulted in a better performance than an augmentation factor of 2.

5. Deep learning model descriptions

5.1. Introduction

Within the last decade, many advances have been made within image segmentation. Initially, image segmentation techniques relied mainly on traditional machine learning algorithms, where hand-crafted features and heuristic rules were used to partition images into different regions. However, the emergence of deep learning, in particular convolutional neural networks, revolutionized image segmentation by enabling automatic feature extraction and learning of hierarchical representations.
Deep learning based image segmentation is becoming increasingly crucial in the medical field because of its ability to accurately and efficiently analyze medical images.

Semantic segmentation involves classifying and labeling individual pixels in an image based on their semantic meaning. The goal is to categorize each pixel in the image.

The most commonly known algorithms for image segmentation are the convolutional neural network (CNN) and the fully convolutional network (FCN). However, they have several limitations that newer algorithms don't have. An important limitation of CNNs is that they tend to overfit, particularly when there is limited data. This means the model becomes too focused on the training data and struggles to perform well on new, unseen data, which is undesirable when using the model in clinical settings. For FCN's, an important limitation is that they often require large amounts of memory (GPU/CPU).

In the next subsections, the 4 different models that are used for tracheal segmentation are explained. Figure 11 shows how U-Net and Depth-Anything are used in the 4 models.

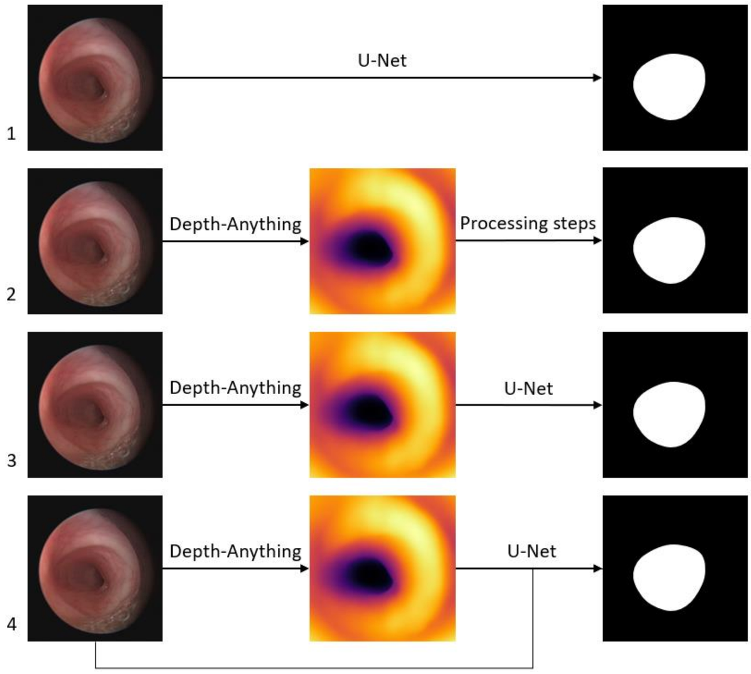


Figure 11. Diagram showing how Depth-Anything and U-Net are used in the 4 different deep learning models.

5.2. Standard U-Net model

Using FCN as a foundation, Ronneberger et al. created the U-Net network in 2015, which is a specialized tool that excels in handling images in the medical industry or any field where detailed image analysis is important. U-Net was developed to break down an image into smaller components, analyze those components, and reassemble the image. This is particularly helpful for finding and highlighting specific

objects in medical images. U-Net is known for its exceptional capacity to simultaneously see the big picture and concentrate on small details. A limitation is that the performance might be affected by its inability to reliably distinguish under-represented classes (categories or classes in a dataset that have fewer samples compared to others). In the case of severe tracheomalacia, which may occur less frequently in the dataset and/or in real-world scenarios than moderate or mild tracheomalacia, the model might struggle to correctly segment these images due to their lower representation.
Generally, successful training of deep-learning networks requires many thousand annotated training samples. However, Ronneberger et al (2015) have showed that with the use of data augmentation an end-to-end network can be trained from very few images, while still outperforming other methods like a basic CNN [47].

Hyperparameters are settings or configurations that are chosen before training a deep learning model. These parameters control how the model learns from the data and can significantly affect the model's performance. Unlike model parameters, hyperparameters are set manually and do not change during the learning process. Examples of hyperparameters that will appear in the next sections are: model-specific hyperparameters (number of layers, number of neurons per layer, kernel size), training-related hyperparameters (learning rate, batch size, number of epochs), regularization-related hyperparameters (dropout rate, L1 or L2 regularization strength) and optimization hyperparameters (optimizer choice, momentum).

The U-Net model is known for its "U" shaped architecture, which consists of a contracting path (encoder), a bottleneck and an expansive path (decoder). The U-Net structure allows for precise localization and segmentation, combining high-resolution features from the encoder with upsampled outputs from the decoder. The structure of the U-Net model is illustrated in Figure 12 and will be discussed in the following section.

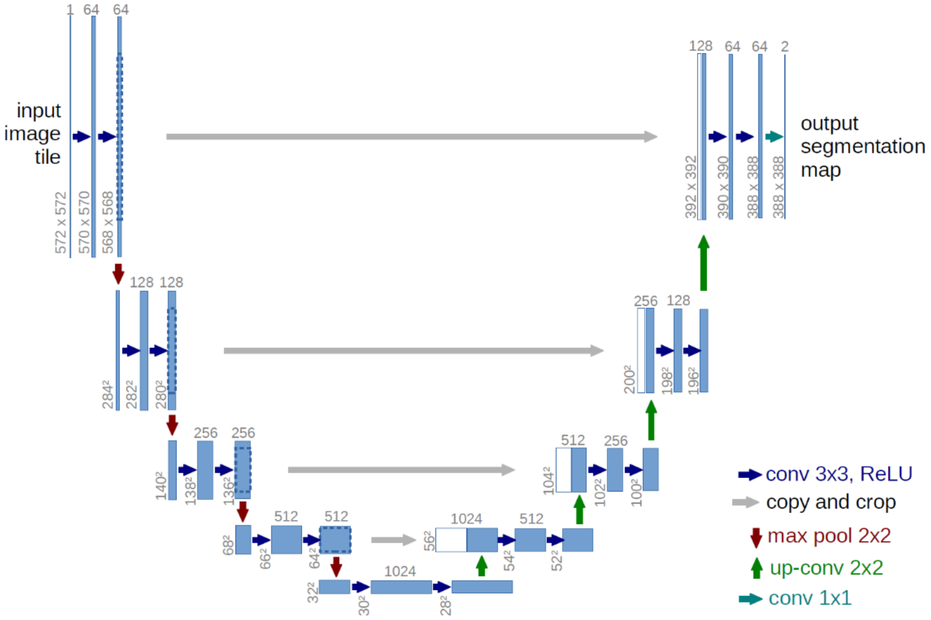


Figure 12. U-net architecture. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided as the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. Image reproduced from Ronneberger et al. (2015) [47]

Contracting path

The encoder (contracting path) captures the context of the input image by downsampling it.
Each block in the encoder consists of two 3x3 convolutional layers, each followed by a rectified linear unit (ReLU) activation. A convolutional layer works by sliding a small window (3x3 pixels), called a filter,

over the image and performing a mathematical operation called convolution. This helps identify features like edges, textures, and shapes. ReLU activation helps the network learn complex patterns by making sure that any negative values are set to zero while keeping positive values the same. This helps the network handle non-linearity and learn better.

After each convolutional block, a 2x2 max pooling layer is applied with a stride of 2, reducing the size of the image and making the network faster and more efficient. It works by taking a small window and sliding it over the image, but instead of doing a convolution, it picks the maximum value from that window. This helps to keep the most important information while discarding less important details.

Bottleneck

The bottleneck is the deepest part of the network, where the image is represented in the smallest spatial dimensions but with the highest number of feature maps. It consists of two 3x3 convolutional layers followed by ReLu activations, similar to the other convolutional blocks.

Expansive path

The decoder (expansive path) upsamples the feature maps back to the original input size, allowing for precise localization. Each block in de decoder starts with a 2x2 transposed convolution (upconvolution) that doubles the spatial dimensions. The upsampled feature map is than concatenated with the corresponding feature map form the contracting path. These so called ‘skip connections’ help in recovering spatial information lost during downsampling. Following concatenation, two 3x3 convolutional layers with ReLU activations are applied.

Final layer

The final layer is a 1x1 convolution that reduces the number of feature maps to the number of classes to be segmented. Since within this algorithm we want a binary output, a sigmoid activation is applied.

5.3. Depth-Anything model

Depth-Anything is a deep learning model for monocular depth estimation, developed by Yang et al (2024) [34]. Stereo vision requires two images taken from slightly different viewpoints, similar to human binocular vision. By comparing the displacement (disparity) of objects between these two images, it calculates the distance to various points in the scene. It generally provides a high accuracy as the disparity between images offers a direct quantitative basis for calculating depth.

Monocular depth estimation, on the other hand, is a computer vision task that involves predicting the depth value (distance relative to the camera) of each pixel given a single (monocular) red-green-blue (RGB) image. Unlike stereoscopic techniques, monocular depth perception algorithms must extract depth cues from various image features such as texture gradients, object sizes, shading, and perspective. The challenge lies in translating these inherently ambiguous cues into accurate depth maps, which has seen significant advancements with the advent of deep learning.

The Depth-Anything model was developed with the aim to build a simple yet powerful monocular depth estimation model dealing with images under any circumstances.

The model takes a regular 2D image as input. It uses a deep neural network to analyze the image, looking for visual cues that humans use to perceive depth, such as: object size (smaller objects usually appear farther away), occlusion (objects in front blocking parts of objects behind), texture gradients (textures appear finer at a distance), lighting and shadows. The model also looks at the image at different scales, from fine details to the overall scene structure.

The model has been trained on a vast dataset of images paired with their corresponding depth information (1.5 million labeled images) and more than 62 million unlabeled images [34]. Through this training, it learns to associate visual patterns with depth values. For each pixel in the input image, the model estimates its depth, creating a depth map where closer objects are represented as brighter and farther objects as darker.

Recent experiments have shown that the Depth-Anything algorithm performs better than the previous best monocular depth estimation model named the mixed data sampling (MiDaS) model [34, 35]. In Figure 13, a visual example is provided of both the Depth-Anything model and the MiDaS model.

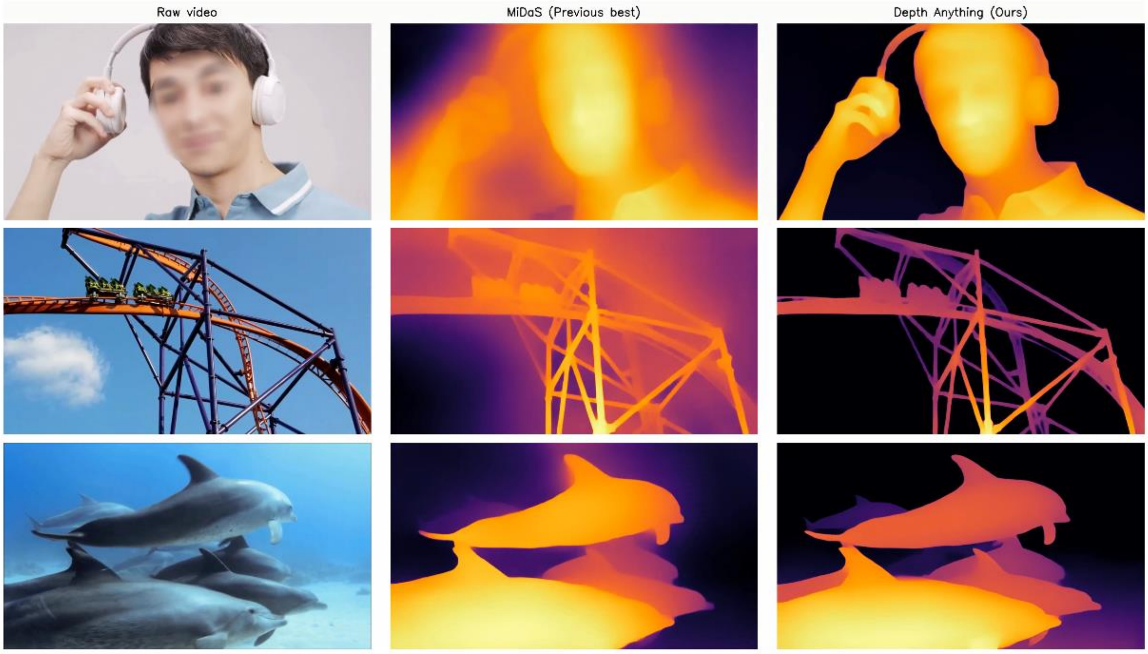


Figure 13. Visual example of both the Mixed Data Sampling model and the Depth-Anything model [38].

For the Depth-Anything model, some processing had to be done to go from depth map to segmentation, see Figure 14. After the output of the Depth-Anything model was generated, there was not yet a segmentation. In order to do this, the optimal amount of clusters needed to be determined. Two different methods for determining the optimal amount of clusters were evaluated, namely the elbow method (inertia method) and the co-occurrence method.

The inertia method measures the sum of squared distances between each point in a cluster and the centroid of that cluster, which quantifies how tightly the points are packed together. This process involves running a clustering algorithm (such as K-Means used in this thesis) for a range of values for k and calculating the inertia for each k. In contrast, the co-occurrence method analyzes the frequency of pairs of items or features appearing together in a dataset. The procedure begins by creating a co-occurrence matrix which can then be analyzed using clustering techniques, like K-means, to group similar items based on their co-occurrence patterns. Since the co-occurrence method yielded a higher dice score, it was selected as the preferred method for determining the optimal number of clusters.

Once the optimal number of clusters was established, the clusters were identified, and a Canny edge mask was applied to detect the edges in the output. Each edge was stored as a different contour, and all contours were sorted from largest to smallest based on pixel count. A function was implemented to determine which contours touched the border of the image and which did not. The largest contour that did not touch the border was filled and used as a mask. Finally, the outline of this segmentation was drawn onto the original image to visualize the result.

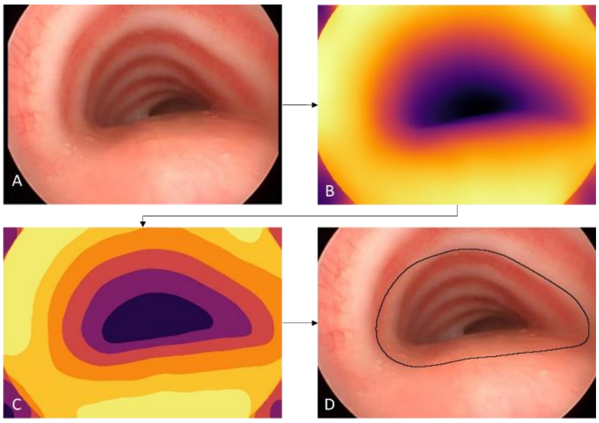


Figure 14. (A) Original bronchoscopy image showing the airway. (B) Depth image produced by the Depth-Anything algorithm (without the use of extrapolation). (C) Clustered depth image. (D) Outline of the biggest segment of which the outline is completely in view.

5.4. U-Net model using Depth-Anything images

In this model, all pre-processed images were first processed through the depth estimation model, and the resulting depth maps were saved in a list. These depth maps then served as input images for the U-Net model. This U-Net model used the same architecture-specific hyperparameters as the previous U-Net model but differed in its training, regularization, and optimization hyperparameters.

5.5. U-Net model using two images

For this model, all pre-processed images were first processed through the depth estimation model, and the resulting depth maps were saved in a list. Next, these depth maps and their associated pre-processed images were used as input pairs for the U-Net model. This U-Net model, therefore, took two input channels, while still producing a single prediction mask as output. The architecture of this U-Net model is shown in Figure 15.

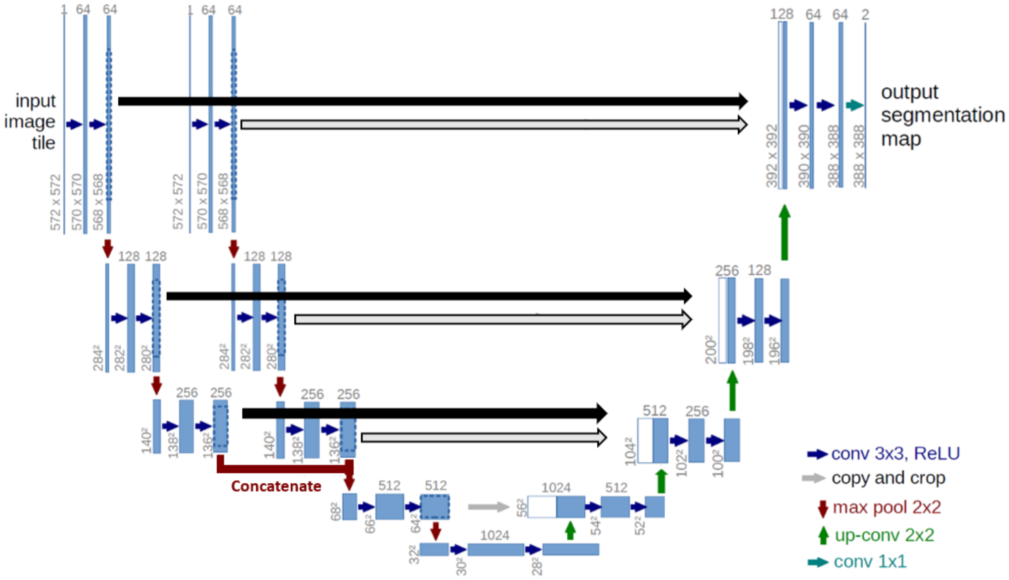


Figure 15. U-Net architecture with two input image tiles, concatenated at the deepest layer. Each blue box corresponds to a multi-channel feature map, with the number of channels indicated on top. The x-y size is shown at the lower left of each box. White boxes represent copied feature maps, and arrows denote different operations: convolution, max-pooling, and up-convolution. Image adapted from Ronneberger et al. (2015). [47]

6. Post-Processing

6.1. Introduction

In post-processing, the primary task is to convert the grayscale output of the model into a binary image to accurately define the segmented region. The grayscale image generated by the model indicates the probability of each pixel belonging to the mask, with brighter pixels representing higher probabilities. The post-processing steps performed will be discussed in the following sections.

6.2. Thresholding

The first step in post-processing is thresholding, where a specific threshold value is applied to the grayscale image to create a binary mask. Pixels with values above the threshold are classified as belonging to the mask and to white, while those below are set to the background (black). This step is essential for enabling a direct comparison with the ground truth mask.

6.3. Geometric parameters

Multiple geometric outcome parameters are extracted from both the ground truth and prediction masks, including dimensions, area, perimeter, and shape characteristics. The process begins by identifying the largest contour within each segmentation mask. The dimensions of the smallest bounding box around this contour are then calculated, specifically the distances between midpoints of opposite sides, representing the diameters (dA and dB), see Figure 16. Additionally, the contour's area and perimeter are determined. These measurements are used to compute further shape metrics, such as the aspect ratio and roundness. Roundness is defined by the following formula:

$$Roundness = \frac{4\,\pi * area}{perimeter^2}$$

This formula is based on the idea that a circle is the most efficient shape in terms of the relationship between perimeter and area—it maximizes the area for a given perimeter.

To understand this formula:

1. For a given perimeter, the radius of an equivalent circle can be estimated by using the formula $P = 2\pi R$, where P is the perimeter and R is the radius of the circle.
2. Once the radius is known, the area of the equivalent circle can be calculated using the formula $A_{circle} = \pi R^2$.
3. Finally, the roundness is determined by comparing the actual area of the shape to the area of this equivalent circle.

The results are organized into two dataframes: one for predicted segmentation masks and another for ground truth masks. The computed metrics are saved as Excel files. This allows for a comprehensive comparison between the ground truth and prediction mask outcomes, providing alternative performance evaluation methods besides the dice score.

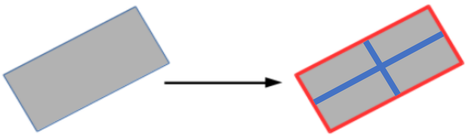


Figure 16. Bounding box principle. The smallest diameter is dA and the largest dB

7. Deep learning segmentation results

7.1. Introduction

When evaluating a binary segmentation model, several metrics can be used to assess its performance, each offering unique insights. The dice score is particularly valuable as it emphasizes the overlap between the predicted and actual segments, making it ideal for situations where accurate overlap measurement is critical, such as in medical imaging.

Additionally, IoU (Intersection over Union) is a standard metric that measures the ratio of overlap between predicted and ground truth regions to their combined area. IoU is widely used and provides a straightforward interpretation of model accuracy.

Moreover, precision measures the accuracy of positive predictions, which is important if there is a need to minimize false positives, while recall focuses on ensuring that all true positives are identified, essential in scenarios where missing a positive case is costly. The F1 Score combines precision and recall, offering a balanced metric that captures the trade-off between the two. However, these three metrics can be less informative on their own in segmentation tasks compared to dice and IoU.

Because of the need for accurate overlap between the predicted mask and ground truth mask, the dice score is utilized as the primary metric to assess performance. During training, the model employs dice loss, defined as 1 minus the dice coefficient, as its loss function. A loss function measures the difference between a model's predictions and the actual values, providing a numeric error score. Since the objective of training is to minimize this loss, the model's parameters are adjusted to maximize the dice coefficient.

As described in Section 3.2, multiple frames were captured at the same location, with one taken during inspiration and another during expiration. The primary objective was to assess the degree of tracheal collapse. To achieve this, the outcome of the algorithms was compared to percentages as determined by the physicians. The correlation between various outcome parameters and the percentage of remaining lumen, was then analyzed to understand how well these parameters align with the clinical evaluations.

7.2. Pre-processing

See Figure 8 for an overview of the different models and the corresponding pre-processing steps and Table 2 for the outcome scores based on which the choices are made whether to include a pre-processing step or not. Initially, only the basic pre-processing steps were applied, and the model's performance was noted. Then, the first pre-processing step was tested. If this resulted in an improved score, the step was incorporated into the entire pre-processing pipeline; otherwise, the previous pipeline was maintained. This process was repeated for each new pre-processing step, with the step being retained only if it enhanced the model's performance, otherwise, it was not included in the pipeline. In Table 2 the outcome scores based on which the choices are made whether to include a pre-processing step or not.

Table 2. Preprocessing steps and corresponding dice score results for the four models: U-Net, Depth Anything, U-Net with Depth-Anything images, and U-Net using two images. Highest scores are underscored.

	U-Net	**Depth-Anything**	**U-Net with Depth-Anything images**	**U-Net using two images**
Basic	0.79	0.57	0.73	0.79
Extrapolation	-	0.66	0.73	-
Histogram equalization	0.78	0.66	0.73	0.79
Anisotropic blurring	0.79	0.67	0.71	0.78

7.3. Standard U-Net model

Training results

The Hyperband algorithm was employed for hyperparameter optimization. Hyperband operates by running multiple rounds, or brackets, each starting with a large number of hyperparameter configurations (trials). In the initial round, each configuration is trained for a limited number of epochs. After evaluating performance, only the best-performing configurations are selected to proceed to the next round, where they are trained for additional epochs. This process continues until only a few configurations remain, which are then trained for the maximum number of epochs specified. The optimal hyperparameters identified through this process can be found in Table 3.

Table 3. Table showing the range of hyperparameter options explored during optimization and the corresponding optimal values identified by the Hyperband algorithm.

Hyperparameter	**Options**	**Optimal value**
Kernel size	1, 3, 5, 7	7
Convolution 1 filters	16, 32, 48, 64	16
Convolution 2 filters	32, 64, 96, 128	32
Convolution 3 filters	64, 128, 192, 256	256
Bottleneck filters	128, 256, 384, 512	384
Learning rate	0.01, 0.001, 0.0001, 0.00001	0.01
Optimizer	Adam, Rmsprop, Sgd	Rmsprop
Batch size	8, 12, 16, 20, 24, 28, 32	32
Epochs	10, 15, 20, 25, 30, 35, 40, 45, 50	35

Figure 17A shows the dice score for both training and validation data over 35 epochs. The training dice score increases rapidly in the initial epochs and stabilizes around a value of 0.82, indicating that the model is effectively learning to segment the images. The validation dice score also shows a steady increase, though it lags slightly behind the training score, stabilizing around 0.78. The shaded regions represent the variability across folds, with a noticeable variance in the early epochs of validation, likely due to differences in the folds or the model's initial learning phase. Overall, the high mean dice score suggests that the model achieves a strong overlap between the predicted segmentations and the ground truth.

Figure 17B illustrates the loss values for both training and validation across the same epochs. The training loss decreases sharply in the first few epochs and then gradually approaches a stable point around 0.18, reflecting effective model convergence. The validation loss follows a similar trend but remains slightly higher than the training loss, leveling off near 0.22. The shaded areas indicate variability across folds, with the validation loss showing more fluctuation early on, which decreases as the model stabilizes. The close proximity of the training and validation loss curves indicates good generalization, with the model performing consistently across different data subsets.

In summary, the U-Net model demonstrates strong performance in segmenting images, with a mean dice score of 0.78 and stable loss values, suggesting it is well-suited to the task. The consistency between training and validation metrics also indicates minimal overfitting and reliable generalization.

After training the final model, the mean dice score across all training images was 0.79.

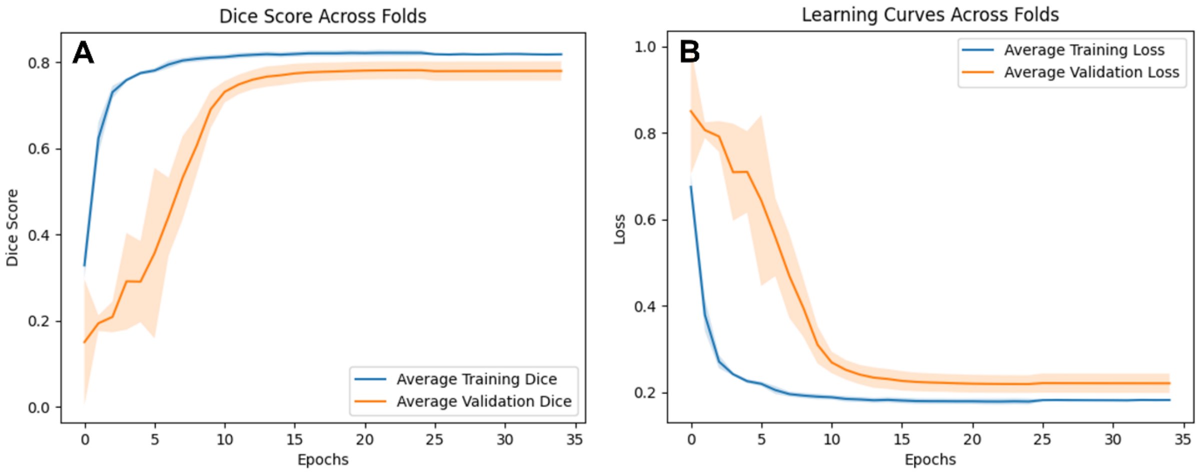


Figure 17. Performance of the model using the original images as input. (A) Dice score as a function of training epochs, averaged across k-folds. The blue line represents the training dice score, while the orange line represents the validation dice score. Shaded regions represent the standard deviation across folds. (B) Corresponding loss curves across epochs for training (blue) and validation (orange).

The agreement between the ground truth and prediction masks on these values were evaluated by calculating Pearson correlation coefficient (r).

Based on the training results, the parameter roundness (r = 0.878) showed the strongest correlation between the ground truth and prediction masks, see Figure 18.

<u>Test results</u>
After obtaining acceptable validation results, the model was tested on the test data. The dice score for the test data was 0.75.

The parameter ratio exhibited the strongest correlation between the ground truth and prediction masks (r = 0.93), see Figure 18.

Scatter plots, Bland-Altman plots and distribution plots for the parameter outcomes of the U-Net model can be found in Appendix B. Analysis was only repeated for the other deep learning models if these models outperformed the U-Net model. It can be seen that the U-Net generally has a comparable distribution to the ground truth. However, some differences can be seen. For example, the U-Net model generally predicts the segmented area to have a way higher roundness score. Roundness scores below 0.6 almost never occur, while the distribution for the ground truth data is more distributed. In the scatter plot it can be seen that the U-Net model especially deviates from the ground truth data in cases where the ground truth roundness is low. Moreover, the parameter area shows a similar trend, whereby the algorithm deviates most from the ground truth during ground truth areas that are small. Lastly, the distribution of diameter A (dA) shows that the prediction data often over-estimates the dA and that small dA's are almost never predicted correctly.

Train and test results for the correlation with the metrics are listed in Appendix C for all models.

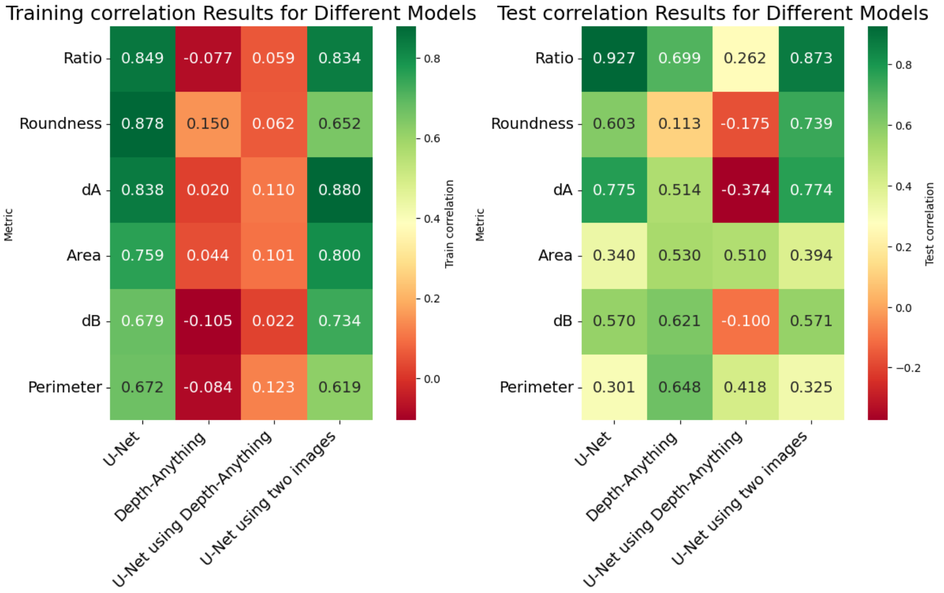


Figure 18. The heatmap displays the correlation coefficients between the predictions of different models (U-Net, Depth-Anything, U-Net using Depth-Anything, and U-Net using two images) and ground truth measurements across six metrics: Ratio, Roundness, Area, dA, dB, and Perimeter. Higher correlations (in green) indicate better agreement with ground truth, while lower or negative correlations (in red) reflect poorer performance.

7.4. Depth-Anything model

Training results

For the Depth-Anything model, no training was performed on the model itself. However, processing steps to convert the depth maps to segmented images were developed using the training data. All training images were processed through the model and following processing steps, and a mean dice score was calculated. In cases where no contour could be detected for an image, this was recorded and factored into the overall performance evaluation. The decision to include a pre-processing step was not based solely on the dice score. In addition to the dice score, the number of times a mask could be successfully generated and the frequency with which the dice score exceeded 0.1 were also recorded and taken into account, see Table 4 for these results. These factors were carefully considered before determining whether to incorporate a particular pre-processing step into the final pipeline. For this model, a dice score of 0.67 was reached using the basic preprocessing steps and extrapolation. The distribution of dice scores can be found in Figure 19.

Table 4. Evaluation of different preprocessing steps for the Depth-Anything model. The evaluation includes three key metrics: the mean dice score, the number of successfully detected masks, and the number of prediction masks with dice scores greater than 0.1. For each pre-processing step, its impact on these metrics was recorded. The combination of metric outcomes considered best is underscored.

	Dice score	**Number of successfully detected masks (%)**	**Number dice scores >0.1 (%)**
Basic	0.57	97 (82.9)	89 (91.8)
Extrapolation	0.66	95 (81.2)	88 (92.6)
Histogram equalization	0.66	80 (68.4)	74 (92.5)
Anisotropic blurring	0.67	89 (76.1)	83 (93.3)

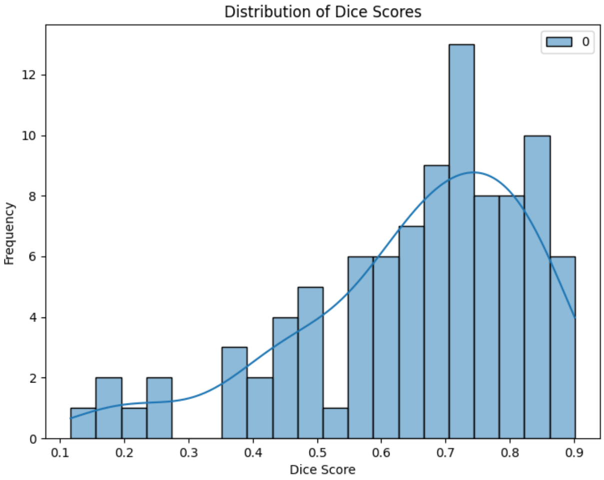


Figure 19. Histogram representing the distribution and frequency of dice scores obtained during training.

Besides the dice score, the parameters mentioned in the previous model were also calculated. Here, the parameter roundness (r = 0.150) exhibited the strongest correlation between the ground truth and prediction masks, see Figure 18.

Test results

After obtaining acceptable validation results, the model was tested on the test data. This resulted in a mean dice score of 0.60.

Out of the 10 test images, 8 (80.0%) masks were successfully detected. For these masks, 7 (77.8%) dice scores were >0.1. The distribution of the dice scores can be seen in Figure 20.

For the other parameters, the parameter ratio exhibited the strongest correlation with a correlation of 0.70, see Figure 18.

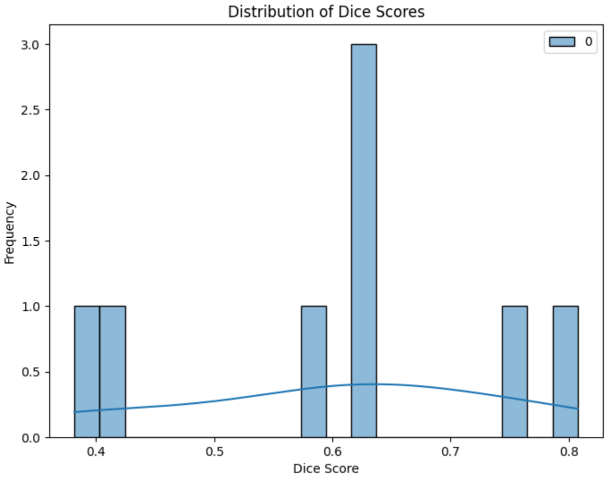


Figure 20. Histogram representing the distribution and frequency of dice scores obtained during testing.

7.5. U-Net model using Depth-Anything images

Training results

The Hyperband algorithm was again employed for hyperparameter optimization. The optimal hyperparameters identified through this process can be found in Table 5.

Table 5. Table showing the range of hyperparameter options explored during optimization and the corresponding optimal values identified by the Hyperband algorithm.

Hyperparameter	**Options**	**Optimal value**
Kernel size	1, 3, 5, 7	7
Convolution 1 filters	16, 32, 48, 64	48
Convolution 2 filters	32, 64, 96, 128	128
Convolution 3 filters	64, 128, 192, 256	192
Bottleneck filters	128, 256, 384, 512	512
Learning rate	0.01, 0.001, 0.0001, 0.00001	0.00001
Optimizer	Adam, Rmsprop, Sgd	Rmsprop
Batch size	8, 12, 16, 20, 24, 28, 32	24
Epochs	10, 15, 20, 25, 30, 35, 40, 45, 50	45

In Figure 21A, the performance of the U-Net model with Depth-Anything input images across different epochs during a 5-fold cross-validation is illustrated. Plot A shows the dice score for both training and validation data. As training progresses, both the average training and validation dice scores increase, indicating that the model is improving its ability to overlap predicted segmentations with the ground truth. The learning curve begins to plateau around 20 epochs, suggesting that the model reaches its peak performance early, with a mean dice score stabilizing round 0.55.

Figure 21B presents the learning curves in terms of loss. Both the training and validation losses decrease rapidly during the first 10 epochs, indicating that the model is learning efficiently. The losses continue to decrease at a slower rate, stabilizing after around 20 epochs. The close alignment between the training and validation curves suggests good generalization, although the higher validation loss towards the end of training could indicate slight overfitting. The shaded regions around the curves in both plots represent variability across the folds, with a larger variability in both the training and validation data than in model 1. This variability could be attributed to the small dataset size or the inherent variability in the folds used for cross-validation. Moreover, a slight drop in validation dice scores is observed after approximately 35 epochs, potentially indicating overfitting or a need for early stopping.

After post-processing, the mean training dice during cross-validation was 0.80, while after training the final model, the mean dice score across all training images was 0.70. It is somewhat uncommon for the cross-validation score to be higher than the training score, but it can occur depending on the data, model, and training conditions. In this case, the difference could be attributed to the variability in the quality of the images used in the different folds of the cross-validation. Since the input for the U-Net model comes from the output of the Depth-Anything model, the quality of these depth maps may vary across different images. It's possible that the images in the test set within each fold are relatively of higher quality or provide more accurate depth information, making them easier for the model to segment, thus leading to better performance on those particular images. On the other hand, when the model is trained on the entire dataset, it has to learn from a broader range of images, some of which might have less accurate depth information or other challenges, leading to a lower mean dice score.

The values of all other parameters were compared between the ground truth and prediction masks to evaluate their agreement. The parameter perimeter (r = 0.12) exhibited the strongest correlation between the ground truth and prediction masks, see Figure 18.

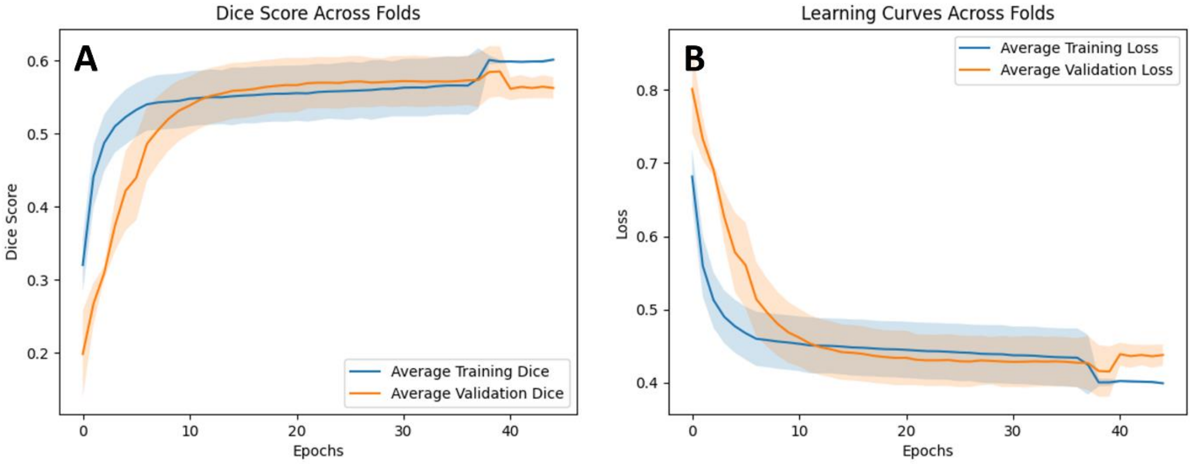


Figure 21. Performance of the model using the depth images as input. (A) Dice score as a function of training epochs, averaged across k-folds. The blue line represents the training dice score, while the orange line represents the validation dice score. Shaded regions represent the standard deviation across folds. (B) Corresponding loss curves across epochs for training (blue) and validation (orange).

<u>Test results</u>

After obtaining acceptable validation results, the model was tested on the test data. The dice score for the test data was 0.56.

The parameter area (r = 0.51) exhibited the strongest agreement between the ground truth and prediction masks, see Figure 18.

7.6. U-Net model using two images

<u>Training results</u>

The Hyperband algorithm was again employed for hyperparameter optimization. The optimal hyperparameters identified through this process can be found in Table 6.

Table 6. Table showing the range of hyperparameter options explored during optimization and the corresponding optimal values identified by the Hyperband algorithm.

Hyperparameter	**Options**	**Optimal value**
Kernel size	1, 3, 5, 7	5
Convolution 1 filters	16, 32, 48, 64	48
Convolution 2 filters	32, 64, 96, 128	64
Convolution 3 filters	64, 128, 192, 256	256
Bottleneck filters	128, 256, 384, 512	384
Learning rate	0.01, 0.001, 0.0001, 0.00001	0.01
Optimizer	Adam, Rmsprop, Sgd	Sgd
Batch size	8, 12, 16, 20, 24, 28, 32	8
Epochs	10, 15, 20, 25, 30, 35, 40, 45, 50	25

Figure 22A displays the dice score progression for both training and validation data over 25 epochs. The training dice score exhibits a rapid increase in the initial epochs and stabilizes around 0.8, demonstrating the model's effective learning in segmenting the images. The validation dice score shows a similar upward trend, though it plateaus slightly lower at approximately 0.75. The shaded regions represent variability across folds, with notable variance in the early epochs, particularly for validation, likely due to differences in the folds or the model's initial learning phase. The consistent high mean dice scores for both training and validation suggest strong overlap between predicted segmentations and ground truth.

Figure 22B illustrates the loss curves for training and validation across the same 25 epochs. The training loss decreases sharply in the first few epochs and then gradually stabilizes around 0.2, indicating effective model convergence. The validation loss follows a similar trend but remains slightly higher than the training loss, leveling off near 0.25. The shaded areas show variability across folds, with the validation loss exhibiting more fluctuation early on, which diminishes as the model stabilizes. The proximity of the training and validation loss curves suggests good generalization, with the model performing consistently across different data subsets. The overall trend of decreasing loss values correlates with the improving dice scores seen in Figure 22A.

In the last few epochs of Figure 22A, there is a slight upward trend in the training dice score, while the validation dice score shows a minor downward trend. Similarly, in Figure 22B, we can see a small increase in the gap between training and validation loss towards the end.

This divergence between training and validation performance is often an indicator of potential overfitting. Overfitting occurs when a model learns to perform very well on the training data but fails to generalize as effectively to new, unseen data (represented by the validation set).

The trend that can be seen during the last few epochs suggest that the model might be starting to overfit to the training data in the later epochs. It's learning patterns specific to the training set that don't generalize well to the validation set. Early stopping or regularization are methods that might can address this potential overfitting.

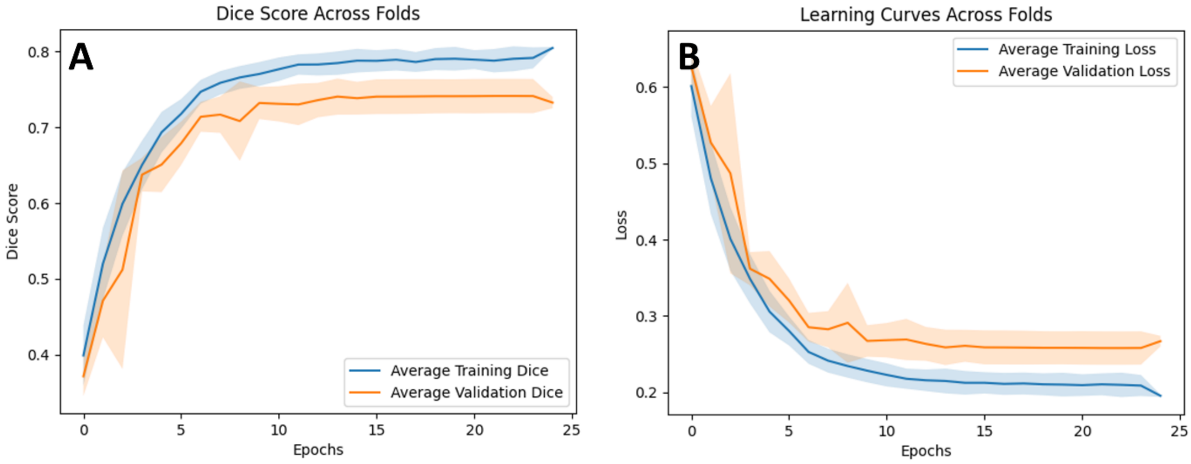


Figure 22. Performance of the model using both the depth and original images as input. (A) Dice score as a function of training epochs, averaged across k-folds. The blue line represents the training dice score, while the orange line represents the validation dice score. Shaded regions represent the standard deviation across folds. (B) Corresponding loss curves across epochs for training (blue) and validation (orange).

After training the final model, the mean dice score across all training images was 0.85.

The values of all other parameters were compared between the ground truth and prediction masks to evaluate their agreement. The parameter dA (r = 0.88) exhibited the strongest correlation between the ground truth and prediction masks, see Figure 18.

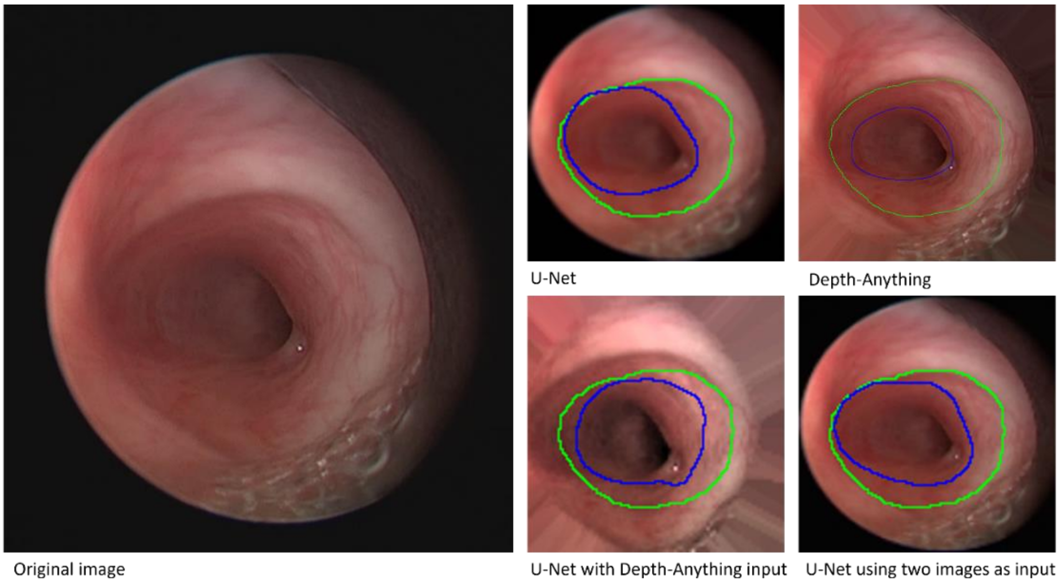
Test results

After obtaining acceptable validation results, the model was tested on the test data. The dice score for the test data was 0.73.

The parameter ratio exhibited the strongest agreement between the ground truth and prediction masks, with a correlation of 0.87, see Figure 18.

7.7. Examples

In Figure 23, an example is visualized where there was a significant difference between the predicted segmentation and the ground truth segmentation. However, the prediction in this case might not be wrong for every model, since different correct segmentations may be drawn on this image.



Original image

U-Net

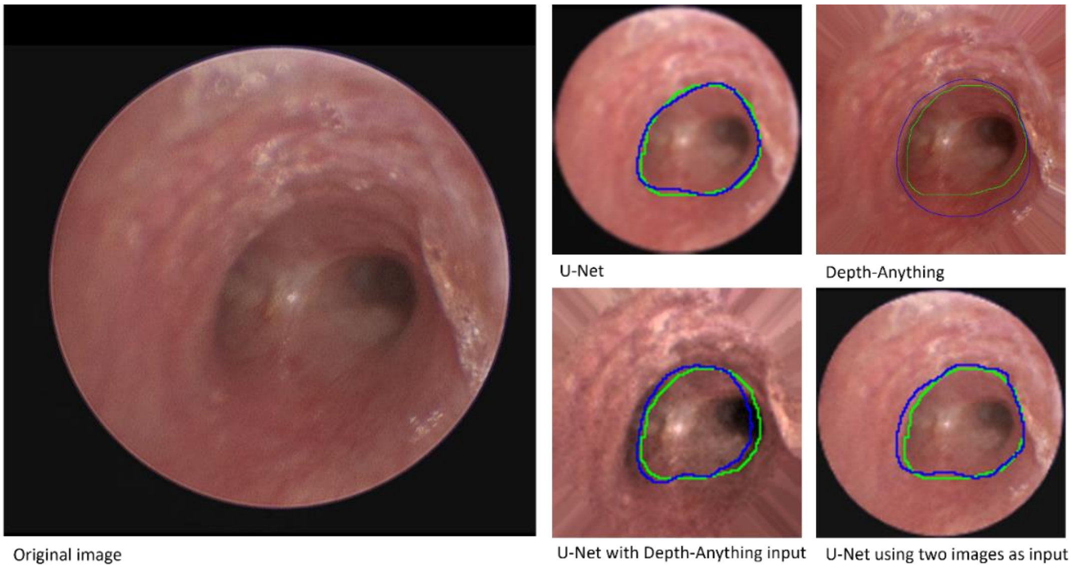
Depth-Anything

U-Net with Depth-Anything input

U-Net using two images as input

Figure 23. Example image where the outline of the predicted segmentation is displayed in blue and the outline of the ground truth segmentation is displayed in green.

In Figure 24, an example is visualized where the dice score was quite high for every model. Still, the segmentation results are different for each model.



Original image

U-Net

Depth-Anything

U-Net with Depth-Anything input

U-Net using two images as input

Figure 24. Example image where the outline of the predicted segmentation is displayed in blue and the outline of the ground truth segmentation is displayed in green.

In Figure 25, an example is visualized where the dice score was quite high for the U-Net model and the U-Net model using two images, but low for the other two models.

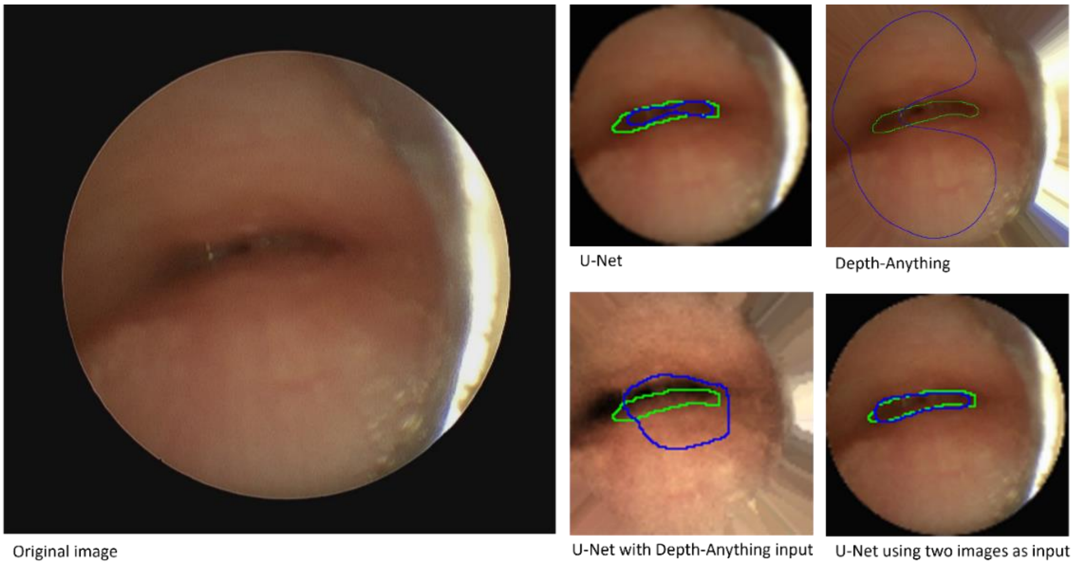

U-Net
Depth-Anything
Original image
U-Net with Depth-Anything input
U-Net using two images as input

Figure 25. Example image where the outline of the predicted segmentation is displayed in blue and the outline of the ground truth segmentation is displayed in green.

8. Clinical results

8.1. Introduction

In this sub-study, the aim is to better understand the extent to which the dice score is a reliable metric in this study. Although the dice score is commonly used to quantify the quality of a segmentation, it does not always provide a complete picture of accuracy. This is because there can be multiple acceptable segmentations, such as at different depth levels of the tracheal rings. In other words, there may be several "correct" answers, which complicates the interpretation of the dice score.

8.2. Methods

In this experiment, 27 image pairs were randomly selected, and clinicians were asked to estimate the percentage of remaining lumen of the airway based on paired inspiration and expiration images. These estimations are used in Chapter 9.

For the inspiration images of the 27 pairs, the clinicians were tasked with indicating whether two visible segmentations were to be considered correct or incorrect. One segmentation represented the ground truth, while the other was the model's prediction, but the clinicians were not informed which was which. To reduce the workload, only the segmentations from the highest-performing model, the U-Net model, (based on the best test dice score) were presented and evaluated for these 27 images. The goal of this evaluation was to compare how often the ground truth segmentations were judged as correct versus the model's predictions. We aimed to determine whether these results were comparable or if there was a statistically significant difference between them. To assess this, a McNemar statistical test, designed for paired categorical data, was applied to determine if the observed difference was meaningful.

8.3. Results

In total, 17 out of 27 (63%) images were considered correct for the manual segmentation (the ground truth segmentation) and 15 out of 27 (56%) images were considered correct for the model's prediction.

The results indicated that although the ground truth segmentations were judged as correct more often than the model's predictions, the difference was not statistically significant (p = 0.616), suggesting that the model's performance was comparable to the ground truth in this particular evaluation. Results are listed in Table 7. While the model's predictions align closely with the ground truth, it's also important to consider potential limitations in the ground truth itself. In 10 out of 27 images (37%), the physician considered the ground truth segmentation as incorrect. This indicates that the training dataset's ground truth was not perfect, and therefore, this imperfection could have contributed to a lower "correct" score for the predicted segmentations. The model's performance might also be affected by these incorrect ground truth segmentations. The discrepancies between the predicted segmentations and the ground truth could stem not only from the model's limitations but also from potential issues in the ground truth labeling process, which in turn could lead to a less accurate evaluation of the model's performance.

Table 7. Contingency table displaying the amount of times the ground truth and prediction segmentations were accepted or not accepted.

		Ground truth segmentation		
		Correct	Not correct	*Total*
Prediction segmentation	Correct	8	7	15
	Not Correct	9	3	12
	Total	17	10	27

9. Collapse percentage calculations results

9.1. Introduction

In this substudy, the goal was to transform segmentation data and corresponding metrics into a meaningful percentage of airway collapse (or percentage of remaining lumen), as this is the desired output for the physicians rather than the raw segmentation outputs themselves. The focus was on creating models that could predict the percentage of collapse based on these metrics.

9.2. Correlation metrics and percentage remaining lumen

A Linear Regression model was fitted to the data to evaluate the correlation, and the mean absolute error (MAE) was calculated to quantify the accuracy of the predictions, measuring the deviation between predicted percentages and actual values. The analysis revealed that the two metrics with the highest correlation were roundness ($r = 0.84$, $R^2 = 0.70$, MAE = 13.31) and ratio of largest to smallest diameter ($r = 0.82$, $R^2 = 0.68$, MAE = 13.79), see Figure 26. The red line in Figure 26 represents the regression line, which is a line that minimizes the sum of squared differences between the actual data points and the predicted values on the line.

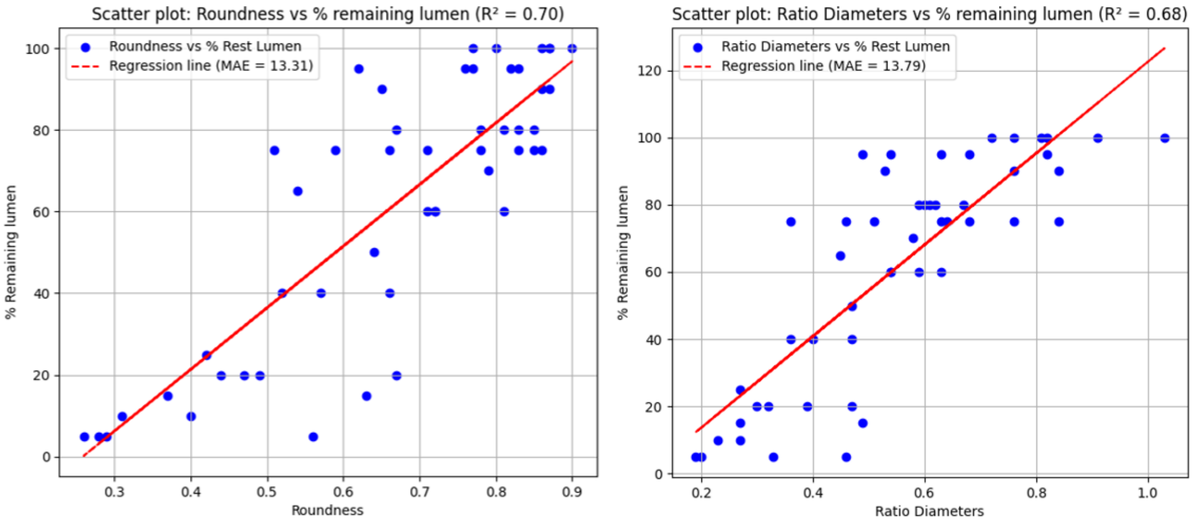


Figure 26. Scatter plot with regression line of the two metrics with the highest correlation.

9.3. Correlation PCA and percentage remaining lumen

In this case, the PCA combined the metrics (roundness, diameter ratio, diameters, area, and perimeter) into new variables that represent the most important underlying structures. These components help simplify the data while retaining essential information and can be used for more accurate prediction.

A Linear Regression model was fitted to the data to evaluate the correlation, and the mean absolute error (MAE) was calculated to quantify the accuracy of the predictions, measuring the deviation between predicted percentages and actual values. The analysis revealed that principal component 1 explained a variance of 0.69 ($r = 0.75$, $R^2 = 0.57$, MAE = 16.56) and principal component 2 explained a variance of 0.27 ($r = 0.4$, $R^2 = 0.16$, MAE = 25.37), see Figure 27. The red line in Figure 27 represents the regression line, which is a line that minimizes the sum of squared differences between the actual data points and the predicted values on the line.

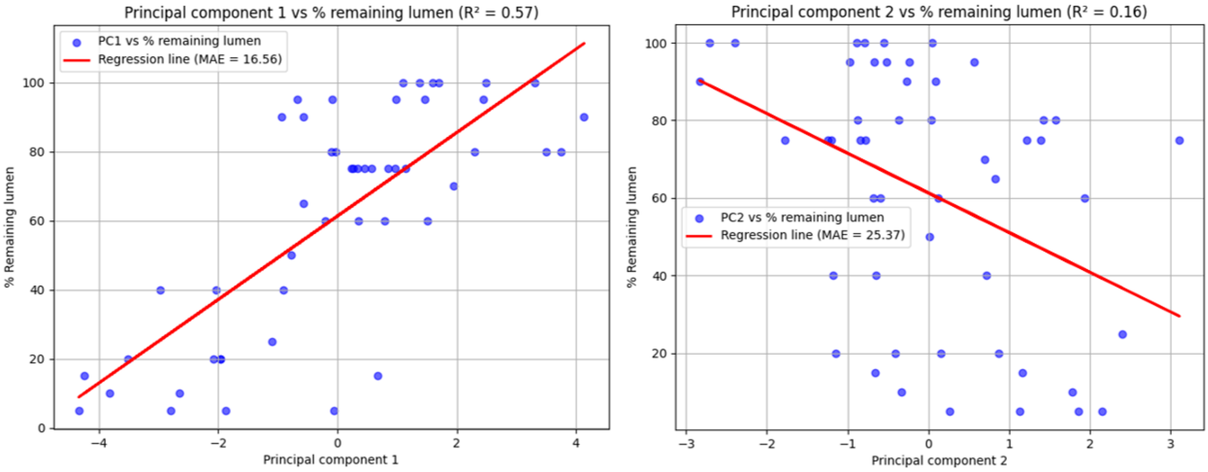


Figure 27. Scatter plot with regression line of the two principal components.

9.4. Cross validation model for estimating percentage of remaining lumen

After investigating the correlation between the roundness, ratio, principal components (PCA) and the percentage of remaining lumen, two models were trained to predict the collapse percentages. The dataset was divided into a training group and a test group, ensuring a more reliable evaluation of the model's performance. Cross-validation (k = 5) was employed within the training group to assess the model's robustness and generalizability.

The PCA-based model showed slightly better performance, achieving a correlation of 0.84 ($R^2 = 0.70$) compared to the roundness model's correlation of 0.81 ($R^2 = 0.66$). While both models exhibited Mean Absolute Errors above 10% (PCA model: 12.73%; roundness model: 14.40%), their mean differences were remarkably close to zero (PCA model: 0.09%; roundness model: -0.15%). These near-zero mean differences indicate that neither model shows systematic bias in its predictions, even though individual predictions may deviate considerably from actual values. Figure 28 and Figure 29 provide visual representations of each model's performance through scatter plots and Bland-Altman analyses.

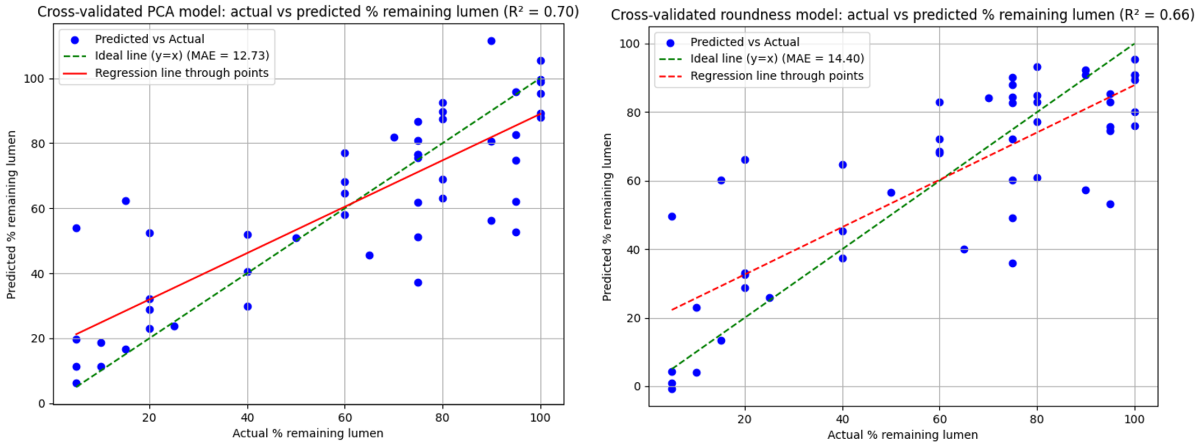


Figure 28. Scatter plot of the cross-validated models for predicting the percentage of remaining lumen. Left: model using the principal components to predict the percentages. Right: model using the parameter 'roundness' to predict the percentages

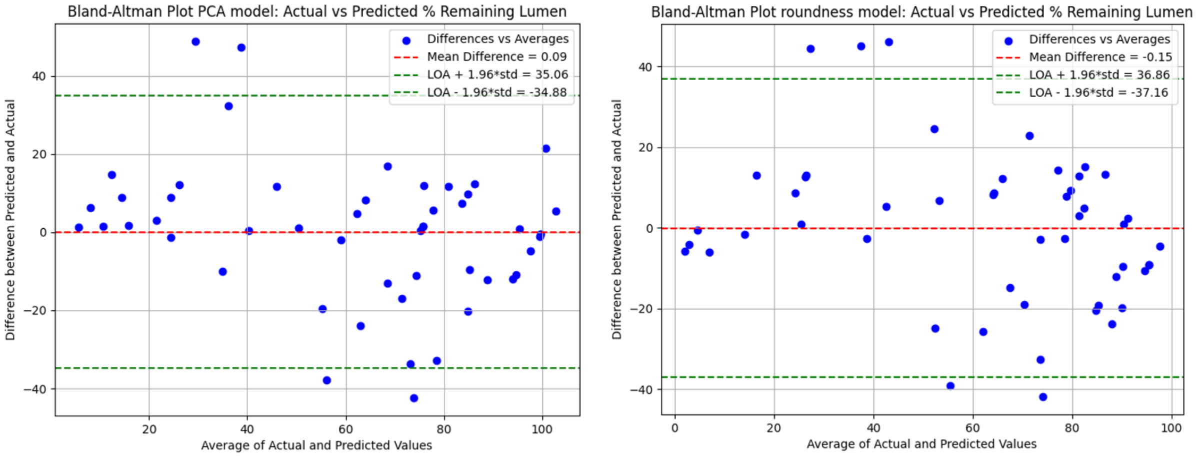


Figure 29. Bland-Altman plot of the cross-validated models for predicting the percentage of remaining lumen. Left: model using the principal components to predict the percentages. Right: model using the parameter 'roundness' to predict the percentages

10. Discussion

This study has explored the application of deep learning models, particularly U-Net and Depth-Anything, for the segmentation of airways, with the goal of assessing airway collapse in tracheomalacia patients. The findings provide valuable insights into both the potential and the challenges of automating airway collapse segmentation. In this section, we discuss the key results of the study, highlight challenges encountered during the study, and propose directions for future work to enhance the clinical applicability and robustness of the segmentation models.

Model performance

The primary goal of this study was to evaluate the performance of different deep learning models in segmenting airways, particularly in the context of airway collapse. The U-Net model was the best performer, achieving a dice score of 0.75 on test images and demonstrating a strong correlation (r = 0.93) between predicted outputs and the ground truth for the "ratio" parameter. The model's robustness is evident in its minimal overfitting, as the dice scores for training and testing were very close (0.79 vs. 0.75), which suggests that the model can generalize well to unseen data. This makes U-Net a promising candidate for clinical applications where rapid and accurate airway analysis is crucial.

Further analysis of the U-Net model's performance revealed some important trends. In particular, while the model performed well overall, there were notable deviations in specific parameter outcomes. As illustrated in the scatter plots, Bland-Altman plots, and distribution plots in Appendix B, the U-Net model generally predicts the segmented area to have a higher roundness score compared to the ground truth. Roundness scores below 0.6 were rarely predicted, whereas the ground truth distribution showed a broader spread. The scatter plots showed that the U-Net deviated most significantly from the ground truth when the ground truth roundness values were low.

Similarly, the parameter "area" displayed a trend where the algorithm's predictions deviated more for smaller ground truth areas. The distribution of the area predictions was skewed toward higher values, particularly for cases where the true area was small. Lastly, the predictions for the parameter dA consistently overestimated its value, with smaller dA values being almost never accurately predicted. These discrepancies suggest that while the U-Net model performs well in general, it struggles with certain outlier cases, particularly when the ground truth exhibits smaller or more irregular values.

In comparison, the Depth-Anything model performed less well, with a lower dice score of 0.60. Furthermore, it could not generate a mask for every image, rendering it unsuitable for this specific task. The lower performance can likely be attributed to the challenges inherent in relying solely on depth information, which may not capture the fine details required for accurate segmentation of airway structures. Interestingly, combining depth data with standard images (e.g., U-Net with Depth-Anything images or using two images) provided slight improvements, but these models still did not outperform the standard U-Net model. This indicates that adding complexity with additional data modalities does not always guarantee better results, and may even hinder model generalization.

Pre-processing and post-processing

The pre-processing pipeline used in this study was crucial in preparing high-quality input data. Key steps, such as noise reduction, contrast enhancement, and image normalization, were essential in improving the clarity and consistency of airway structures, which directly impacted segmentation accuracy. Data augmentation techniques were beneficial in expanding the dataset and reducing the risk of overfitting. The post-processing steps, which involved thresholding and geometric analysis, helped refine the segmentation results. The geometric analysis of the segmented area also provided for more ways to compare the results with the ground truth than the dice score alone.

Correlation with collapse percentage and clinical relevance

A central objective of this study was to correlate segmentation results with clinical indicators of airway collapse, expressed as the percentage of remaining lumen. The analysis revealed a strong correlation between geometric metrics (e.g., roundness and the ratio of the largest to smallest diameter) and the degree of collapse. Notably, principal component analysis (PCA) was effective in reducing data dimensionality, revealing that combinations of certain metrics provided more accurate predictions of collapse percentage. This highlights the importance of these metrics in assessing the degree of collapse and suggests that such approaches could aid clinical decision-making.

Although the mean absolute error (MAE) was still relatively high—12.73% for the principal component and 14.40% for the roundness metric—the MAE between different bronchoscopy users is unknown. Previous research has highlighted significant inter-observer variability, but these studies typically assessed collapse using subjective grading systems rather than precise percentage measurements. Therefore, it is unclear whether the mean absolute errors of 12.73% and 14.40% fall within or outside the range of typical inter-observer variability, as this range has not been precisely defined. Additionally, Murgu et al. (2012) demonstrated that physicians tend to underclassify tracheomalacia severity 47% of the time [26]. Knowing the mean absolute error of these physicians would be valuable for comparing their performance with that of the algorithm.

Reliability of the gold standard and statistical testing

A notable limitation in this study was the reliability of the ground truth data. The ground truth segmentations were created by a researcher with limited experience in bronchoscopy image segmentation, which could introduce potential errors. For instance, when assessing a random subset of the data of 27 images, a physician found that only 63% of the ground truth segmentations were acceptable. It would have been beneficial if all ground truth segmentations had been reviewed and graded before training the models, with adjustments made where necessary.

Additionally, the collapse percentage was estimated by a single physician, despite the known inter-observer variability in such measurements. Addressing these limitations in future research could involve using more experienced annotators and applying statistical methods to assess the reliability of the gold standard. Techniques like inter-rater reliability testing or correlation analysis could be employed to quantify the agreement between different observers and assess segmentation consistency.

Moreover, statistical tests such as logistic regression could be useful for identifying the airway collapse parameters or segmentation features that most strongly predict the need for tracheopexy. This would enable the development of predictive models that assist clinicians in identifying patients likely to require surgical intervention, ultimately improving clinical decision-making.

Multiple segmentation options

In medical image segmentation, it is important to recognize that multiple valid segmentations may exist for the same image, especially when delineating complex anatomical structures. Future research could develop models that account for multiple correct segmentations by incorporating annotations from multiple experts into the training process. This would allow the model to capture variability and generate more robust segmentation results. Additionally, incorporating uncertainty estimation into the model could enable it to quantify the confidence in the segmentation, allowing clinicians to assess the reliability of the results, particularly in complex cases where multiple plausible segmentations exist.

Future directions

While this study has demonstrated the potential of deep learning models for airway collapse segmentation, several areas for future research and improvement remain.

One significant challenge encountered was the manual selection of frames, a time-consuming and error-prone process that highlights the need for automation in clinical workflows. Automating frame selection, such as choosing frames with the greatest or smallest cross-sectional areas, could enhance efficiency and reduce human intervention, particularly in dynamic imaging scenarios like video sequences, where the algorithm could automatically select frames corresponding to critical phases of airway collapse.

Another crucial aspect is to have the segmentations performed by a physician, or at the very least, evaluated and modified by them when necessary. It is preferable to involve multiple physicians in this process to minimize inter-observer variability. When a single physician evaluates all ground truth segmentations, there is a risk that another physician may not agree with which segmentations are considered correct. Involving multiple physicians helps ensure that the model is trained on more accurate and consistent data, ultimately improving the model's performance and the reliability of its output.

Additionally, the variability in image orientation and positioning presented another challenge, underscoring the importance of standardizing imaging protocols to ensure consistent alignment. This would reduce variability and improve segmentation accuracy, enabling the model to perform more reliably across different patients and clinical settings.

Future research could also focus on refining pre-processing methods, such as experimenting with various noise reduction techniques, sharpening filters, or advanced methods like anisotropic Gaussian blurring to optimize image clarity. Post-processing improvements, such as morphological operations (dilation or erosion) to smooth mask boundaries or remove noise, and adaptive thresholding for diverse anatomical variations, would further enhance segmentation quality.

To improve model performance, expanding training datasets to include a wider range of patient anatomies and imaging conditions, along with hyperparameter optimization and the incorporation of advanced techniques like multi-modal data fusion or attention mechanisms, could improve the model’s generalization.

Strengthening the connection between segmentation results and clinical outcomes, particularly the degree of airway collapse and the need for surgical intervention, is another important direction. A retrospective analysis comparing segmentation results with clinical interventions could identify the most predictive metrics for surgical decisions.

Incorporating statistical methods, such as logistic regression, could further enhance the model. In clinical settings, logistic regression can analyze various predictors—such as patient demographics, imaging findings, severity markers, and other clinical features—to estimate the probability of a patient requiring surgical intervention.

By assigning weights to each predictor, logistic regression generates a risk score that helps clinicians gauge the likelihood of needing surgery based on the patient's specific characteristics. This can guide more personalized treatment plans, potentially identifying patients who may benefit from early intervention or, conversely, those who could avoid surgery.

Finally, developing models that account for multiple valid segmentations, along with uncertainty estimation, would improve the robustness and clinical reliability of the system, enabling clinicians to assess the confidence in segmentation results, especially in complex cases where multiple segmentations may be acceptable.

11. Conclusion

This research presents a deep learning-based model for the segmentation of airway collapse in bronchoscopy images, with a focus on improving accuracy and clinical relevance. Through the application of U-Net and Depth-Anything models, significant advancements were made in automatic segmentation, providing a useful tool for clinicians in evaluating the extent of airway collapse. The standard U-Net model showed the highest performance with a dice score of 0.75 for the test images and a correlation of 0.93 (parameter: ratio) between the ground truth and model output. The preprocessing pipeline, although contributing positively to model performance, revealed that the improvements were incremental, suggesting that further refinement and exploration of additional techniques may be necessary for more substantial gains. Post-processing, including thresholding and geometric analysis, played a crucial role in refining the model's output and providing valuable parameters for clinical interpretation.

However, several limitations, including the reliance on a single physician's annotations for the ground truth and the inherent subjectivity in segmenting medical images, must be considered. Future work should address these challenges by incorporating multiple expert annotations and exploring methods that handle the inherent uncertainty in segmentations. Moreover, the correlation between segmentation results and clinical outcomes, such as the need for tracheopexy, needs further exploration through statistical analyses to identify reliable predictive markers.

Future perspectives highlight great potential for integrating this model into clinical settings to assist in early diagnosis of severity of tracheomalacia and personalized treatment planning. Expanding the dataset, improving the model's generalizability, and exploring advanced post-processing techniques could enhance its clinical applicability. Ultimately, the goal is to create a model that not only segments the airway collapse accurately but also serves as a reliable tool for predicting clinical outcomes and guiding decision-making.

References

1. Boogaard R, Huijsmans SH, Pijnenburg MWH, et al (2005) Tracheomalacia and bronchomalacia in children: incidence and patient characteristics. Chest 128:3391–3397. https://doi.org/10.1378/CHEST.128.5.3391
2. Fischer AJ, Singh SB, Adam RJ, et al (2014) Tracheomalacia is associated with lower FEV1 and Pseudomonas acquisition in children with CF. Pediatr Pulmonol 49:960. https://doi.org/10.1002/PPUL.22922
3. Hysinger EB (2018) Laryngomalacia, Tracheomalacia and Bronchomalacia. Curr Probl Pediatr Adolesc Health Care 48:113–118. https://doi.org/10.1016/J.CPPEDS.2018.03.002
4. Kamran A, Jennings RW (2019) Tracheomalacia and Tracheobronchomalacia in Pediatrics: An Overview of Evaluation, Medical Management, and Surgical Treatment. Front Pediatr 7:. https://doi.org/10.3389/FPED.2019.00512
5. Deacon JWF, Widger J, Soma MA (2017) Paediatric tracheomalacia - A review of clinical features and comparison of diagnostic imaging techniques. Int J Pediatr Otorhinolaryngol 98:75–81. https://doi.org/10.1016/J.IJPORL.2017.04.027
6. Fraga JC, Jennings RW, Kim PCW (2016) Pediatric tracheomalacia. Semin Pediatr Surg 25:156–164. https://doi.org/10.1053/J.SEMPEDSURG.2016.02.008
7. Carden KA, Boiselle PM, Waltz DA, Ernst A (2005) Tracheomalacia and Tracheobronchomalacia in Children and Adults: An In-depth Review. Chest 127:984–1005. https://doi.org/https://doi.org/10.1378/chest.127.3.984
8. van Tuyll van Serooskerken ES, Tytgat SHAJ, Verweij JW, et al (2021) Primary Posterior Tracheopexy in Esophageal Atresia Decreases Respiratory Tract Infections. Front Pediatr 9:. https://doi.org/10.3389/FPED.2021.720618
9. Durkin N, De Coppi P (2022) Management of neonates with oesophageal atresia and tracheoesophageal fistula. Early Hum Dev 174:105681. https://doi.org/https://doi.org/10.1016/j.earlhumdev.2022.105681
10. Dodge-Khatami A, Deanovic D, Sacher P, et al (2006) Clinically relevant tracheomalacia after repair of esophageal atresia: the role of minimal intra-operative dissection and timing for aortopexy. Thorac Cardiovasc Surg 54:178–181. https://doi.org/10.1055/S-2005-872954
11. Kugler C, Stanzel F (2014) Tracheomalacia. Thorac Surg Clin 24:51–58. https://doi.org/10.1016/j.thorsurg.2013.09.003
12. Pelaia C, Bruni A, Garofalo E, et al (2021) Oxygenation strategies during flexible bronchoscopy: a review of the literature. Respir Res 22:253. https://doi.org/10.1186/s12931-021-01846-1
13. Wallis C, Alexopoulou E, Antón-Pacheco JL, et al ERS statement on tracheomalacia and bronchomalacia in children. https://doi.org/10.1183/13993003.00382
14. Majid A, Gaurav K, Sanchez JM, et al (2014) Evaluation of tracheobronchomalacia by dynamic flexible bronchoscopy: A pilot study. Ann Am Thorac Soc 11:925–932. https://doi.org/10.1513/ANNALSATS.201312-435BC/SUPPL_FILE/DISCLOSURES.PDF
15. Tan JZY, Ditchfield M, Freezer N (2012) Tracheobronchomalacia in children: review of diagnosis and definition. Pediatr Radiol 42:906–915. https://doi.org/10.1007/S00247-012-2367-5
16. Nemes R-M, Postolache P, Cojocaru D-C, Nitu M-F (2014) TRACHEOMALACIA IN CHILDREN AND ADULTS - NOT SO RARE AS EXPECTED. The Medical-Surgical Journal 118:608–611

17. Wright CD (2003) Tracheomalacia. Chest Surg Clin N Am 13:349–357. https://doi.org/10.1016/S1052-3359(03)00036-X
18. Masters IB, Chang AB (2009) Tracheobronchomalacia in children. Expert Rev Respir Med 3:425–439. https://doi.org/10.1586/ERS.09.29
19. Austin J, Ali T (2003) Tracheomalacia and bronchomalacia in children: pathophysiology, assessment, treatment and anaesthesia management. Paediatr Anaesth 13:3–11. https://doi.org/10.1046/J.1460-9592.2003.00802.X
20. Snijders D, Barbato A (2015) An Update on Diagnosis of Tracheomalacia in Children. Eur J Pediatr Surg 25:333–335. https://doi.org/10.1055/S-0035-1559816
21. Children's Minnesota Esophageal Atresia Symptoms & Surgery. https://www.childrensmn.org/services/care-specialties-departments/fetal-medicine/conditions-and-services/esophageal-atresia/. Accessed 3 Apr 2024
22. Tytgat SHAJ, van Herwaarden-Lindeboom MYA, van Tuyll van Serooskerken ES, van der Zee DC (2018) Thoracoscopic posterior tracheopexy during primary esophageal atresia repair: a new approach to prevent tracheomalacia complications. J Pediatr Surg 53:1420–1423. https://doi.org/https://doi.org/10.1016/j.jpedsurg.2018.04.024
23. Polites SF, Kotagal M, Wilcox LJ, et al (2018) Thoracoscopic posterior tracheopexy for tracheomalacia: A minimally invasive technique. J Pediatr Surg 53:2357–2360. https://doi.org/https://doi.org/10.1016/j.jpedsurg.2018.08.004
24. van Tuyll van Serooskerken ES, Tytgat SHAJ, Verweij JW, et al (2021) Primary Posterior Tracheopexy in Esophageal Atresia Decreases Respiratory Tract Infections. Front Pediatr 9:720618. https://doi.org/10.3389/FPED.2021.720618
25. Tytgat SHAJ, van Herwaarden-Lindeboom MYA, van Tuyll van Serooskerken ES, van der Zee DC (2018) Thoracoscopic posterior tracheopexy during primary esophageal atresia repair: a new approach to prevent tracheomalacia complications. J Pediatr Surg 53:1420–1423. https://doi.org/10.1016/J.JPEDSURG.2018.04.024
26. Murgu S, Colt H (2013) Subjective assessment using still bronchoscopic images misclassifies airway narrowing in laryngotracheal stenosis. Interact Cardiovasc Thorac Surg 16:655. https://doi.org/10.1093/ICVTS/IVT015
27. Mohammed S, Kamran A, Izadi S, et al (2024) Primary Posterior Tracheopexy at Time of Esophageal Atresia Repair Significantly Reduces Respiratory Morbidity. J Pediatr Surg 59:10–17. https://doi.org/10.1016/J.JPEDSURG.2023.09.028
28. Schneider CA, Rasband WS, Eliceiri KW (2012) NIH Image to ImageJ: 25 years of image analysis. Nature Methods 2012 9:7 9:671–675. https://doi.org/10.1038/nmeth.2089
29. Data Translation Inc (2003) GLOBAL LAB Image/2 ®
30. IPLEX Viewer Plus Software. https://www.olympus-ims.com/en/news/100-id.184549850.html. Accessed 24 Jun 2024
31. Norman G (2012) Bersoft Image Measurement Review
32. Iris — Iris 3.9.0 documentation. https://scitools-iris.readthedocs.io/en/stable/. Accessed 24 Jun 2024
33. Woo Mason, Woo Mason, OpenGL Architecture Review Board. (1999) OpenGL programming guide : the official guide to learning OpenGL, version 1.2. 730
34. Yang L, Kang B, Huang Z, et al Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data
35. Ren´ R, Ranftl R, Lasinger K, et al (2020) Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. IEEE Trans Pattern Anal Mach Intell XX:1

36. Monocular Depth Estimation: Exploring Techniques and Applications | AtharvMalusare | Nov, 2023 | Medium. https://medium.com/@atharvmalusare/a-comparative-study-on-monocular-depth-estimation-a12f6b847087. Accessed 8 Nov 2024

37. Gil D, Ortiz RM, Sánchez C, et al (2017) Objective Endoscopic Measurements of Central Airway Stenosis: A Pilot Study. Respiration 95:63–69. https://doi.org/10.1159/000479888

38. Yang L, Kang B, Huang Z, et al (2024) Depth Anything V2

39. Weng W, Zhu X (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. IEEE Access 9:16591–16603. https://doi.org/10.1109/ACCESS.2021.3053408

40. Clark AE, Biffi B, Sivera R, et al (2020) Developing and testing an algorithm for automatic segmentation of the fetal face from three-dimensional ultrasound images. R Soc Open Sci 7:201342. https://doi.org/10.1098/RSOS.201342

41. Wilson SM, Bautista A, Yen M, et al (2017) Validity and reliability of four language mapping paradigms. Neuroimage Clin 16:399–408. https://doi.org/10.1016/J.NICL.2016.03.015

42. Boehringer AS, Sanaat A, Arabi H, Zaidi H (2023) An active learning approach to train a deep learning algorithm for tumor segmentation from brain MR images. Insights Imaging 14:. https://doi.org/10.1186/S13244-023-01487-6

43. Understanding Evaluation Metrics in Medical Image Segmentation | by Nghi Huynh | Medium. https://medium.com/@nghihuynh_37300/understanding-evaluation-metrics-in-medical-image-segmentation-d289a373a3f. Accessed 14 Nov 2024

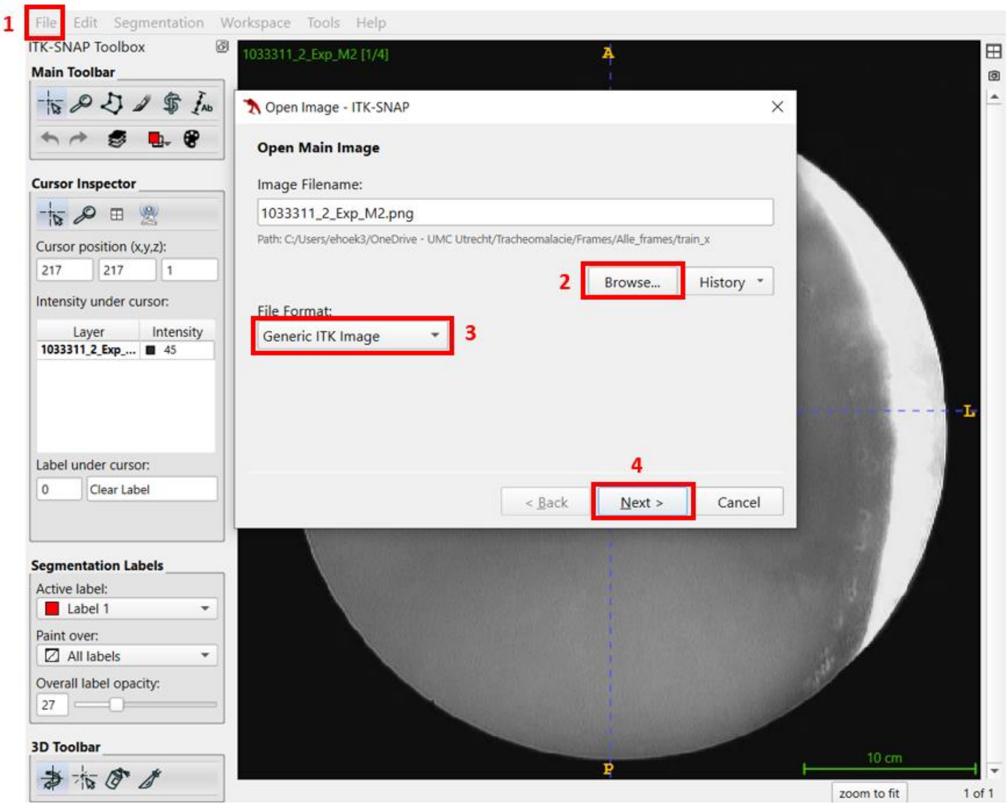
44. Yushkevich PA, Piven J, Hazlett HC, et al (2006) User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. Neuroimage 31:1116–1128. https://doi.org/10.1016/J.NEUROIMAGE.2006.01.015

45. NIfTI: — Neuroimaging Informatics Technology Initiative. https://nifti.nimh.nih.gov/. Accessed 8 Nov 2024

46. 3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.5.2 documentation. https://scikit-learn.org/stable/modules/cross_validation.html. Accessed 21 Oct 2024

47. Ronneberger O, Fischer P, Brox T U-Net: Convolutional Networks for Biomedical Image Segmentation

Appendix A. Workflow segmenting manually in ITK-SNAP



1033311_2_Exp_M2 [1/4]

Open Image - ITK-SNAP

Open Main Image

Image Filename: 1033311_2_Exp_M2.png

Path: C:/Users/ehoek3/OneDrive - UMC Utrecht/Tracheomalacie/Frames/Alle_frames/train_x

File Format: Generic ITK Image

Cursor position (x,y,z): 217 217 1

Intensity under cursor:

Layer	Intensity
1033311_2_Exp_...	■ 45

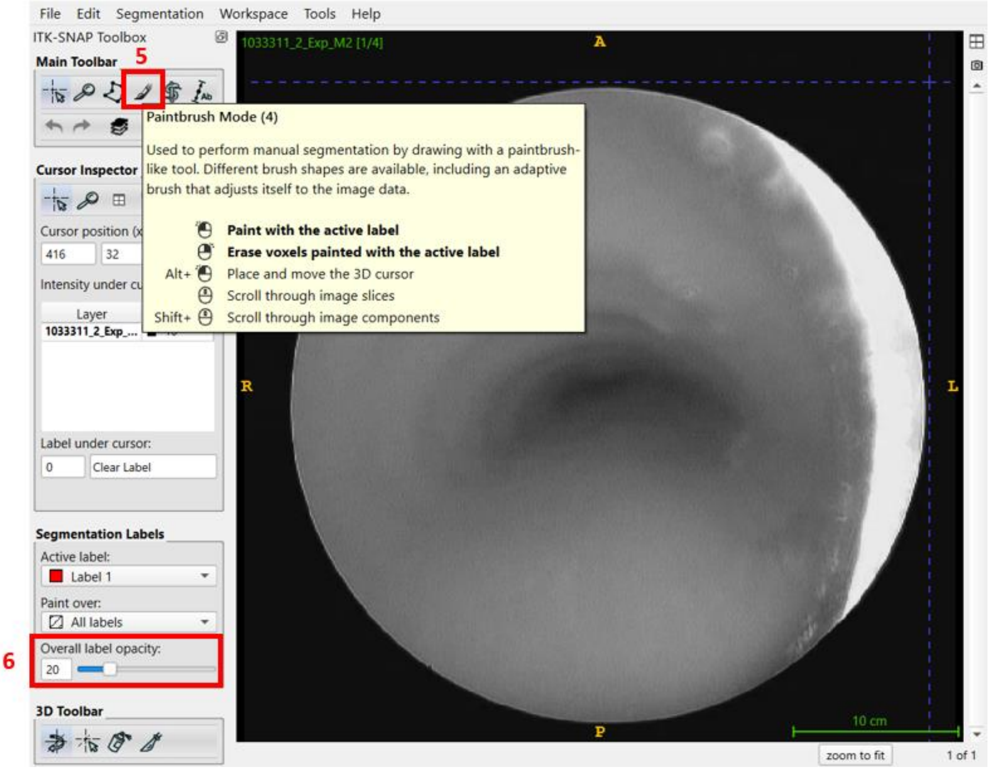
Label under cursor: 0 Clear Label

Active label: Label 1

Paint over: All labels

Overall label opacity: 27

Figure 30. Open a file in ITK-snap by clicking on 'file' in the upper bar (1), browse the wanted image (2), set file format to 'Generic ITK Image' (3) and click twice on 'Next >' (4).



1033311_2_Exp_M2 [1/4]

Paintbrush Mode (4)

Used to perform manual segmentation by drawing with a paintbrush-like tool. Different brush shapes are available, including an adaptive brush that adjusts itself to the image data.

- Paint with the active label
- Erase voxels painted with the active label
- Alt+ Place and move the 3D cursor
- Scroll through image slices
- Shift+ Scroll through image components

Cursor position (x: 416 32

Intensity under cu

Layer
1033311_2_Exp_...

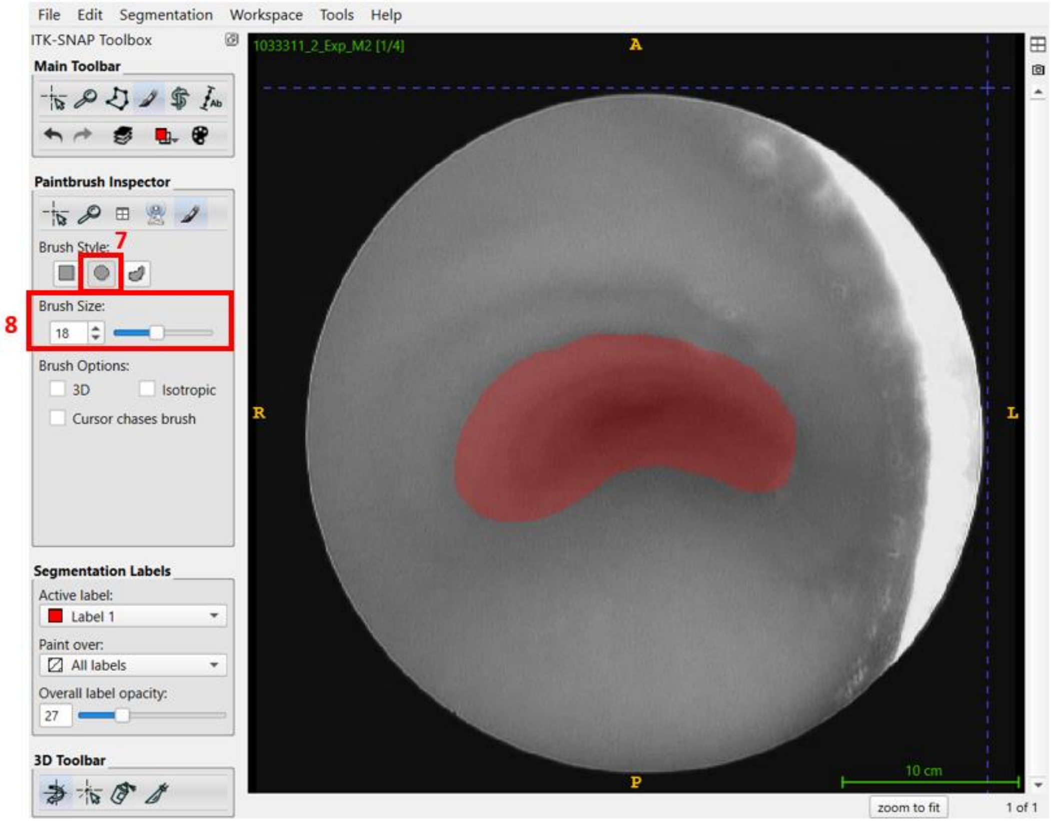
Label under cursor: 0 Clear Label

Active label: Label 1

Paint over: All labels

Overall label opacity: 20

Figure 31. To segment the wanted area, click on the paintbrush to switch to paintbrush mode (5). Set the overall label opacity to +-20 to have visualization of both the image and the segmented area (6).



Brush Size: 18

Active label: Label 1

Paint over: All labels

Overall label opacity: 27

Figure 32. Within the paintbrush mode, switch the brush style to 'round' (7). Adjust the brush size as wanted during segmentation (8).

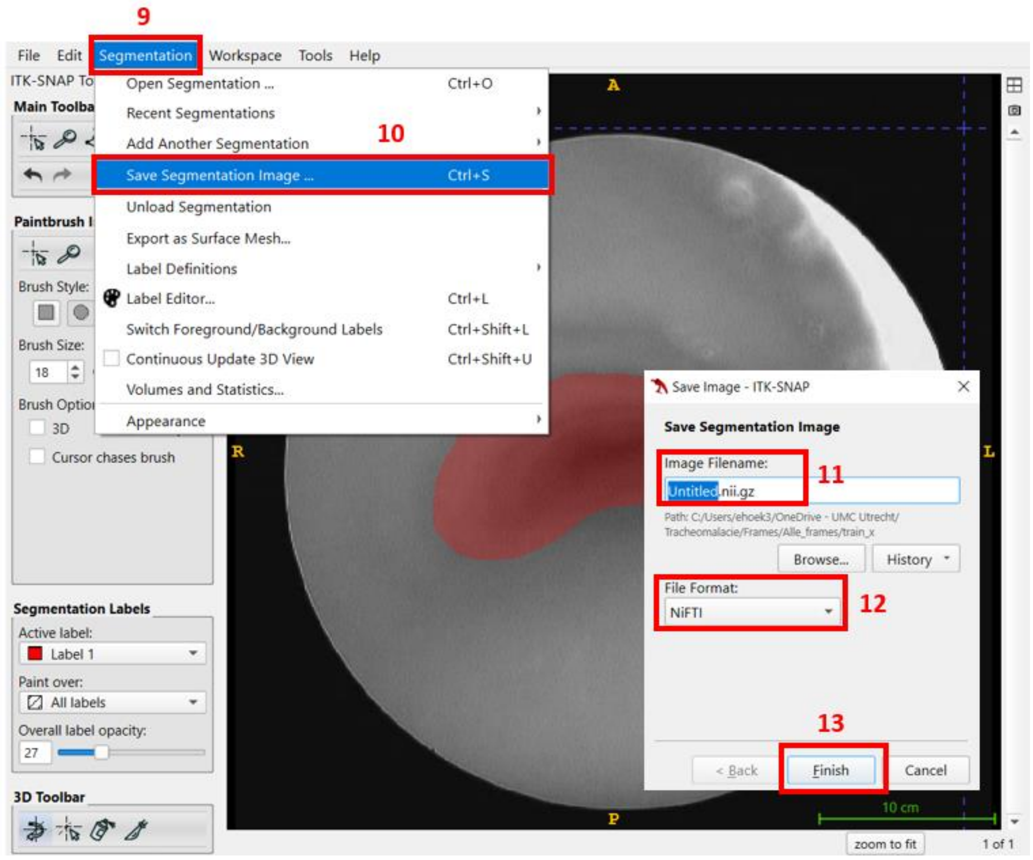


Image Filename: Untitled.nii.gz

Path: C:/Users/ehoek3/OneDrive - UMC Utrecht/Tracheomalacie/Frames/Alle_frames/train_x

File Format: NiFTI

Figure 33. Save the segmentation by clicking on 'Segmentation' in the upper bar (9), select 'Save Segmentation Image' (10), specify the file name (11), check if the file format is set to NiFTI (11) and click on 'Finish' (13).

Appendix B. Visualization of the outcomes of the U-Net model

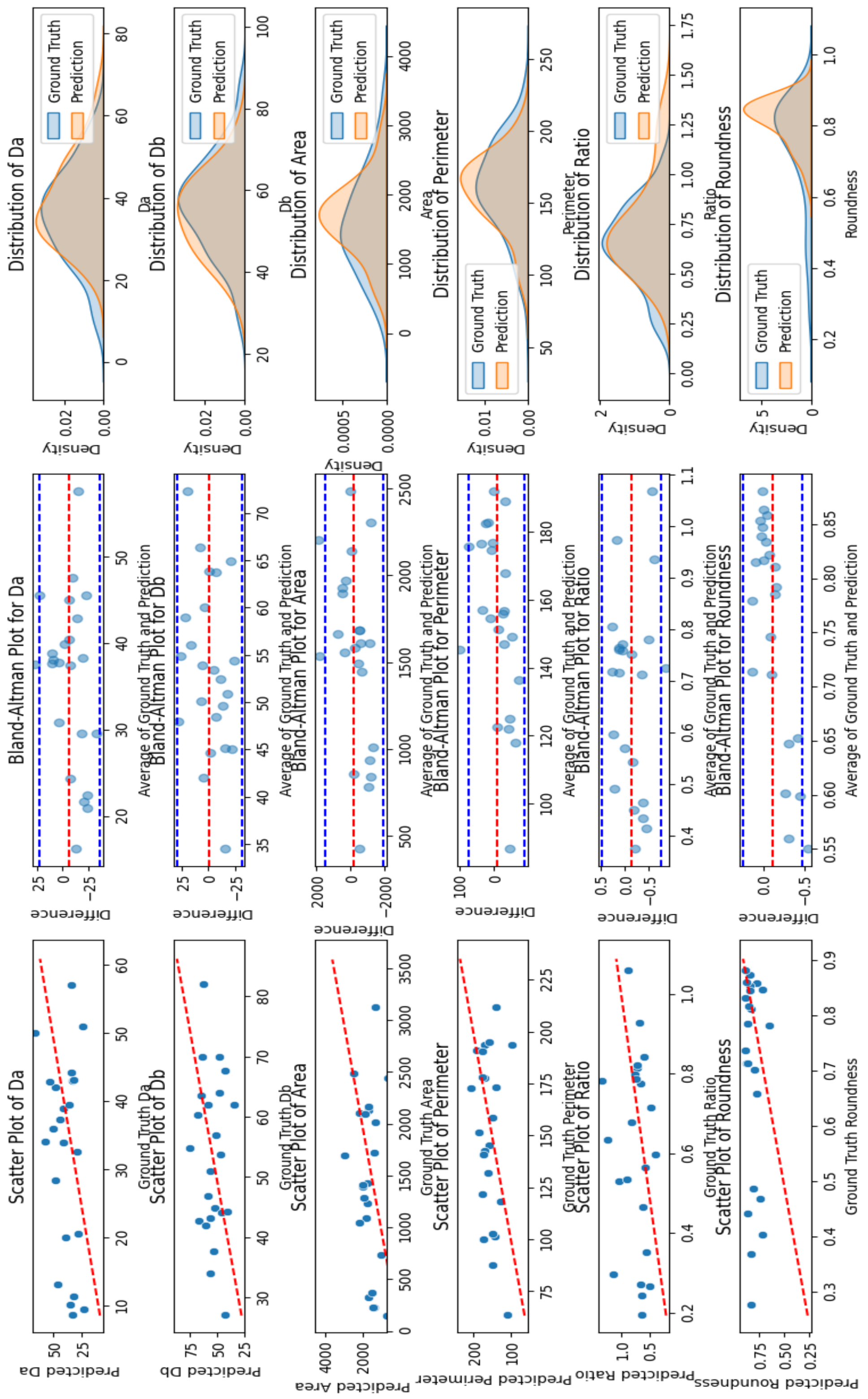


Figure 34. Scatter plots, Bland-Altman plots and Distribution of parameters for the U-Net algorithm

Appendix C. Train and test ranked correlation results

Standard U-Net model

Train results:

Roundness:	0.87812
Ratio:	0.84855
dA:	0.83774
Area:	0.75878
dB:	0.67945
Perimeter:	0.67204

Test results:

Ratio:	0.92712
dA:	0.77478
Roundness:	0.60315
dB:	0.57008
Area:	0.33900
Perimeter:	0.30055

Depth-Anything model

Train results:

Roundness:	0.150439
Area:	0.043903
dA:	0.020469
Ratio:	-0.07742
Perimeter:	-0.08425
dB:	-0.10536

Test results:

Ratio:	0.69850
Perimeter:	0.64817
dB:	0.62107
Area:	0.52916
dA:	0.51398
Roundness:	0.11292

U-Net model using Depth-Anything images

Train results:

Perimeter:	0.12282
dA:	0.11004
Area:	0.10085
Roundness:	0.06170
Ratio:	0.05939
dB:	0.02234

Test results:

Area:	0.50952
Perimeter:	0.41780
Ratio:	0.26190

dB:	-0.0966
Roundness:	-0.1746
dA:	-0.3741

U-Net model using two images

Train results:

dA:	0.87982
Ratio:	0.83427
Area:	0.79512
dB:	0.73379
Roundness:	0.65200
Perimeter:	0.61931

Test results:

Ratio:	0.87337
dA:	0.77407
Roundness:	0.73922
dB:	0.57064
Area:	0.39371
Perimeter:	0.32473