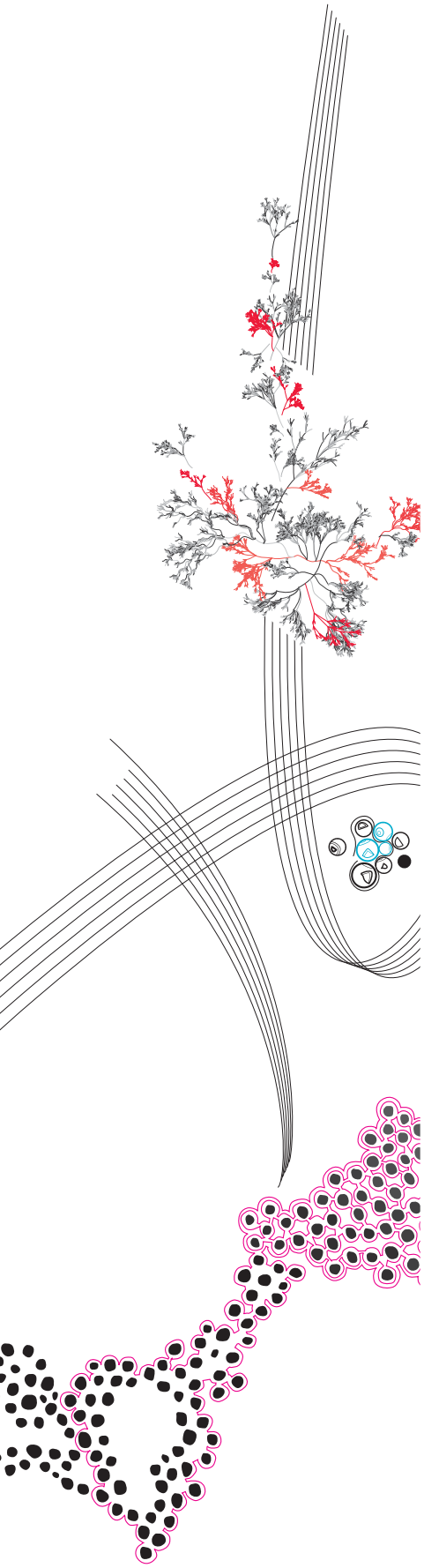


BSc Thesis Applied Mathematics

# Bayesian uncertainty estimation of deep learning carotid artery vessel wall segmentation

Sophie Buurman

Supervisors: Dieuwertje Alblas, MSc.
Dr. Jelmer M. Wolterink

July, 2022

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

## Preface

When I started this bachelor thesis, I did not even know what a neural network was, let alone how to train one and analyse its uncertainty. Still, I was determined to learn more about deep learning and uncover its mathematical secrets. At the same time, I felt motivated to help out doctors and patients suffering from atherosclerosis by contributing to this project. It has been a steep but entertaining learning process and I could not have done it without the help of the following people.

First and foremost, I want to thank my supervisor Dieuwertje Alblas for the pleasant and very helpful weekly meetings. Thank you for always being there to answer questions, to provide suggestions and to think along with me. You really helped me in the right direction and assisted me to improve the quality of this bachelor assignment. Many thanks to Jelmer Wolterink and Christoph Brune for taking the time to critically listen to my presentation, both complimenting my work and giving constructive feedback. You pointed out the limitations to my research and helped me question my knowledge. Also, I would like to thank my mother, dr. ing. Louise Wipfler, for thinking along with me and motivating me to go on, when I got stuck again whilst debugging code. Last but not least, thanks to all friends and family interested in my work, for having me explain what I was doing in layman's terms, for motivating me and for forcing me to think about the essence and aim of this assignment.

# Bayesian uncertainty estimation of deep learning carotid artery vessel wall segmentation

Sophie Buurman*

July, 2022

**Abstract**

Monitoring patients with atherosclerosis demands measurements of the thickness of the carotid artery vessel wall. An accurate segmentation is essential for these measurements, however manual acquisition is extremely time consuming. Recently, Alblas et al. [2] proposed a fully automatic method for vessel wall segmentation on 3D MRI images, ensuring ring-shaped segmentations. The method returns contour points describing two nested circles, representing the lumen and outer wall on each axial slice of the image. Although very successful, the model returns a prediction regardless of the underlying image quality. This can be problematic in medical images that contain regions of noise or artefacts, as the model should indicate the segmentations are uncertain around those regions. Therefore, we propose the use of dropout layers in the convolutional neural network of Alblas et al., introducing stochasticity in the network. These dropout layers can be used to approximate the posterior predictive distribution by passing multiple stochastic inferences through the network. The predictive mean and variance are calculated for each of the predictive contour points. As we hypothesized, we observe a substantial higher variance for low quality image data, as well as near the carotid bifurcation.

*Keywords*: atherosclerosis, Bayesian neural networks, dropout, convolutional neural networks, deep learning, carotid artery, segmentation, uncertainty quantification, magnetic resonance imaging, anatomic prior

## 1 Introduction

Atherosclerosis is a cardiovascular disease that is characterized by plaque build-ups on the vessel wall. It causes narrowing of the artery and can lead to ischemic stroke, one of the leading causes of death in the modern developed world [14]. In Figure 1, the artery system in the neck region is depicted. The left and right carotid arteries split at the bifurcation into the internal and external carotid arteries.

When the build-up of plaques is suspected to take place in a carotid artery, the progression can be visualized using black-blood magnetic resonance imaging (BB-MRI), which produces three dimensional scans of the neck region. From these scans, the thickness of the plaque lesions can be measured by segmenting the vessel wall. Segmentation is the identification of regions of interest in image data. In our case this concerns the carotid artery vessel wall in 3D MR images encompassing the neck region. Manual segmentation of the carotid artery vessel wall is a time-consuming task, because the MRI scans consist of hundreds of axial slices that ideally should

*Email: s.buurman@student.utwente.nl

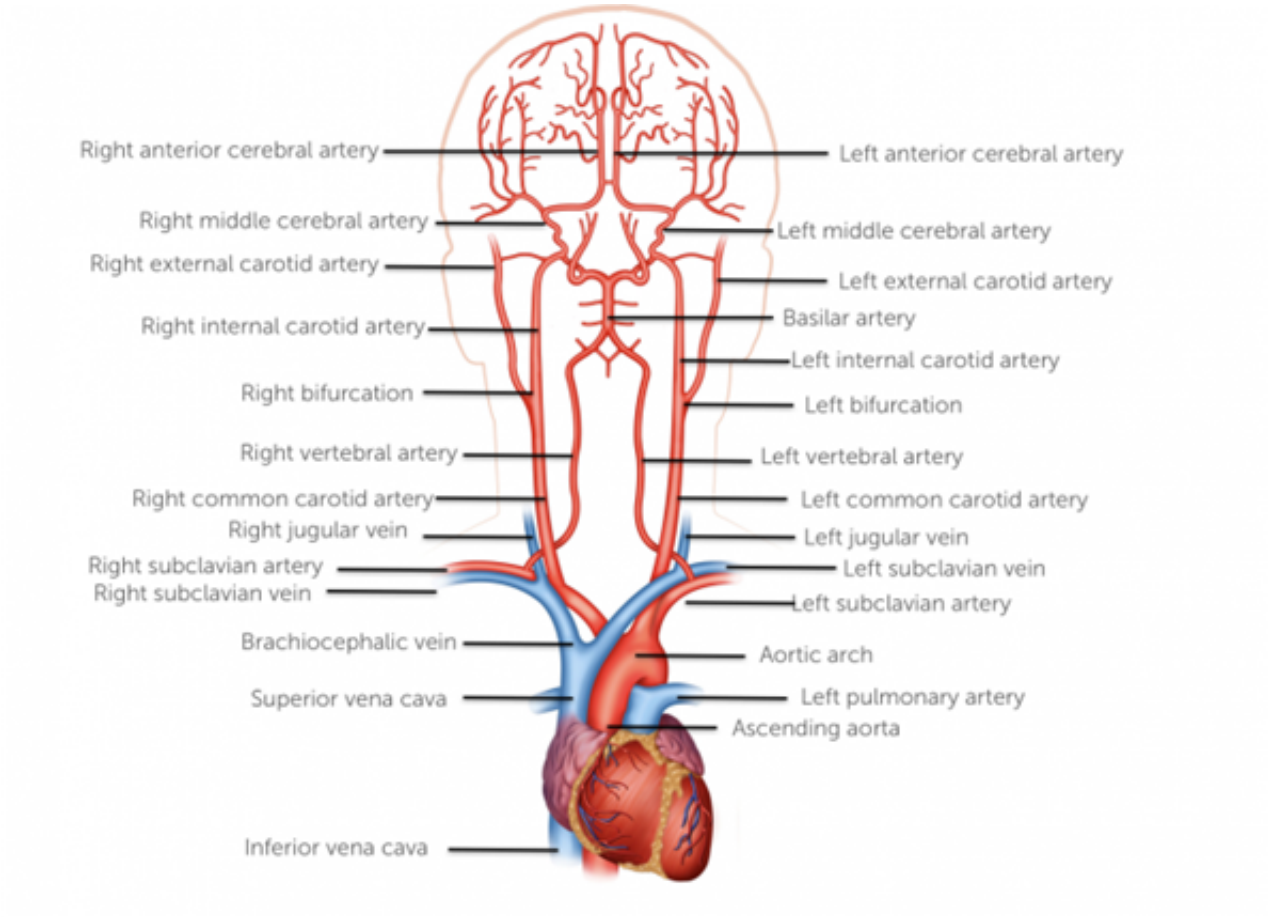



Figure 1: Arteries going from the neck to the head. In green is our region of interest: the left and right common carotid arteries, splitting into the internal and external carotid arteries at the bifurcation. Image adapted from [4].

all receive annotations. Moreover, these annotations are prone to inter- and intra-observer variability.

Automating the segmentation process rapidly decreases segmentation time and omits variability introduced by human operators. Traditionally, model-driven segmentation techniques are used for segmentation. These techniques utilize prior shape and region information but are strongly parameter and initialization dependent [13]. On the other hand, deep learning models take large datasets to automatically learn features from the data. Deep learning segmentation typically involve two steps: in step one the centerlines of the artery are located and in step two the inner wall (lumen) and outer wall of the artery are predicted [5]. Many deep learning algorithms, among which fully connected CNNs, yield voxel masks in the second step. These voxel masks form a binary classification for each pixel or voxel in the image, and therefore do not guarantee preservation the topology of the artery [12]. The resulting segmentation is likely to show gaps and patches outside the region of interest.

Recently, Alblas et al. [2] developed a method using two nested circular shapes without intersection as anatomical priors, combining the model-based and data-based approaches to improve the voxel-mask approach. This anatomy-aided method has induced the aspiration to quantify the uncertainty per angle. If the input image data is of questionable quality, the CNN

still confidently produces a segmentation. Obtaining estimates for uncertainty of the estimations allows for improving the reliability of the vessel wall thickness measurements. Segmentations based on slices of poor quality can for example be excluded from further measurement of the vessel wall thickness.

Thus the aim of this paper is to extend the automatic vessel wall segmentation method developed by Alblas et al. [2] with a measure of uncertainty. To this end, a Bayesian CNN is adopted by adding dropout layers to the CNN used in the second step of the segmentation.

In the next section (section 2) we start with introducing the main concepts and explain the mathematical connection of Bayesian neural networks and dropout more detail. We then provide information on the dataset in section 3. After this, some background is given on the model that is to be improved, and we propose our modelling method in section 4. Subsequently, we will show the results of the implementation of the proposed uncertainty estimations in section 5, and end with a discussion of the results and implications in sections 6, 7 and 8.

# 2 Preliminaries

We provide some preliminary theoretical background on neural networks, dropout, CNNs and Bayesian CNNs. This will give a more thorough understanding of our problem and will later be used to justify the implementation of the proposed adapted model in section 4. We will show that the use of dropout after each layer can be seen as an approximation of Bernoulli variational inference in a Bayesian CNN and introduces stochasticity in the model. We can calculate the mean of $K$ forward passes through the dropout network during test time to obtain the posterior predictive distribution.

## 2.1 Neural networks

Neural networks (NNs) are often represented as a graph, where the nodes are called neurons and the connections between neurons called the weights. Deep neural networks consist of layers of neurons; an input layer, an output layer and $L$ hidden layers in between, as can be seen in Figure 2.

A neural network can mathematically be defined as a function $f_w : \mathcal{X} \longrightarrow \mathcal{Y}$, where $\mathcal{X}$ is the input space, $\mathcal{Y}$ the output space and the parameters $w^1 \in \mathbb{R}^n$ the weights of the network. The output of the $l^{\text{th}}$ layer of the network is denoted by $y^l$. Each layer processes the output of its preceding layer by a linear transform, followed by a nonlinear activation:

$$y^{l+1} = \sigma(w^{l+1}y^l + b^{l+1}) \quad l \in \{1, ..., L\},$$

where $w_l$ and $b_l$ are called the weights and biases of layer $l$ and $\sigma$ is a nonlinear activation function, e.g., ReLU or Tanh.

The weights and biases of the network are initialized randomly. The network learns parameters that can be used to optimally carry out a task by so-called *training*. The desired output of the network given an input is given by ground truth data. During training, the discrepancy between the output of the network and the desired output is measured by a *loss function*. The loss function is then minimized using the gradient descent method, proceeding backward

[1]The parameters are also classically denoted by $\theta$, however we decided to stick to the notation by [18] in this paper.

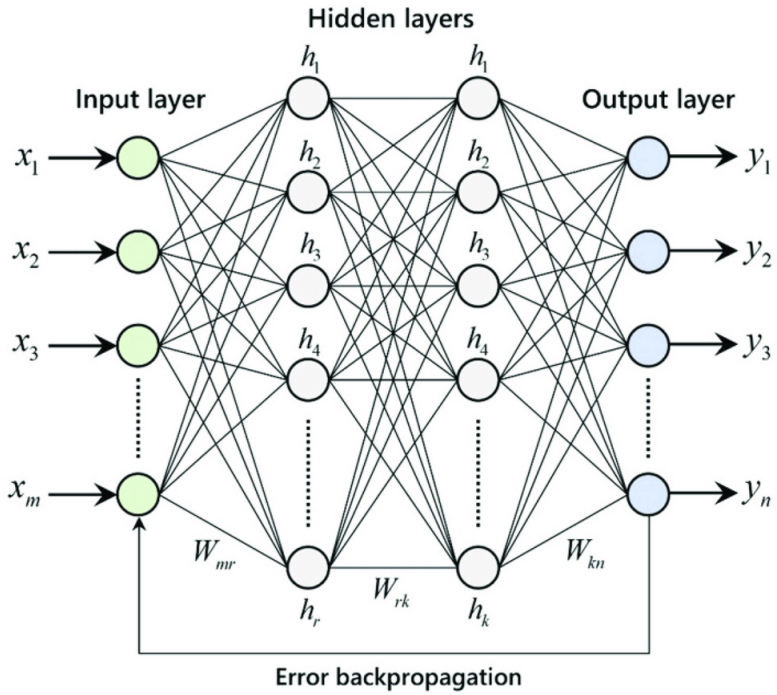



Figure 2: An example of a deep neural network. The function takes m inputs and passes it through a number of hidden layers which perform a linear transformation and a non-linear activation to produce n outputs. Training of the parameters is done by propagating backwards through the network, minimizing the loss function using the gradient descend method. Image adapted from [6].

through the network and adjusting the values of the parameters in each step in the direction of the gradient. This process is called *backpropagation* and the size of the gradient descent step is called the learning rate. After training, the parameters are fixed and the model can hopefully generalize its performance to data it has not seen during training.

## 2.2 Dropout

Dropout is a regularization technique normally used to prevent overfitting. When a deep learning model is trained on a dataset with little variability, the parameters are prone to adjust to the specifics of that dataset. This improves performance on the train data yet deteriorates performance on unseen test data, a phenomenon we refer to as *overfitting*. Dropout decreases neuron dependence by eliminating neurons with a certain probability during training time, introducing stochasticity in the model. This limits the possibility of overfitting and causes the model to have similar performance on train and validation data [7, 18].

When implementing dropout in the network, the value of a neuron is set to 0 with probability $p^l$. This equates to sampling a vector of variables from a Bernoulli distribution for each layer and performing element-wise multiplication with the output of each layer:

$$r_j^l \sim Bernoulli(1 - p^l)$$
$$y^{l+1} = \sigma(w^{l+1}(y^l * r^{l+1}) + b^{l+1}) \quad l \in \{1, ..., L\}.$$

An example of a deep neural network including dropout can be found in Figure 3. Literature suggests a value of 0.5 for the value of each $p^l$ [8, 18].

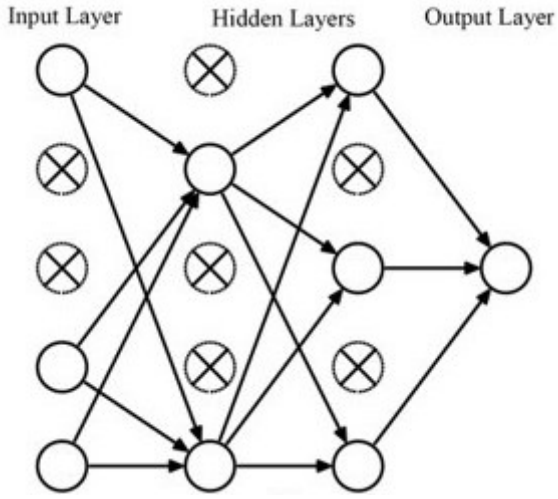



Figure 3: An example of dropout in a neural network. Neurons are eliminated in each layer with a fixed probability per layer. Dropout reduces overfitting on the training data and introduces stochaticity in the model. Image adapted from [19].

### 2.3 Convolutional neural networks

Convolutional neural networks (CNNs) are developed for structures with strong spatial dependencies, such as images. A CNN is a neural network that takes an image as input, and processes local image information using convolutional operations. Convolutional operations pass a symmetrical kernel over each position of the image, performing a dot product operation for all overlapping voxels. An example of a convolutional operation can be seen in Figure 4. The kernels middle value will be replaced with the calculated value in the image. Similar to regular neural networks, the values of the kernel are trained by the CNN. Placing several convolution layers subsequently allows for a growing receptive field and enables the network to extract distinctive features from the image, that can be used for classification, segmentation or regression. As we will see later, CNNs are used in both steps of the segmentation model by Alblas et al.[2].

### 2.4 Bayesian neural networks

We introduce Bayesian neural networks as a way of including stochasticity in the model. In Bayesian neural networks, the weights are modeled by a probability distribution instead of a single value. The network then attempts to learn this distribution from the distributions of the input and output data of the training network. This allows us to obtain the predictive distribution of the output, where the predictive variance can be used as an uncertainty measure.

But before we formally define a Bayesian neural network, we distinguish two types of predictive uncertainty used in Bayesian modelling [10]:

- **Aleatoric uncertainty**: This is the uncertainty that can be attributed to noise in the data. This is the uncertainty that we aim to quantify and cannot be reduced. It can be either independent of the data (homoscedastic) or, in our case, dependent on the data (heteroscedastic).
- **Epistemic uncertainty**: This is the uncertainty that represents the limitations of the model. This uncertainty can be reduced by controlling hyperparameters and observing

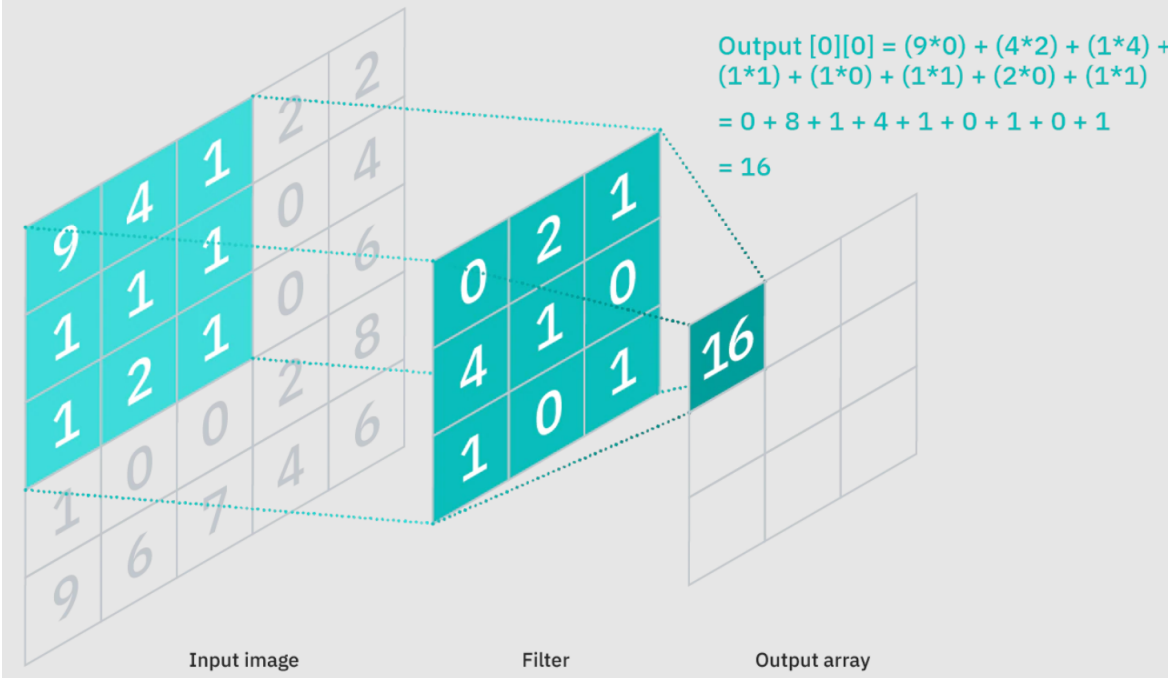



Figure 4: An example of a convolutional operation. A dot product operation between the kernel and the input image is performed to obtain the final value in the output array. In a CNN, the kernel values will be the weights of the network. Image adapted from [9].

more data.

In Bayesian modeling, the total predictive uncertainty is given by the sum of the aleatoric and epistemic uncertainty.

Mathematically, a Bayesian neural network is defined as follows: Given a neural network with weights $w$, we place a prior distribution over the weights, denoted by $p(w)$. From a dataset consisting of input $x_i$ and corresponding labels $y_i$, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$, we want to find the predictive posterior distribution $p(Y|x, \mathcal{D})$, which is the distribution of the output given the training data and a new, unseen, set of inputs $x$. We obtain the predictive posterior distribution by conditioning on the weights:

$$p(Y|x, \mathcal{D}) = \int p(Y|x, w)p(w|D)dw. \tag{1}$$

The predictive posterior gives us the distribution of the most likely output for $x$ given the data. We can use Bayes rule to calculate the posterior distribution $p(w|D)$, which is the probability of the weights given the training data:

$$p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{p(\mathcal{D})},$$

where $p(\mathcal{D}|w)$ is the probability that the training data is generated from certain weights, also called the likelihood. The model evidence $p(D)$ represents the distribution of the input and output data [18].

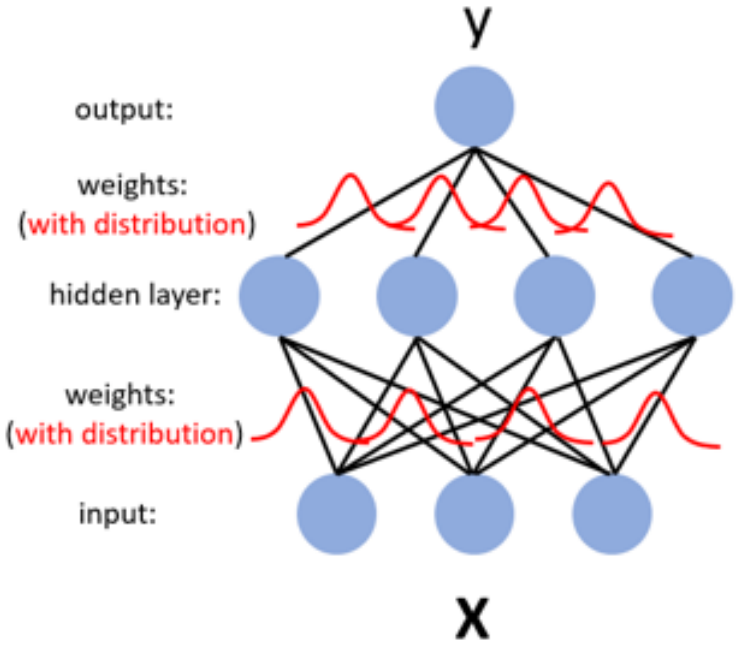



Figure 5: An example of a Bayesian neural network. Instead of single values, the network trains weight distributions to obtain the output distribution. Image adapted from [11].

.

### 2.5 Dropout as a Bayesian approximation

Bayesian neural networks are exceedingly useful for uncertainty quantification. However, since the network has to train a distribution instead of a single value, the complexity of the network increases, making the network notoriously hard to implement in practice. We will approximate the Bayesian neural network by applying dropout after each layer of the CNN. Then the predictive distribution is obtained as the average of $K$ forward passes through the network during test time.

We approximate the posterior distibution $p(w|D)$ by a variational distribution $q(w)$ and minimize the Kullback-Leibler divergence $KL(q(w)||p(w|D))$, a measure for the similarity between the distributions [8]. This is a well-known technique in statistics called variational inference. In order to show that applying dropout after each layer is the same as approximating a Bayesian NN, we define our $q(w)$ with weights $w = (W_l)_{l=1}^{L}$ for each layer $l$ as follows:

$$\begin{aligned} W_l &= \tilde{W}_l \cdot \text{diag}([z_{l,j}]_{j=1}^{K_i}) \\ z_{l,j} &\sim Bernoulli(1 - p_l), \end{aligned} \tag{2}$$

where $W_l$ is a matrix of dimension $K_i \times K_{i-1}$ for each layer $l$. Here, the diag() operator maps the elements $z_{l,j}$ to the diagonals of a matrix of zeros. $\tilde{W}_l$ are the variational parameters to optimize, corresponding to the weights in a network without dropout. Note that defining $q(w)$ (the approximation of our posterior distribution) this way equates to applying dropout to each layer as described in section 2.2. The $j^{\text{th}}$ neuron of layer $l$ is eliminated with probability $p_l$.

Using this definition of the variational distribution, it can be shown that also the minimization objectives of both networks are the same [7]. Then, we can evaluate the integral in equation

1 using a Monte-Carlo approximation [10]:

$$p(Y|x,\mathcal{D}) = \int p(Y|x,w)p(w|D)dw \approx \int p(Y|x,w)q(w) \approx \frac{1}{K}\sum_{j=1}^{K} p(Y|x,w_j),$$

where $w_j$ are sampled from the variational distribution $q(w)$ as described in equation 2. It is shown by Gal and Ghahramani [7] (appendix C, proposition 3) that the expectation of the predictive distribution can then be approximated as the average over $K$ model outputs $y^*$:

$$\mathbf{E}(p(Y|x,\mathcal{D})) \approx \frac{1}{K}\sum_{j=1}^{K} y^*(x, z_{1,j}, ..., z_{L,j}).$$

This expression can be evaluated easily by performing $K$ forward passes through the network at test time, with dropout after each layer. For the remainder of this paper, we refer to this method as Monte-Carlo dropout (MC dropout).

# 3 Data description

The dataset as provided by the Carotid Artery Vessel Wall Segmentation Challenge [1] consists of 26 black-blood magnetic resonance images (BB-MRI) of patients with various degrees of atherosclerosis, obtained from the larger CARE II dataset with data from 13 different hospitals in China. The data are obtained from a 3D Motion Sensitized Driven Equilibrium prepared Rapid Gradient Echo (3D-MERGE) sequence, optimized for visualizing atherosclerosis lesions in carotid arteries rapidly with high spatial resolution [3]. The images contain the left and right common carotid arteries, splitting into the internal and external carotid arteries at the bifurcation.

In medical images, there is often notion of the *axial*, *sagittal* and *coronal* plane. These are anatomical terms, describing the orientation of 2D cross-sections in relation to the patient. The Z-axis is considered to pass through the patient from the feet to the head. In this dataset, the $X, Y$ plane is called the axial plane, taking a cross-section orthogonal to the $Z$-axis. Similarly, the $X, Z$ plane is called the coronal plane and the $Y, Z$ plane is called the sagittal plane. Examples of these planes can be found in Figure 6. For the segmentation method described in this paper, we use axial planes as image input slices to identify the lumen and outer wall. The images have axial dimensions 720 or 640. The isotropic voxel spacing varies between 0.27 $mm^3$ and 0.39 $mm^3$ [2]. Some axial slices were provided with closed contour annotations, 2533 of the internal carotid artery and 122 of the external carotid artery. These annotations are used to train the model. They were extended by Alblas et al.[2] with lumen centerlines to be used in the first step of the model.

An important observation is the difference in quality of the slices. Usually at the beginning and ending of the z-axis, the image quality tends to be low due to practical measurement issues. Because of this, the carotid artery cannot be distinguished everywhere in the image volume, resulting in axial slices without ground-truth annotations. This is resolved by only training the network on the annotated slices. During inference time image quality is unknown, so predictions are made for all images. This leads to our main motivation for the development of uncertainty quantification: the necessity to automatically filter segmentations according to image quality.

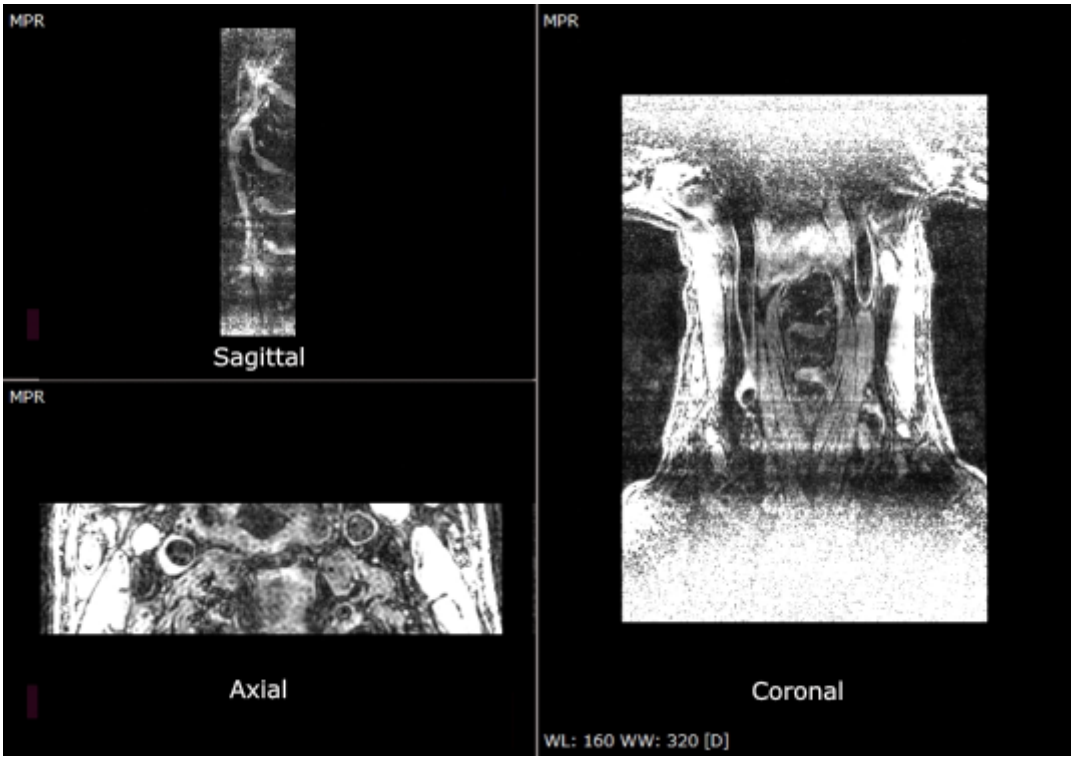



Figure 6: Orientation of 2D patient cross-sections. In our segmentation model, axial slices are used. The axial slice in this image shows plaque buildup in the left artery, due to atherosclerosis. In the upper and lower part of the coronal slice the poor data quality is clearly visible as white noise.

# 4 Method

In this section we describe the method by Alblas et al. [2] in more detail. Then, our approach for quantifying the uncertainty will be explained, based on the preliminary results from section 2.

As previously mentioned, Alblas et al. [2] uses the two-step method for artery vessel wall segmentation. In the first step, the centerlines of the internal and external carotid arteries are located. A cost function that represents the proximity of each voxel to the centerlines is defined. The cost function for Dijkstra's algorithm is learned using a U-Net, which is a convolutional neural network architecture typically used for biomedical image segmentation [16]. The result is a proximity map for the 3D image volume. From the maxima in the proximity maps every 50 slices, Dijkstra's algorithm then finds the shortest path through the cost image, resulting in a continuous centerline.

Secondly, using the estimated centerlines the Cartesian images are transformed into local polar image patches by performing ray-casting at equidistant angles. These polar images serve as input for a dilated CNN, that predicts the radius of the lumen and the vessel wall thickness for each predetermined angle. Converting back to Cartesian coordinates, this results in a segmentation mask of two non-intersecting nested contours for each axial slice. Since axial slices are typically correlated, two distinct methods are compared. In the single-slice method, 2D polar images are acquired without using additional context information in the axial direction. In the multi-slice method, polar images from both sides are stacked to obtain a 3D polar image. This last approach achieved the top result in a public challenge [1], yielding accurate and anatomi-

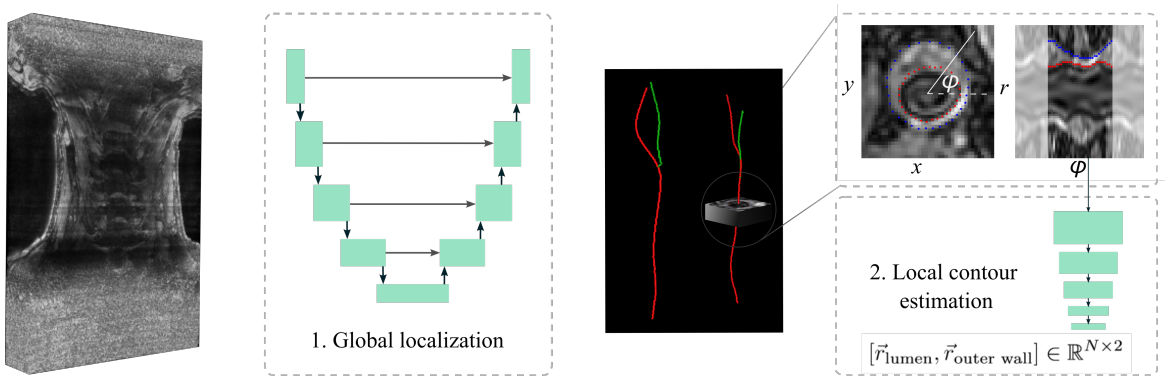



Figure 7: The carotid artery vessel wall segmentation method used by Alblas et al. [2]. In the first step, the centerlines of the internal and external carotid arteries are predicted globally using Dijkstra's algorithm, predicting the cost function with a U-net. In the second step, the centerlines are used to predict the location of the lumen and outer wall locally, exploiting the circular shape of a cross-section of the artery using polar coordinate systems. Image adapted from [2].

cally plausible results. See figure 7 for a graphic representation of the used method. Note that, unlike with voxel masks, the shape and continuity of the vessel wall segmentation is guaranteed.

In addition to the method above, we would like to include uncertainty estimation. Because although the method is accurate for annotated slices, non-annotatable slices should still be excluded from the final result. Since high noise slice predictions are of poor quality due to high aleatoric uncertainty, we are motivated to investigate uncertainty quantification methods. To this end, as explained in the preliminaries, we would like to introduce Bayesian modelling to obtain the posterior predictive distribution. Implementing this in our network, allows us to not only calculate the predictive expectation, but also calculate the predictive variance as a measure of the uncertainty (aleatoric and epistemic) of the model. Gal and Ghahramani [8] have shown the predictive variance to be equal to the sample variance of $K$ stochastic forward passes through the neural network plus the inverse model precision $\tau^{-1} = \frac{2N\lambda}{pl^2}$, which is zero for our model, since our CNN has a weight decay $\lambda$ of zero.

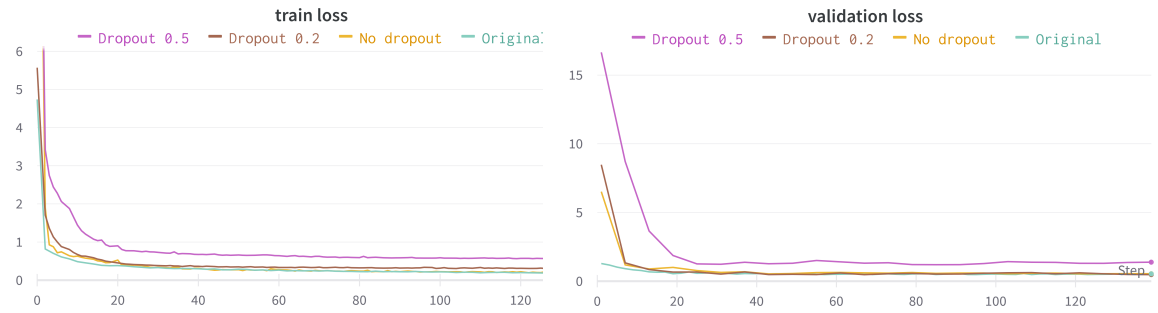
We approximate a Bayesian neural network by implementing dropout after each layer of network. Due to practical considerations, for our implementation we choose the single-layer CNN in the second step. Although dropout is usually only employed during training, we will implement dropout both during training and testing in order to obtain the expectation and variance of the posterior predictive distribution. In this way implementing dropout correctly will give us a measure for the uncertainty in the network and also improve the robustness of the model to new data, as mentioned in section 2.2.

# 5 Experiments and results

## 5.1 Experimental setting

The model as described in section 4 is trained on a dataset of 21 patients from the training set of the Carotid Artery Vessel Wall Segmentation Challenge [1], 5 patients from this dataset are used for validation.

Dropout is implemented after each layer of the CNN in the second step of the model as




(a) Train losses

(b) Validation losses

Figure 8: Loss curves as a function of the number of epochs. Loss for the architecture with a dropout value of 0.2 nearly coincides with the model without dropout, implying similar performance.

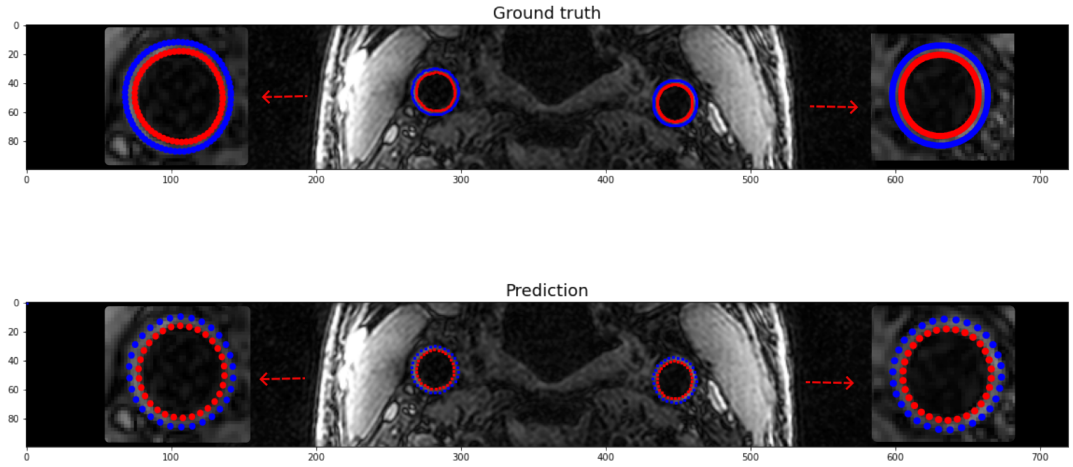
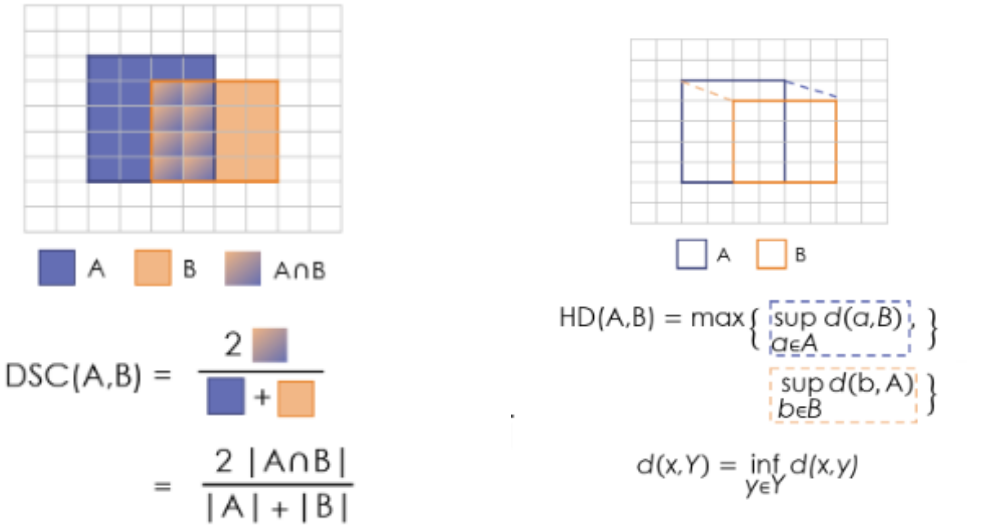



Figure 9: Qualitative comparison of the ground truth to the prediction of the internal carotid artery of a testing patient for the model with dropout probability 0.2. The segmentation is to be almost indistinguishable from the ground truth, suggesting that the segmentation is of high quality.

outlined in section 4, when the location of the lumen and outer wall is predicted. To validate the implementation, the training of the original single-layer CNN is compared to the performance of the new model with a dropout value of zero. During testing time, we perform 20 stochastic forward passes and calculate the mean and the variance for the location of the lumen and the thickness of the wall for each predicted contour point on each axial slice.

## 5.2 Model training and testing

Four different models were trained and compared to each other: a model with dropout value 0.5, a model with dropout value 0.2, a model without dropout and the original single-slice model with augmentation as described by Alblas et al. [2]. These models were trained for 120 epochs with a learning rate of 0.01 and a batch size of 100, using an Adam optimizer and a smooth L1 loss function. The train loss was evaluated after every epoch and the validation loss every 5 epochs, the results for are shown in Figure 8a and Figure 8b.



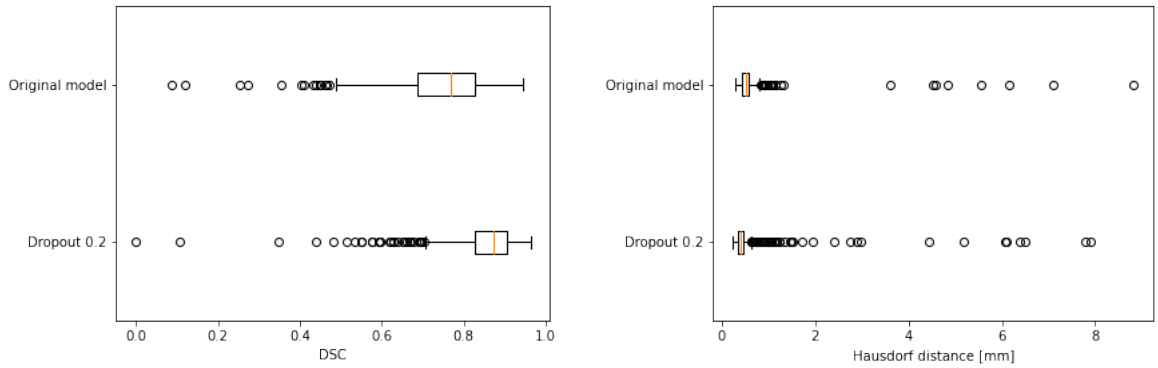

(a) Dice Similarity Coefficient (DSC)

(b) Hausdorf Distance (HD)

Figure 10: A visual explanation of the performance metrics used in assessing the quality of the results. Image adapted from [15].

The loss curves of the original model and the model without dropout are nearly indistinguishable, validating the implementation of the model. Furthermore, it was found empirically that using a dropout probability $p_l = 0.5$ resulted in slow convergence and poor-quality results, even though this value is generally recommended in the literature. In all likelihood it takes longer to train the model because a dropout rate of 0.5 deactivates half the units of the hidden layers. Dependence on nodes is decreased, resulting in more variance in the results and a necessity to perform more forward passes to stabilize. We have chosen to use a dropout rate of 0.2 to avoid these drawbacks. This second model not only showed convergence for both the train loss as well as the validation loss almost as fast as the original model, but also resulted in high-quality segmentations as can be seen in Figure 9. Once chosen, we trained the model for a total of 200 epochs to increase accuracy.

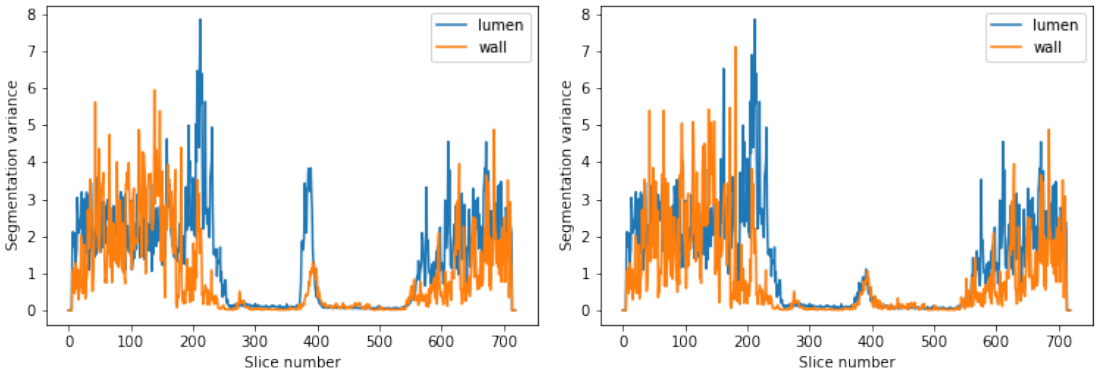
The results were further assessed quantitatively by calculating two performance metrics for the validation set: the Dice Similarity Coefficient (DSC) and the Hausdorff Distances (HD). The performance metrics were calcultated for the validation set consisting of 5 patients. The DSC gives a measure of the overlap of the ground truth and the segmentation. As can be seen in figure 10a, the DSC calculates the area of the intersection, and compares it to the area of both shapes. If the shapes overlap completely, the DSC is one. If they are disjunct, the DSC is zero. In the other hand, as seen in figure 10b, the HD measures the smallest distance between the shapes and takes the maximum of both distances. If the shapes overlap completely, this value is 0. Since the metrics measure the difference between two shapes differently, using both metrics should give a fair view of the quality of the segmentation [15]. The results can be found in Figure 11b. The median of the DSC of the model including dropout is 0.871 and the median of the DSC of the original single-layer model is 0.769. For the HD, the medians of the model with and without dropout are 0.411 and 0.522, respectively. The new model seems to have improved performance for the validation set, likely because the model is less prone to overfitting.




(a) Box plot comparison of the DSC

(b) Box plot comparison of the HD

Figure 11: Dice Similarity Coefficient (DSC) and Hausdorff Distances (HD) for the lumen of the model without dropout and the model with dropout rate 0.2, calculated for all testing patients. The new model has improved for both performance metrics.




(a) Without bifurcation compensation

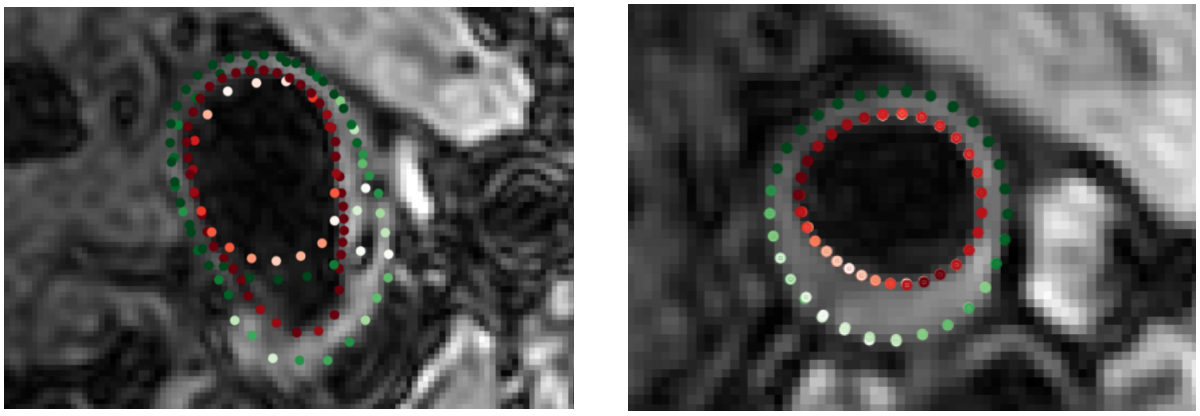
(b) With bifurcation compensation

Figure 12: Average variances for the lumen and wall thickness of a patient from the test dataset. The variance peaks at the first 200-250 slices, at the final 150-200 slices and in the middle around the bifurcation. In the second image, intersecting predictions are removed from the variance.

### 5.3 Model uncertainty

The chosen model (with $p_l = 0.2$) was tested on the set of validation patients, performing $K = 20$ stochastic forward passes as suggested in [7], calculating the predictive mean and predictive variance for each angle of each segmentation as described before. Figure 12a shows the average variance over all angles for the predictions of the lumen and the wall thickness for the left internal carotid artery of a patient from the test dataset.

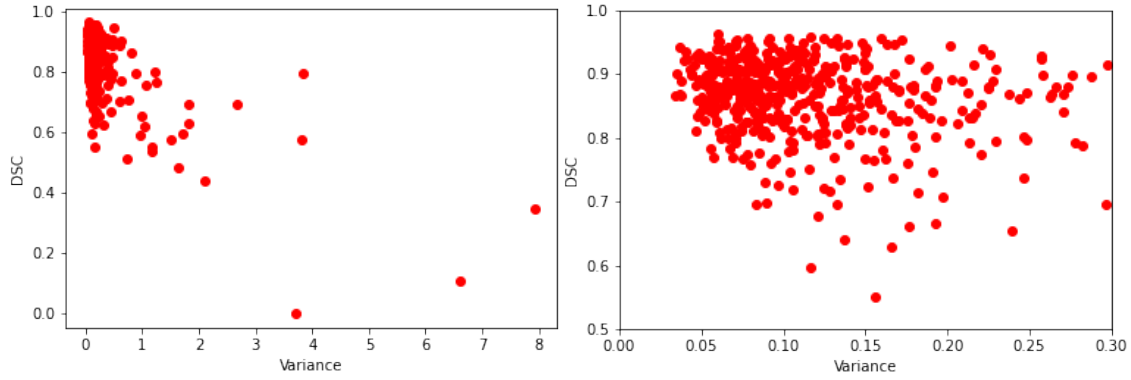
As was observed for all testing patients, the variance peaks at the first 200-250 slices and at the final 150-200 slices. This was to be expected, as the aleatoric uncertainty should be high for poor quality data. However, unanticipated was the peak in the middle of the figure, corresponding to the location of the bifurcation. This has to do with the merging process around the bifurcation, a low peak around the bifurcation remains. Since a part of the polar rays are directed into the lumen, the lumen radius may grow too large, causing it to fall outside the polar image and contributing to high uncertainty. As can be seen in Figure 13a, the variance is



(a) At the bifurcation

(b) In a slice with plaque buildup

Figure 13: Predictive mean and variance of the lumen (red) and wall (green), where brighter dots correspond to higher variance.




(a) Entire plot

(b) The low variance cluster

Figure 14: Variance of all annotated slices in the validation set plotted againts the DSC. The variance seems to be mostly independent from the DSC. High variance outliers correspond to bifurcation slices.

especially high on the broadest side of the bifurcation wall. After all, the model performs best on more circular shapes, because the training set contains relatively less images with non-circular annotations, so performance of these shapes is less optimized. Since this is a limitation of the model, it is explained by the epistemic uncertainty. This also explains why variance is higher at slices with a lot of plaque build-up as can be seen in Figure 13b.

To assess whether high uncertainty corresponds to poor segmentation results, we also looked into the correlation between the Dice Similarity Coefficient and the variance, as can be seen in Figure 14b. We should mention that the dice score can only be calculated for annotated slices, so poor quality slices are not included in this plot. The DSC seems to be high, even for slices with a lot of uncertainty such as around the bifurcation. Note that no correction was performed for the merging process around the bifurcation. But even when excluding high variance results, the correlation between the variables is low.

# 6 Discussion

Our initial goal was to improve the segmentation by excluding contours predicted on poor-quality slices. Although we have successfully quantified the predictive uncertainty of the model, we had not anticipated the high variance at the bifurcation. As of now it is not possible to filter segmentation results on the basis of the prediction variance, without either rejecting the bifurcation region or including poor quality slices. Stated differently, we have not been able to separate the aleatoric and epistemic uncertainty. Furthermore, it was found that the model has a higher uncertainty on slices with a lot of plaque build-up. This is concerning, as this is the part of the carotid artery that we would like to predict with high accuracy. Also, dropout was only implemented in the second step of the model, since we predicted the uncertainty to mainly take place there. Moreover, it was only implemented on the single-layer version of the CNN, although the multi-layer version has been shown to produce better results. It was also found that the DSC and the variance are not correlated, even when excluding high variance results, which was unanticipated. Finally, we should note that the dropout hyperparameter of 0.2 might not be optimal.

# 7 Conclusion

In this paper we improved the carotid artery segmentation model developed by Alblas et al. [2]. This was done by implementing dropout after each layer of the convolutional neural network and performing forward passes through the model to calculate the predictive variance as a measure of uncertainty. It was found experimentally that a dropout parameter value of 0.2 instead of 0.5 resulted in faster convergence of the train and validation loss. Adjusting this value resulted in a model with a higher median Dice Similarity Coefficient and Hausdorff Distance on the validation set than the original model. This suggests substantial improvement in model performance. Moreover, we obtained the variance for each predicted point as a measure of the uncertainty and observed peaks at the first 200-250 slices and the last 150-200 slices.

However, no variance-based separation could be performed yet. This is due to the unexpected high variance in the bifurcation region, where the internal and external carotid artery split. The variance peak around the bifurcation could partly be explained by the merging process used in the model, and partly by the epistemic uncertainty. For the same reason, we also noted high variance for angles where plaque build-up can be observed.

Summarizing, we successfully implemented dropout and performed an in-depth analysis of the uncertainty. Uncertainty quantification has proven to be a promising direction in the advancement of this segmentation model and can help to significantly improve the medical care of patients with atherosclerosis.

# 8 Outlook

For future research, it would be interesting to investigate the separation of the aleatoric and epistemic uncertainties, as for example described by Kendall and Gal [10]. If necessary, a quicker estimate could perhaps be made by smoothing the variance function, or by rejecting pointwise for each angle and use spline interpolation to find the final contour.

Implementing dropout in the first step of the model could also result in additional interesting results with respect to the quality of the model and the variance of the predictive distribution.

Implementing dropout in the multi-layer version of the model could further improve the accuracy of the segmentations. Furthermore, the model could perhaps be improved if the dropout parameter is somehow optimized or regularized, as researched by Theobald et al. [18].

The developed model could serve as a starting point for future research on improving deep learning carotid artery segmentation and quantifying the model uncertainty. Often, segmentation is a necessary first step for building a 3D mesh of the artery. This model can then be analyzed using computational fluid dynamics (CFD), yielding important biomarkers for cardiovascular disease progression. For example for atherosclerosis, the wall shear stress (WSS) has been found to correlate with plaque development. Using CFD to model arteries, biomarkers can be estimated in an accurate and non-invasive manner [17].

We recommend to take the uncertainty in slices with high plaque build-up into account when calculating the values of the biomarkers, such that decisions can be made on the basis of the resulting confidence interval. It should then be investigated whether the predictive distribution follows a normal distribution. Perhaps the uncertainty in these slices can also be used to our advantage, identifying slices with plaque build-up by looking at the variance distribution of that slice.

# References


[1] Carotid artery vessel wall segmentation challenge. `https://vessel-wall-segmentation.grand-challenge.org/`.

[2] Dieuwertje Alblas, Christoph Brune, and Jelmer M. Wolterink. Deep-learning-based carotid artery vessel wall segmentation in black-blood MRI using anatomical priors. In *Medical Imaging 2022: Image Processing*, volume 12032, pages 237 – 244. SPIE, 2022.

[3] Niranjan Balu, Vasily L. Yarnykh, Baocheng Chu, Jinnan Wang, Thomas Hatsukami, and Chun Yuan. Carotid plaque assessment using fast 3d isotropic resolution black-blood mri. *Magnetic resonance in medicine*, 65:627–637, 2011.

[4] Thomas Binder. carotid ultrasound - anatomy, Feb 2021.

[5] Li Chen, Jie Sun, Gador Canton, Niranjan Balu, Daniel S. Hippe, Xihai Zhao, Rui Li, Thomas S. Hatsukami, Jenq Neng Hwang, and Chun Yuan. Automated artery localization and vessel wall segmentation using tracklet refinement and polar conversion. *IEEE access : practical innovations, open solutions*, 8:217603–217614, 2020.

[6] Mohammad Hemmat Esfe, S. Ali Eftekhari, Maboud Hekmatifar, and Davood Toghraie. A well-trained artificial neural network for predicting the rheological behavior of mwcnt–al2o3 (30–70 *Scientific Reports 2021 11:1*, 11:1–11, 8 2021.

[7] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. 6 2015.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *33rd International Conference on Machine Learning, ICML 2016*, 3:1651–1660, 2016.

[9] Guymonahan. Basic overviews on convolutional neural networks, Jul 2021.

[10] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 2017-December:5575–5585, 3 2017.

[11] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. A batched scalable multi-objective bayesian optimization algorithm, 11 2018.

[12] G. Litjens, Francesco Ciompi, Jelmer M. Wolterink, B. D. de Vos, Tim Leiner, J. Teuwen, and I. Išgum. State-of-the-Art Deep Learning in Cardiovascular Image Analysis, 8 2019.

[13] Lu Liu, Jelmer M. Wolterink, Christoph Brune, and Raymond N.J. Veldhuis. Anatomy-aided deep learning for medical image segmentation: a review. *Physics in Medicine Biology*, 66:11TR01, 5 2021.

[14] Christopher J.L. Murray and Alan D. Lopez. Measuring the global burden of disease. *New England Journal of Medicine*, 369:448–457, 8 2013.

[15] Annika Reinke, Minu D. Tizabi, Carole H. Sudre, Matthias Eisenmann, Tim Rädsch, Michael Baumgartner, Laura Acion, Michela Antonelli, Tal Arbel, Spyridon Bakas, Peter Bankhead, Arriel Benis, M. Jorge Cardoso, Veronika Cheplygina, Beth Cimini, Gary S. Collins, Keyvan Farahani, Ben Glocker, Patrick Godau, Fred Hamprecht, Daniel A. Hashimoto, Doreen Heckmann-Nötzel, Michael M. Hoffmann, Merel Huisman, Fabian Isensee, Pierre Jannin, Charles E. Kahn, Alexandros Karargyris, Alan Karthikesalingam, Bernhard Kainz, Emre Kavur, Hannes Kenngott, Jens Kleesiek, Thijs Kooi, Michal Kozubek, Anna Kreshuk, Tahsin Kurc, Bennett A. Landman, Geert Litjens, Amin Madani, Klaus Maier-Hein, Anne L. Martel, Peter Mattson, Erik Meijering, Bjoern Menze, David Moher, Karel G. M. Moons, Henning Müller, Felix Nickel, Jens Petersen, Gorkem Polat, Nasir Rajpoot, Mauricio Reyes, Nicola Rieke, Michael Riegler, Hassan Rivaz, Julio Saez-Rodriguez, Clarisa Sanchez Gutierrez, Julien Schroeter, Anindo Saha, Shravya Shetty, Bram Stieltjes, Ronald M. Summers, Abdel A. Taha, Sotirios A. Tsaftaris, Bram van Ginneken, Gaël Varoquaux, Manuel Wiesenfarth, Ziv R. Yaniv, Annette Kopp-Schneider, Paul Jäger, and Lena Maier-Hein. Common limitations of image processing metrics: A picture story. 4 2021.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9351, pages 234–241. Springer Verlag, 2015.

[17] Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, and Jelmer M Wolterink. Mesh convolutional neural networks for wall shear stress estimation in 3d artery models. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 93–102. Springer, 2021.

[18] Claire Theobald, Frédéric Pennerath, Brieuc Conan-Guez, Miguel Couceiro, and Amedeo Napoli. A bayesian neural network based on dropout regulation. 2 2021.

[19] Zong Sheng Wang, Jung Lee, Chang Geun Song, and Sun Jeong Kim. Efficient chaotic imperialist competitive algorithm with dropout strategy for global optimization. *Symmetry*, 12:1–16, 4 2020.