Automatic Segmentation of the Aortic Wall from CT Scans with U-Net

Yucheng Fu[1] and Christian Ellehammer Andersen[1]

Technical University of Denmark
{s194241, s194255}@student.dtu.dk

Abstract. The aorta is the largest artery in the human body, but prone to disease if the aortic wall is weakened. Segmenting the wall of the aorta from a medical scan is therefore of great interest in the medical field. We propose using a 2D convolutional neural network (CNN) inspired by U-Net for segmenting the aortic wall, trained on sparsely annotated data, where only six 2D slices were annotated per scan, instead of the entire 3D volume. Evaluating on a test set which also consisted of sparsely annotated data yields a Dice score of 0.58 ± 0.07 and a Hausdorff distance of 10.53 ± 19.98 mm which shows that our model is able to segment the wall in 2D with reasonable accuracy. However, our attempts to improve the performance by creating artificial training ground truths with interpolation were unsuccessful. Passing information about the previous slice and its annotation was not useful either, we found. Further research is needed to explore more effective ways of handling the sparsity of the data, either by improving the interpolation idea or using completely alternative methods to enhance the sparse training data.

Keywords: U-Net · Aortic wall segmentation

1 Introduction

The aorta is the largest artery in the human body. It originates from the heart and then extends upwards to form the ascending aorta, before descending to form the descending aorta. The aorta extends further down into the abdomen, forming the abdominal aorta, and ends at the aortic bifurcation. The main task of the aorta is to supply the rest of the body with oxygenated blood.[5]

The aorta can be prone to illnesses such as aortic aneurysms, which is clinically defined as a $\geq 50\%$ increase in aortic diameter compared to the expected normal diameter[2], or aortic dissections, where a tear in the inner layer of the aortic wall occurs, causing the aortic wall to be filled with blood. These illnesses are potentially fatal in case of rupture and require urgent surgical intervention.

Early diagnosis and intervention of aortic aneurysms and dissections are therefore of the utmost importance. An important part of this process is segmenting the aortic wall to inspect it further, as factors such as wall thinning, aortic enlargement and wall stress are leading causes for aortic rupture and dissection, according to Erbel & Eggebrecht (2006)[1].

In Ronneberger et al. (2015)[4], they propose using a deep convolutional neural network (CNN) for image segmentation, called U-Net. The architecture of the U-Net model is shaped like the letter U: First is a contracting network with convolutional layers and max pooling operations that extract features and downsize the image. Then an expanding network uses transposed convolutions to upscale the image back to its original size. Additionally, there are skip connections between the contracting and expanding network to preserve lower-level features.

Although originally used for biomedical image segmentation, U-Net has also been used in medical image analysis, specifically for segmenting the wall of the aorta. In Piri et. al (2021)[3], they use a CNN based on U-Net for segmenting the whole aorta. By comparing the automatic segmentations with manual segmentations of 49 healthy individuals with low cardiovascular disease risk, they find that the automatic segmentations from the CNN are 13-17% smaller than the manually segmented volumes. Piri et. al find the aortic wall on the automatic segmentation by including the voxels at a certain distance from the edge of the aorta segmentation: within 3 mm of the edge on the inside and within 2 mm of the edge on the outside.

In this paper, we attempt to automatically segment the aortic wall of the abdominal region directly from CT scans of healthy individuals. We will be using a 2D U-Net which will perform segmentations slice-by-slice in the axial plane. However, training the U-Net requires ground truth segmentations. Given that the wall is so difficult to see, the ground truth segmentation has to be carved out manually slice-by-slice, which is a slow and tedious process. Therefore, obtaining the ground truth segmentation of the entire aortic wall is unfeasible. Instead, we propose to only manually segment some of the axial plane slices, train and validate on those and then feed the model the entire 3D scan slice-by-slice during inference.

This approach is expanded upon by creating artificial training samples by interpolating between the manual segmentations. With the interpolated samples, we can provide the U-Net with contextual information about the 3D structure of the aorta, such that the network does not output completely independent predictions of the aortic wall for consecutive slices. We want to answer the following:

- How well does the U-Net perform at segmenting the aortic wall from a 3D scan?
- How does training the model on the artificial training samples affect the performance?
- Does providing the U-Net with additional context improve the performance of the network?

2 Data

The dataset consists of 30 CT scans obtained from Rigshospitalet. The scans contain the entire abdominal part of the aorta and are from individuals with

healthy aortas. They all have the isotropic voxel spacing 0.25 mm $\times$ 0.25 mm $\times$ 0.25 mm.

In the ideal world, we would have full 3D segmentations of the aortic wall for all 30 CT scans made by an experienced annotator, but those have not been possible to obtain due to limited resources. Instead, we manually annotated six slices where the wall was prominent enough to see in each scan. On the scans, the lumen of the aorta is the most visible tissue, due to the contrast agent that was administered to the patient. The lumen is surrounded by a barely visible band, separating the aorta from the surrounding tissue. For healthy individuals, the wall is less than 4 mm thick [1].

For the annotation process, we used the software 3D Slicer. Figure 1 below shows a slice of the aorta with and without the manual annotation.

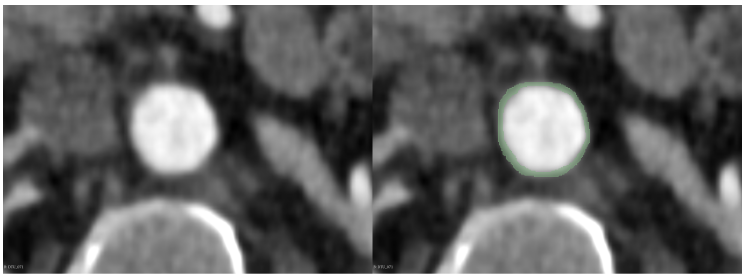


Fig. 1. (a): Example of aorta slice viewed from the axial plane and (b): same slice with manual annotation.

In total, we have 180 annotated slices in the dataset, where 108 are used for training, 36 are used for validation and 36 are used for testing.

3 Methods

3.1 Interpolation of data

As mentioned earlier, we wish to get more training samples by interpolating between the six annotated slices for each scan. The idea is illustrated in figure 2. After interpolation, lots of artificial wall segmentations are created between the manual segmentations. On figure 2, the black arrow illustrates the z-axis. Before interpolating, we *only* have wall segmentations for six, discrete values of z. The idea of interpolation is to create a function $f(z)$ that can output the wall segmentation for *any* given value of z. Such function can be used to estimate the segmentations for z-coordinates where no manual segmentation exists.

A wall segmentation is shaped like a ring which consists of an outer and inner band edge. We use two different functions $f(z)$, one that keeps track of

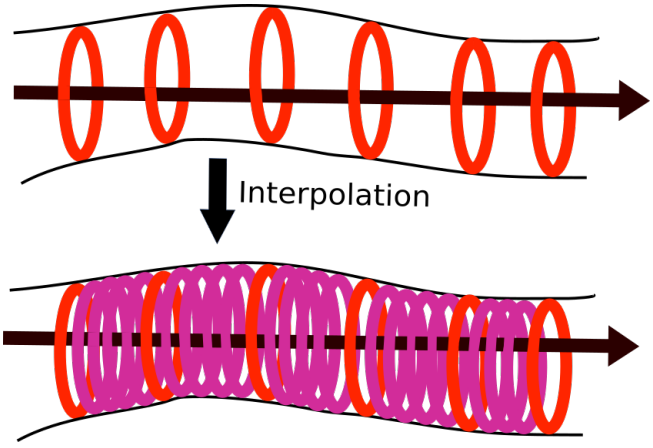


Fig. 2. Interpolation creates artificial segmentations between the manually annotated ones.

the inner edge and one for the outer edge. We can obtain the wall segmentation by predicting the inner and outer edges, and then filling out the space between them.

Each band edge is modelled as a fixed number N of points. This is illustrated on figure 3 where the purple and blue dots make up the inner and outer band edges respectively. After fitting $f(z)$ on, say, the outer band edge, we can call it for any value of z and receive N points that make up the outer band edge at that value of z. Doing that for both band edges and filling out the space in between for all values of z, yields the full segmentation of the entire aortic wall. To create the inner and outer band edges, we use $N = 2000$ points to ensure

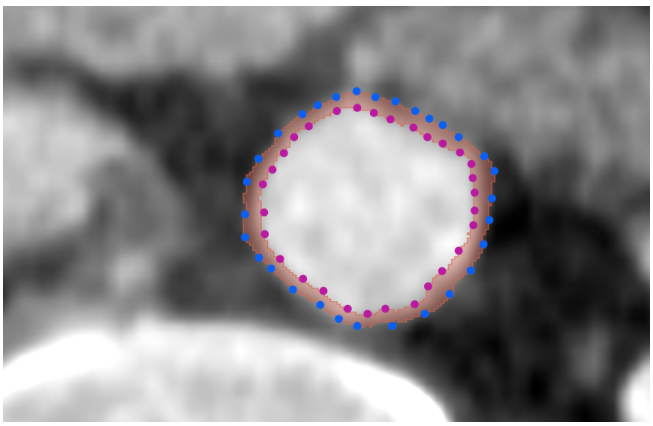


Fig. 3. Band edges modelled as a fixed number of $N = 30$ points.

that there are no holes.

After interpolation, we have in total 2066 slices with "ground truths" in the training set and 746 in the validation set, but keep the original 36 slices in the test set.

3.2 Models

The model is inspired by the U-Net architecture in [4]. However, instead of contracting the input using max pooling operations, the filters in the convolutional layers are applied with a `stride` of 2. The architecture is shown in figure 4.

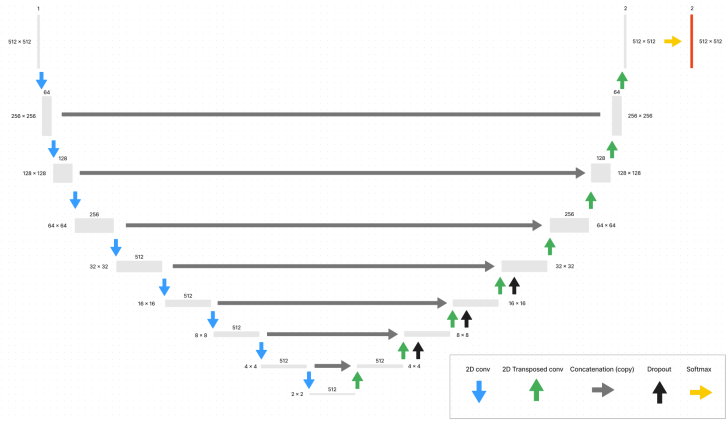


Fig. 4. U-Net. The numbers show the dimensionality of the input at each stage in the network.

With the interpolated segmentations, it is possible to propagate additional context of the 3D structure of the aorta to the 2D U-Net. The contextual U-Net is inspired by Zheng et al. (2018)[6]. In addition to a regular U-Net, a contextual input in the form of the preceding slice and label map is put through a contracting path and concatenated with the input just before the bottleneck of the U-Net, as seen in figure 5

The idea of adding contextual information is to help the model understand the 3D structure of the aortic wall better. With contextual information, there is some dependency along the z-dimension which could mitigate the variation in predictions for consecutive slices. In total, we train and evaluate three different models.

- Model A: 2D U-Net trained only on the manually annotated data.
- Model B: 2D U-Net trained on the interpolated data.
- Model C: 2D Contextual U-Net trained on the interpolated data.

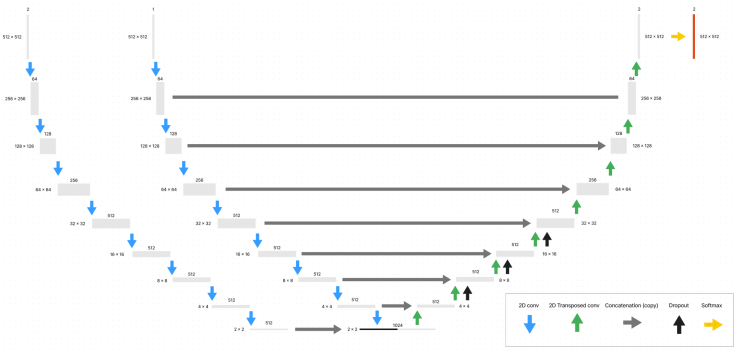


Fig. 5. Contextual U-Net. The numbers show the dimensionality of the input at each stage in the network.

Since we only have one class, the wall, this could be modelled as a binary problem. We opted not to do that because it would require us to manually adjust a confidence threshold. Instead, we simply have two different classes, wall with label 1 and non-wall (background and lumen) with label 0. The shape of the network's output tensor is $(N_{batches} \times 2 \times 512 \times 512$ because there are 2 classes and the spatial dimensions are 512×512.

The manual segmentations, called the target, and the output from the network, called the prediction, are used during training in the following loss function:

$$\text{loss} = CE(p, t) + \lambda \cdot D(p, t)$$

where p is the prediction, t is the target, CE is the cross-entropy loss between prediction and target, $D(p, t)$ is the Dice score between the prediction and target, and λ is a weight factor for the Dice score. The target is severely imbalanced, as there are many more zeros than ones, which affects the cross-entropy loss. To make up for this, each class is weighted according to how often it appears in the training set. Furthermore, we set $\lambda = 10$, such that the Dice score is more heavily favoured in the loss term.

The Hounsfield units (HU) of the slices that are fed to the model are clamped between -100 and 500, meaning HU values below -100 are set to -100 and HU values above 500 are set to 500. HU values are then rescaled to the range $[0, 1]$. Additionally, each slice is scaled to the size 512×512. All models are trained with the same parameters: `batch size` $= 16$, `lr` $= 3 \cdot 10^{-4}$, `epochs` $= 1000$ and the weights are initialised with Xavier initialisation.

Inference in 3D During the test/inference phase, the masks are made by argmaxing along the channel dimension. The model simply selects the class with the highest probability.

To infer in 3D, models A and B (which do not use contextual information) make predicted segmentation on each axial plane slice in the scan individually. These outputs are then stacked on top of each other to form a 3D segmentation.

For models A and B, the order in which the slides are fed is irrelevant. For model C, the slices must be fed in a specific order, starting from the top and moving down. The model's output in the previous slice will be included in the contextual tensor when segmenting the current slice.

3.3 Experimental design

Our experiment will be about comparing Model A (the one with no interpolation), B (interpolation but no context) and C (interpolation and context). We will evaluate the performance based on six test scans, with six manually annotated slices each, yielding 36 test slices in total. During inference, the contextual input for Model C comes from the preceding interpolated slice.

For quantitative evaluation, we use the Dice score and Hausdorff distance as metrics. The Dice score is defined as

$$\text{Dice} = \frac{2 \cdot |A \cap B|}{|A| + |B|}$$

It is a measure of the amount of overlap between the predicted segmentation A and ground truth B. The Dice score is in the interval [0,1] where 1 means complete overlap between A and B.

The Hausdorff distance is defined as

$$H(A, B) = \max\{\max_{a \in A} \min_{b \in B} \mathrm{d}(a, b), \max_{b \in B} \min_{a \in A} \mathrm{d}(a, b)\}$$

where $\mathrm{d}(a, b)$ is the distance between point a and b. Briefly put, it measures the greatest minimum distance between the prediction A and the ground truth segmentation B. The lower the Hausdorff distance, the closer the prediction is to the ground truth.

4 Results

The experiment compares three models: A (the one with no interpolation), B (interpolation but no contextual) and C (interpolation and contextual). The results are presented in table 1.

Table 1. Dice Score and Hausdorff Distance for models A, B and C. The values are given as mean ± standard deviation.

model	Dice Score	Hausdorff Distance
A	$\mathbf{0.58 \pm 0.07}$	$\mathbf{10.53 \pm 19.98}$ mm
B	0.28 ± 0.05	20.56 ± 20.19 mm
C	0.24 ± 0.06	42.47 ± 31.46 mm

Table 1 shows the performances of the three models on the test set. Model A outperformed the other models on the Dice score, with a mean value of 0.58 compared to 0.28 for model B and 0.24 for model C. A similar difference is reflected in the Hausdorff distance.

Models B and C have relatively similar performance based on the Dice score. However, when looking at the Hausdorff distance, model B scores on average 21.91 mm lower than C. It is also worth noticing that the Hausdorff distance has a very high standard deviation. This is an interesting phenomenon that will be discussed later.

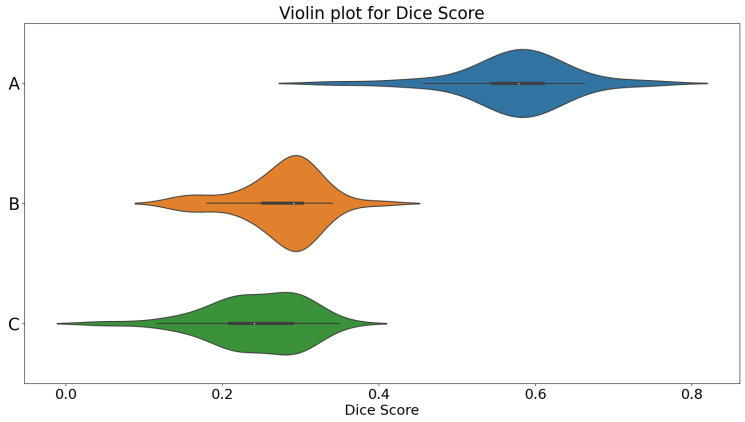


Fig. 6. Violin plot comparing the performance of models A, B, and C using the dice score as a metric

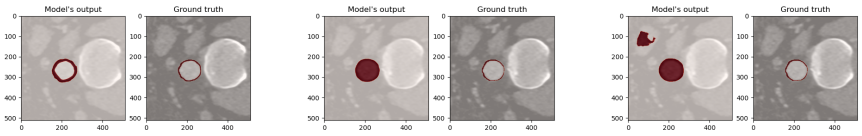
Figure 6 is a violin plot over the performances of the models when using the dice score as a metric. Model A seems to be performing better than both models B and C. Model A has the most consistent performance, with most of its values between 0.55 and 0.60. Model B has a peak around 0.3 and even its highest value, 0.40, is quite far from most of the scores of model A. Model C seems to be performing worse than B because it has more scores around 0.2.

Table 1 and figure 6 both show a clear conclusion: Model A performs the best. Based on the experiment, we can conclude that interpolating between the slices does *not* improve the performance.

5 Discussion

5.1 Lack of improvement in model performance when using interpolated slices

Interpolating between the manually segmented slices for extra training data did not improve performance. As seen on figure 7, the output of model A is a ring *around* the lumen which approximates the wall. However, models B and C end up segmenting the lumen itself. That is why B and C achieve such low Dice scores. Figure 8 shows an example of a full 3D wall segmentation that has been



(a) Prediction of model A (b) Prediction of model B (c) Prediction of model C

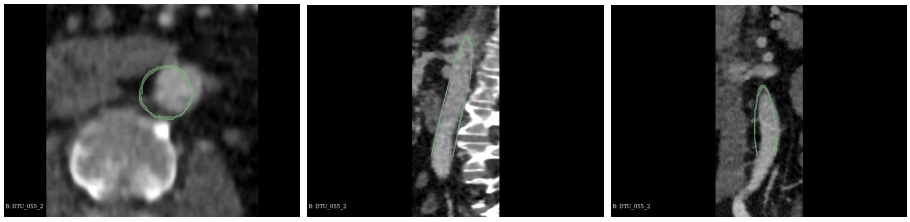
Fig. 7. Model predictions on one slice.

created by interpolating between the manually segmented slices. In most places, the interpolations are a good approximation of where the wall really is. However, the interpolation is not able to adapt to the curves. This may have caused models B and C to learn a biased or incorrect representation of the structures in the scans, leading to poor performance when applied to the test data. Instead of adding more real-world variance to the training phase, it seems the interpolated scans just added more noise. It seems they have learned to look for the gradient when going from wall to background, but not when going from lumen to wall.

Model A, on the hand, only received good segmentations. When training, it does not try to learn to emulate a faulty ground truth which results in a better model.

Interpolation might still be applicable, though. Making more manual annotations and placing them around curves could be a way to make the interpolation error smaller. Hopefully, this would mean the artificial segmentations added diversity to the training phase, instead of just adding noise.

Apart from this, we could add a term to the loss function that encourages the model to output a ring-like shape instead of a circle. Every pixel classified as wall by the model should have a distance to the centre of the ground truth segmentation more or less equal to the radius of the ground truth. The scaling of this term, however, must not be so large that it forces the model to output arbitrary rings.



(a) (b) (c)

Fig. 8. Instance of where interpolation fails, seen from the (a) axial (b) coronal and (c) sagittal view.

5.2 Impact of false positives on Hausdorff distance

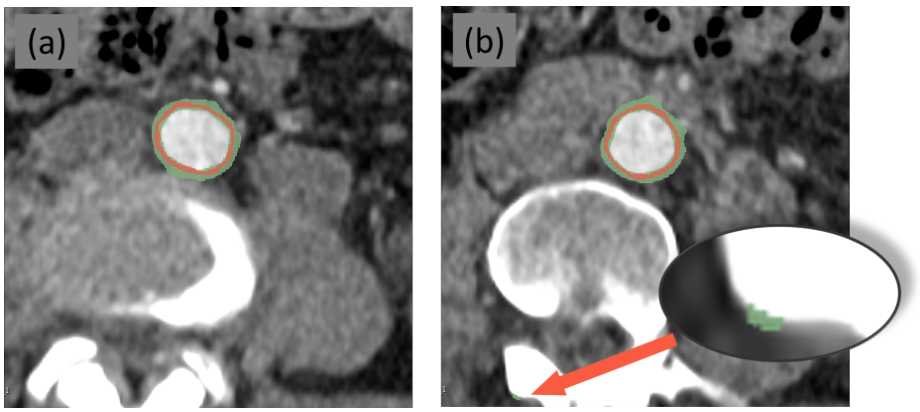


Fig. 9. Scan (a) has Dice score 0.53 and Haussdorf distance 14. Scan (b) has Dice score 0.54 and Hausdorff distance 416

On table 1, we observed very large standard deviations for the Hausdorff distance. On figure 9, there are two predicted segmentations plotted against the ground truth. The Dice scores are almost similar but the Hausdorff distance is much higher on figure 9b. This is due to the presence of false positives far away from the ground truth. The Hausdorff distance essentially measures the distance to the point that is furthest away from the ground truth. A false positive - an "island" - far very far away from the ground truth can therefore have a tremendous effect on the Hausdorff distance. The Dice score, on the other hand, is not nearly as dramatically affected by the presence of false positives. This explains the large standard deviations on Hausdorff distances in table 1. The false positives are not problematic as they can easily be removed using post-processing techniques, where only the largest connected component is kept. This would likely also improve the Dice scores and Hausdorff distances.

On table 1, we observed that model B outperformed C on Hausdorff distance. We believe this difference is due to the fact that model C had a bigger tendency than B to produce false positives far away from the ground truth (see figure 8c). The contextual tensor that model C receives is based on the interpolations. Since they are erroneous, the contextual tensor will merely add noise.

It is possible that adding the Hausdorff distance as a term in the loss function could help remove those undesired false positives. Penalising the model for predicting the wall class on pixels far away from the ground truth might make the model less prone to do so. However, adding the Hausdorff term would make the loss function harder to scale.

5.3 Qualitative assessment of the performance of model A in 3D

The U-Net only outputs segmentations in 2D, but we can extract the full 3D volume by feeding slices one at a time and combining the outputs. It is not possible to perform a quantitative test on this, though, since we do not have ground truth segmentations in 3D. Figure 10 shows an example of the output of model A in 3D. On figure 10, (a) is the 3D-view and (b) the sagittal plane.

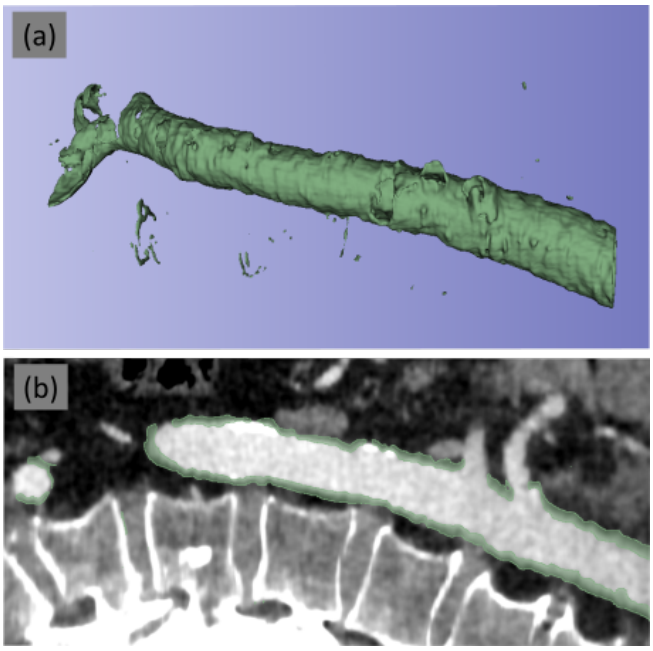


Fig. 10. Segmentation of test slice by model A (a) 3D view and (b) sagittal plane

Figure 10 demonstrates that the output of the U-Net is relatively smooth. One

might be worried that the U-Net would miss the wall on a few slices, making the 3D output inconsistent, however that is not the case.

On figure 10 a, there are some isolated islands. These are the false positives discussed earlier, which the model wrongly assumes to be part of the aortic wall.

On figure 10b, we see that the segmentation follows the edge of the aorta all the way down. Despite not knowing anything about the previous and following slices, the U-Net manages to output a consistent 3D surface. The thickness of the wall seems to be stable along the aorta, although there are some variations.

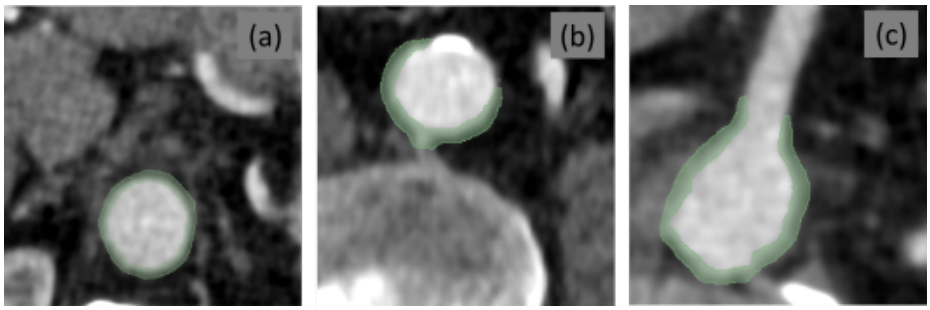


Fig. 11. (a): The behaviour on a typical slice. (b): When plaque is present. (c): When an artery exits

On figure 11, there are slices from the axial plane which is the plane the models get as input. Figure 11a is an example of the behaviour we want, where the predicted segmentation surrounds the entire aorta. This is what most slices in the axial look like for the axial plane. General feedback from clinical experts indicates that the automatically segmented wall is thicker than what is typically observed in healthy individuals.

There is plaque on figure 11b and that confuses our model, it seems. To be anatomically correct, the plaque would have to be included in the wall; but here, the predicted wall segmentation "stops" when it encounters the plaque. As a result, it does not encircle the entire aorta as it should. The model interprets the plaque as a different element, likely because of the dramatic difference in HU value between the two. One could speculate that adding more training data with slices including plaque would correct this behaviour. The presence of plaque means that the outputted 3D segmentation has undesired holes which were more apparent in the 3D viewing tool.

Figure 11c shows that the predicted segmentation accurately follows the out-branching artery for a few millimetres before stopping, as expected. This is a sign that our model does *not* merely draw arbitrary circles, but that it adapts its behaviour to match the shape of the wall.

A limitation of the current approach is that the model is only trained on the aortic walls of healthy individuals. The aortic wall for a sick individual can look vastly different and the model might not generalise well to those.

To sum up, the segmented wall is generally smooth, with a few undesired holes where plaque is present. The wall is a little too thick compared to the aortas of healthy individuals which also to some degree explains the low Dice score for model A. Similar remarks could be made about the other test scans. Although this is *not* a quantitative study, it does indicate that training a U-Net to segment the aortic wall using only sparse annotations is feasible.

6 Conclusion

Our experiments have shown that a U-Net trained on sparse annotations can learn to segment the wall of the aorta from CT scans of healthy individuals. When tested on 2D slices with ground truth segmentations, we achieved an average Dice score of 0.58 and an average Hausdorff distance of 10.53 mm. These results show that the model is able to segment the aortic wall in 2D with reasonable accuracy.

Qualitative inspections of the model's ability to extract the full 3D volume showed promising results. The segmentation output is smooth and consistent for the most part. The wall was placed accurately, although too thick compared to the aortas of healthy individuals. It struggles with plaque which it wrongly assumes to be a different structure. However, ground truth annotations in 3D are needed to give a satisfactory assessment of the model's performance in 3D.

To combat the sparsity of the training date, we tried to interpolate segmentations between the manually segmented slices. Unfortunately, this did not improve the performance, as the model started outputting solid circles instead of rings. We believe the quality of the interpolated segmentations was too poor, and that they added noise rather than useful variance. Adding a contextual tensor with the previous slice and its annotation, only made matters worse. Further investigation is required to make this data enhancement work.

These findings only apply to CT scans of healthy individuals, and further research is required to adequately assess the model's performance on other populations.

References

1. Raimund Erbel and Holger Eggebrecht. Aortic dimensions and the risk of dissection. *Heart*, 92(1):137–142, 2006.
2. KW Johnston, RB Rutherford, MD Tilson, DM Shah, L Hollier, and JC Stanley. Suggested standards for reporting on arterial aneurysms. subcommittee on reporting standards for arterial aneurysms, ad hoc committee on reporting standards, society for vascular surgery and north american chapter, international society for cardiovascular surgery. *Journal of vascular surgery*, 13(3):452—458, March 1991.
3. Reza Piri, Lars Edenbrandt, Måns Larsson, Olof Enqvist, Amalie Horstmann Nøddeskou-Fink, Oke Gerke, and Poul Flemming Høilund-Carlsen. Aortic wall segmentation in f-sodium fluoride pet/ct scans: Head-to-head comparison of artificial intelligence-based versus manual segmentation. *Journal of nuclear cardiology : official publication of the American Society of Nuclear Cardiology*, 29(4):2001—2010, August 2022.

4. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
5. Clement Tran, Cheng Wu, Stephen Bordes, and Forshing Lui. *Anatomy, Abdomen and Pelvis, Abdominal Aorta*. 06 2022.
6. Qiao Zheng, Hervé Delingette, Nicolas Duchateau, and Nicholas Ayache. 3d consistent amp; robust segmentation of cardiac images by deep learning with spatial propagation, 2018.