Trude Jostad

Coronary Artery Segmentation using Deep Learning

Master's thesis in Computer Science
Supervisor: Frank Lindseth
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Trude Jostad

Coronary Artery Segmentation using Deep Learning

Master’s thesis in Computer Science
Supervisor: Frank Lindseth
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Measuring the fractional flow reserve (FFR) is one method used for diagnosing coronary artery disease (CAD), which involves a small invasive surgical procedure. By using both CT scans to extracting the geometric structure of the coronary artery and fluid dynamics, the measuring of FFR can be done automatically. This, in turn, will reduce the total cost, resources, and health risks. This thesis is focusing on extracting the geometrical structure of the coronary artery using deep learning models.

Models used in this thesis are BVNet, BVNet3D, and SegCaps. The BVNet model is using more channels to get a 3D effect, and it has less computation cost because all the mathematical operations are in 2D. This is exploried in experiment 1, which consists of finding the optimal channel size and stride for getting the best BVNet model. The dice score is increased with some exceptions by increasing the channel size and by using stride 2. The model with 7 channels and stride 2 had the best mean Dice$_{post}$ score and smaller gaps between the true positive.

The general BVNet model with 5 channels and stride 1 are used in the other experiments, which had relatively low computational cost and the best mean Dice$_{raw}$ score.

The BVNet3D model is the BVNet model with all the 2D operations replaced with 3D, which increases the computational cost of this model.

The SegCaps model came in 2018 and uses CapsuleNet for segmentation. CapsuleNet is using vectors instead of scalers and routing strategies for avoiding the usage of maxPooling. The scalars is used by the conventional networks models, such as BVNet and BVNet3D.

Next, the models were tested on different data splits with the labels for the coronary artery, Aorta, and portal vein in order to find the benefits and disadvantages of these binary classification models. The Aorta has a large tubular structure, which should be easy to find. The portal vein is used in order to check if these models could be used on other vessels. The BVNet model was the best at predicting the Aorta with dice score in most cases around 90%, where the other models contained a lot of false positive. BVNet3D model predictions have less false negative and more false positive for the coronary artery and portal vein.

The SegCaps model was the worst but showed huge improved potential.

The last experiment consisted of exploring the Frangi filter, which is used to enhance the vessel structure.

Past one of the experiment consisted of using The Frangi filtered CT slices as input for the BVNet model. There was a huge difference between the dice score of the test images, which indicates that the hyperparameters used to create the Frangi filtered CT images was not optimal for all the CT images.

Part two of the experiment consisted of using a combination of Frangi filtered CT and regular CT images as input. The result did not increase the dice score, but the variation between the best and worst prediction was reduced and had fewer gaps between the true positive.

These models seems promising, but none of the models tested in this thesis is good enough to measure the FFR without further development.

Sammendrag

En metode for diagnostisering av hjerte-karsykdommer er å måle FFR (fractional flow reserve), som innebærer et lite kirurgisk inngrep. Ved å ekstrahere den geometriske strukturen av hjertets kransarterier fra CT bilder sammen med fluiddynamikk kan FFR bli målt automatisk. En automatisk måling av FFR vil kunne spare ressurser, kostnader og redusere helseriskoer for pasienten. Denne masteroppgaven omhandler hvordan den geometriske strukturen til kransarteriene kan ekstraheres automatisk gjennom CT bilder ved bruk av dyplæring modeller.

Modellene brukt i denne oppgaven er BVNet, BVNet3D og SegCaps. BVNet modellen bruker 2D skiver av CT bildene, men benytter seg av flere kanaler for å få en 3D effekt og den har lavere minnebruk fordi alle de matematiske operasjonene er 2D. Dette utforskes i ekperiment 1 som besto av å finne det optimale antall kanaler og steglengde for å forbedre BVNet modellen. I dette eksprimentet økte dice verdien med noen untak ved å øke kannal størrelsen og steglengde 2. Den modelen med best gjennomsnittlig Dice$_{post}$ verdi var med 7 kanaler og stride 2, men den inneholdt noen sprik mellom de sanne positive.

Den generelle BVNet modellen har 5 kanaler og steglengde 1 og er brukt i de andre eksprimentene, fordi den har lavere minnebruk og har den beste gjennomsnittlig Dice$_{raw}$ verdien.

BVNet3D modellen har samme arktitektur som BVNet modellen bortsett fra at alle de 2D matematiske operasjonene er byttet ut med 3D, som gjør at den får større minnebruk. SegCaps modellen bruker CapsuleNet til segmentering, og ble først brukt til dette i 2018.

CapsuleNet bruker vektorer framfor skalarer og egne strategier for å bestemme hvilke av capsulene i det neste laget som skal aktiveres, slik at de kan unngå å bruke maxPooling. Skalarer blir brukt av convolutional nettverk modeller som BVNet og BVNet3D.

Videre er modellene testet på forskjellige data spliter av kransarteriene, Aorta og portalvenen for å finne fordeler og ulemper ved de ulike modellene for binær klassifikasjon. Aorta er valgt på bakgrunn av at den har en stor tubulær struktur som betyr at den i utgangspunktet skal være lett å finne. Portalvenen er valgt for å undersøke om modellene kan bli brukt på årer generelt. BVNet modellen var best til å prediktere Aortaen best dice verdien i den fleste tilfeller rundt 90 %, hvor de andre modelene hadde mange falske positive prediksjoner. BVNet3D modelen hadde færre falske negative and flere falske positive prediksjoner for segmenteringen av kransearteriene og portal venene.

SegCaps modelen var dårligst, men viste et stort forbedringspotensial.

Det siste eksperimentet utforskes bruken av Frangi filteret, som blir brukt til finne årestrukturer. Målet med dette eksperimentet er for å finne ut om bruken av Frangi filteret kan gjøre BVNet modellen mer presis i å finne kransarteriene.

I del en av det siste eksperiment blir Frangi filterte benyttet på CT bildene som input på BVNet modellen. Resultatet hadde stor forskjell mellom dice verdiene for de ulike testbildene, som indikerer at de beste hyperparameterne ikke ble brukt til å generere Frangi filteret i fleste tilfeller.

I del to av Frangi filter eksperimentet blir både CT bilder og Frangi filteret CT bilder brukt som input på BVNet modellen. Resultatet økte ikke dice verdien, men forskjellen mellom den beste og værste prediksjonen ble redusert.

Modellene brukt i denne oppgaven ser lovende ut, men ingen av dem er gode nok til å bli brukt til å måle FFR uten videre utviklig.

Preface

This thesis is written as my master thesis for the Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU) that extends the work done in my specialization project [15], where some of the theory and methods are still relevant.

First, I would like to thank my supervisor Frank Lindseth for given good advises and allowed me to work on this exciting field. Next, I would like to thank Fredrik Eikeland Fossan for proving me with information about the FFR project and the dataset from the St.Olav hospital, which has become the basis for the motivation, and for applying the Frangi filter on each CT image with the VMTK.

Table of Contents

List of Tables

List of Figures

Abbreviations

LM	=	left main coronary artery
RCA	=	right coronary artery
CA	=	coronary artery
CAD	=	coronary artery disease
FFR	=	fractional flow reserve
PV	=	portal vein
CT	=	computed tomography
CNN	=	convolutional neural network
SGD	=	stochastic gradient descent
DSC	=	dice similarity coefficient
TP	=	true positive
FP	=	false positive
TN	=	true negative
FN	=	false negative

Chapter 1

Introduction

Computer vision with deep learning has become a popular field, which is used for object recognition. The motivation is to replace many of the manual tasks, like face recognition, that is time-consuming. One of the more complicated tasks in computer vision are image segmentation because it needs to both locate and classify each pixel belonging to a given class.

Image segmentation has recently been used on medical images to find complex geometrical structures like vessels and arteries. The goal in this thesis is to perform automatic segmentation of the coronary artery using deep learning methods that will be described in section 1.2.

The main motivation of this thesis is to extract the geometrical structure of the coronary artery from CT images, for making the procedure of measuring FFR with fluid dynamics automatic. This will be explained further in section 1.1.

This chapter ends with a description of the contribution in section 1.3 followed by an overview of the outline in section 1.4.

1.1 Motivation

Coronary artery disease (CAD) is the leading cause of death globally [21], and its relevance will increase as the global population ages. An effect of the CAD is that the blood supply to the heart muscle is impaired due to stenosis in one or more of the larger coronary arteries. Stenosis is a narrowing or obstruction of the artery.

Fractional Flow Reserve (FFR) is an index to characterize the functional significance of coronary artery stenosis. The index is used for measuring the ratio between the actual blood flow over the hypothetical blood flow, which would be observed in the absence of the stenosis under examination.

In clinical practice, however, a proxy for the ratio is obtained through invasive pressure measurements, upstream and downstream of the stenosis. Even though FFR is considered the gold standard for diagnosing CAD, it remains underused, mainly due to its need for a specialist, and its relatively high cost. As a consequence, many attempts have been made to predict FFR non-invasively through computational fluid dynamics. An important step in such an attempt is the definition of the patient-specific coronary anatomy/geometry, which is commonly performed through segmentation of the coronary artery from Computed Tomography (CT) images. Most existing methods rely on manual or semi-automatic segmentation, which is time-consuming and highly dependent on the observer performing the segmentation. This thesis focuses on automatic methods for coronary artery segmentation based on CT images.

1.2 Project Goals and Research Questions

The goal of this project is to compare different models and processing techniques in order to get better predictions of the coronary artery. The different models used are BVNet, BVNet3D, and SegCaps, which will be explained in chapter 3. These models are also tested on the Aorta and portal vein. The Aorta label is using the same CT input images as the coronary artery. It is used to checks if the models are good to detect large tubular structures, which should be easier than detecting the coronary artery. The portal vein is used for finding out if the models can be used for detecting arteries/vessels in general. For reaching these goals, the following research questions were created:

RQ1: Is it possible to improve the BVNet model's understanding of the predicted slice surroundings?

RQ2: What is the advantages and disadvantages of BVNet, when used for segmenting the coronary artery, Aorta, or portal vein?

RQ3: What is the advantages and disadvantages of BVNet3D model, when used for segmenting the coronary artery, Aorta, or portal vein?

RQ4: What is the advantages and disadvantages of SegCaps model, when used for segmenting the coronary artery, Aorta, or portal vein?

RQ5: Can a Frangi vessel filter help increase the dice score for the BVNet model?

1.3 Contribution

The first contribution of this thesis is finding the optimal number of channels and stride used for the BVNet. Stride is in this context the step size of which of the slices are going to be added as channels.

Next, it compares the BVNet, BVNet3D, and SegCaps model. The SegCaps model has been used for lung segmentation. This thesis explores if SegCaps can be used on more complex structures. The last contribution consists of combining a traditional method with a modern model. This is done by using Frangi filtered CT images as input on the BVNet model, and using both CT images and Frangi filtered CT images as input on the BVNet model.

1.4 Outline

This section gives a short overview of the structure of this thesis.

Introduction

This chapter begins with an explanation of the primary motivation for developing methods for automatic segmentation of coronary arteries based on CT images. Next, it covers the goals, the research questions, the contribution, and the structure of the thesis.

Background

The background chapter explains some basic medical terms, some general deep learning techniques, and related work. The medical terms include CT images, FFR, and the different structures of the coronary arteries. The related work section gives an insight into the different methods used in the field. This includes some traditional methods consisting of vessel enhancement filters, a more modern method of using deep learning models, and the SegCaps model that uses capsuleNet for segmentation.

Methodology

This chapter describes the methods used for performing the different experiments. It contains details of the dataset, how it is preprocessed and the training strategy used, in addition to a description of the evaluation metrics. The chapter finishes with a detailed description of the different experiments.

Result

The result chapter contains the result of the different experiments. The result is consisting of both tables and visualizations.

Discussion

This chapter discusses the results of the experiments, which are presented in the previous chapter, in relation to the research questions.

Conclusion and Future Work

The final chapter gives the conclusions of the results and the discussion and finishes with suggestions for future work.

Chapter 2

Basic Theory

This chapter will first explain some medical terms in section 2.1 in order to get a deeper understanding of the motivation and the dataset itself. Next, different concepts of deep learning will be described in section 2.2 and the aim is to provide a greater understanding of the network used for the automatic segmentation of the coronary artery. The chapter finishes with an overview of the related work in the field.

2.1 Coronary Artery

The dataset in this thesis consists of CT-scans of coronary arteries. In subsection 2.1.1, the general anatomy of the coronary artery, its functionality, and the coronary artery disease are described. The definition of FFR can be found in subsection 2.1.2. This section finishes with a basic introduction of CT imaging in subsection 2.1.3.

2.1.1 Structure and Functionality

The function of the coronary artery is to supply the heart muscles with blood. It consists of two main branches, which is the left coronary artery (LM) and the right coronary artery (RCA) shown in Figure 2.1. The coronary arteries originate from the Aorta, immediately downstream of the aortic valve. The LM splits into two main branches, and further into smaller arteries, which mainly supplies the left side of the heart. The RCA branches into smaller arteries, which mainly supplies the right side of the heart [26].

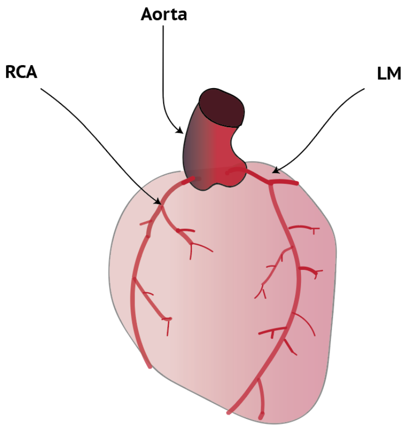


Figure 2.1: Structure of the coronary artery which consists of RCA and LM.

Coronary artery disease (CAD) is caused by a plaque build-up on the inside of the artery wall, which mainly consists of cholesterol and white blood cells. The plaque build-up introduces a narrowing of the artery wall, often denoted stenosis, which may impair the blood supply to the downstream, if the plaque blocks the blood flow completely, or if part of the plaque ruptures, the person could suffer from a heart attack [10].

2.1.2 FFR

The fractional flow reserve (FFR) is defined as

$$FFR = \frac{\bar{P}_{distal}}{\bar{P}_{proximal}} \tag{2.1}$$

where $\bar{P}_{distal}$ is the cardiac cycled averaged pressure down-stream of the stenosis, and is measured with a sensor-tipped wire. $\bar{P}_{proxinal}$ is the cardiac cycle averaged pressure at the Aorta and is measured by a catheter. The purpose is to measure the FFR in the stenosis that may have CAD [6].

2.1.3 Computed Tomography (CT)

A doctor can take a CT or MRI scan if a visualization of bones, organs, vessels, or other tissues of the human body is necessary for analysis or surgery planning. A CT-scan consists of a patient laying on an X-ray sensitive plate with a ring of detectors that encircles the patient and an X-ray source. The X-ray source rotates with the ring of detectors around the patient. During the rotation, the X-ray goes through the patient to the detectors on the opposite end of the X-ray. When the X-ray has rotated to the starting point, a 2D slice of the patient is constructed. For generating the CT-scan, several 2D slices of different depth are stacked together [8, p.28].

In a CT-scan, different tissue will have its value given by the Hounsfield scale. The Hounsfield scale is centralizing water with 0 intensity. In this case, the minimum intensity is the air with -1000 and the maximum is the bone with 1000.

In order to increase the contrast in the CT-scan for a given tissue, a more suitable window size and level can be used. A Hounsfield window and level are used to map the intensity of the Hounsfield scale into new intensity values. The level is used to centralize the grey scale image of the CT-scan. The window size is the range of the extended values in the window, which consists of an interval. If the window level is 300, and the window width is 500, the interval of the window is *(50, 550)*. The values not in the window are clipped to the minimum or maximum value [1]. In this thesis is the term CT image used as a CT-scan.

2.2 Deep Learning

Deep learning is a field in artificial intelligence consisting of deep neural networks. This chapter begins with introducing a basic neural network in subsection 2.2.1 followed by how the optimizer tells the network to update the weights discussed in subsection 2.2.2.

The next topic is the common practice of creating data splits for training the network explained in subsection 2.2.3. This is followed by some techniques for helping the learning process described in sections 2.2.5 - 2.2.8. The vessel enhancement filters will be described in subsection 2.2.6, which can be used as a preprocessing or post processing step. The section ends with a description of the convolution neural network and capsuleNet.

2.2.1 Artificial Neural Network

The artificial neural network was inspired by the biological neural network and its ability to learn. Like the biological neuron gets signals from dendrites the artificial neuron gets input z from input data *x*, weights *w*, and bias *b* shown in Figure 2.2. The weights tell the neuron how much the given input should count, and the bias is some additional value that is added to the values of inputs and weights. Inside the artificial neuron the input z transform to the output *a* with an activation function ϕ [20].

Two common activation functions are sigmoid and ReLU, defined as

$$sigmoid = \frac{1}{1+e^{-z}} \tag{2.2}$$

$$ReLU = max(0, z). \tag{2.3}$$

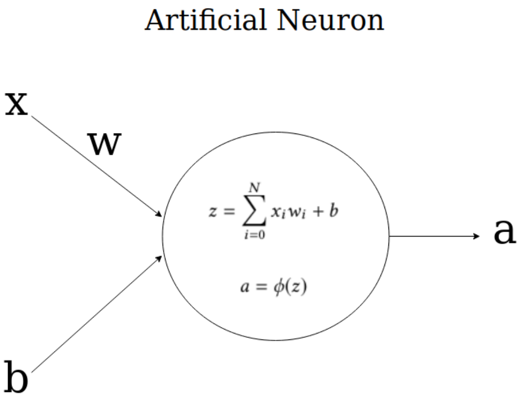


Figure 2.2: An artificial neuron with input x, weight w, bias b, activation function ϕ.

Sigmoid has an output between 0 and 1, which makes it suitable for a probability distribution for binary classification. A problem with sigmoid is that the gradient sometimes becomes so small that it approximates to zero [9, p.65,178]. This problem is called vanishing gradient and will be explained further in section 2.2.2.

The functionality of the ReLU is that it removes all the negative values by setting them to zero. It does not have the vanishing gradient problem like Sigmoid but will stop learning where the inputs are negative[9, p.187].

An artificial neural network consists of connected neurons in layers. The different layers are the input, hidden, and output layer, where the hidden layers are all layers between the input and output layer. A network where all the neurons are connected to all the neurons in the next layer is called a fully connected network, as shown in Figure 2.3.

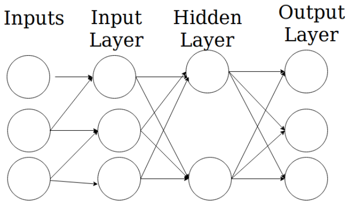


Figure 2.3: The different layer in an artificial neuron network

To speed up the different calculates in the artificial neural network, the bias trick is applied. It consists of only doing matrix calculations, and it is done by adding an extra input of value 1 and using the bias as its weight, as shown by

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{bmatrix} \times \begin{bmatrix} w_1 & \dots & w_n & b \end{bmatrix} = z. \tag{2.4}$$

A forward pass is when the inputs have gone through the artificial neural network, and a prediction is given from the output layer. When learning, it uses a loss function to calculate the loss of the predicted and the actual value.

The next step is an optimization that is done by an optimizer like stochastic gradient descent, which will be described in subsection 2.2.2. The optimizer's job is to update the weights and biases. An artificial neural network uses this process to train. An epoch is when the artificial neural network has gone through all of the training samples once.

2.2.2 Optimizer

As mention in subsection 2.2.1, the job of the optimizer is to update the weights and bias in neurons to reduce the training error, a process that makes the network able to learn. The optimizers use the delta rule, defined by

$$\theta = \theta + \Delta\theta, \tag{2.5}$$

for updating the weights and bias (θ). The different types of optimizers are distinguished by how they calculate $\Delta\theta$, and some examples are given below.

Stochastic Gradient Descent(SGD): In deep learning is all the training samples typically divided into groups called batches of size *m*, for updating θ. SGD first calculated the mean gradient of the loss function (*L*) for a batch defined by

$$g = \frac{1}{m}\nabla_\theta \sum_{i=0}^{m} L(x_i). \tag{2.6}$$

It then uses

$$\Delta\theta = -\eta g \tag{2.7}$$

to calculate $\Delta\theta$, where η is the learning rate, which is how much the network should learn from each error.

After the neural network has calculated the forward pass for all the samples in a given batch, it uses back-propagation to calculate the different gradients in the neural network. Back-propagation uses the chain rule for getting all the gradients of the loss function on all the weights and biases of the network. A problem with SGD is finding the optimal parameters for not getting stuck in local minima, such as the learning rate [9, p.147-149, 286-288].

Momentum can be used to speed up the training for SGD. The momentum takes the previous gradient (*v*) into consideration when calculating the $\Delta\theta$. The α constant is used to decide how much *v* should count in the new calculation of $\Delta\theta$

$$\Delta\theta = \alpha v - \eta g \tag{2.8}$$

[9, p.288-290].

Adam: An adaptive learning optimizer is an optimizer where each parameter has separate learning rate parameters. These parameters are p_1 for the first momentum *s* and p_2 for the second momentum *r* when using the Adam optimizer. The first and second momentum are zero initialized and updated by

$$s = p_1 s + (1 - p_1)g \tag{2.9}$$

$$r = p_2 r + (1 - p_2)g \odot g \tag{2.10}$$

The gradients *g* are calculated in the same way as for SGD. Next step is to make estimates for *s* and *r*

$$\hat{s} = \frac{s}{1 - p_1^t} \tag{2.11}$$

$$\hat{r} = \frac{r}{1 - p_2^t} \tag{2.12}$$

with *t* being the current epoch. $\Delta\theta$ is calculated from

$$\Delta\theta = -\eta \frac{\hat{s}}{\sqrt{\hat{r}} + \delta}, \tag{2.13}$$

where δ is a small constant used to prevent division by zero. A benefit of using Adam is that learning is less sensitive again chosen hyperparameters than SGD [9, p.301-302].

Problems with optimization: The problems of under- and overfitting, described in subsection 2.2.3, are two common problems in deep learning. Another problem is vanishing and exploding gradients. The optimizer sometimes gets stuck in a local minimum when trying to find the global minimum for the loss during the training of the model. One solution is to use an adaptive learning optimizer, such as Adam that is described above. The vanishing gradient problem is when the derivative of the gradient is approximate to zero, with the result of parameters not being updated [20, chap.5]. Exploding gradients are when the gradients become too big and result in large updates, which makes the predictions more unreliable. One solution to this problem is to use cliffs gradients, which consists of clipping all gradients over an upper limit to the max value [9, p.281]

2.2.3 Training and validation set

In deep learning, the dataset is typically divided into training, validation, and test set. First is the dataset is usually divided into 80% samples in training and 20% for the test set. Next, the training set is typically divided into 80% samples for training and 20% for validation. The functionality of the validation set is to monitor the generalization error during training. The generalization error is the error the network has on unseen samples and makes it easier to observe when overfitting has occurred.

The problem of overfitting and underfitting is common in deep learning. Underfitting occurs when the model has problems finding any patterns in the training data usually caused by having a small dataset with too much variation. Overfitting is when the model finds patterns that are only in the training set [9, p.109].

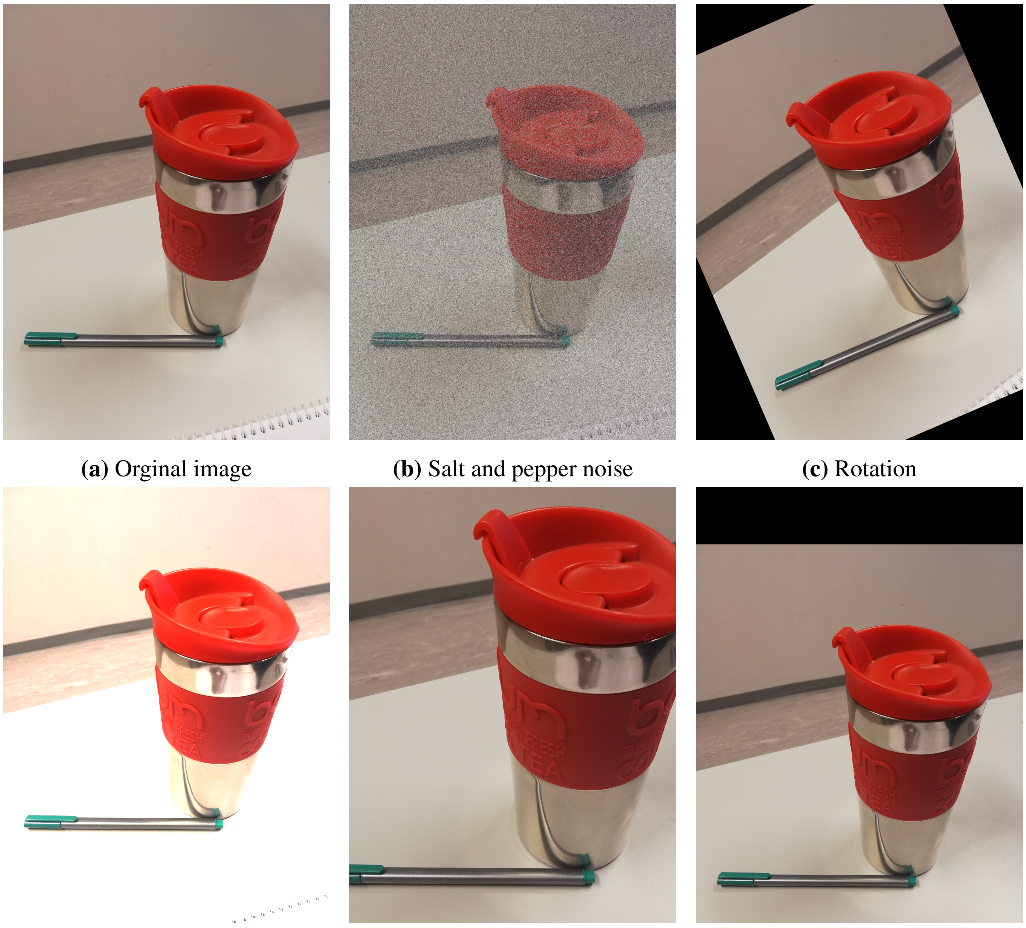
Some data augmentation and regularization techniques can be used for reducing the chance of overfitting, which will be explained in the sections below.

2.2.4 Dataset Augmentation:

This technique consists of getting more data by applying affine transformations to some of the training data, as shown in Figure 2.4. It is important that these transformations are realistic and could occur in the dataset for reducing the generalization gap. The transformations can be shear, rotation, scale, flip, and translation [9].

Another strategy is to apply noise in order to make the neural network more robust. Noisy images is a common problem in some cases, such as medical imaging or old images. A typical noise is salt and pepper noise, which makes white and black dots occur on the image[8, p.178].

Another strategy is to use elastic deformations as used in U-Net [23], which is used for making variations in the surface. It is done by generating a random distribution vector from a Gaussian distribution on a coarse grid of 3x3 grid and using bicubic interpolation afterward.



(a) Orginal image **(b)** Salt and pepper noise **(c)** Rotation

(d) Brightness **(e)** Scale **(f)** Translation

Figure 2.4: Examples of some types of augmentation data

2.2.5 Regularization

Some regularization techniques can be used for reducing the chance of overfitting [9, chap.7], where some of the most common techniques are explained below.

Weight decay: A way of reducing the chance of overfitting is to regularize the weight update in the optimization by adding a weight penalty ($\alpha\Omega(\theta)$).

α has values in the interval [0,1] and is used to denote how much the weights is contributing, where 0 is none. The weight penalty is only used on weights because a regularization on bias can cause underfitting [9, p. 223].

$$L_2 = \frac{1}{2}||w||^2 \tag{2.14}$$

$$L_1 = \frac{1}{2}||w|| \tag{2.15}$$

are the two common weight decay($\Omega(\theta)$) strategies. When using L_2, the weights will shrink on the features that have a high variance on the features of the target [9, p.224-227].

L_1 has another effect on the weight decay than L_2. The weights will become sparse and can be used for feature selection. The reason is that a subset of the weights will be zero, and the non-zero can be used for feature selection [9, p.227-230].

Early Stopping: Sometimes the generalization error keeps on increasing during training, and the neural network gets overfitted. Early stopping ends the training if the validation loss has not decreased after a x number of epochs, where x is called patience [9, p.242-245].

Dropout: Another method, called dropout, is used for preventing overfitting by applying dropout layers. In these layers are neurons deactivated with the probability p and the active neurons are multiplied with $1-p$ to scale down the active weights [9, p.251-262].

Batch Normalization: A method that achieves the same effect as dropout is batch normalization. In these layers, the output is normalized between 0 and 1 for each batch [9, p.260,309-312].

2.2.6 Vessel Enhancement Filters

A way of preprocessing or post processing an image is to apply a filter to extract wanted features. A vessel enhancement filter is typically used to enhance vessels or other tubular structures, and it is categorized either as Hessian-based or diffusion-based filters. The most popular Hessian-based filters are Frangi and Sato. All the Hessian-based filters calculate the Hessian matrix to get the eigenvalues. This can be used to enhance vessels because it gathers both the direction and the brightness. The Hessian matrix is the derivative in the vertical and horizontal direction of the image.

In the Frangi filter, the eigenvalues λ are sorted by increasing absolute value, with λ_1 being the smallest. The α, β, γ are constants set to control the sensitivity values of

$$R_A = |\frac{\lambda_2}{\lambda_3}| \tag{2.16}$$

$$R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}} \tag{2.17}$$

$$R_S = \sqrt{\sum_j \lambda_j^2}. \tag{2.18}$$

The Frangi filter uses

$$V_F(\sigma) = \begin{cases} 0, & if\ \lambda_2 > 0\ or\ \lambda_3 > 0 \\ (1 - e^{-\frac{{R_A}^2}{2\alpha}})e^{-\frac{{R_B}^2}{2\beta}}(1 - e^{-\frac{{R_S}^2}{2\gamma}}), & otherwise \end{cases} \tag{2.19}$$

to enhance the vessels.

The Sato filter differs from the Frangi in that it only uses two eigenvalues. The first eigenvalue is the largest of the eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$. The second eigenvalue λ_c is $min(-\lambda_2, -\lambda_3)$. It uses

$$V_S(\sigma) = \begin{cases} e^{\frac{-\lambda_1{}^2}{2(\alpha_2\lambda_c)}}\lambda_c, & if\ \lambda_1 \leqq\ 0,\ \lambda_c \neq 0 \\ e^{\frac{-\lambda_1{}^2}{2(\alpha_2\lambda_c)^2}}\lambda_c, & otherwise \end{cases} \tag{2.20}$$

for finding the vessels, where the α_1 and α_2 are used to optimize the ratio between the two eigenvalues.

The diffusion filters uses the partial differentiation $u_t = div(D, \nabla u)$. The D is the diffuse tensor and the ∇u is the gradient of the image. This method can be divided into anisotropic and isotropic diffusion.

The anisotropic method changes locally magnitude, the direction of the smoothing with diffusion tensor, where isotropic diffusion uses the local gradient magnitude to decide the smoothing. An example of isotropic diffusion is RPM and anisotropic is HDCS [18].

2.2.7 Post processing

The predicted image needs to be thresholded to get binary values. This threshold value is usually *0.5*, but finding the best threshold value can be difficult. The Otsu method tries to find the optimum global threshold for the image. This method uses the histogram of the image and chooses the threshold that minimizes the weighted sum of the group variance [8, p.764-769].

Sometimes the predicted images contain some unconnected or noisy parts, like small dots that do not belong to the main object. A way of solving this problem is to use morphological operations like erosion, dilation, opening, and closing. The functionality of erosion is thinning the border of the image.

It is

$$A \ominus B = \{z|(B)_z \cap A^c = \emptyset\}, \tag{2.21}$$

which means all the possible translates of the filter *B* within the image *A* containing uncommon elements is set to zero. The dilation expand the border of the image and are defined as

$$A \oplus B = \{z|(\hat{B})_z \cap A^c \neq \emptyset\}, \tag{2.22}$$

where $\hat{B}_z$ is B reflected around its origin and shifted by z. The result of the dilation is that all the uncommon elements is set to one.

Opening is erosion followed by dilation and is used for removing small areas, like vertical lines in an image. Closing is dilation followed by erosion and are used for filling holes, for example, an image with a circle with spots [8, p.652-654, 657-660].

Another way of doing post processing is to find the connected components and remove unwanted connected components. Region growth is a method for finding connected components. It starts by setting out some seed points in the image. Next, it collects all the neighbors over a given intensity threshold. Then it finds the neighbors of the neighbors over a given intensity threshold until there are no more to be found [8, p.174-176].

2.2.8 Transfer Learning

In deep learning, the initial model weights are usually zero or random. The purpose of transfer learning is to give the model a better starting point by using the weights from another model as the starting point. Transfer learning is most beneficial when the transfer weights have been trained on a similar data form.

By using transfer learning, a small dataset may use longer time to overfit because it already has learned some of the special features from the transfer model. Another benefit is that the training may converge faster to the desired value, which reduces the training time, and less data may be needed [22].

2.2.9 Convolutional Neural Network(CNN)

For a neural network to be a convolution neural network, it has to have at least one convolution layer. A convolution layers consisting of x filters to extract different features by using convolution defined by

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n), \tag{2.23}$$

where the input is referenced as *I* with size *(m, n)* and the filter as *K* with a size of *(i, j)*. *S* is called the feature map and contains the detected features. The size of the filter is essential for the feature detection. A small filter will find more specific features, and a large filter will find more general features. A feature can be corners or intensity changes in an image. The filters in a convolution layer are the weights at that layer, which are updated during training.

The advantages of using CNN is that it will still recognize the object if its location change and the use of parameter sharing. The benefit of using parameter sharing is that some of the computation cost will be reduced [9, p. 321-330].

In the CNN architecture is it normal to have some convolution layer with nonlinear activation functions followed by a pooling layer. The functionality of the pooling layer is downsampling its input. This is done for making the input more robust against small, large, or both small and large changes because it replaces a given location with a summary of the nearby inputs. The summary can be the maximum, minimum, or average value of the nearby inputs.

The nearby inputs and the replaced point are selected similar to the convolution filter moving across the image with the stride as the step size [9, p. 330-335].

For making the upsampling learnable transposed convolution can be used. It uses the the principle that convolution can be translated to matrix operations between the flattened input *I* of size *(Nx1)* and the filter *K* with size *(MxN)*. For performing a forward pass the input is multiplied with *K* and back-propagation is computed by multiply the loss with K^T. The transpose convolution multiplies the input with K^T for the forward pass and the loss with *K* for back-propagation [5].

2.2.10 CapsuleNet

CapsuleNet was introduced in 2011 and is a new way of making the recognition more viewingpoint invariant and will be discussed in Transforming Auto-encoders. The pooling reduces the computational cost by downsampling its input parameters, but in this process are a lot of information lost. The CapsuleNet uses routing agreement techniques for reducing some of the computational cost without losing information, which will be explained below.

Transforming Auto-encoders: One of the difficult tasks in computer vision is to recognize the object with all possible angles and shapes. One problem with convolution neuron networks is that it does not know the relationship between the features. An example is in face recognition, where CNN will still predict a face even if the nose and mouth switched place.

The neurons in the convolution network produce scalars for the feature detectors as a summary of the activity. An alternative is to use ”capsules”, which produce a vector with the probability of the different viewing conditions, such as rotation, brightness, and deformations. These parameters are called instantiation parameters and are used to give information about the rendered object, as shown in Figure 2.5. An advantage is that the instantiation parameters can be used to debug where the recognition went wrong.

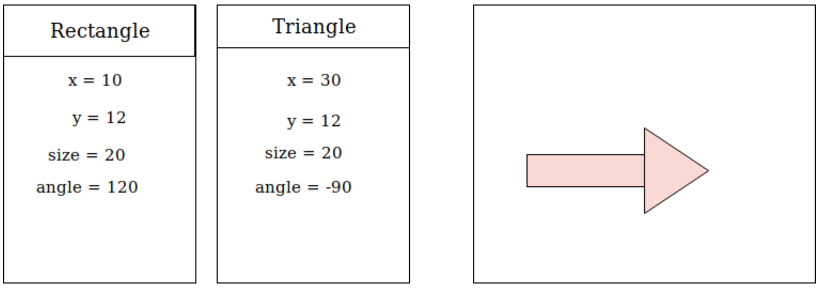


Figure 2.5: Some instantiation parameters of the rectangle and triangle.

This concept is similar to the affine transformations from computer graphics, where it is possible to apply different linear transformations to end up with the wanted pose. An example is an image of a smaller rotated hand-written number. It needs to be rotated back and scaled up to the general representation of the object and is then viewpoint-invariant.

Transforming auto-encoders are used to create the first capsule layers from the pixel intensities. This is done by each capsule having a recognition and generation unit. The recognition unit consisting of p(probability of the object being present), x, y, that is passed forward to the next layer. The generation is $p(x + \Delta x)$ and $p(y + \Delta y)$, and by multiplying with p decides how much of the generation unit to consider when generating the output. The model created by Hilton had the weakness that the capsules could only represent one representation of the object at the same time [11].

Dynamic Routing Between Capsules: Routing agreement technique is used for selecting which of the capsules in the next layer to activate without losing any information like pooling layers do. Sabour [25] chose to use the capsules output vector v_j length for denoting the probability for whether the entity of the object was present. An example is shown in Figure 2.6.

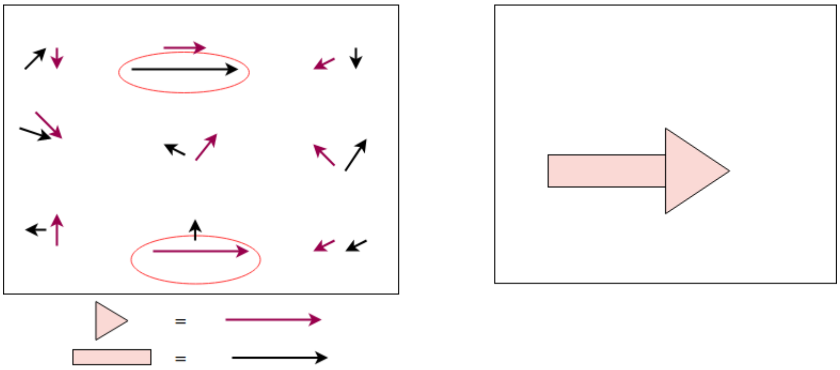


Figure 2.6: It is an example on some of capsules output vectors for finding rectangles and triangles with their pose. The black arrows belong to the rectangle, and the purple arrows belong to the triangle. The different poses of the arrow are the different values of the rotation instantiation parameter. The red rings are around the longest output vector of the rectangle class and longest output vector of the triangle class.

The capsule output vector is calculated by

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||}, \tag{2.24}$$

where s_j is the the total input for that capsule that is calculated by

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{2.25}$$

except for the first capsule layer. The first capsule layer is created by resampling its input and ”squashing” it to have values below 1.

The idea is that ”routing-by-agreement” can lean the relationship between the features by making the lower level capsule only activates all the possible parents at the next layer. In the Figure 2.6 is the rectangle and triangle the low level capsules and the possible parents is the different ways of putting them together. The wanted possible parents are the rectangle and triangle, forming an arrow such as in the figure.

The activation is done by first sending the scaled down output vector of the low level capsule to each of the possible parents. It is scale down by the coupling coefficient, which is calculated by

$$c_{ij} = \frac{\exp b_{ij}}{\sum_k \exp b_{ik}}, \tag{2.26}$$

where b_{ij} is the softmax of the prior log probability that capsule$_i$ should be routed to capsule$_j$. Next each parent capsule multiplies its own output u_i with the weight matrix W_{ij}, which is called the ”prediction vector”

$$\hat{u}_{j|i} = W_{ij} u_i. \tag{2.27}$$

The possible parent with a large scalar of the prediction vector and output vector get an increment of the coupling coefficient, and all the others get a decreasing. This is done by updating b_{ij}

$$b_{ij} = b_{ij} + \hat{u}_{j|i} \cdot v_j. \tag{2.28}$$

After some iterations are some of the possible parents become so small that they are denoted as deactivated capsules.

EM Routing: Hilton [12] published a new routing agreement between capsules in 2018 called EM Routing. In this cases is the capsules in layer L is denoted as Ω_L, and each of them has a 4x4 pose matrix M and an activation probability a as shown in Figure 2.7. For training, a 4x4 transformation matrix W_{ij} is used as weights. Each of the capsules in Ω_L is voting for finding which of the capsules in Ω_{L+1} that should be activated. The voting is by calculating $V_{ij} = M_i W_{ij}$.

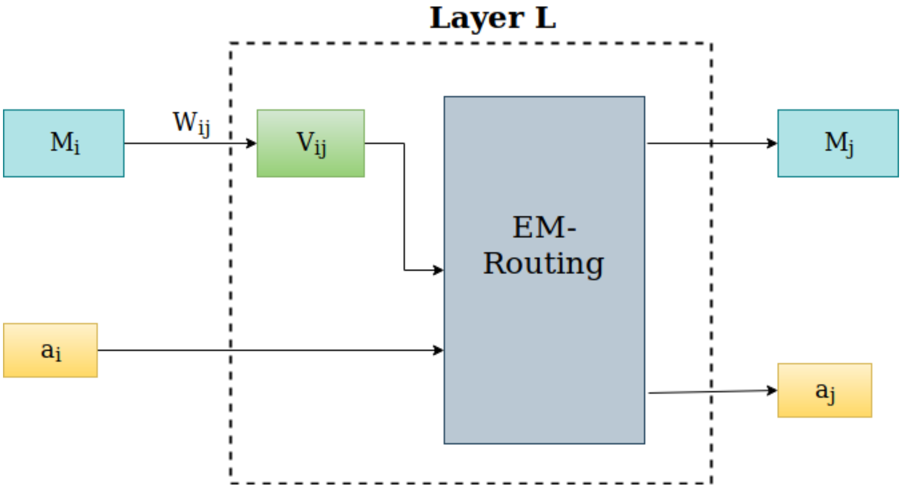


Figure 2.7: Overview of which values the EM-Routing gets from the capsules

This algorithm consists of an E-step and M-step. The E-Step calculates the probability of a datapoint of Ω_L belonging to a given gaussian distribution of the Ω_L datapoints to a given capsule of Ω_{L+1}, which is used as weights. The M-step is used to calculate the mean and variance of the datapoints, and is approximating the gaussian distribution to best fit the datapoints assigned to it. This is done by making each gaussian distribution maximize the sum of weighted log probability of these datapoints. The idea is that after three iterations of EM-routing is only the capsules in Ω_{L+1} with higher probability than the activation cost will be activated.

2.3 Related Work

In this section, different architectures of segmentation on medical images with main focus on vessels and artery structures will be explored. First, the traditional methods consisting of using vessel filters will be described in Vessel filters-based segmentation models. Next, a more modern method of using different architectures of the fully conventional network for segmentation found in subsection 2.3.2. The chapter finishes off with the SegCaps, which uses new deep learning network, capsule network explained in subsection 2.2.10, for segmentation.

2.3.1 Vessel filter-based segmentation models

There are many different ways of doing vessel segmentation. Jimenez-Carretero [14] created a method for lung vessel segmentation consisted of the Frangi filter described in subsection 2.2.6 for detecting lung vessels and the inverse to detect airways in the vessel. Segmentation was calculated by

$$V_{\mathbf{F}vessel}(x \in WallCandidates(y)) = max(0, V_{\mathbf{F}vessels}(x) - V_{\mathbf{F}airway}(y)). \tag{2.29}$$

Next, is a kind of a region-growth method from Luu [18] for segmenting liver vessels. The method is summarized in Figure 2.8. The first step is to apply a filter that can be either Hessian or diffusion based on the input image. This is followed by generating liver masks on both the input and filtered version image.

The Masking Liver step consists of making contours on some slices where the liver vessel are. The Histogram-based Estimation step is used to calculate the threshold value for the seed points in the Region growing step and the Regional maxima step.

The Regional maxima step is used to remove some of the seed points used in the Region Growing step. The Region growing step works like the connected component analyses explained in subsection 2.2.7. The postprocessing step is applied to remove noise, which consists of using closing explained in subsection 2.2.7 with a filter

5 x 5 x 3. The best result was with the Frangi and HDCS filters.

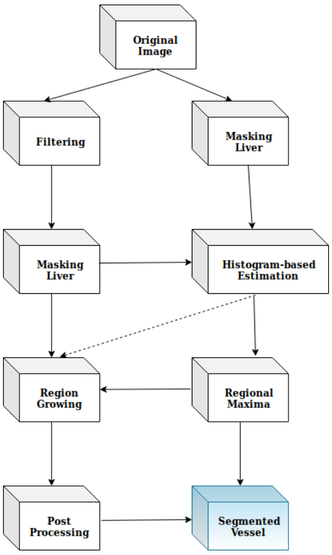


Figure 2.8: A summarization of the liver segementation method of [18]

A method consisting of combining the Frangi filter with deep learning model was created by Fu [7] and used on Retina vessels. This method involved using Frangi filter as pre-trained weights in the neural network. The first step was to convert the output of the filter to a binary mask. This was done by subtracting a threshold and asymmetrically scale it and last apply a Sigmoid layer to get a probability distribution. The architecture is designed to be equivalent to the Frangi filter described in subsection 2.2.6. The idea is that the weights from the convolution net can be used in another network to improve the segmentation on tubular structures.

Bhattacharya [2] created a method for coronary artery segmentation. It consists of the Frangi filter and an anisotropic diffusion technique called ”Optimized Anisotropic Rotational Invariant Diffusion Scheme”. The first step of the method used an angiography image as input and set the values for α, β, and σ_{max}. The σ_{max} values are the max limit for the standard derivative. Second step is to apply the Frangi filter and use the result that is $F(x) = max_{\sigma} f(x, \sigma)$. Next part of the method is Optimized Anisotropic Rotational Invariant Diffusion Scheme”. It begins by applying the Gaussian smoothing on the output of the Frangi filter, then calculating the Hessian for all the pixels and finding the eigenvalues of it afterward. Next, the diffusion tensor is calculated with the eigenvalues as directions. The method finishes with a finite Difference scheme before it applies all of the steps one more time. This method had some improvement in finding smaller arteries.

2.3.2 Fully Convolutional Networks Segmentation models

Ronnebergers U-Net [23] has become one of the gold standards for segmenting biomedical images. It won the ISBI challenge in 2015, which consisted of segmenting cells and membranes. In this challenge, U-Net overcomes the limitation of the training data size by using strong data argumentation and an autoencoder architecture. Shift, rotation, greyscale variations, and elastic deformations were used as data augmentation to make the training data more robust to unseen samples.

The architecture has an encoder and decoder path with depth four shown in Figure 2.9. The upsampling layer at depth$_x$ concatenate the last layer of the encoding block at depth$_x$ and the last layer of the decoder block at depth$_{x-1}$.

This is used for getting both low and high-resolution features. The U-Net applies a convolution layer with valid padding, which means the output size will become smaller than the input size of that layer. Another benefit of the U-Net model is that it can be trained end-to-end.

One of the challenges with segmenting cells and membranes was separating touching objects. This was solved by using a weighted loss function, where the weight was calculated using morphological operations.

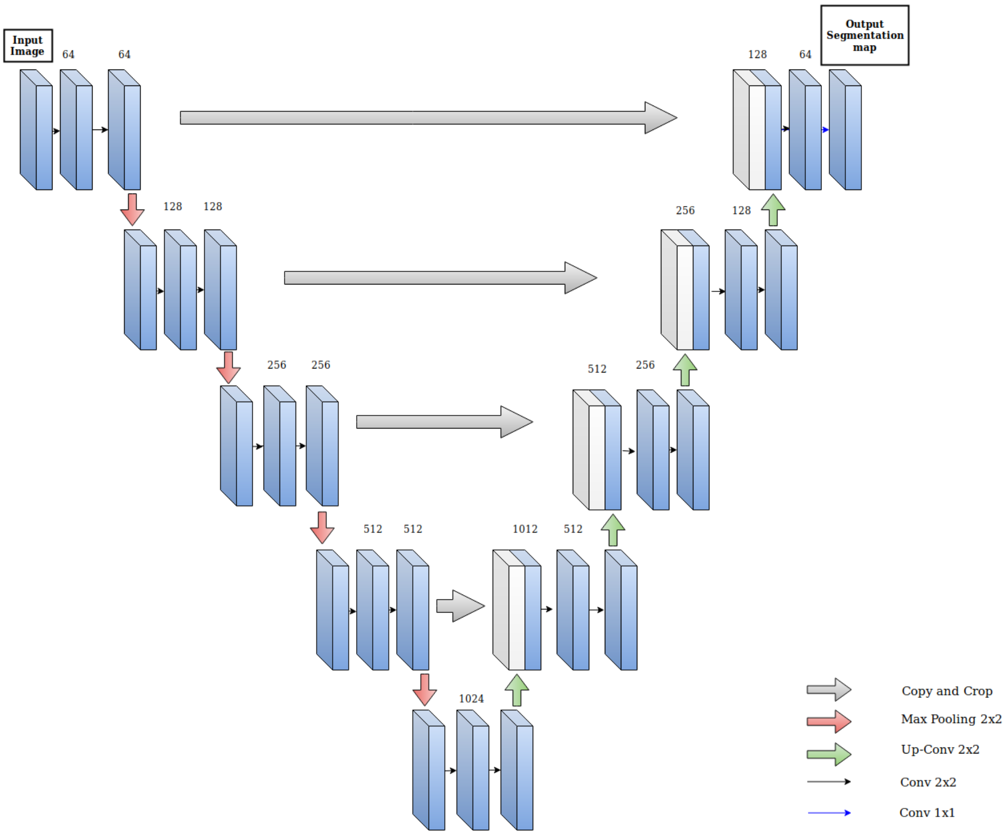


Figure 2.9: The architecture of U-Net

The 3D U-Net model came in 2016 [3], which has the same architecture as U-Net except all the 2D operations are replaced with 3D operations. This architecture is, therefore, more computationally expensive. Between the convolution layer and the activation function was batch normalization applied. This was done for making the model converge faster and decrease the change of overfitting. The model was trained on the Xenopus kidney dataset, which was divided into voxels before passing through the model.

Another segmentation architecture for volumetric images came in 2016 called V-Net [19], which was used for segmenting the prostates from MRI images. It has an autoencoder architecture with a depth of four. The loss function used is dice loss that handles the strong imbalance between foreground and background voxels. Instead of pooling is downsampling done with the usage of convolution layer with 2x2 filters and stride of two. The upsampling is done by using convolution transpose layers a 2x2 filters with stride two. The activation function used between the convolution layers is PReLU. It uses $max(\alpha x, x)$, which treats α as a learnable parameter.

Roth [24] created a two-stage pipeline for multiclassification, which is used for segmenting different anatomical structures from CT images. The anatomical structures include vessels and different organs.

The first stage is to make the FCN separate the foreground and background voxels from each other. The last stage helps the model segmenting the smaller classes, like vessels, with focus on the boundary. The model used for the different stages is the 3D U-Net with a modified loss function.

Kjerland [16] used the Deep Medic model for coronary artery segmentation. This architecture consists of two residual paths connecting at the end, where one of the residual paths being a scaled-down version of the image. This method used data refinement, in the form of connected component analysis, on the predicted image to improve the result.

Joacobsen [13] used the U-Net, BVNet, and 3D U-Net model in his thesis for segmenting coronary arteries. For making the training data more robust, data augmentation were applied in the form of elastic deformation, brightness, rotation, and scale.

The two next and previous slices were added as channels to help the model get a better understanding of the predicted slice surroundings. By using this technique, the model is training on the channel in the middle.

The BVNet differ from U-Net having depth three and an extra convolution layer at each encoding block located at the end and decoding block locating at the beginning, which is shown in Figure 2.10. The extra convolution layer differs from the other convolution layers in that block with having twice as many parameters.

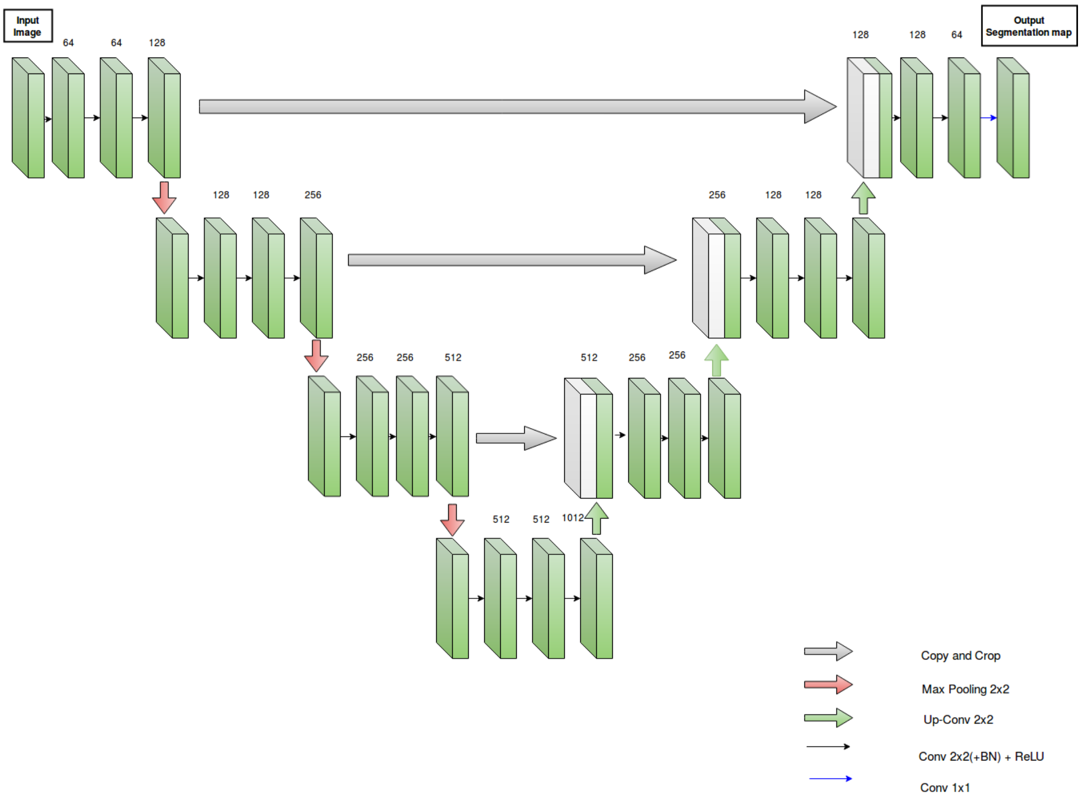


Figure 2.10: Architecture of BVNet

2.3.3 SegCaps

LaLonde and Bagci [17] were the first to use capsules for segmenting medical images, and the architecture is called SegCaps. It has a convolutional-deconvolutional, architecture that is shown in Figure 2.11, and was used on the LUNA16 dataset that consists of CT images of lungs. The capsules pass forward to the next layer capsules using dynamic routing between capsules as described in subsection 2.2.10. Both the capsule architecture and the routing algorithm are memory expensive. In an effort to solve this all capsules of a given type share the transformation matrix T, which cut down the number of parameters learned by back-propagation. The capsule type is a user-defined kernel to the parent capsule. For calculating the segmentation mask the length of the final layer, capsules are used for assigning the class if the magnitude is over a given threshold.

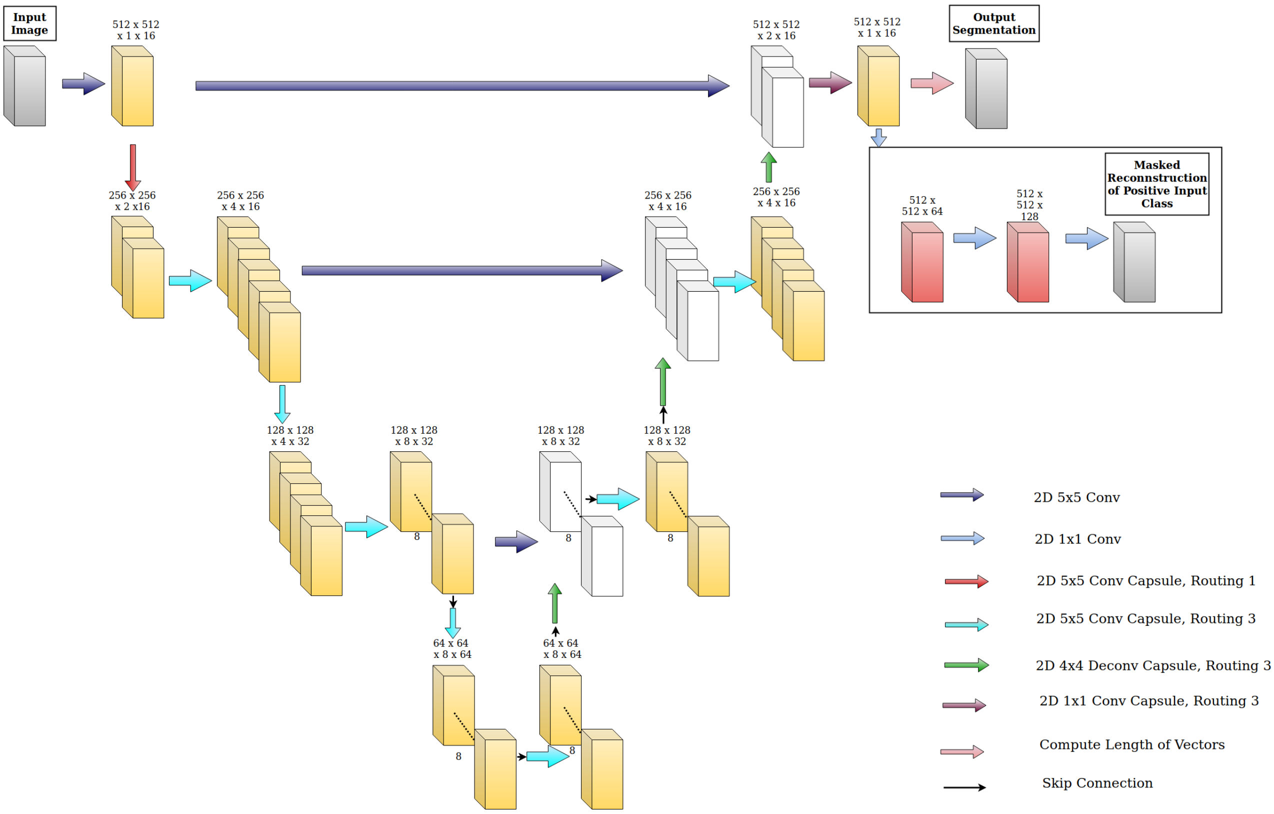


Figure 2.11: Architecture of SegCaps

Chapter 3

Methodology

This chapter explains the implementation details for performing the experiments. First, an introduction to the dataset is given in section 3.1. This is followed by a description of the various technical tools needed to perform the experiments. In section 3.3 will some of the general implementation details be explained, which is common for all the experiments. The chapter finishes with a description of all the different experiments.

3.1 Dataset

The main dataset is provided by st.Olav hospital as a part of the FFR project. The dataset consists of 3D CT images explained subsection 2.1.3 of size $(512, 512, z)$, where z is the depth that has a value around 300 for most of the images. There is some variation between the voxel spacing, which is around $(0.38, 0.38, 0.3)$. The images are having different Hounsfield levels and window size, which makes them have different contrast and intensity.

Each input image has three labels, namely RCA, LM, and Aorta. In this thesis, the RCA and LM label is combined into one ground truth known as the coronary artery label.

At the beginning of this thesis, the dataset consisted of 82 CT images, but because this is still an ongoing project, 8 more images were available through out the thesis.

The dataset was divided into three different splits, which is shown in Table 3.3, for seeing more general patterns in the prediction. The size of the dataset is quite small, which makes it sensitive for edge cases.

The split-0 contains 66 training images because it was created before more images were available. Split-1 and split-2 contain 72 training images. All the images for training, validation, and testing were selected randomly for the splits.

Another dataset called 3Dircadb1 is used to check if the models can be used to segmenting other vessels. The dataset is created by IRCAD for the 3D-ircadb 01 project[1], which consists of 20 CT images with labels from the different parts of the liver, where 75% has hepatic tumors. The images have a Hounsfield level at -0.5 and window size 2047. The voxel spacing has values around 0.6 for the x- and y-axis and 1.6 for the z-axis. The size of the images is $(512, 512, z)$, where z is the depth that has a value around 125 for most of the images.

In this thesis, the portal vein label is used, and the dataset is divided into three splits. Each data split consisted of 16 training, 2 validation, and 2 test images, as shown in Table 3.3.

	Split-0	Split-1	Split-2
Training images	66	72	72
Validation images	8	8	8
Testing images	8	8	8

Table 3.1: Dataset from st.Olav

	Split-0	Split-1	Split-2
Training images	16	16	16
Validation images	2	2	2
Testing images	2	2	2

Table 3.2: 3Dircadb1 dataset

Table 3.3: Information about the different splits.

[1] https://www.ircad.fr/research/3d-ircadb-01/

3.2 Technical Tools

This section gives a description of all the different technical tool used in this thesis.

GPU Cluster A cluster where each user can connect to a node that can have one or more GPUs connected to it. In this thesis two different GPU clusters were used. One containing Tesla V100 GPUs with 34 GB GPU and another with two Tesla v100 GPUs with 50GB. In this thesis the GPU is used to speed up the calculations and shorten the training time.

Docker[2] A platform used for making it easier to develop, deploy and run applications, by using containers that is a running docker image. An docker image contain all the dependencies needed to run an application. The benefit of this is that the container isolates the software environment, which that makes the application run independently on any computer. In this project, it was used to run programs on the GPU cluster.

Keras[3] A python library used for deep learning. It uses either TensorFlow, CNTK, or Theano as backend for creating and usage of the network. Many of Keras functionality can be seen as a black box, which makes it easy to use, like the fit method used for training the model, where only the hyperparameters and data needs to be specified. Another aspect is that it is modular, as it is easy to add and remove functionally such as performance metric or a layer in the model. In Keras is it possible to make custom metrics and loss functions, which is useful when doing research. In this project, the dice similarity and dice similarity loss are custom objects and TensorFlow is used as backend.

ITK-Snap[4] A program that is written in C++ and provides 3D segmentation. It supports different 3D image domains like NIfTI and DICOM[27]. In this project, it is used to visualize 3D structure, mainly of the ground truth segmentation and the predicted segmentation.

VMTK[5] A program consisting of different libraries and tools for making it easier to analyze and model blood vessels. In this thesis the VMTK program was used for creating the Frangi filtered images. This will be explained further in subsection 3.4.3.

3.3 Implementation

The general implementation details are explained in this section. This is everything from the different preprossing approaches, to the training strategies and the post prossesing. The code implementation[6] is forked from laLonde's implementation of SegCaps[7].

3.3.1 Preprosessing

The 3Dircadb1 dataset is a small dataset, and a new Hounsfield window size and level were applied to increase the contrast at the portal vein. The portal vein has an intensity value around 200, which became the new Hounsfield level. Next, an assumption was taken about the intensities, which consisted of that the intensity range of the portal vein would not change more than $+/-$ 500. This resulted in a new Hounsfield window of 1000.

The next step is common for both the 3Dircadb1 and st.Olav dataset. It consisted of subtracting each image by its mean and dividing it by its standard deviation.

3.3.2 Training

The training is set to 500 epochs with early stopping with monitoring the validation loss for preventing overfitting as explained in subsection 2.2.5. The batch size is set to the maximum possible size, which is 8 for BVNet and BVNet3D and 1 for SegCaps. The optimizer is Adam, described in subsection 2.2.2, with a starting learning rate

[2] https://docs.docker.com/get-started/
[3] https://keras.io
[4] www.itksnap.org
[5] http://www.vmtk.org
[6] https://github.com/Trudelutt/SegCaps
[7] https://github.com/lalonderodney/SegCaps

of 0.0001. To help the model with getting a better learning rate, a learning scheduler is used. It decreases the learning rate by 0.05 if the validation loss has not decreased after five epochs. After each epoch, all the training and validation images are shuffled to create more variation. The slices of the image are also shuffled to make the model less sensitive to the order of the slices in the image.

For centralizing the training, all the image slices and patches only containing background are removed from the training set.

3.3.3 Evaluation Metrics

An evaluation metric is used for evaluating the prediction result. For using the evaluation metrics, the prediction needs to be binary. For making the predictions become binary, a threshold of 0.5 is used during training, and the Otzu threshold for testing. As mention in subsection 2.2.7, the Otzu threshold is the value that minimizes the weighted sum of the group variance of the image's histogram. The result of the prediction can either be true positive (TP), true negative (TN), false positive (FP), or false negative (FN), as explained in Table 3.4.

Term	Desciption
True positive(TP)	Predicts that it is the given class when it is the given class.
True Negative(TN)	Predicts that is not the given class when it is not the given class.
False Positive(FP)	Predicts that it is the given class when it is not the given class.
False Negative(FN)	Predicts that it is not the given class when it is the given class.

Table 3.4: Predictions Outcome

The segmentation task of this thesis consists of binary classification. It consists of classifying all of the pixels to either be the background or the wanted class.

Dice Score The dice similarity coefficient metric takes the class imbalance problem into consideration when measuring the performance of the network, by valuing the true predictions twice as much. It both encourages the model to find the true positive without predicting all to positive [4]. The dice score is defined by

$$Dice\ Score = \frac{2TP \cap TN}{2TP \cup FP \cup FN}. \tag{3.1}$$

Recall Recall measures the ratio of the correct target values detected and is defined by

$$Recall = \frac{TP}{TP + FN}. \tag{3.2}$$

In the case of class imbalance, the recall will be high if the model only predicts the given class [9, p.411].

Precision The precision metric measures the amount of correct predictions defined by

$$Precision = \frac{TP}{TP + FP}. \tag{3.3}$$

In the case of class imbalance the prediction will be high if the model identifies none of the target values of TP, when background is the dominating class [9, p.411].

3.3.4 Loss Function

The loss function explained in subsection 2.2.1 is used to measure how good the model is at making correct predictions. The dice loss was used in this thesis, and is calculated by

$$Dice\ Loss = 1 - Dice\ Score(T, P), \tag{3.4}$$

where P contains values between 0 and 1.

The dice loss is beneficial to use when there are large differences in the number of pixels belonging to the different classes. In this thesis, the background class is much larger than the ground truth class.

3.3.5 Post Prosessing

The post processing step consists of applying some image processing techniques, with the goal of improving the prediction, most of these techniques are explained in subsection 2.2.7. All the predictions are binary after applying the Otzu threshold. This is the raw test predictions in chapter 4. Next step, used connected components analysis.

Three different cases are used to decide which of the connected components that should be removed.

The first case involves removing all the connected components except the largest. The condition is that the largest component needs to be a lot larger than the second largest component. The assumption is that the largest component is the Aorta or the portal vein.

The second case involves removing all the components except the two largest components. Here is the condition that the second largest component needs to be a lot larger than the third component. In this case, the assumption is that the two largest components are RCA and LM.

The third case removes all the components of size less than 500 when none of the cases above are accurate. The removing threshold is set to 500 because a higher threshold is more likely to remove some of the TP of the smaller arteries.

3.4 Experiment

In this section a detailed description is given for all the experiments. All of these experiments are created to gather information in order to answer the research questions from section 1.2. In the table below gives an overview of the step up of each of the experiments.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
RQ	RQ1	RQ2	RQ3	RQ4	RQ5
Label	{CA}	{CA, Aorta, PV}	{CA, Aorta, PV}	{CA, Aorta, PV}	{CA}
Model	BVNet	BVNet	BVNet3D	SegCaps	BVNet
Channels	{1,3,5,7,9}	{5}	{1}	{1}	{5, 10}
Stride	{1,2,5}	{1}	{1}	{1}	{1}
Batch size	4	8	8	1	8
lr	0.001	0.0001	0.0001	0.0001	0.0001
lr scheduler	-	✓	✓	✓	✓
Split	{0}	{0,1,2}	{0,1,2}	{0,1,2}	{0,1,2}
Input	{CT}	{CT}	{CT}	{CT}	{Frangi, CT & Frangi}
Metric	{dice score, recall, precision}	{dice score, recall, precision}	{dice score, recall, precision}	{dice score, recall, precision}	{dice score, recall, precision}

Table 3.5: A summary of the different elements in each of the experiments, where CA = coronary artery, PV = portal vein, lr = learning rate, and Frangi = Frangi filtered image.

3.4.1 Experiment 1: Channels and Stride

In this experiment is the BVNet model from subsection 2.3.2 used with transpose convolution for upsampling. The images are divided on the z-axis into slices before they are used as input on the 2D model. The BVNet model's input shape is *(512, 125, c)* where c is the number of channels. The channel size needs to be an odd number because the model only trains on the middle channel. The channels are put together by adding the x previous and x next slices. The x value is calculated by $\lfloor \frac{c}{2} \rfloor$. The stride value is how many steps to move from the given value when adding the previous and next channels. If stride is two, every other slice before and after the given slice is added.

This experiment consists of experimenting with a different number of channels and stride of the BVNet model. The goal of this experiment is to check if it is possible to improve the BVNet model's understanding of the predicted slice surroundings to answer RQ1.

The experimenting consists of using the channel size of $\{1, 3, 5, 7, 9\}$ and the different stride of $\{1, 2, 5\}$.

3.4.2 Experiment 2: BVNet

In this experiment, the BVNet model with the channel size and stride of the best mean $Dice_{raw}$ with a relatively low computational cost model form experiment 1 used, which was 5 channels and stride 1. The BVNet model will be tested on the coronary artery, Aorta, and portal vein. This experiment gathers information in order to answer RQ2. For being able to answer RQ2, the advantages and disadvantages of BVNet need to be found. This is done by checking how good the model is at finding large tubular structures like the Aorta, and how good it is on a smaller dataset that is not from st.Olav containing the portal vein. The reason for using the portal vein is to check how good the model is at finding arteries/vessels in general.

3.4.3 Experiment 3: BVNet3D

The BVNet3D model replaces all the 2D operations of the BVNet model with 3D operations, similar to the transformation from U-Net to 3D U-Net. This experiment is created for answering RQ3 by finding the advantages and disadvantages of BVNet3D used for segmenting the coronary artery, Aorta, and portal vein. For using BVNet3D, the images need to be divided into patches before they can be used as input because of the computational cost. For this experiment, the patches have the size $(64, 64, 64)$, where the image is padded with zero on the axese not dividable on 64.

3.4.4 Experiment 4: SegCaps

This experiment gathers information about the SegCaps in order to answer RQ4, which consists of finding the benefits and disadvantages of the SegCaps model when used for segmenting the coronary artery, Aorta, and portal vein. The SegCaps model uses CapsuleNet in an encoder-decoder architecture for segmentation as described in subsection 2.3.3. In this experiment, dynamic routing is used to decide which of the capsules to activate in the next layer with three iterations.

3.4.5 Experiment 5: Frangi Filter

This experiment is created to check if the Frangi filter can help increase the dice score of the BVNet model when segmenting the coronary artery, which is RQ5. This experiment consists of two parts; namely using Frangi filtered images as input to the BVNet model and using a combination of both CT and Frangi filtered images as input to the BVNet model. The Frangi filter is explained in subsection 2.2.6 and is created by the VTMK with α 0.3, β 500, and γ 100.

The first part uses Frangi filtered images as input to the BVNet model with channel size 5 and stride 1.

The last part uses both CT and Frangi filtered CT images as input to the BVNet model with 10 channels and stride 1. The five first channels consist of CT slices, and the rest is the result of applying the Frangi filter on these slices as shown in Figure 3.1.

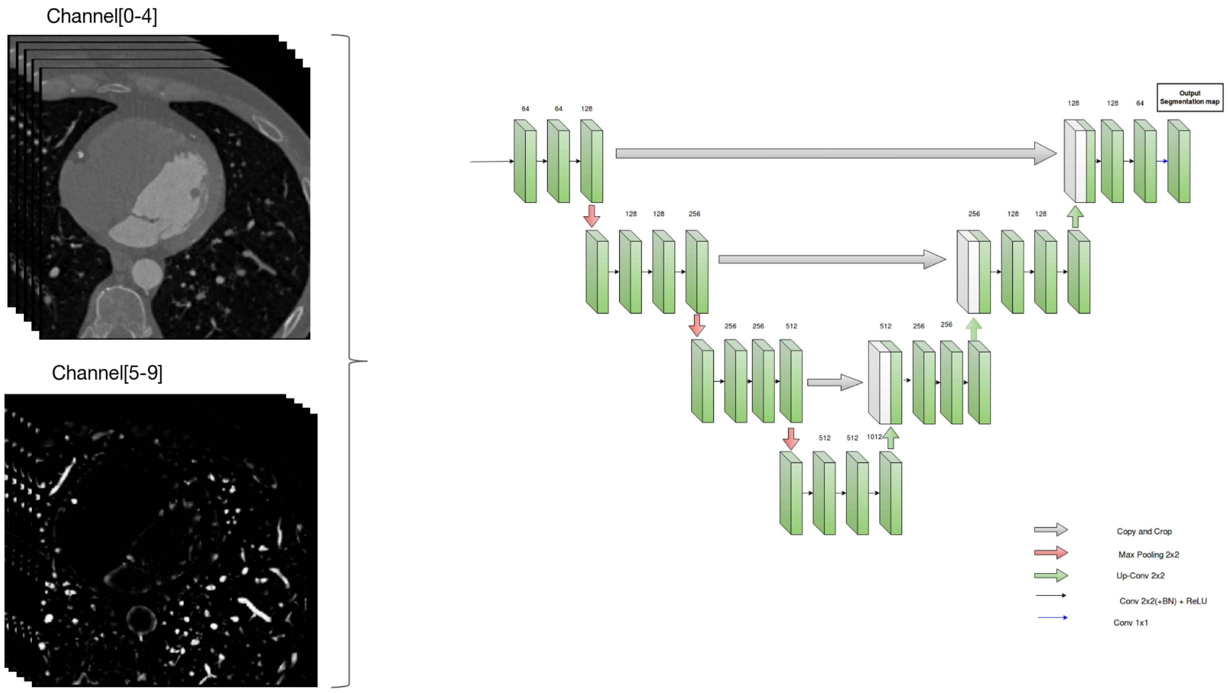


Figure 3.1: A visualization of combined CT and Frangi as input for the BVNet model

Chapter 4

Result

This chapter gives an overview of the different result from the experiments explained in section 3.4. The visualizations are color-coded prediction with blue for TP, green for FP, and red with low opacity for FN.

4.1 Experiment 1: Channels and Stride

In this experiment, the goal is to gather enough information to answer the first research question (RQ1). It entails finding the optimal number of channels and stride of the step size of the next slice, to be added as a channel on the BVNet model described in subsection 3.4.1. Table 4.1 shows the result of the mean Dice$_{raw}$ score of the models with different channels and stride. The main focus in this experiment is going to be on the Dice$_{raw}$ because the Dice$_{post}$ has similar to the Dice$_{raw}$. The mean Dice$_{raw}$ and Dice$_{post}$ is gathered from the tables in appendix A, which contains the result of each model in the form of dice score, recall, and precision for each raw and post processed test image.

Table 4.1 shows that the Dice$_{raw}$ is in the interval $(0.324, 0.593)$. From the tables in appendix A is it possible to observe a small increment in the mean Dice$_{raw}$ score with five or more channels. Another observation is that the mean precision is higher than the mean recall in most cases.

Next, are the tables in appendix A showing that an increment of the channel size increases the dice score in general, but the minimum dice score is still relatively low. The stride of 2 seems to entail a minor increase in the mean Dice$_{raw}$ score for 3 and 7 channels, but the stride of 5 slightly decreases the mean dice score. The best mean Dice$_{r}aw$ is BVNet with 5 channels and stride 1.

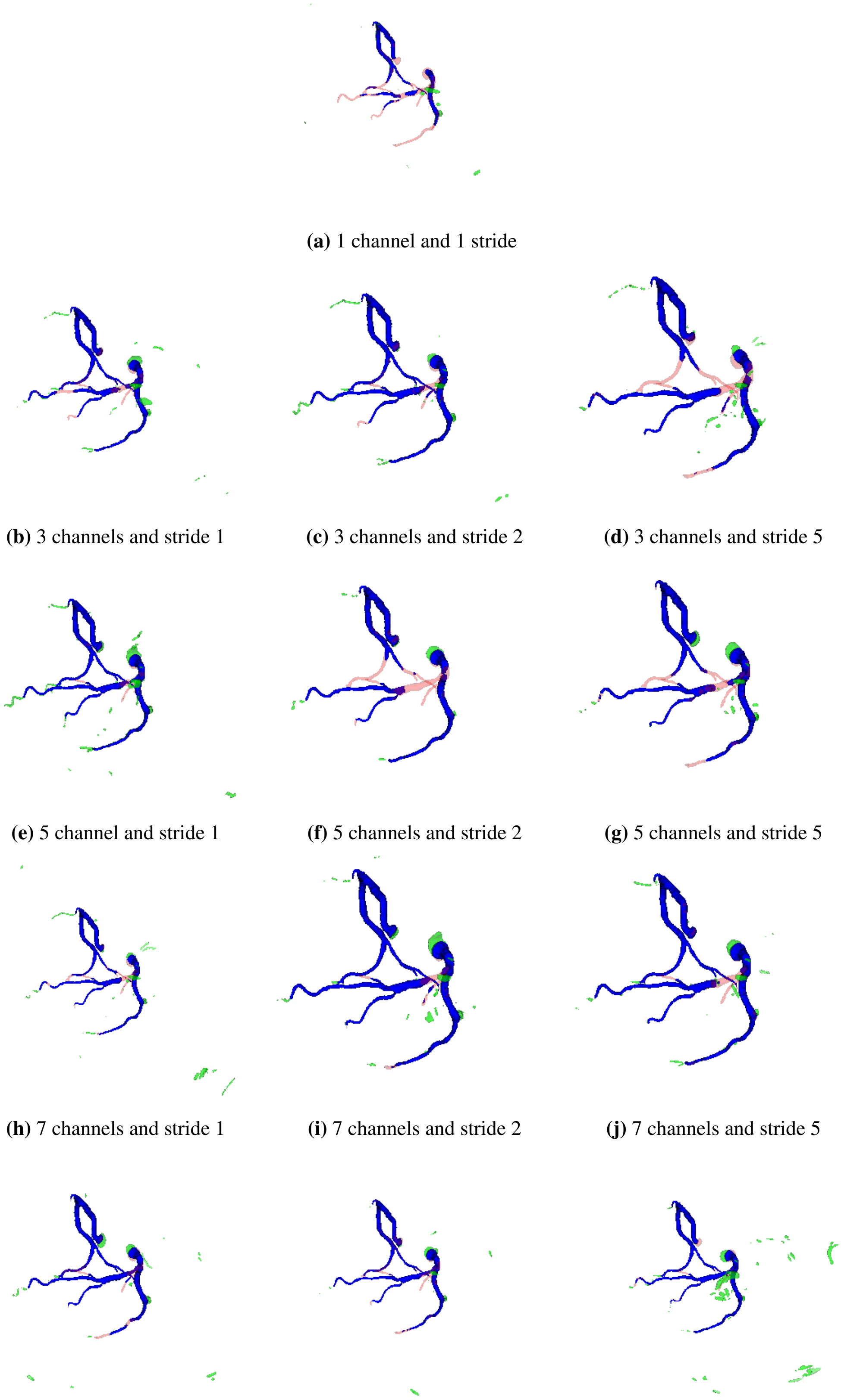
	1-Channel		3-Channels		5-Channels		7-Channels		9-Channels	
Stride	Dice$_{\text{raw}}$	Dice$_{\text{post}}$	Dice$_{\text{raw}}$	Dice$_{\text{post}}$	Dice$_{\text{raw}}$	Dice$_{\text{post}}$	Dice$_{\text{raw}}$	Dice$_{\text{post}}$	Dice$_{\text{raw}}$	Dice$_{\text{post}}$
S-1	0.324	0.304	0.525	0.516	0.593	0.59	0.544	0.538	0.547	0.54
S-2	-	-	0.553	0.543	0.555	0.541	0.59	0.59	0.471	0.449
S-5	-	-	0.51	0.504	0.563	0.551	0.583	0.577	0.494	0.498

Table 4.1: Average dice score from different channels and stride combinations.

In Figure 4.1 is the best predicted image of the raw test images from the different models, which is the same image for all the different channels and stride. Prediction with 9 channels is the only one that is fully connected on the left main branch of the LM. All of the predictions are almost not finding any of the thin arteries in the middle, but 3 channels with stride 2 and 5 and 7 channels with stride 2 did find some pixels of it.

The prediction with the worst Dice$_{raw}$ score from the different models are shown in Figure 4.2, which has more FP and FN than the predictions in Figure 4.1. In Figure 4.2 are all the models having the same worst prediction test image, except for the model with channel 1 and 9 channels with stride 2. These two predictions contain almost none TP and FP. The rest of the predictions have a lot of FP, where some contain large vessel/artery like structures. An observation is that these predictions detect many small connected components of TP on the LM.

Another observation is that all of these models have a problem with predicting the connection to Aorta. The model with 7 channels and stride 1 stands out as being the only one finding some of the parts closest to Aorta in RCA.



(a) 1 channel and 1 stride

(b) 3 channels and stride 1

(c) 3 channels and stride 2

(d) 3 channels and stride 5

(e) 5 channel and stride 1

(f) 5 channels and stride 2

(g) 5 channels and stride 5

(h) 7 channels and stride 1

(i) 7 channels and stride 2

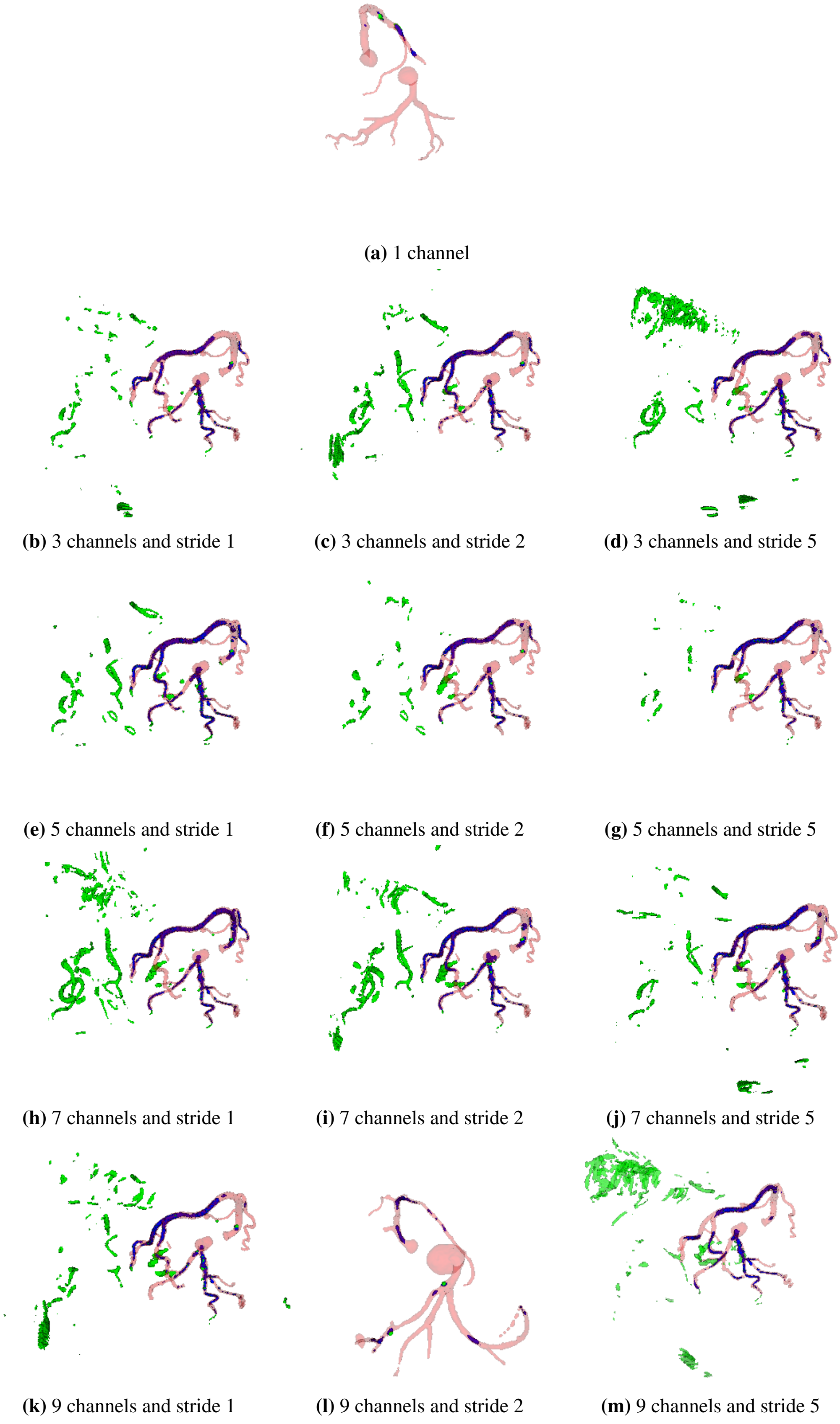
(j) 7 channels and stride 5

(k) 9 channels and stride 1

(l) 9 channels and stride 2

(m) 9 channels and stride 5

Figure 4.1: A visualization of prediction with the best raw dice score for the BVNet model with all the different channels and stride, where the prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN. Some of the subfigure are captured at different levels of zoom.



(a) 1 channel

(b) 3 channels and stride 1

(c) 3 channels and stride 2

(d) 3 channels and stride 5

(e) 5 channels and stride 1

(f) 5 channels and stride 2

(g) 5 channels and stride 5

(h) 7 channels and stride 1

(i) 7 channels and stride 2

(j) 7 channels and stride 5

(k) 9 channels and stride 1

(l) 9 channels and stride 2

(m) 9 channels and stride 5

Figure 4.2: A visualization of prediction with the worst raw dice score for the BVNet model with all the different channels and stride, which is not on the same input image for all of the models. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN. Some of the subfigure are captured at different levels of zoom.

4.2 Experiment 2: BVNet

For this experiment, the BVNet will be tested on three different splits for the coronary artery, Aorta, and the portal vein as described in subsection 3.4.2.

4.2.1 Coronary Artery

Table 4.2 contains the different dice scores from the dataset splits, where all the splits have a mean over 0.5, which is quite high. Another observation is that all the splits have a similar maximum Dice$_{raw}$ score, but there are some large differences between the splits, such as split-0 having the smallest minimum Dice$_{raw}$ of 0.233 and split-2 having the largest minimum Dice$_{raw}$ of 0.547. In section B.1 is the tables of the dice score, recall, and precision for each test image for all of the splits, which shows that the precision is in most cases higher than the recall.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.233	0.223	0.445	0.444	0.547	0.548
Max	0.774	0.795	0.781	0.784	0.803	0.81
Mean	0.54	0.545	0.622	0.642	0.686	0.694

Table 4.2: The dice score of the different splits of the coronary artery using the BVNet model.

Figure 4.3 shows the best Dice$_{raw}$ prediction of the splits and the post processed version. The split-0 prediction is unconnected in the left branch in the LM, where it branches into the two main branches. The FP in this prediction has some quite big FP without the tubular structure and some other making the smaller arteries longer. In this case, post processing helps split-0 by removing some of the FP.

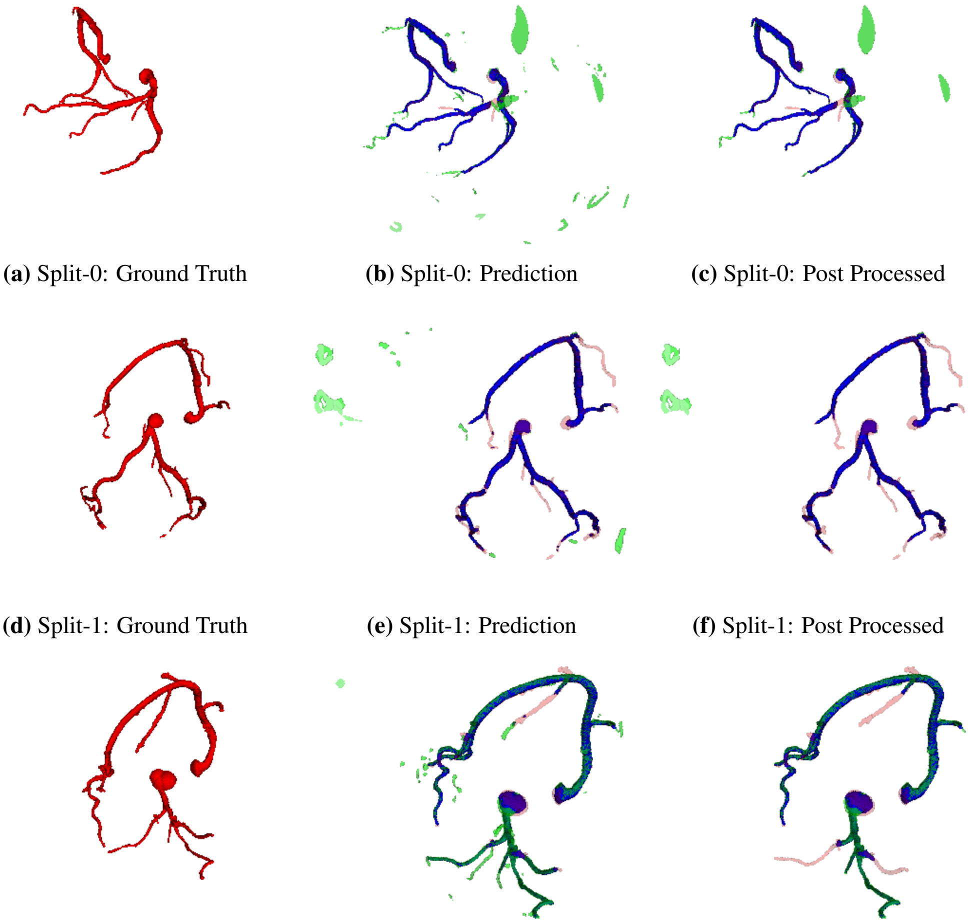
Similar to split-0, the post processing helps split-1 by removing most of the FP not connected to the RCA or the LM. In split-1 most of the ground truth in the main branches of the RCA and the LM is found. There is a small gap in the curved area in the left main branch of LM. In this prediction is almost none of the smaller arteries found.

The best prediction for split-2 distinguishes from the best predictions from the other two split with having few FP around RCA and LM. Instead, some of the FP is connected to the TP, which in some cases extending the branch length.

Figure 4.4 shows the worst raw prediction of the splits and that prediction after the post processing step. It shows that none of the split has predicted the connection point to aorta correctly. Another thing the splits have in common is that they have a lot of gap between the TP, which makes some of the TP connected components very small. The split-1 has a lot more FP than the other two. The FP does not have a tubular structure and is not connected to either the RCA or the LM. The post processing step in split-1 is both removing some FP and a lot of the TP in LM. The split-1 has almost none of the TP in the right branch of LM and the RCA has large gaps. The split-2 consists of large TP connected components. It has some large gaps in the connects part of the branches in the LM and FN, where the thin branches curves. The split-2 is the split with the least amount of FP.

In the CT slices of the best and worst raw prediction shown in Figure 4.5 are the worst CT slices darker than the best. The FN is in the darker part. The slice $237/316$ in the worst prediction have some FP at the bones. The CT slices in section B.1 have, in most cases, low contrast where the FN is located, and there is some intensity difference between the different test images.

The graphs from Figure 4.6 show a similar evolution for the splits. There is a large difference between the training and validation values for the dice score and the loss. The validation values contain many spikes, and the training values has a stable evaluation for both the dice score and loss. The training loss is smallest for split-2, with values around 0.1. The split-2 trains for almost 15 epochs longer than split-0, which has the largest loss. Split-0 flattens around 0.35 and split-1 flattens around 0.3. The validation loss for split-1 is the one that is closest to the training loss with a difference around 0.25. The dice score of split-0 is almost the same as the validation dice score for split-2, except for some spikes. The split-0 validation loss has a large variation in the spikes.

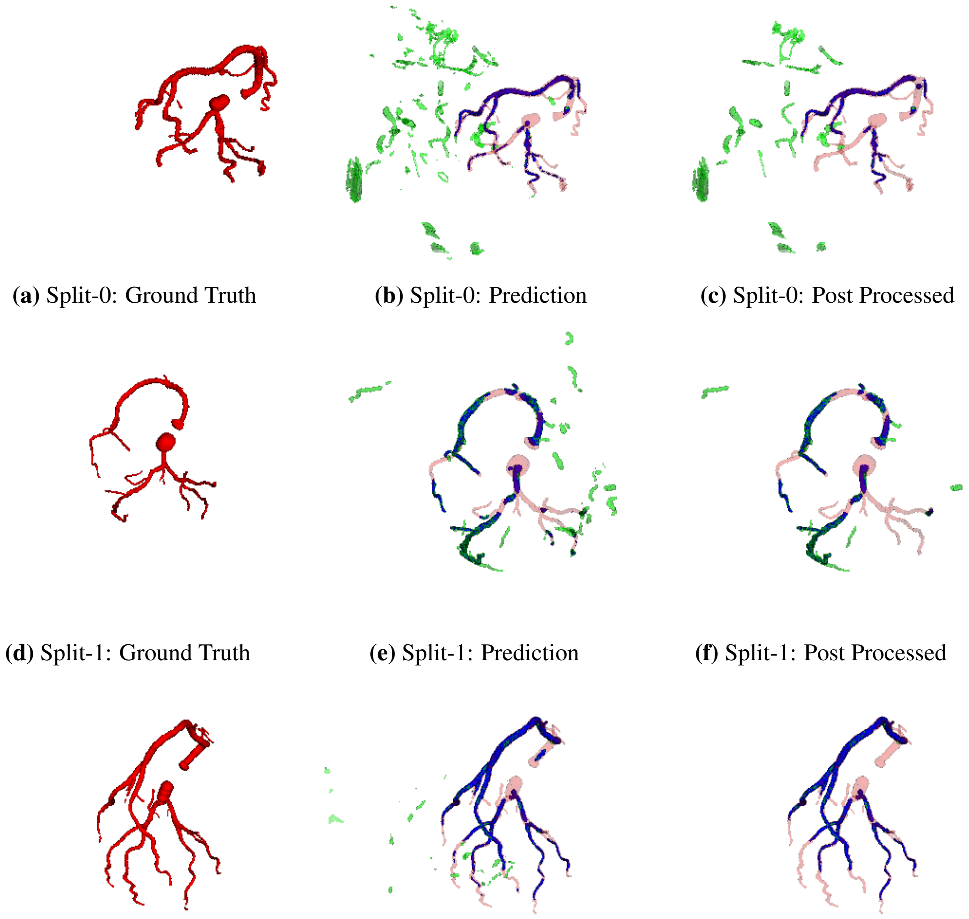


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.3: A visualization of the prediction of the coronary artery with the best raw dice score for the BVNet model with 5 channels and stride 1 where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Split-0: Ground Truth

(b) Split-0: Prediction

(c) Split-0: Post Processed

(d) Split-1: Ground Truth

(e) Split-1: Prediction

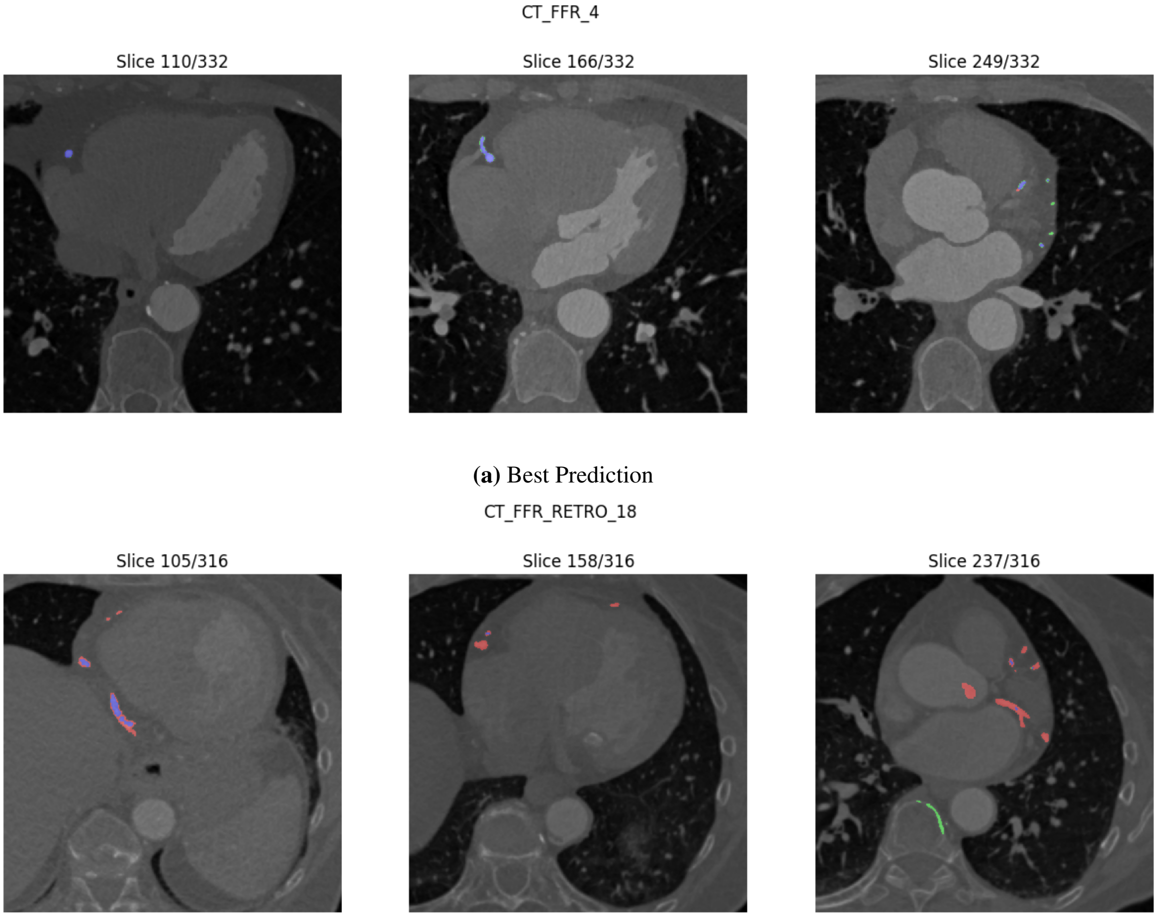
(f) Split-1: Post Processed

(g) Split-2: Ground Truth

(h) Split-2: Prediction

(i) Split-2: Post Processed

Figure 4.4: A visualization of the prediction of the coronary artery with the worst raw dice score for the BVNet model with 5 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Best Prediction

(b) Worst Prediction

Figure 4.5: A selection of some of the CT slices of the BVNet model's best and worst prediction of the coronary artery splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

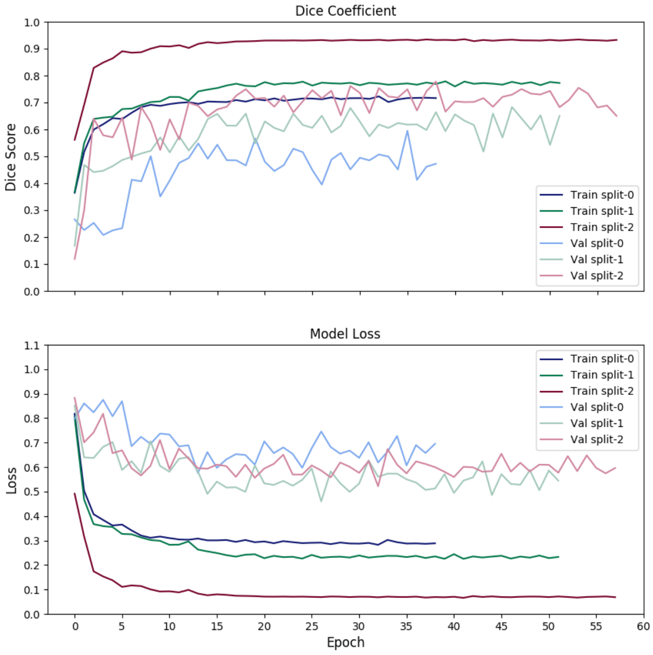


Figure 4.6: A graphs of the dice score and loss over the epochs for the different splits on the BVNet model with the coronary artery label.

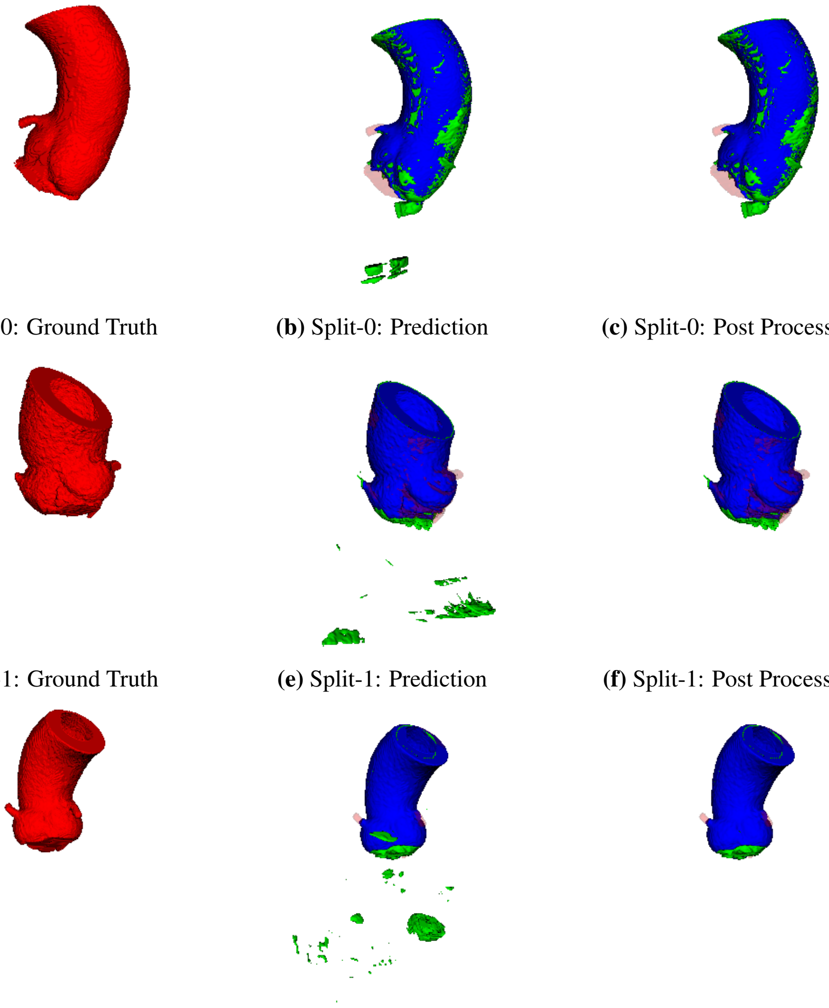
4.2.2 Aorta

In the Table 4.3 are the different dice scores of the splits, which are very high. There are not huge differences between the minimum and maximum Dice$_{raw}$ score, except for the split-0, where the difference is 0.277.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.682	0.728	0.798	0.8	0.91	0.923
Max	0.959	0.963	0.933	0.939	0.969	0.985
Mean	0.909	0.919	0.885	0.926	0.944	0.959

Table 4.3: The dice score of the different splits of the Aorta using the BVNet model.

The visualization of the best raw prediction is shown in Figure 4.7. The FP in split-1 and split-2 have some connected components around the Aorta, which are removed by the post processing step. All the predictions have FP in the bottom part of the Aorta and FN, where it branches into RCA and LM. The split-0 prediction has some FP in the back part as well.



(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

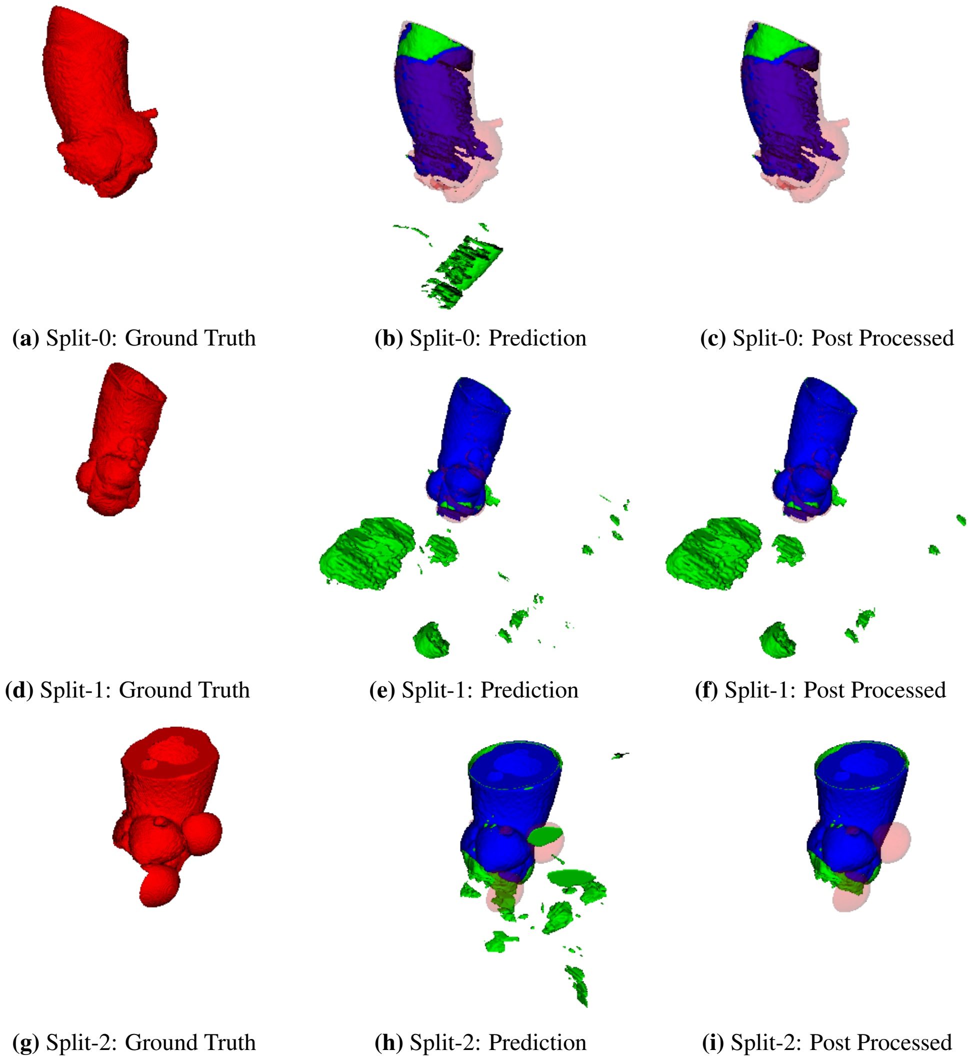
(g) Split-1: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.7: A visualization of the prediction of the Aorta with the best raw dice score for the BVNet model with 5 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

The worst raw prediction is shown in Figure 4.8. The split-0 prediction is missing the bottom part and has some FP connected to the TP and around the Aorta, forming a tubular structure.

In split-1 there is almost none FN, but it has some large FP predictions around the Aorta, which are removed by the post processing.

The ground truth of split-2 distinguishes from the rest of the ground truth by containing some clumps connected to the Aorta, which was predicted to be FN. Most of the FP is connected to the bottom part of the Aorta. The post processing step is removing the FP not connected to the Aorta.


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

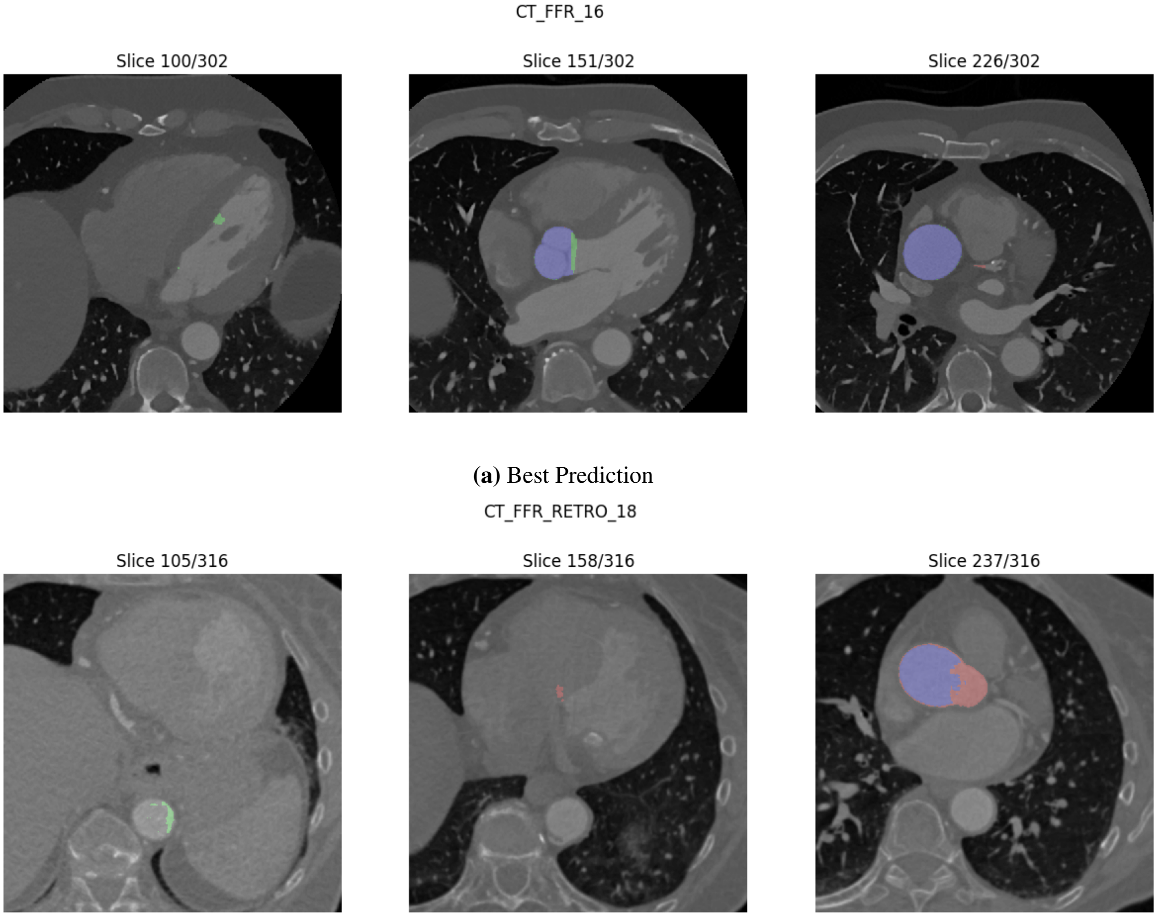
(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.8: A visualization of the prediction of the Aorta with the worst raw dice score for the BVNet model, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

The CT slices in Figure 4.9 show the best and worst raw prediction on these slices. The FP in Figure 4.9a) is either in the slices not containing any ground truth with similar intensity or as a part of the correct predicted Aorta. The Figure 4.9 b) shows that the FP is on another circular shape at the CT slice, which looks like a large vessel close to the spine. The ground truth from section B.2 shows that slices, where the Aorta has a more circular shape, have less FN.

Evaluation of the dice score and loss is similar for the different splits, as shown in Figure 4.10. All the training losses increase and flatten out to around 0.1 already in epoch 15. The validation loss contains small spikes after epoch 15, which is around the value 0.6. Split-1 is training longer than the other splits. The validation dice score of split-1 is the only one that distinguishes from the rest by having greater spikes in the interval of values around $(0.5, 0.7)$.



(a) Best Prediction

(b) Worst Prediction

Figure 4.9: A selection of some of the CT slices of the BVNet model's best and worst prediction of the Aorta splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

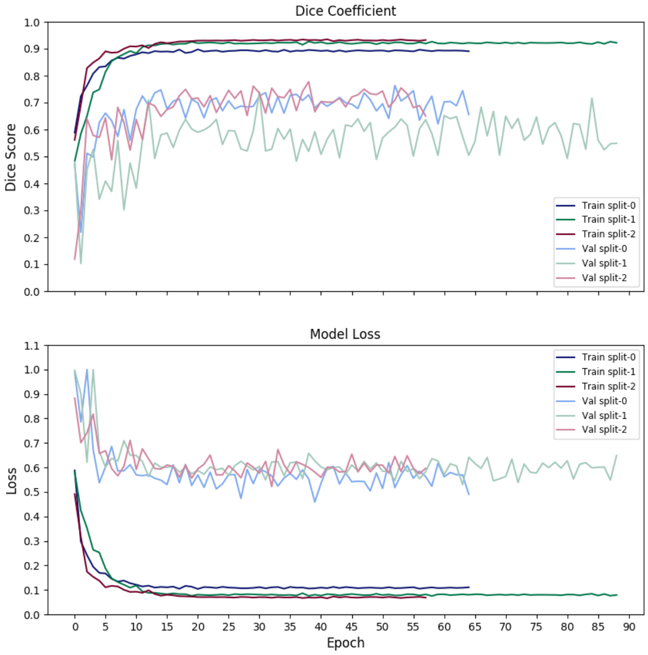


Figure 4.10: A graphs of the dice score and loss over the epochs for the different splits of Aorta using the BVNet model.

4.2.3 Portal Vein

In the Table 4.4 is a summary of the dice score of the portal vein, where split-2 has a very similar minimum and maximum dice score. The split-1 differences from the other two by having a difference between the minimum and maximum of 0.256. All the splits have a relatively low dice score on the two test images from the different splits. The recall and precision results have a lot of variation between each test image, as shown in section B.3, where the recall in most cases is higher than the precision.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.244	0.261	0.141	0.128	0.356	0.371
Max	0.323	0.332	0.406	0.423	0.362	0.368
Mean	0.284	0.296	0.274	0.275	0.359	0.369

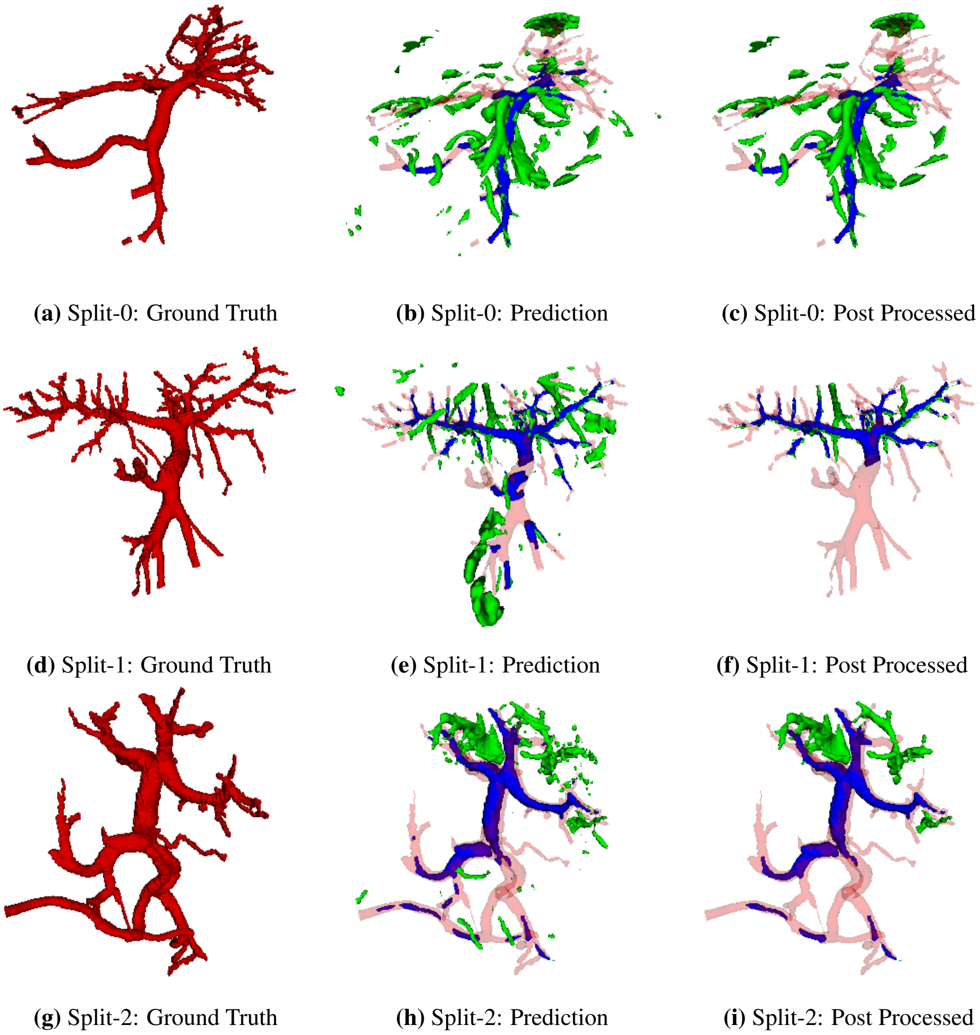
Table 4.4: The dice score of the different splits of the portal vein using the BVNet model.

A visualization of the best raw prediction of the two test images is shown in Figure 4.11. The predictions of split-1 and split-2 are most of the thickest part of the portal vein found. In these splits are many of the FP formed as vessels or clumps. The post processing step does not remove these NP because of there size. The majority of the FN is in the smaller veins. The prediction of split-2 also contains a lot of TP in the bottom part and FP, mostly located in the top area.

In the prediction of split-1 is almost all of the ground truth found in the upper part. The prediction also consists of some large clump formed FP in the bottom part and some around the smaller veins in the top part. Most of these FPs are removed in the post processing step together with the TP in the bottom part.

In Figure 4.12 are the worst raw predictions of the portal vein splits. The split-0 prediction has few TP and a lot of FP, which is located in the top part. There are few TP and FP in the split-0 prediction, and the existing TP are in the smaller veins. The prediction of split-2 differences from the others splits by founding a lot of the ground truth. Most of the FP is located close to the TP veins, and the FN is mainly located in the smaller veins endpoint.

The Figure 4.13 shows the best and worst prediction on the different splits of the portal vein on the corresponding CT slice. Both the best and worst do not find the part of the portal vein inside the liver. The worst is predicting some of the other vessels instead, which is also the case in most of the other predictions in the different splits on the corresponding CT slice, shown in section B.3.

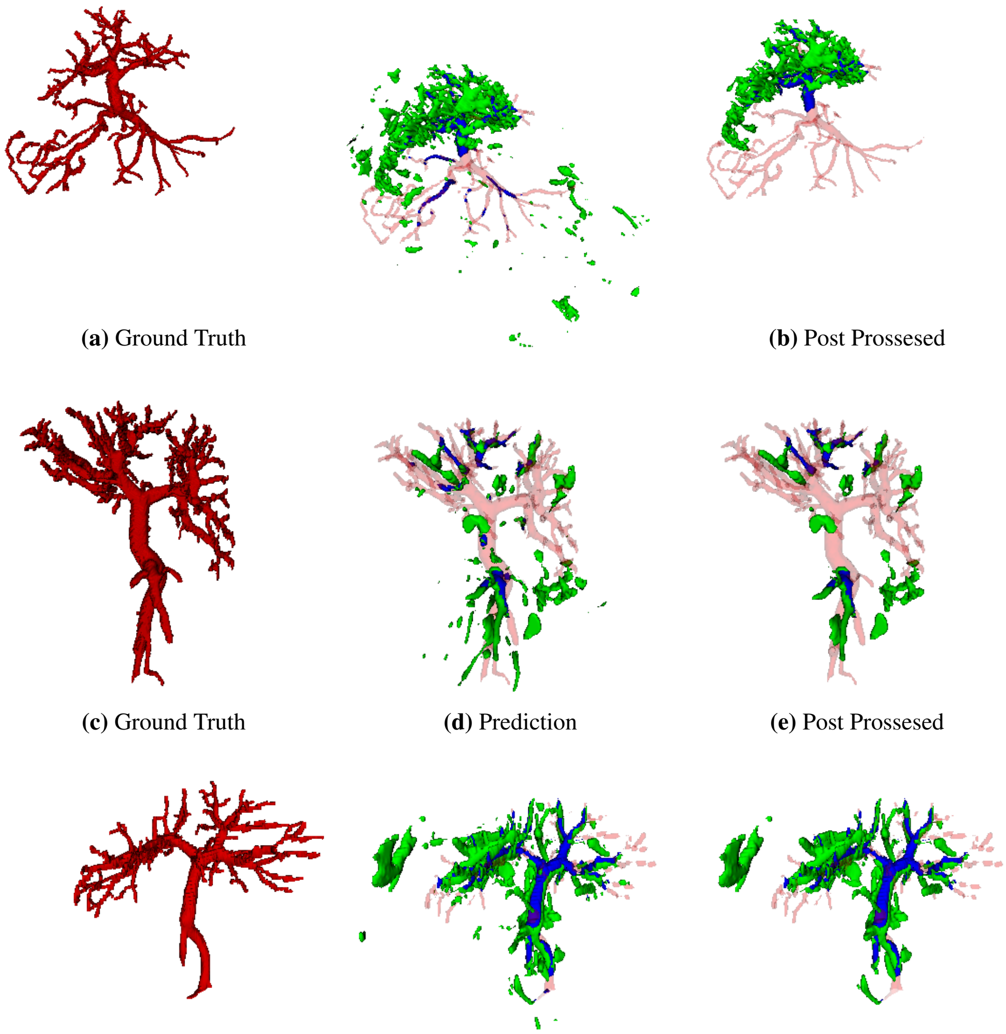


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.11: A visualization of the prediction of the portal vein with the best raw dice score for the BVNet model with 5 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Ground Truth

(b) Post Prossesed

(c) Ground Truth

(d) Prediction

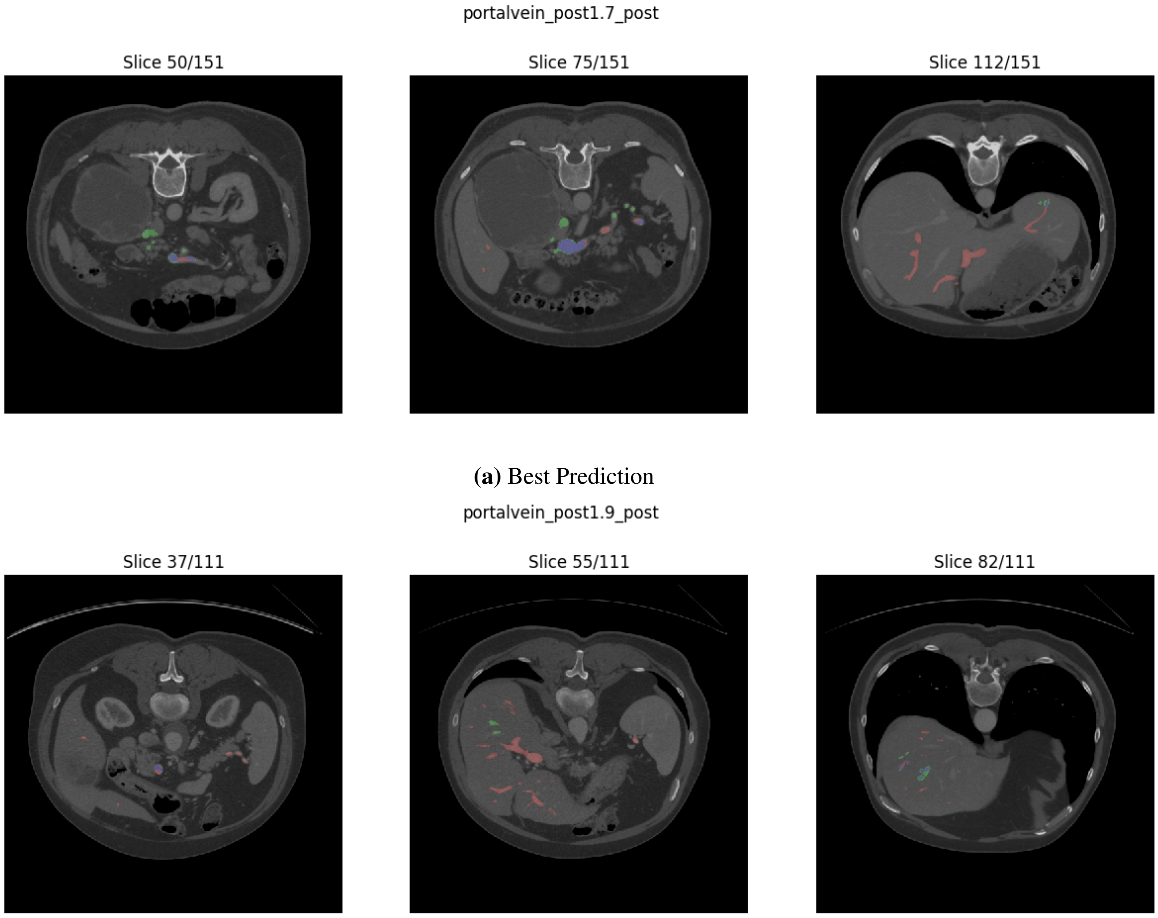
(e) Post Prossesed

(f) Ground Truth

(g) Prediction

(h) Post Prossesed

Figure 4.12: A visualization of the prediction of the portal vein with the worst raw dice score for the BVNet model with 5 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Best Prediction

(b) Worst Prediction

Figure 4.13: A selection of some of the CT slices of the BVNet model's best and worst prediction of the portal vein splits, where the corresponding Prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

The training performance is shown in Figure 4.14. There is a gap between the training and validation results in both the dice score graphs and the loss graphs. In the dice score graphs is the evaluation similar for all the splits, except the validation of split-1, which contains some large spikes in the end. The training loss has a stable decrement that flattens out around 0.55, and the validation loss for split-0 and split-2 is quite unstable with large spikes. The validation loss for split-1 is different with having values around 0.95 with some minor spikes resulting it to training longer than the others.

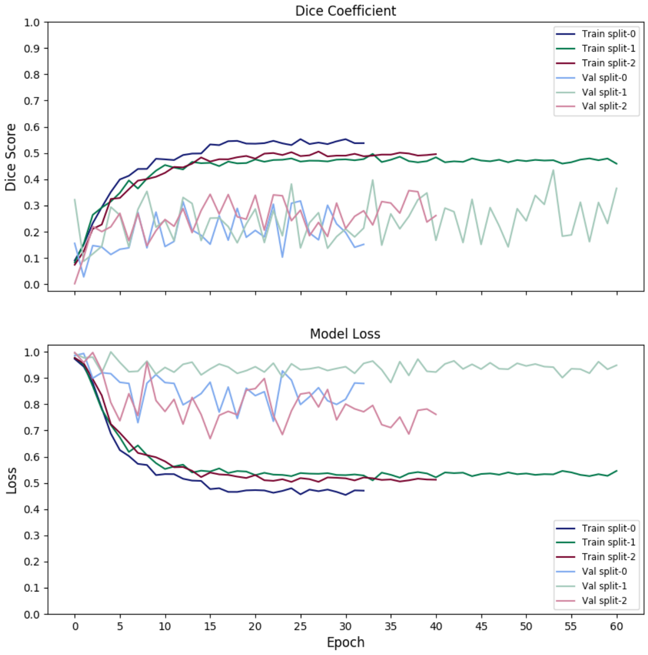


Figure 4.14: A graphs of the dice score and loss over the epoches for the different splits of portal vein on the BVNet model.

4.3 Experiment 3: BVNet3D

In experiment 3, the BVNet3D model is trained on the coronary artery, Aorta, and portal vein in order to find the advantages and disadvantages of the model described in subsection 3.4.3.

4.3.1 Coronary Artery

In Table 4.3 are the different dice scores of the different splits. There is a great variation in the minimum and maximum Dice$_{raw}$ score for each of the splits. The split-0 and split-2 have a difference between 0.228 in the minimum value, where split-2 has the greatest minimum value. The recall and precision values found in section C.1 indicate that the model is finding more of the TP with the cost of having more FP because the recall is higher than the precision.

	Split-0		Split-1		Split-2	
	Dice$_{\mathrm{raw}}$	Dice$_{\mathrm{post}}$	Dice$_{\mathrm{raw}}$	Dice$_{\mathrm{post}}$	Dice$_{\mathrm{raw}}$	Dice$_{\mathrm{post}}$
Min	0.185	0.182	0.302	0.32	0.413	0.448
Max	0.667	0.713	0.73	0.814	0.7	0.723
Mean	0.444	0.472	0.48	0.52	0.519	0.55

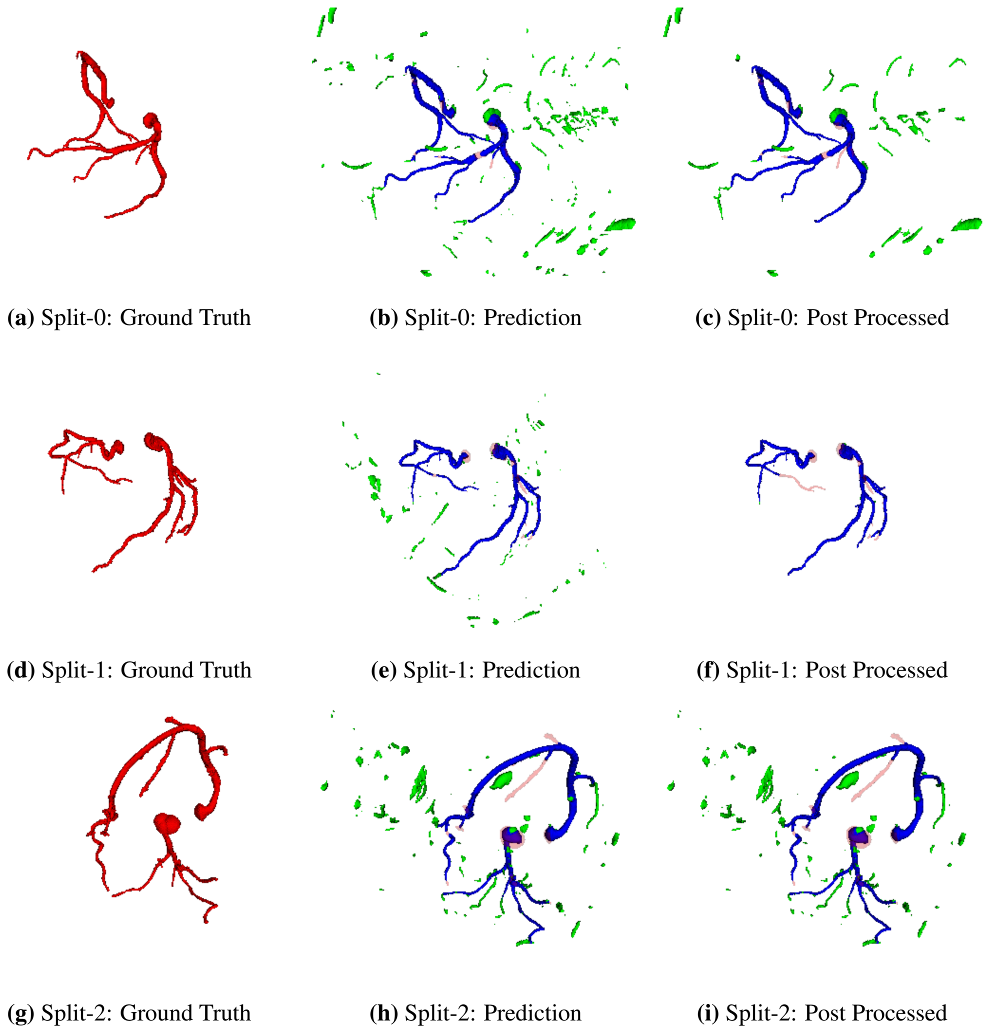
Table 4.5: The dice score of the different splits of the coronary artery using the BVNet3D model.

The best raw predictions of the different splits are shown in Figure 4.15. All the predictions contain few FN, where some are located in the connection point to Aorta and others in the thin branches. A common factor in these predictions is a lot of FP is located around the coronary artery. Some of these FPs are removed in the post processing step.

In Figure 4.16 is the worst raw prediction of the coronary artery from the different splits. In these results are split-0 different from the other splits, because it has few TP. The FP in this prediction has a tubular structure, which makes it look like vessels. In split-1 is some of the FP looking like bones in the form of ribs. It contains few FN, and most of the TP is connected to either RCA or LM. In this case, the RCA is very small with long and thin branches. The split-2 prediction has almost none FN, but some of them are in the area where the coronary artery connects to the Aorta. This prediction contains many FP, which is not connected to RCA or LM without a tubular structure.

In Figure 4.17a) is the best raw prediction on the corresponding CT slices, which contains mostly TP. It has some FP in slice $97/293$ located close to some of the bones. The FP in slice $219/293$ is around the Aorta. In Figure 4.17b) is the worst raw prediction, which has a lot of FN and some FP.

A huge gap between the training and validation loss can be observed in Figure 4.18. The validation loss has values close to 0.95 for all the splits, and the training loss decreases to values between 0.6 and 0.4 for the different splits. The gap between the training and validation dice score is around 0.15, which is moderate. There is a difference in the dice score between split-0 and split-1 at almost 0.2.



(a) Split-0: Ground Truth

(b) Split-0: Prediction

(c) Split-0: Post Processed

(d) Split-1: Ground Truth

(e) Split-1: Prediction

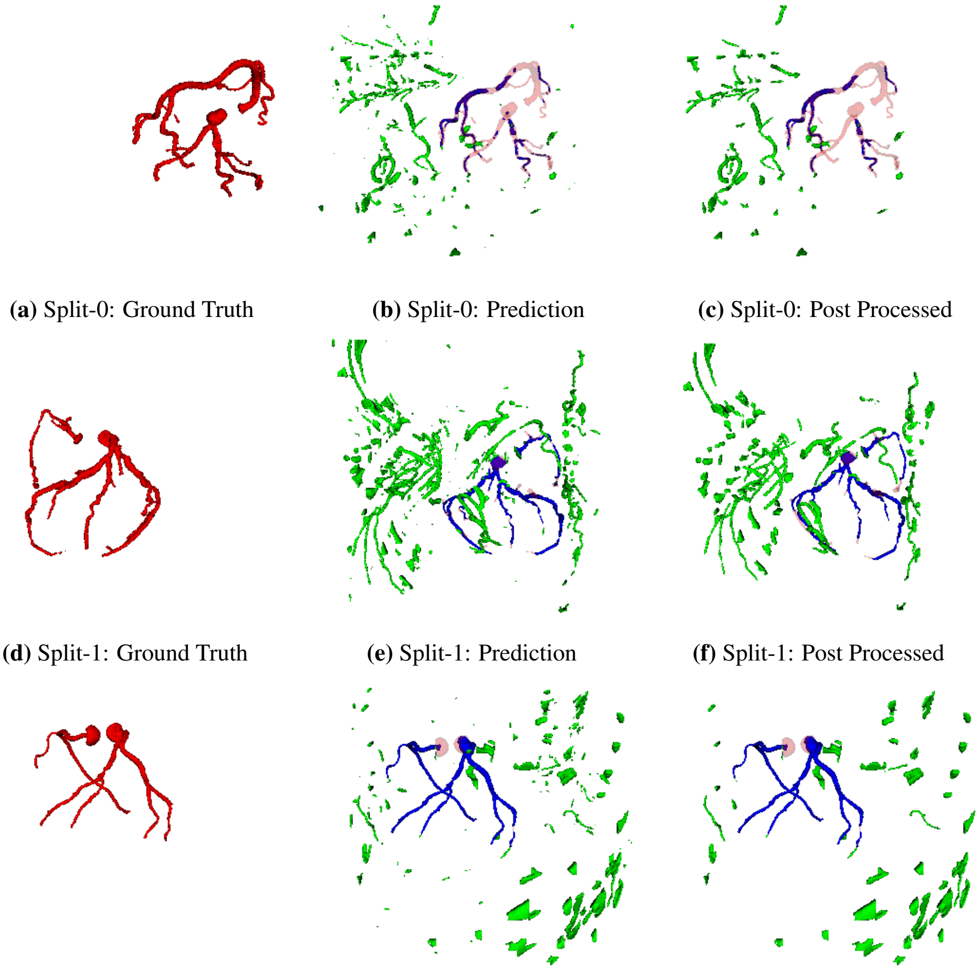
(f) Split-1: Post Processed

(g) Split-2: Ground Truth

(h) Split-2: Prediction

(i) Split-2: Post Processed

Figure 4.15: A visualization of the prediction of the coronary artery with the best raw dice score for the BVNet3D model, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

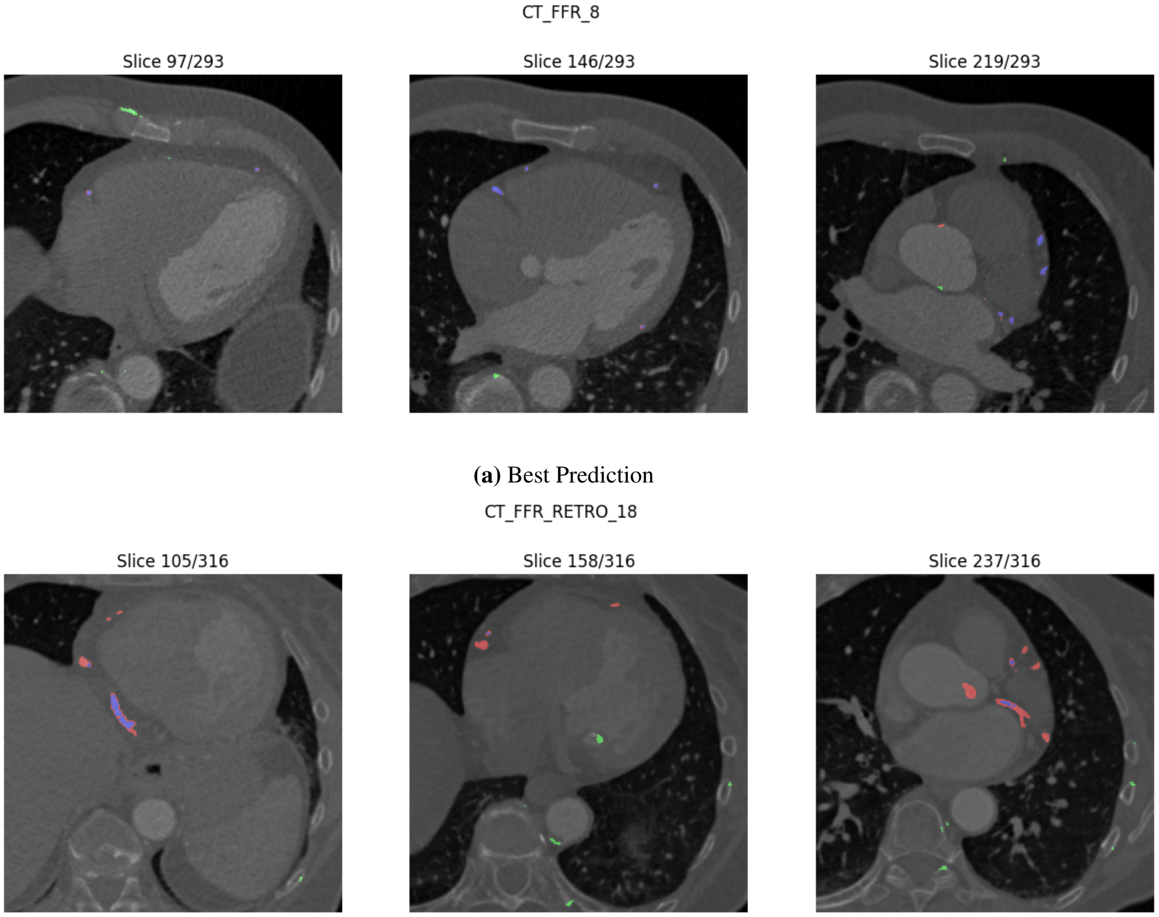


(a) Split-0: Ground Truth (b) Split-0: Prediction (c) Split-0: Post Processed

(d) Split-1: Ground Truth (e) Split-1: Prediction (f) Split-1: Post Processed

(g) Split-2: Ground Truth (h) Split-2: Prediction (i) Split-2: Post Processed

Figure 4.16: A visualization of the prediction of the coronary artery with the worst raw dice score for the BVNet3D model, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Best Prediction

(b) Worst Prediction

Figure 4.17: A selection of some of the CT slices of the BVNet3D model's best and worst prediction of the coronary artery splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

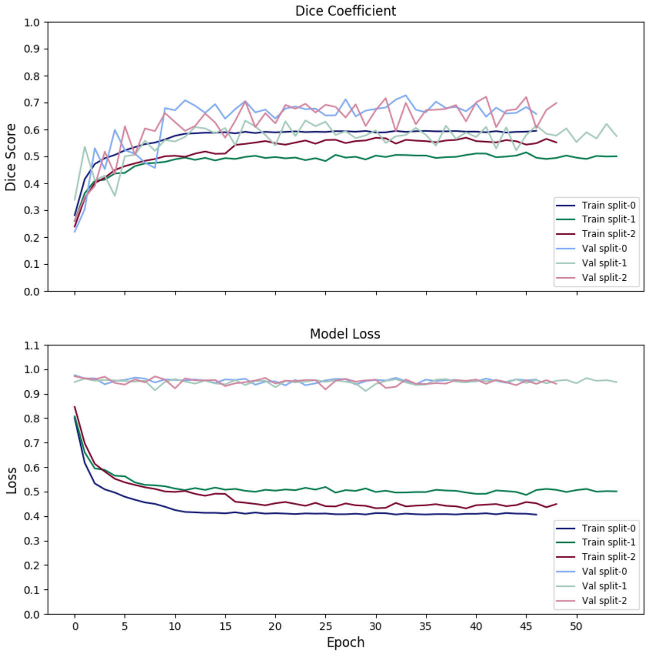


Figure 4.18: A graphs of the dice score and loss over the epochs for the different splits on the BVNet3D model with the coronary artery label.

4.3.2 Aorta

In Table 4.6 is a summary of the test result of the different splits, which shows that all the splits have similar mean values. The split-0 and split-1 have similar minimum dice score values, and the split-1 and split-2 have similar maximum dice score values. The tables in section C.2 contains all the individual test results from the splits, which shows that the majority of the results is closer to the minimum dice score value. This draws down the mean dice score even though it has the greatest maximum value of the splits. The mean recall is high, and the mean precision is low for the splits.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.368	0.369	0.367	0.367	0.44	0.443
Max	0.742	0.781	0.653	0.661	0.659	0.661
Mean	0.492	0.505	0.515	0.521	0.572	0.593

Table 4.6: The dice score of the different splits of the Aorta using the BVNet3D model.

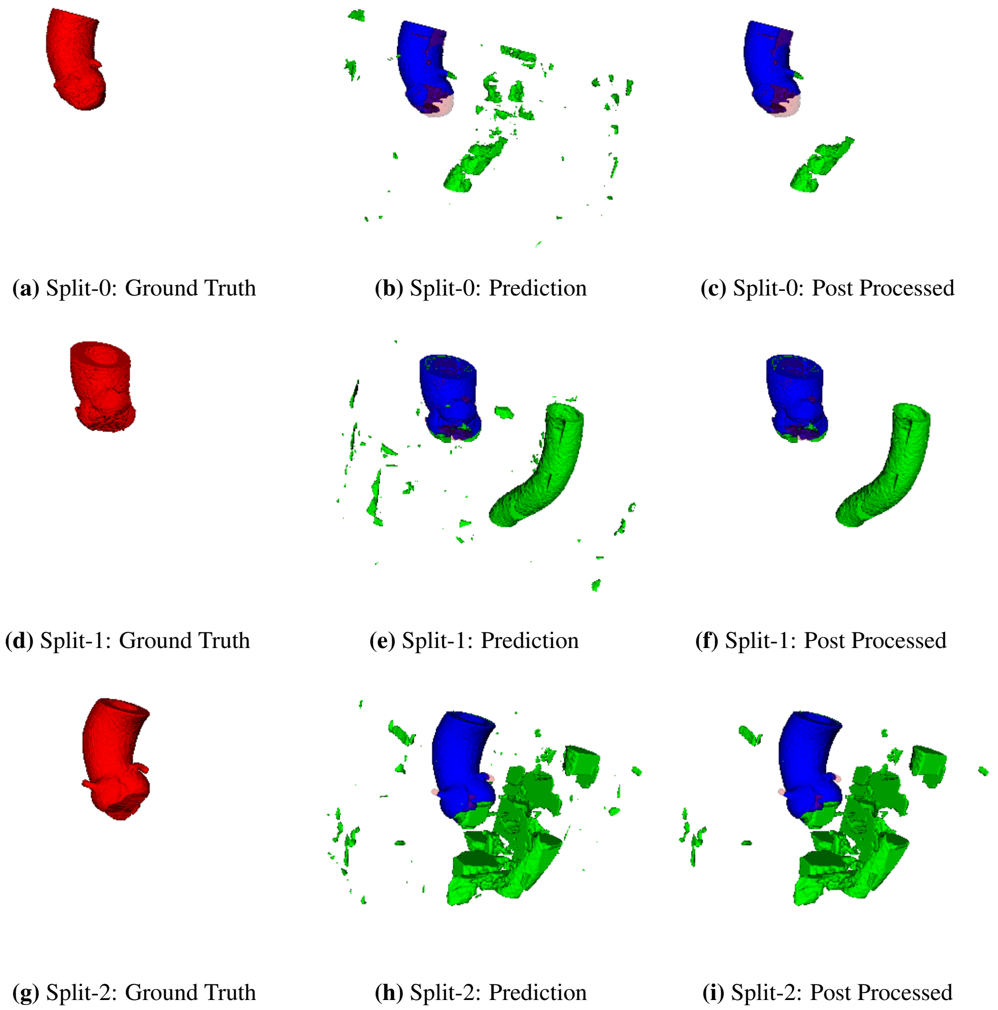
A common factor of the best raw predictions and the worst raw prediction is that they contain a lot of large FP connected components.

In the first row in Figure 4.19 are there many FN in the bottom part and some small FP. One of the FP is forming a large tubular structure with some missing pieces. The second row has almost none FN, and the FP has a large tubular structure shape, which looks like a large artery or vessel. In the last row is the best raw prediction of split-2, which is having a lot of FP with a more cubical structure. The post processing is not making a significant improvement of the best prediction for split-1 and split-2.

Like mention above the worst raw predictions of the different splits contains lots of FP, as shown in Figure 4.20. The Aorta is found in all the worst raw predictions of the splits with some FP extending its structure. The majority of the FP is located around the Aorta in a cubical form with some flat sides.

The worst raw prediction on the corresponding CT slice is shown in Figure 4.17, which have some FP parts consisting of straight lines starting to form a square. The general representation from section C.2 shows that most of the predictions detect some part of the large artery/vessel laying close to the spine and some of the sides are put orthogonal on each other.

The Figure 4.22 shows all the training and validation results of both the dice score and the loss, where the training results have a stable evaluation that flattens out after epoch 15. The decrement in the loss does not seem to affect the validation loss of the splits, which has values around 0.95 during training. The dice score of the validation have some instability in the form of large spikes, but all are increasing and have values around 0.55 after epoch 15. An interesting observation is that the top value of the spikes in the validation dice score of split-2 is very close to the training dice score for the given epoch.

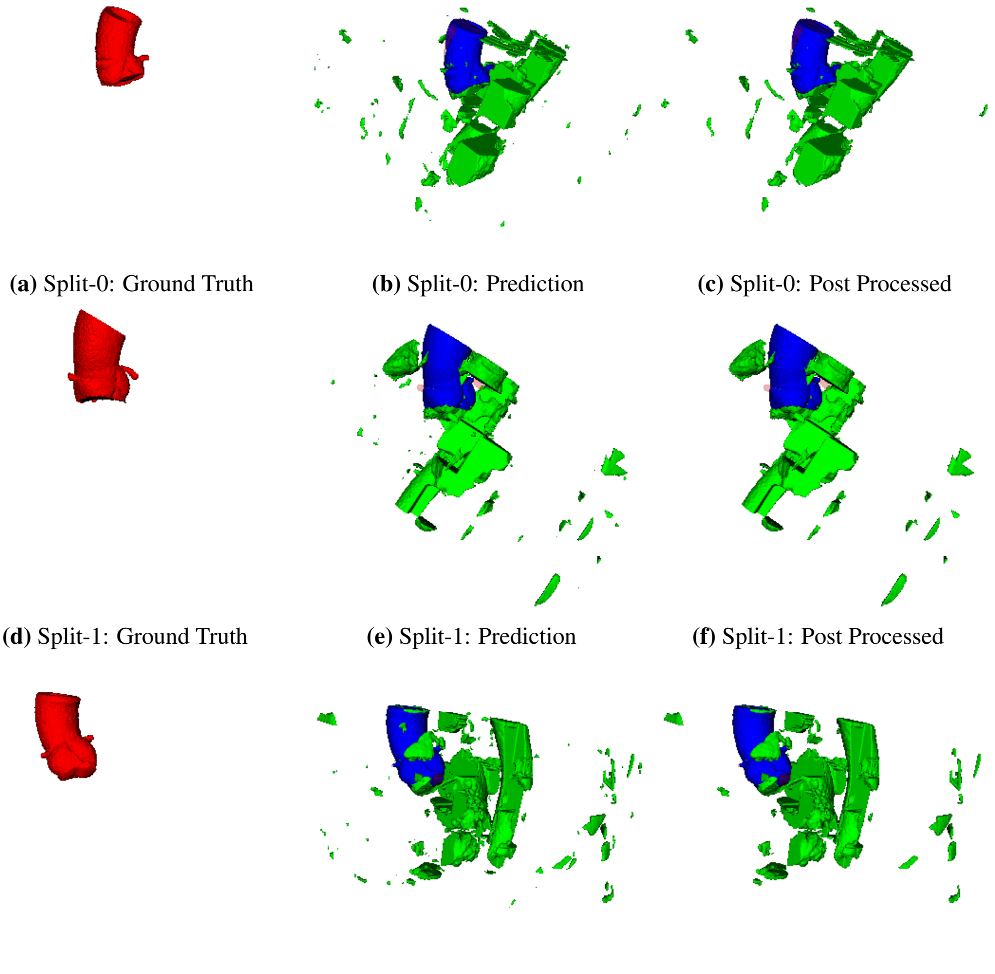


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.19: A visualization of the prediction of the Aorta with the best raw dice score for the BVNet3D model, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

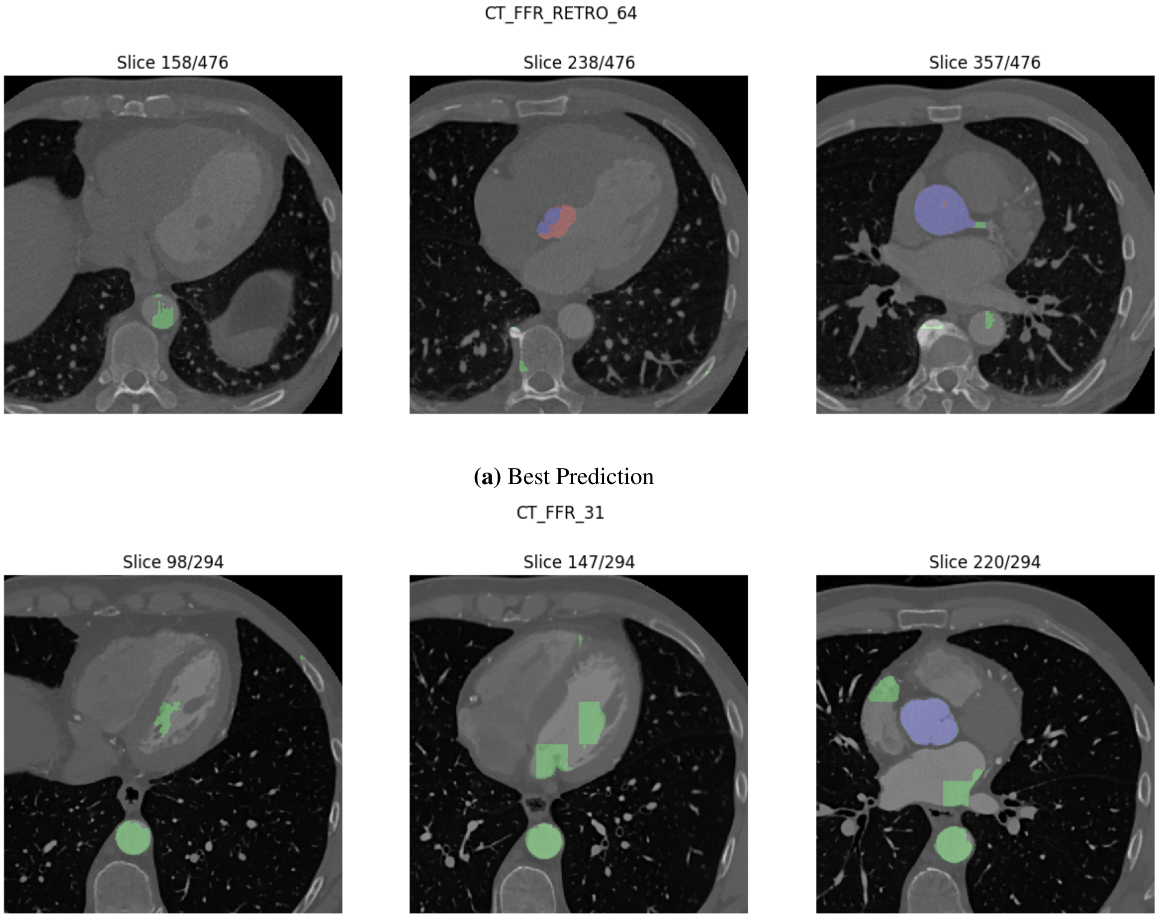


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.20: A visualization of the prediction of the Aorta with the worst raw dice score for the BVNet3D model, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Best Prediction

(b) Worst Prediction

Figure 4.21: A selection of some of the CT slices of the BVNet3D model's best and worst prediction of the Aorta splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

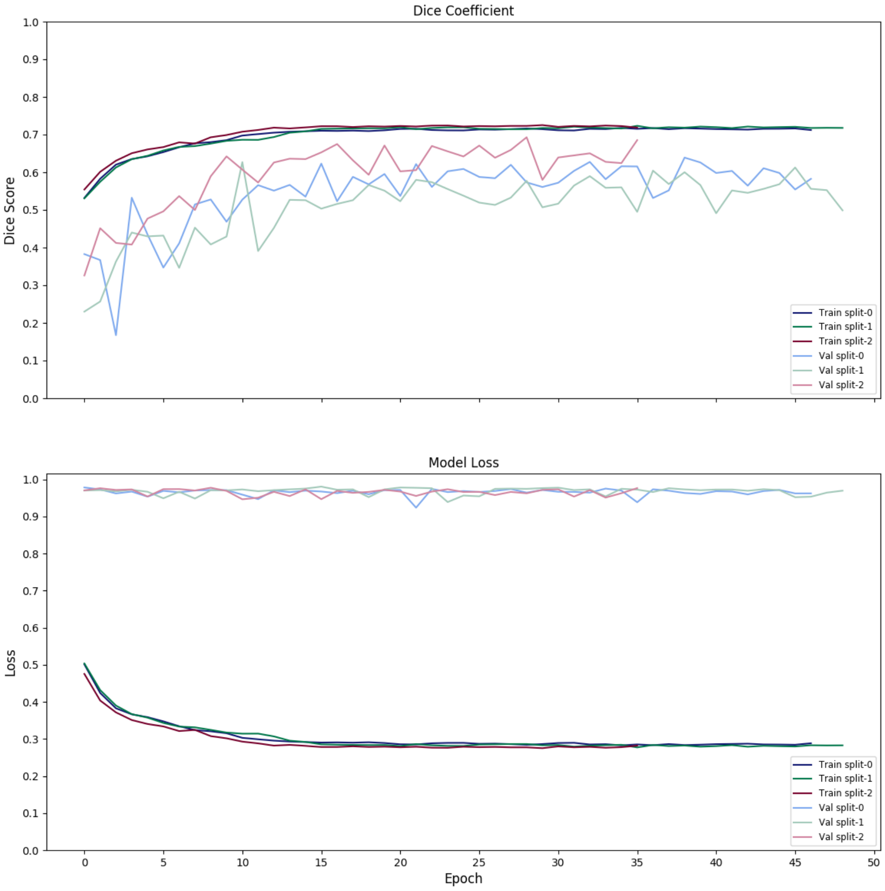


Figure 4.22: A graphs of the dice score and loss over the epochs for the different splits on the BVNet3D model with the Aorta label.

4.3.3 Portal Vein

In the Table 4.7 is a summary of dice score from the different splits. As mention earlier has each test split two samples, which means that the result below is of the hole test split. The table shows some difference in the dice score between the splits. The recall and the precision values are mostly the same, which are around 0.4, but in some cases are the precision better than recall like the *3Dircadb1.9* test image from split-1 shown in section C.3.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.476	0.485	0.533	0.529	0.365	0.359
Max	0.477	0.517	0.678	0.663	0.481	0.494
Mean	0.476	0.501	0.606	0.596	0.423	0.427

Table 4.7: The dice score of the different splits of the portal vein using the BVNet3D model.

The best raw predictions are shown in Figure 4.23. All the splits have TP in the midsection of the portal vein, which is the largest part. The split-0 prediction has some trouble finding the smaller veins and has almost only NP in the bottom. The existing TP is in the lower area, which is removed by the post processing step. In the top part is some of the smaller veins found. In split-1 is most of the ground truth found except the smaller veins. It has some gaps in the lower part of the portal vein. The smaller veins in split-1 are not as long as the one in split-0.

The split-2 ground truth distinguishes from the other two with almost having none smaller veins. The FN is mainly located where the veins branches out and in some of the endpoints. All the splits have some FP, but they are less than the FN.

Most of the worst raw predictions are finding a major part of the portal vein with some small differences, as shown in Figure 4.24. This makes some of them have a similar dice score to the best raw predictions. The worst raw prediction of split-0 is having a couple of FP formed as veins close to the TP. The lower part of the portal vein is having some FN close to where the largest vein branches out at the endpoints without creating gaps. The majority of the FN is in the upper half of the portal vein, where the smallest vein should have been.

The split-1 prediction has almost not found any of the smaller veins and has some connected components of FP, where the post processing step removes most of them. Most of the TP is connected together expect some parts of the smaller veins. The post processing also removes these parts.

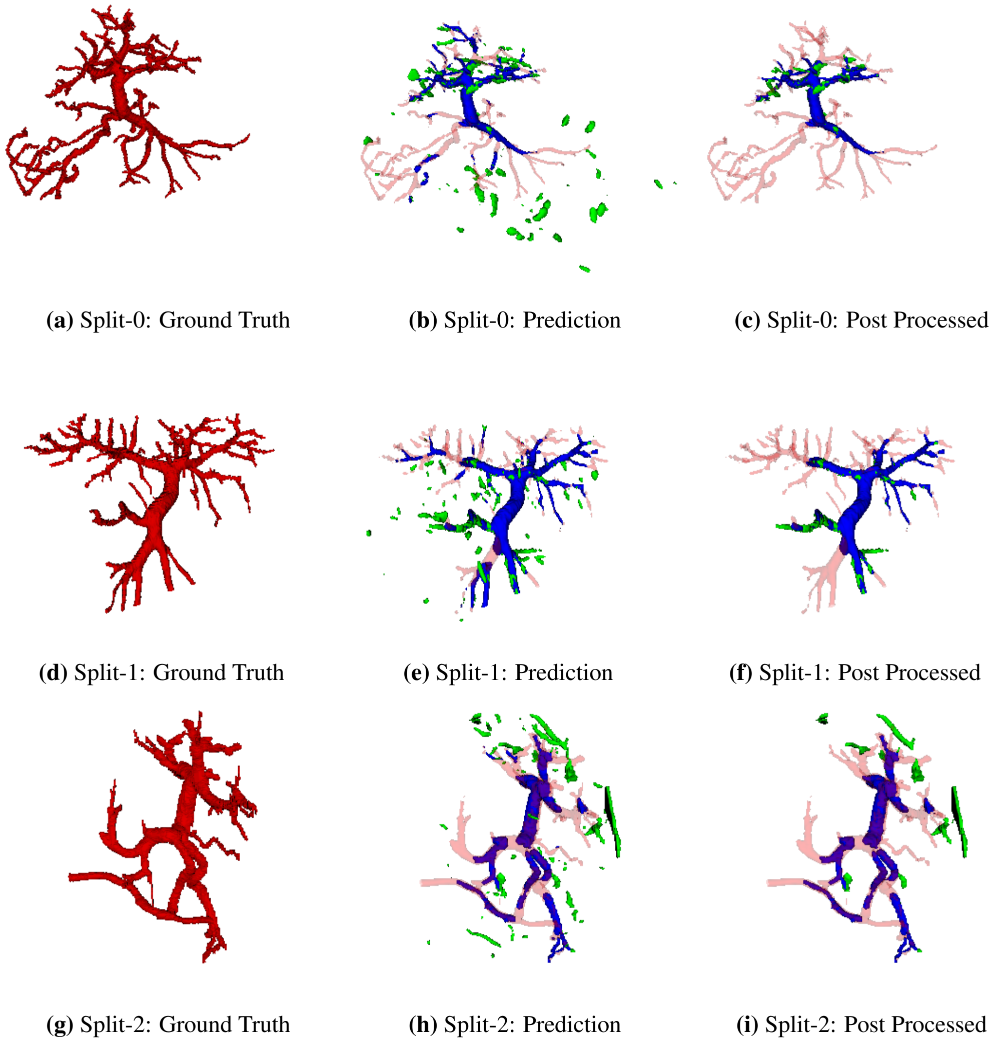
The split-2 has the worst raw prediction, which has fewer TP. These TPs consists of three large connected components in the largest vein and some small TP in the smaller veins. The majority of FP consists of three connected components with some tubular structure not connected to the portal vein. The post processing step removes the TP and FP in the smaller veins.

In Figure 4.25 is the best and worst raw prediction on the corresponding CT image. In general, is it hard to see some of the ground truth in a) because it has a similar intensity to the liver tissue. The $slice112/151$ in Figure 4.25b) shows that some of the FP, FN, and TP can be distinguished from the liver, but it is difficult to separate them because they all have similar intensity.

Figure 4.26 shows some variation in the training performance. The split-2 has the lowest validation loss and the highest training loss of the splits, and only trains for almost 35 epochs, which means that the weights used for testing were from around epoch 10. There is not that most none variation in the validation loss.

All the splits had validation loss values close to 1 with some minor spikes. Another observation is that there is almost 0.2 difference between the training loss values between split-1 and the others.

In the first to the tenth epochs are there some huge spikes from split-1 and split-2 in dice score for the validations. Afterward is the split-0 and split 1 having a similar evaluation with smaller spikes around the 0.8 value. The splits continue with some smaller spikes around the 0.65 value. The training dice score has a stable increment to the tenth epoch, where the split-0 and split-2 flatten out around 0.4 and split-1 around 0.5.

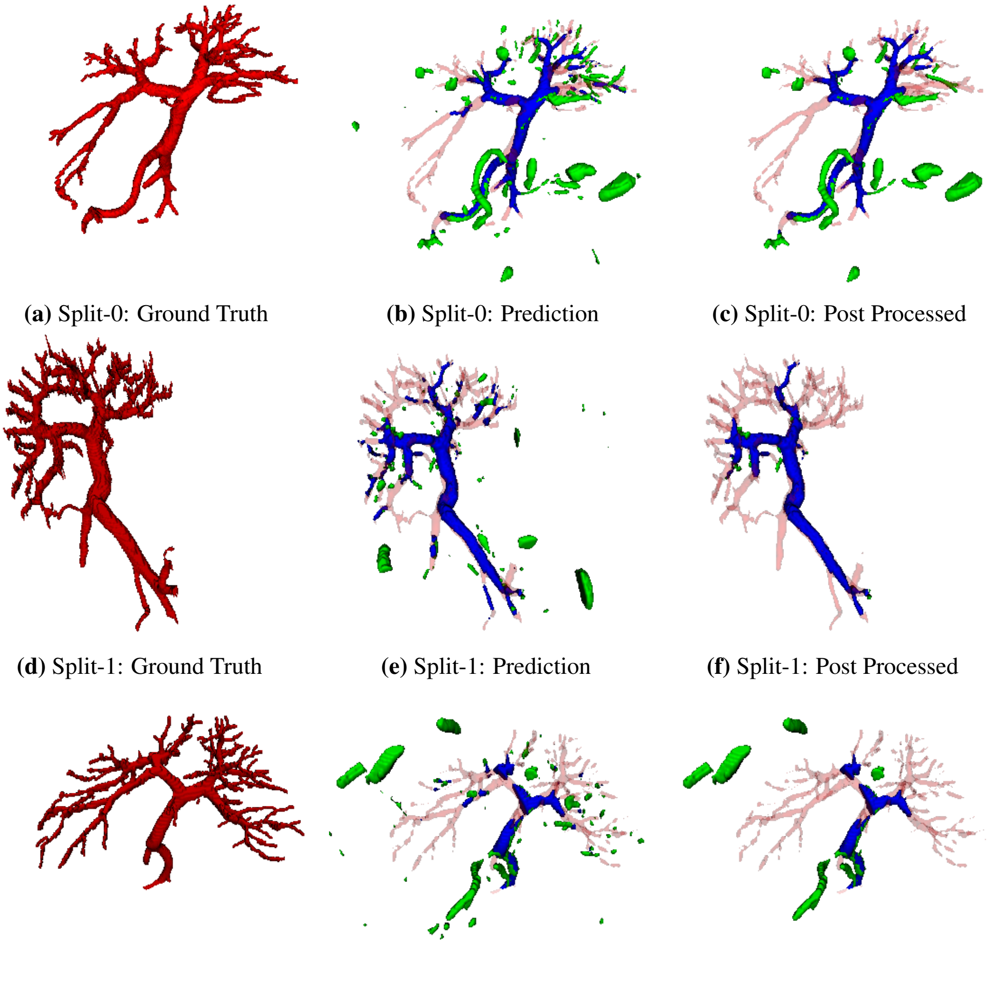


(a) Split-0: Ground Truth (b) Split-0: Prediction (c) Split-0: Post Processed

(d) Split-1: Ground Truth (e) Split-1: Prediction (f) Split-1: Post Processed

(g) Split-2: Ground Truth (h) Split-2: Prediction (i) Split-2: Post Processed

Figure 4.23: A visualization of the prediction of the portal vein with the best raw dice score for the BVNet3D model, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

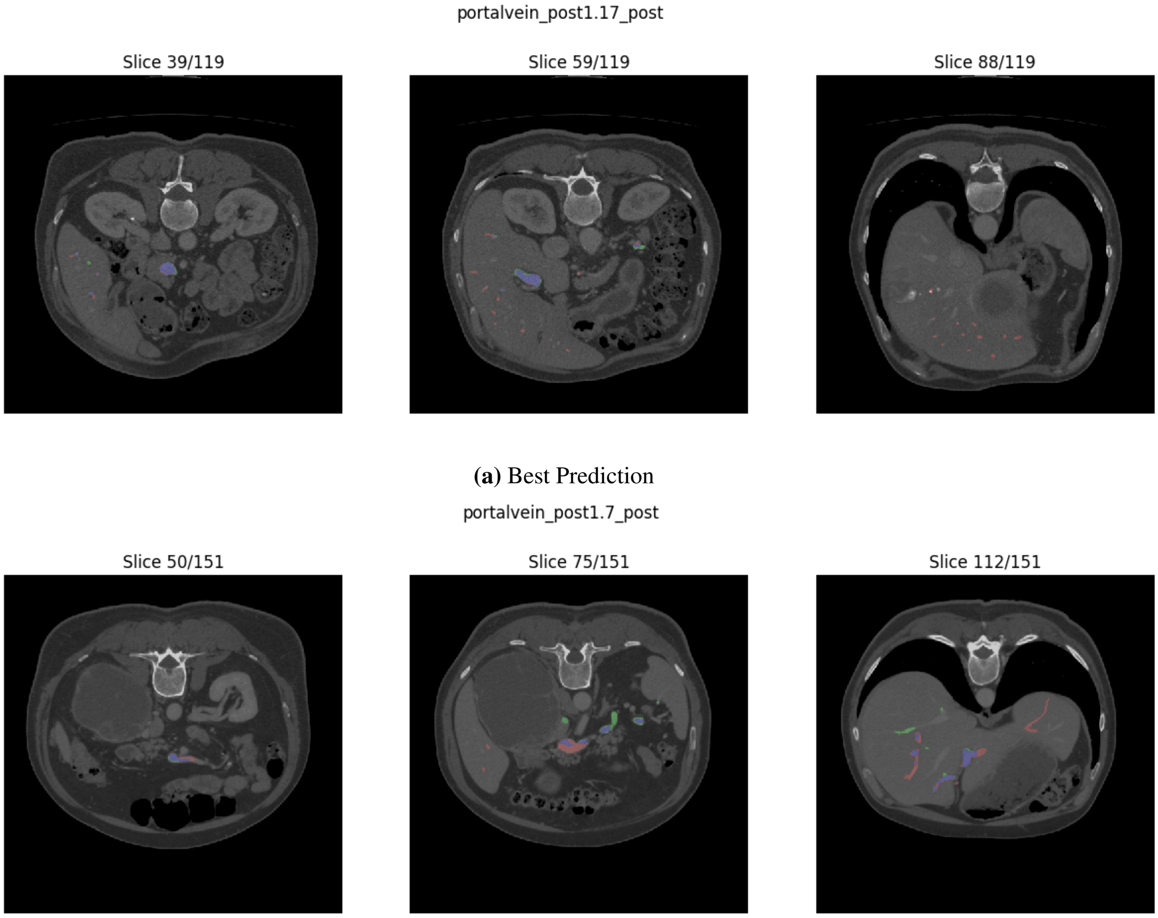


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.24: A visualization of the prediction of the portal vein with the worst raw dice score for the BVNet3D model, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



portalvein_post1.17_post

Slice 39/119

Slice 59/119

Slice 88/119

(a) Best Prediction

portalvein_post1.7_post

Slice 50/151

Slice 75/151

Slice 112/151

(b) Worst Prediction

Figure 4.25: A selection of some of the CT slices of the BVNet3D model's best and worst prediction of the portal vein splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

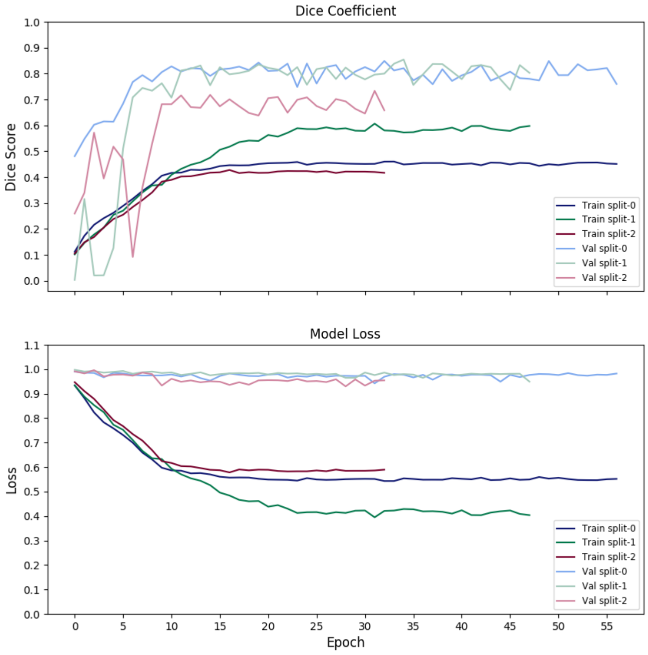


Figure 4.26: A graphs of the dice score and loss over the epochs for the different splits on the BVNet3D model with the portal vein label.

4.4 Experiment 4: SegCaps

In this section is the result of experiment 4, where SegCaps is used for segmenting the coronary artery, Aorta, and portal vein.

4.4.1 Coronary Artery

Table 4.8 is a summary of the dice score of the different splits. The variance between the minimum and maximum dice score from the different splits is around 0.2. This is almost the same as the variance between split-1 dice scores and the rest of the splits dice scores in Table 4.8.

	Split-0		Split-1		Split-2	
	$\mathrm{Dice}_{\mathrm{raw}}$	$\mathrm{Dice}_{\mathrm{post}}$	$\mathrm{Dice}_{\mathrm{raw}}$	$\mathrm{Dice}_{\mathrm{post}}$	$\mathrm{Dice}_{\mathrm{raw}}$	$\mathrm{Dice}_{\mathrm{post}}$
Min	0.46	0.48	0.262	0.294	0.439	0.449
Max	0.77	0.801	0.486	0.501	0.667	0.679
Mean	0.599	0.617	0.365	0.392	0.578	0.593

Table 4.8: The dice score of the different splits of the coronary artery using the SegCaps model.

Figure 4.27 is showing the best raw predictions of the different splits. In the first row is the best raw prediction of split-0, where the TP contains large connected components. In this prediction is most of the RCA found with some FP forming parts of a small artery close to the first curve. The rest of the FP is very small and laying close to the coronary artery except for two large clump formed connected components.

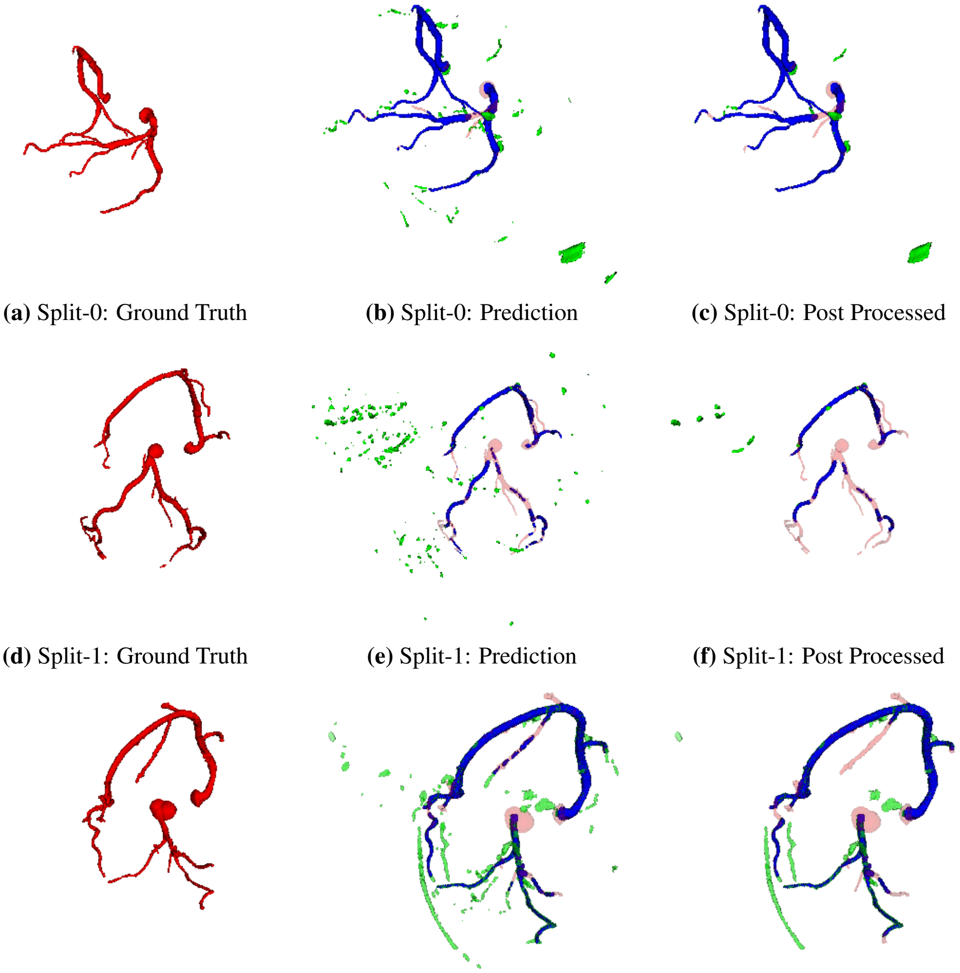
The best raw prediction of split-1 did find most of the RCA but had a gap after the first out branching and is not finding any of the smaller arteries. The left main branch of LM is having a lot of FN, and the TP is mostly tiny connected components. One of the parts not found in LM is where it branches into the two main branches. The FP consists mainly of many small parts close to each other. The post processing step removes almost all of the FP and some TP.

The best prediction of split-2 have almost none FN, and the existing ones are in the LM, where it connects to Aorta area and the smaller arteries. One of the connected components of the FP is a long curved tubular structure and some others being smaller tubular structures or small noisy parts. The post processing steps remove the noisy parts.

The worst raw predictions of the splits are shown in Figure 4.28. These predictions distinguish from the best raw predictions with having a lot more FP and more gaps between the TP, which is best illustrated by the worst raw prediction of split-2.

In Figure 4.29a) is the best raw prediction on corresponding CT slices, which contains almost none FP. On slice $306/409$ is there a thin and small FN located close to the TP. In b) are the worst raw prediction, which contains few TP and many small FP. The FP is mostly located around bones.

Figure 4.30 is showing the evaluation of the training performance of the dice score and the loss. An observation is that there is a small gap between the training and validation for the different splits. The split-0 has the lowest loss and the highest dice score, and the split-1 has the highest loss and lowest dice score. The difference between split-0 and split-1 is around 0.2 for both the dice score and the loss. During the training are the splits having a stable evaluation of the loss and the dice score, which flattens around epoch 15. The validation evaluation contains some minor spikes for both the loss and the dice score. Another observation is that the validation dice score is a bit lower than the training dice score for all the splits.



(a) Split-0: Ground Truth

(b) Split-0: Prediction

(c) Split-0: Post Processed

(d) Split-1: Ground Truth

(e) Split-1: Prediction

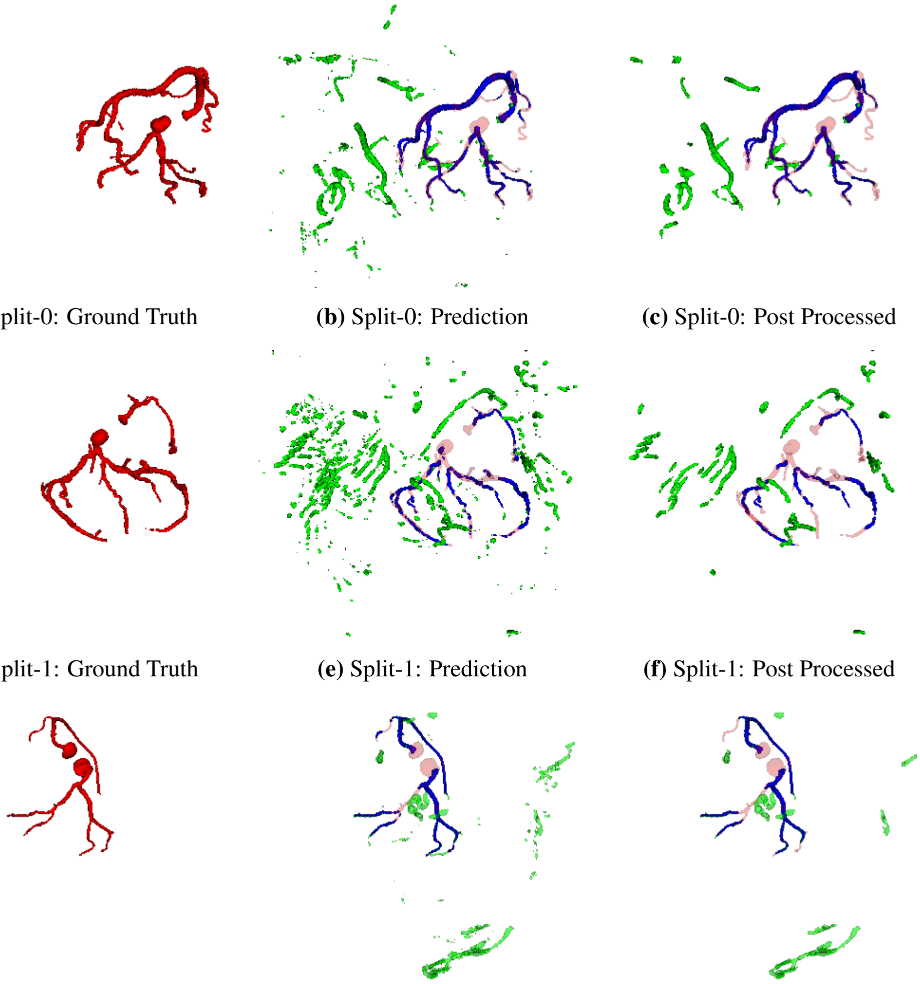
(f) Split-1: Post Processed

(g) Split-2: Ground Truth

(h) Split-2: Prediction

(i) Split-2: Post Processed

Figure 4.27: A visualization of the prediction of the coronary artery with the best raw dice score for the SegCaps model with 1 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

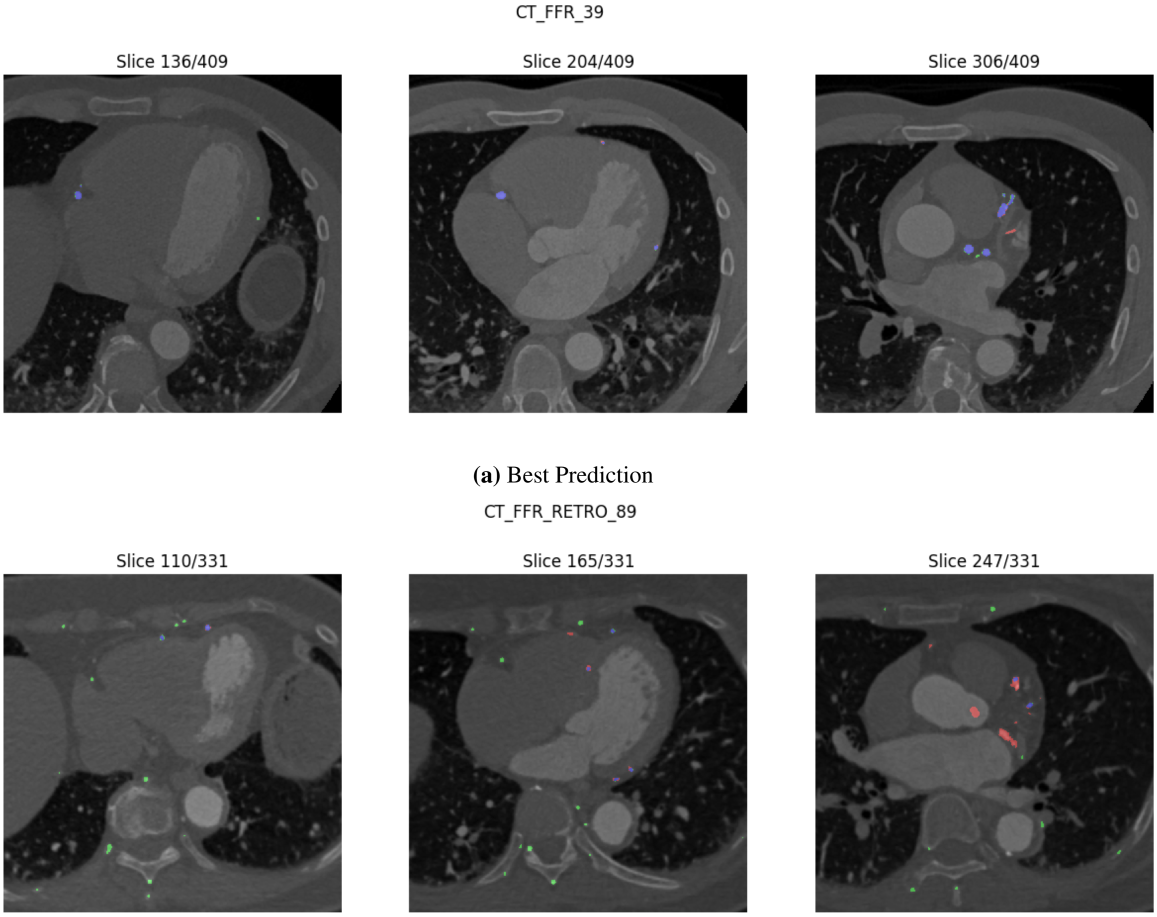


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.28: A visualization of the prediction of the coronary artery with the worst raw dice score for the SegCaps model with 1 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Best Prediction

(b) Worst Prediction

Figure 4.29: A selection of some of the CT slices of the SegCaps model's best and worst prediction of the coronary artery splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

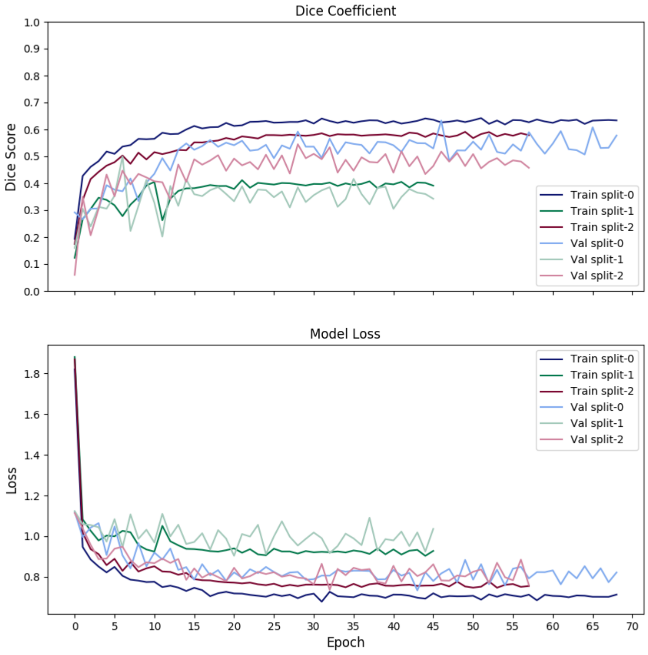


Figure 4.30: A graphs of the dice score and loss over the epochs for the different splits on the SegCaps model with the coronary artery label.

4.4.2 Aorta

Table 4.9 is a summary of the dice score of the different splits of Aorta. The result of split-0 and split-2 is similar with both having dice score 0 after post processing. The split-1 of the other hand has a maximum Dice$_{raw}$ score, which is almost 0.3 higher than the other two splits.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.102	0.0	0.196	0.217	0.047	0.0
Max	0.353	0.387	0.668	0.688	0.335	0.353
Mean	0.212	0.207	0.43	0.446	0.238	0.235

Table 4.9: The dice score of the different splits of the coronary artery using the SegCaps model.

The best predictions of each of the splits are shown in Figure 4.31. Both the best raw prediction from split-0 and split-2 have a lot of FP in most of the tissue close to the Aorta, which creates a large connected component. This connected component does contain some of the TP as well. Some of the FP looks like arteries/vessels or some part of the heart.

Most of the split-2 prediction consists of the Aorta and a large artery/vessel. All of the predictions have in common that they have some FN in the outer layer of the Aorta.

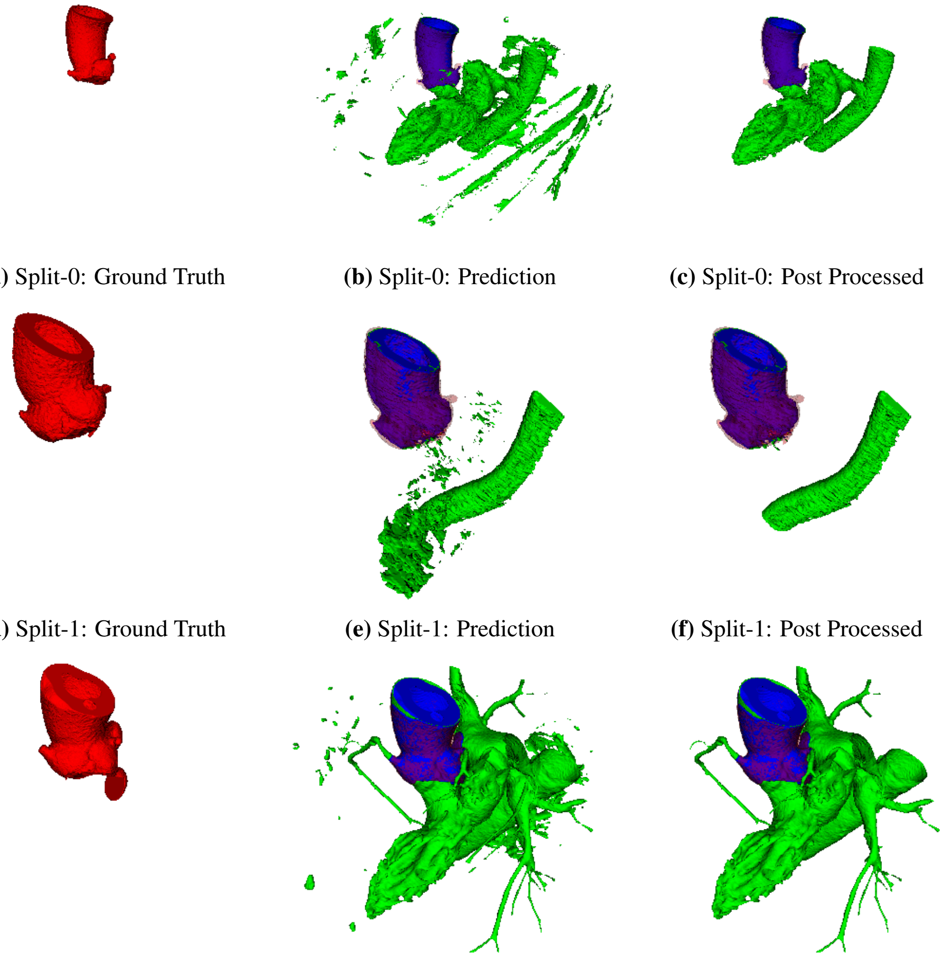
All the worst prediction of the splits is shown in Figure 4.32. The worst prediction of split-1 is similar to the best prediction of split-0 and split-2 with containing some part of the heart, a tubular structure as part of the Aorta, and having the outer layer of the Aorta not being predicted. The post processing step removes the tubular structure prediction.

The worst prediction of the split-0 and split-2 is similar with predicted most of another shape as Aorta, which has some kind of tubular structure. The TP part is so small that it is removed in the post processing step in these two predictions.

The rest of the best and worst raw predictions of the different splits on some of the corresponding CT slices in section D.2 shows that some of the FP are in general located at the bones. In most of these cases is the Aorta found with some FP of similar intensity laying around it.

The graph of the training performance for each split is shown in Figure 4.34. An interesting observation is that the validation loss is much smaller than the training loss. The validation loss has a more unstable evaluation caused by some spikes. The split-2 validation is the one with the largest spikes, which is in the interval $(3.5, 5)$. The training loss is quite similar between the splits, but the split-1 is a little bit lower and stops training around the 35 epoch. All the dice score values flatten out around epoch 15. The validation splits are containing some minor spikes. The training dice score is higher than the validation dice scores, with the split-1 having the highest values.

Figure 4.33 shows the best and the worst raw prediction on some of the corresponding CT slices of the splits. In Figure 4.33a) is a lot of the ground truth found except the ones in the outer layer. The rest of the slices shown of the best raw prediction contains a lot of FP at a circular shape close to the spine. Figure 4.33b) is almost none of the ground truth found, and most of the FP is located at the bones.

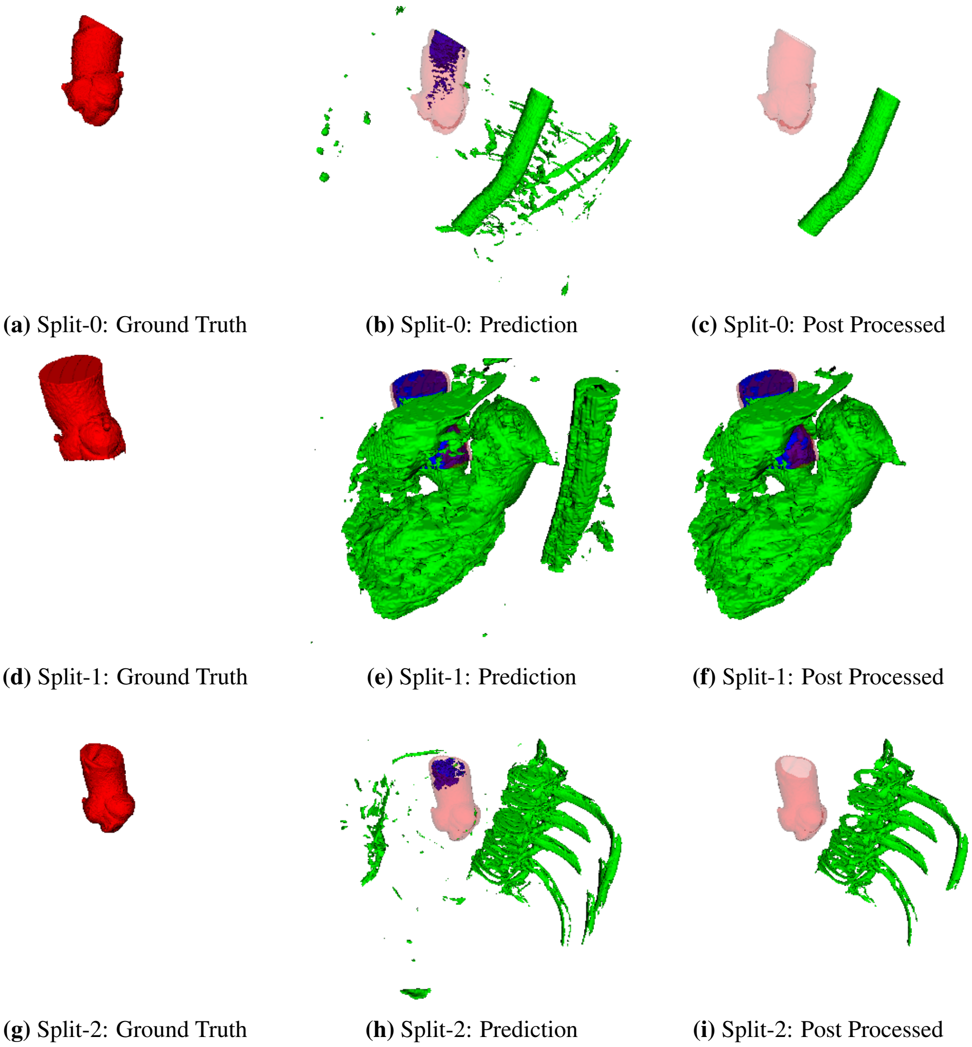


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.31: A visualization of the prediction of the Aorta with the best raw dice score for the SegCaps model with 1 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Split-0: Ground Truth

(b) Split-0: Prediction

(c) Split-0: Post Processed

(d) Split-1: Ground Truth

(e) Split-1: Prediction

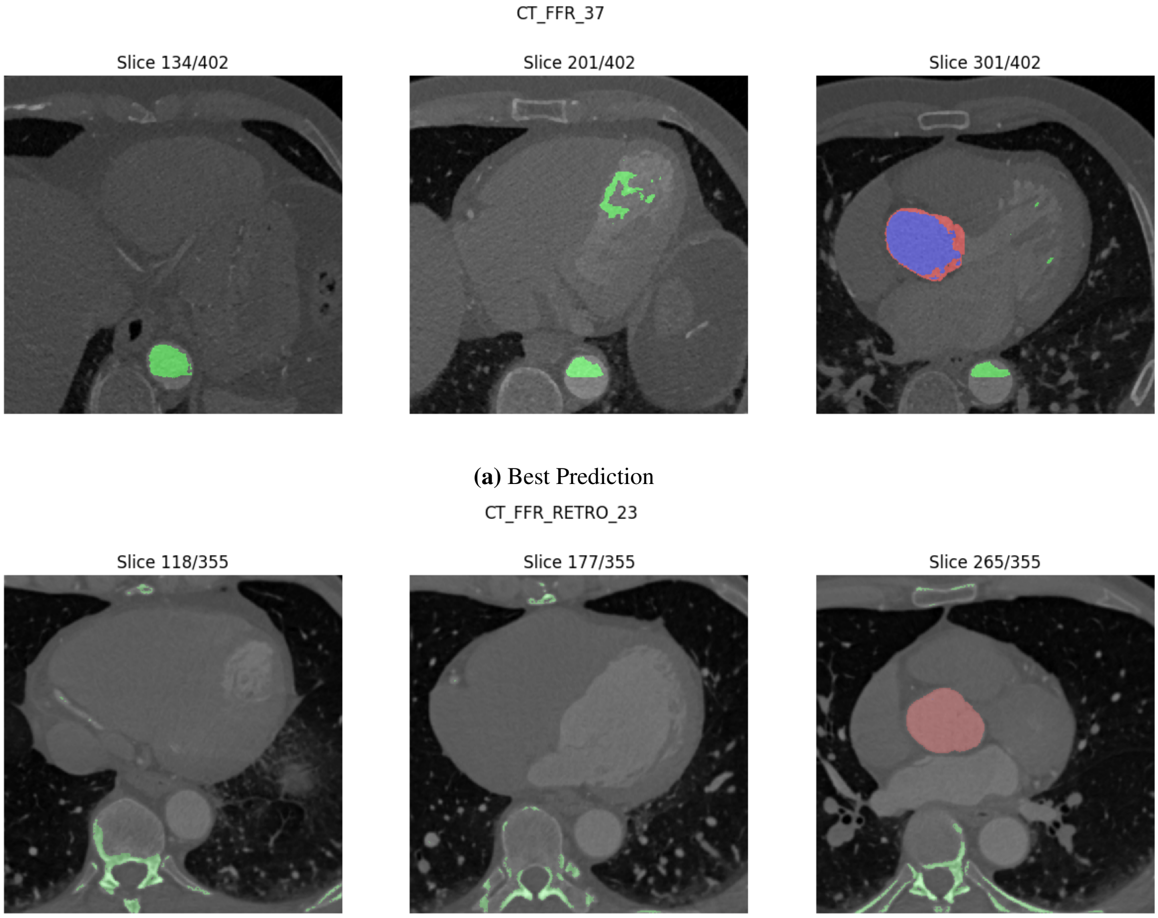
(f) Split-1: Post Processed

(g) Split-2: Ground Truth

(h) Split-2: Prediction

(i) Split-2: Post Processed

Figure 4.32: A visualization of the prediction of the Aorta with the worst raw dice score for the SegCaps model with 1 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Best Prediction

(b) Worst Prediction

Figure 4.33: A selection of some of the CT slices of the SegCaps model's best and worst prediction of the Aorta splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red=FN

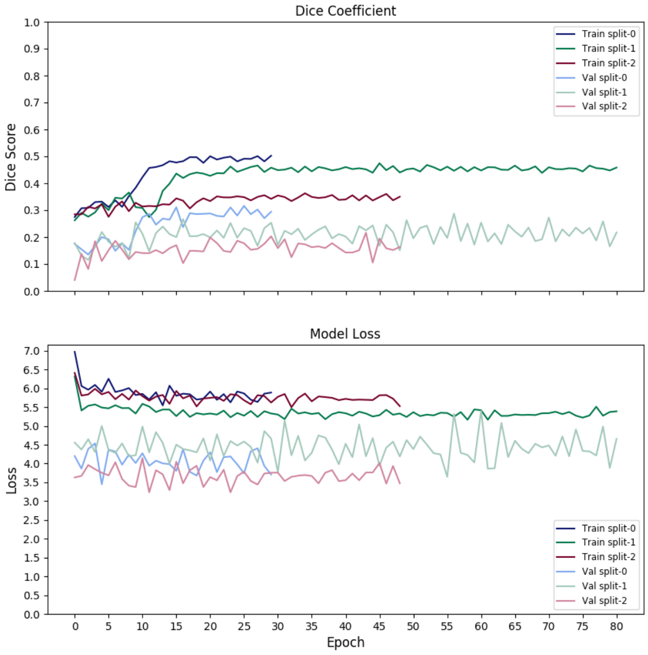


Figure 4.34: A graphs of the dice score and loss over the epochs for the different splits on the SegCaps model with the Aorta label.

4.4.3 Portal Vein

A summary of the dice score of the different splits is found in Table 4.10, which shows that the dice score is low for all the splits.

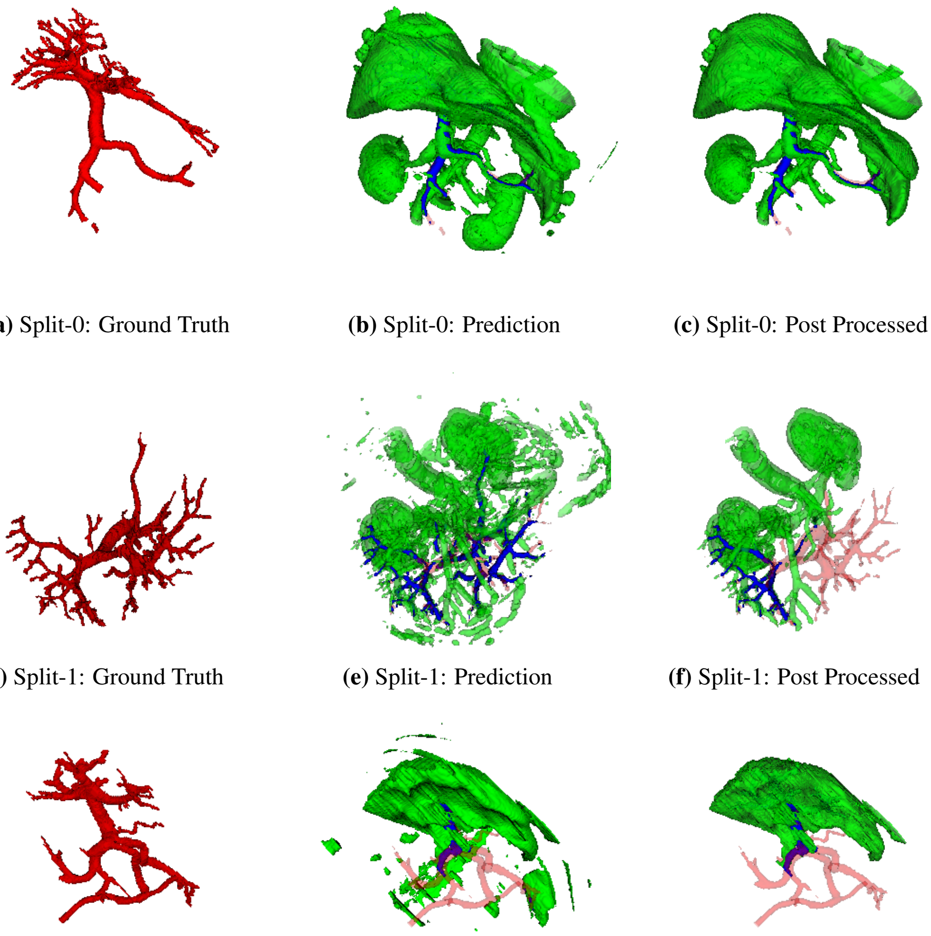
	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.038	0.05	0.021	0.018	0.043	0.045
Max	0.049	0.053	0.114	0.053	0.077	0.083
Mean	0.044	0.052	0.067	0.035	0.06	0.064

Table 4.10: The dice score of the different splits of the portal vein using the SegCaps model.

The visualization of the best raw prediction of the splits are shown in Figure 4.35 and the worst raw in Figure 4.36. All of these predictions have in common that they contain a lot of FP formed as kidneys, liver, or vessels. In the best raw predictions are some of the portal vein found with a lot of FP connected to them. The best raw visualization of split-1 and split-2 are most of the TP in the lower part removed by the post processing.

The worst raw predictions of all the splits have detected the perimeter of the liver shown in Figure 4.37b). The general pattern from the rest of these slices, found in section D.3, is that these models predict almost all of the liver tissue to be the portal vein. This is resulting in a lot of FP connected to the TP. In Figure 4.37a) are some of the FP in the vessel area, but the majority is in the kidneys.

The training preforming graphs in Figure 4.38 shows that there is almost none evolution during the training, except for the dice score of split-1. This split has some spikes in the beginning and flattens out around the 0.2 value, where the rest of the splits have values close to 0. The validation results for all of the splits have almost the same value as the split-0 and split-2.

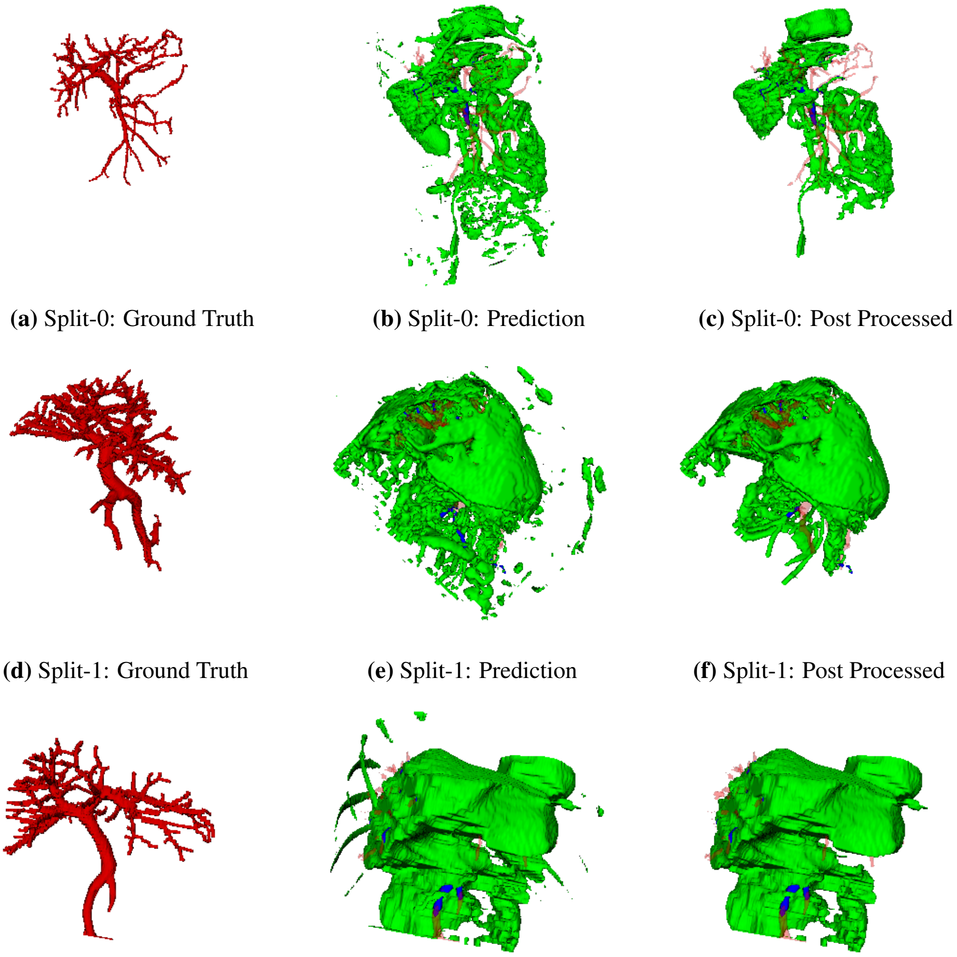


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.35: A visualization of the prediction of the portal vein with the best raw dice score for the SegCaps model with 1 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

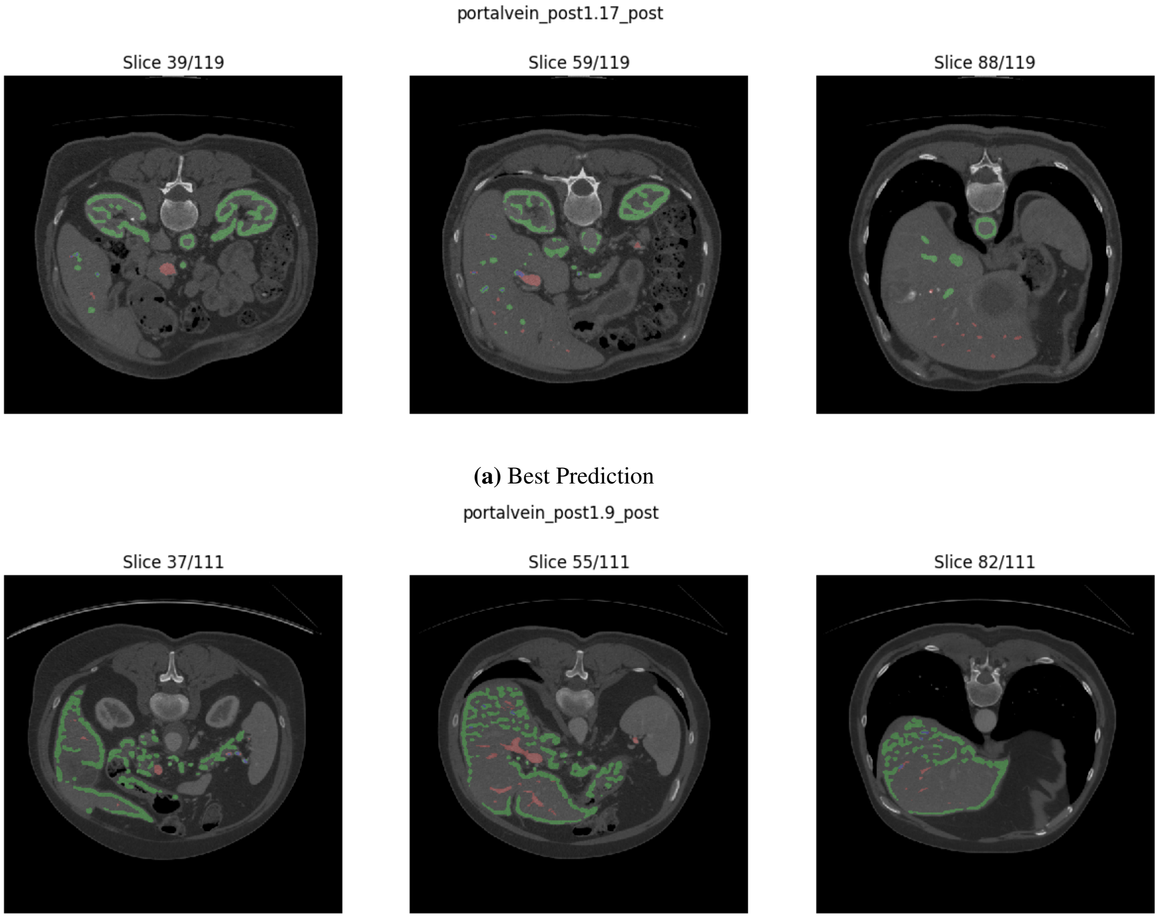


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.36: A visualization of the prediction of the portal vein with the worst raw dice score for the SegCaps model with 1 channels and stride 1, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Best Prediction

(b) Worst Prediction

Figure 4.37: A selection of some of the CT slices of the SegCaps model's best and worst prediction of the portal vein splits, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

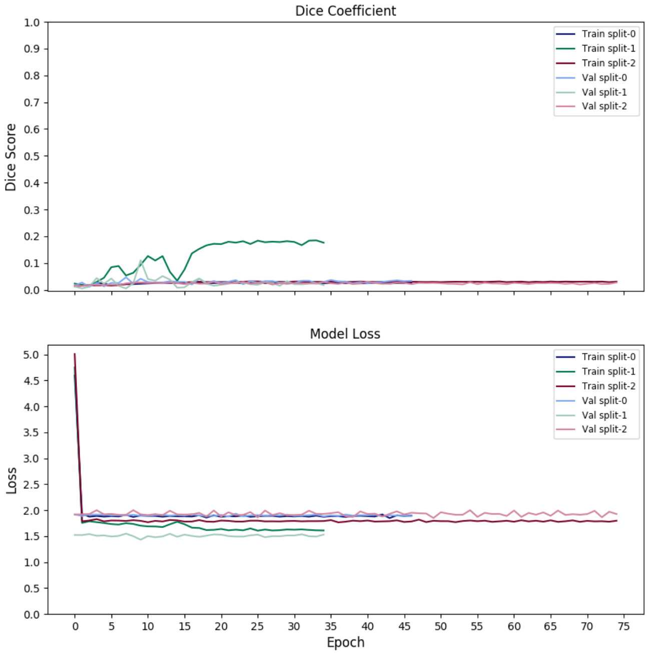


Figure 4.38: A graphs of the dice score and loss over the epochs for the different splits on the SegCaps model with the portal vein label.

4.5 Experiment 5: Frangi Filter

This experiment consists of two parts, as described in subsection 3.4.5. The first part is using Frangi filtered CT images as input to the BVNet model, and the second part is to use a combination of CT and Frangi filtered images as input to the BVNet model.

4.5.1 Frangi Input

The Table 4.11 shows a huge difference in the minimum and maximum dice score for each of the splits. The split-0 and split-1 have minimum Dice$_{raw}$ around 0.03, which means that these predictions almost did not find any ground truth or have a lot of FP. The tables from section E.1 shows a great variation in the dice score for each of the test images, which means that there are a lot of both very low and relatively high dice scores.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.033	0.031	0.014	0.009	0.225	0.232
Max	0.636	0.648	0.493	0.49	0.611	0.609
Mean	0.258	0.267	0.297	0.312	0.443	0.473

Table 4.11: The dice score of the different splits of the coronary artery using the BVNet model with input of Frangi filtered CT images.

The Figure 4.39 shows the best raw predictions for each split. The best raw prediction for split-0 have a few gaps in the smaller arteries of the LM. The post processing step removes most of the TP in the smaller arteries not connected to the main branch.

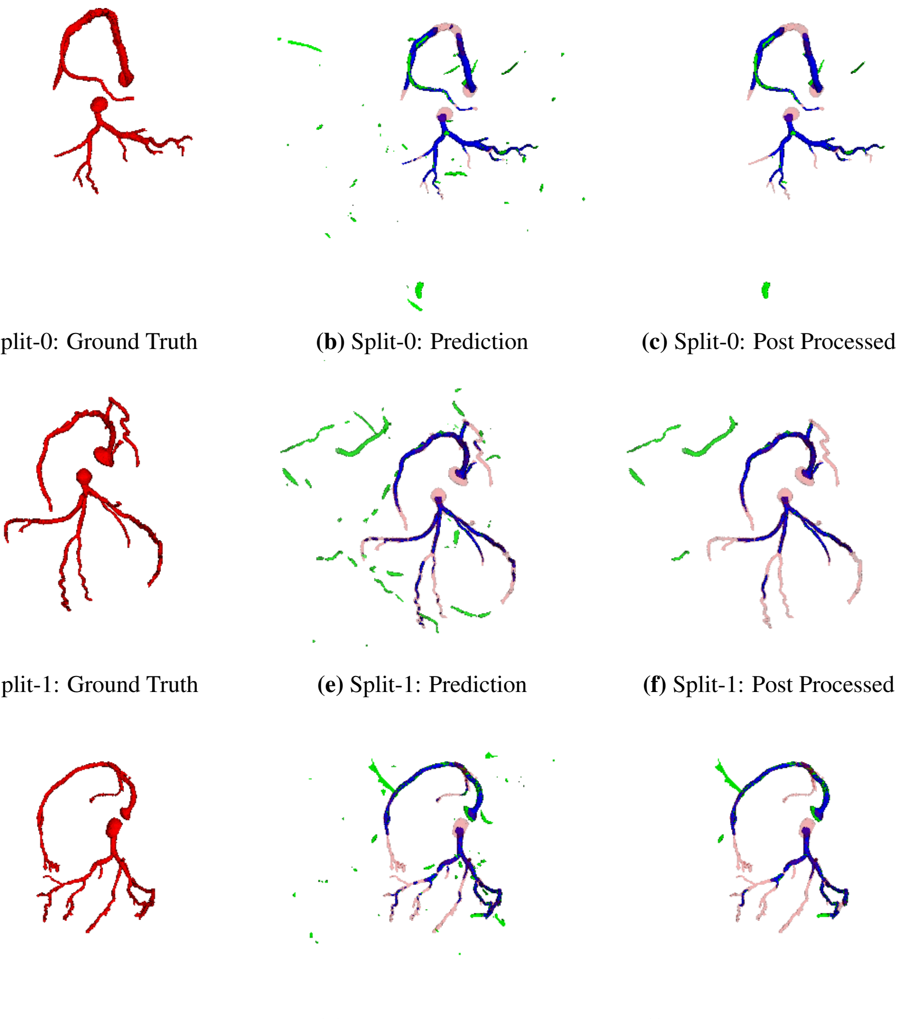
Both in the RCA and LM is the connection to Aorta predicted as NP. The RCA prediction contains a huge gap in the main branch, where it curves.

The split-1's best raw prediction is the one with most FP, which is located around the coronary artery with a tubular structure. Both split-1 and split-2 prediction is finding most of the main branches and having a lot of FN in the smaller branches. The FP in split-2 is mainly consisting of small tubular structure around the coronary artery, where the post processing removes most of them. The RCA's connection to the Aorta is extended with a couple of FP for split-2.

The worst raw prediction of the splits is shown in Figure 4.40. The two first splits have almost none TP and have few FP. The split-2 differences with predicting most of the left main branch in LM correct and having some TP in RCA. This prediction has a lot of FP next to the coronary artery.

The best and the worst raw prediction of the splits on some of the corresponding Frangi filtered CT slices is shown in Figure 4.41. The NP is located in some of the low intensity areas of the slices. An observation is that both the best and the worst prediction slices have higher intensity at the other vessels.

The Figure 4.43 shows the evaluation of the dice score and loss during training for the splits. All of the splits have almost the same dice score and loss values, which have a stable evaluation that flattens out around epoch 15. The validation dice score and loss contains large spikes for all the splits. The split-2 had a validation loss minimum around the value 0.6, which is smaller than the minimum loss value of the others.

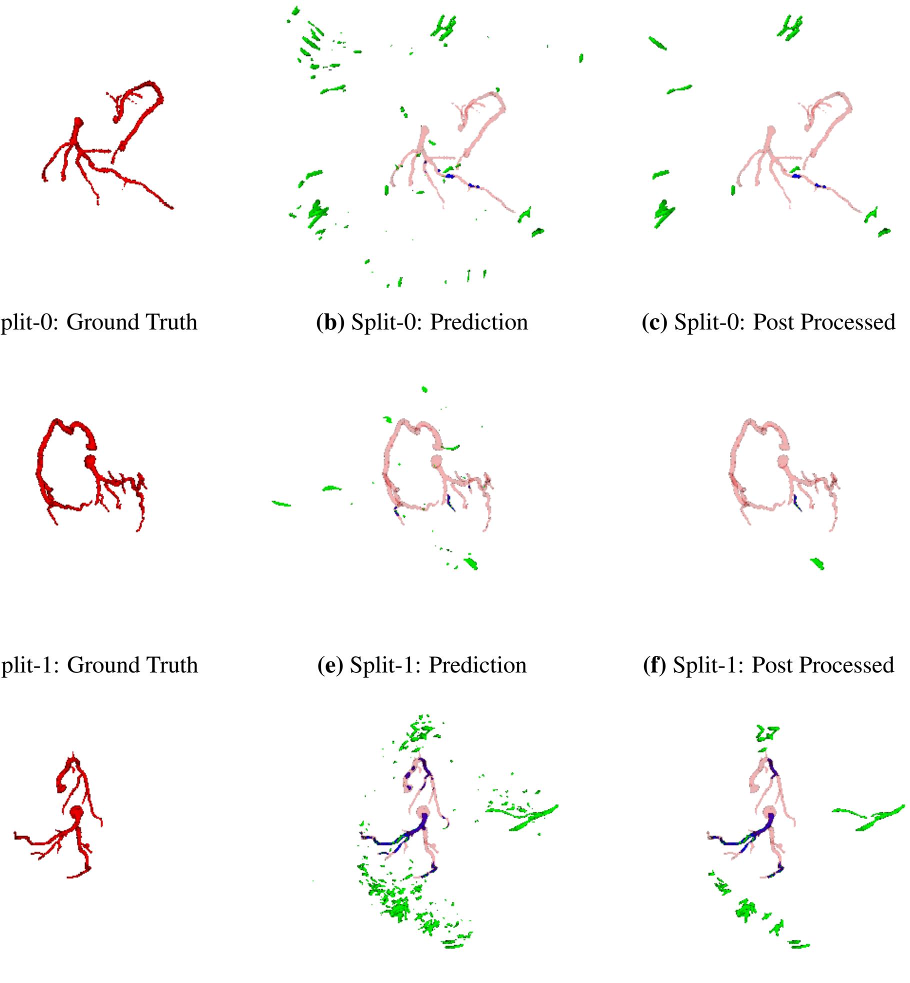


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.39: A visualization of the prediction of the coronary artery with the best raw dice score for the BVNet model with Frangi filtered CT images as input, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

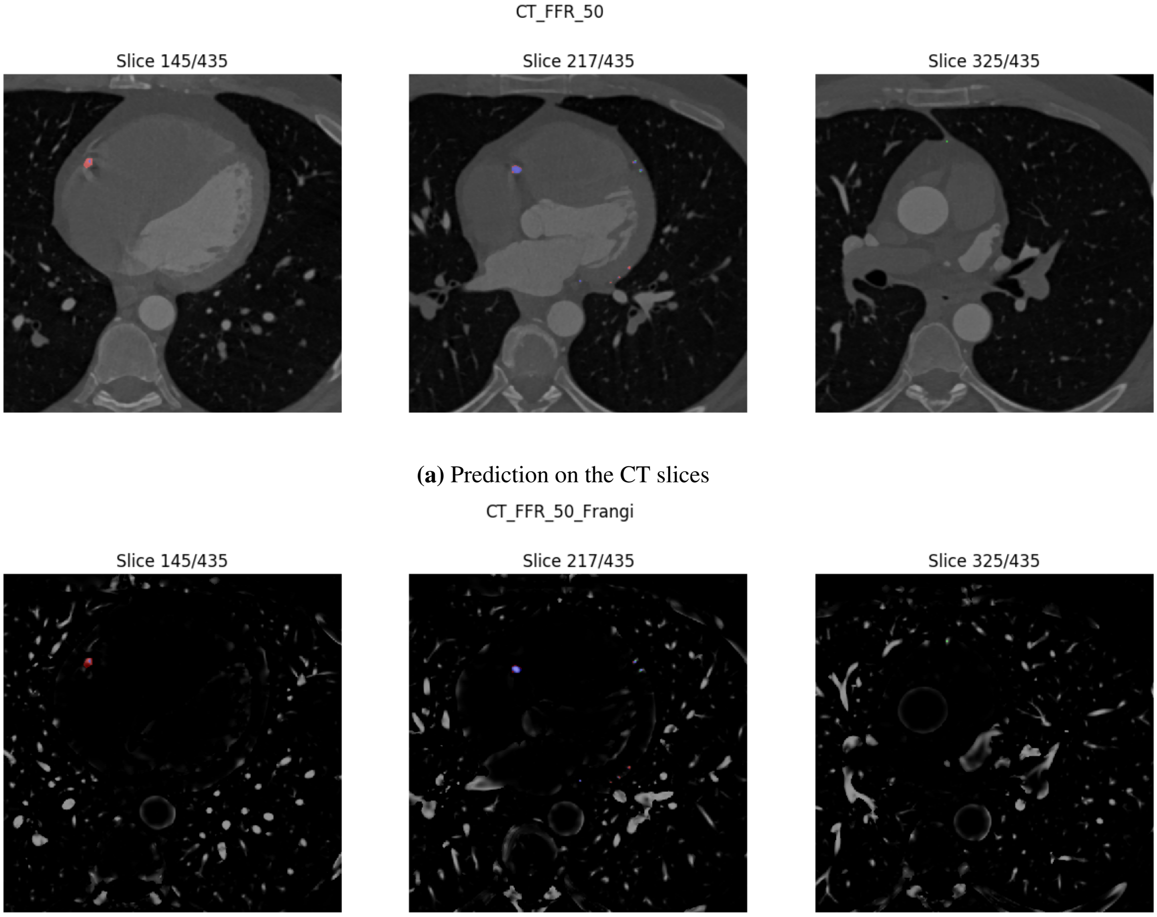


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

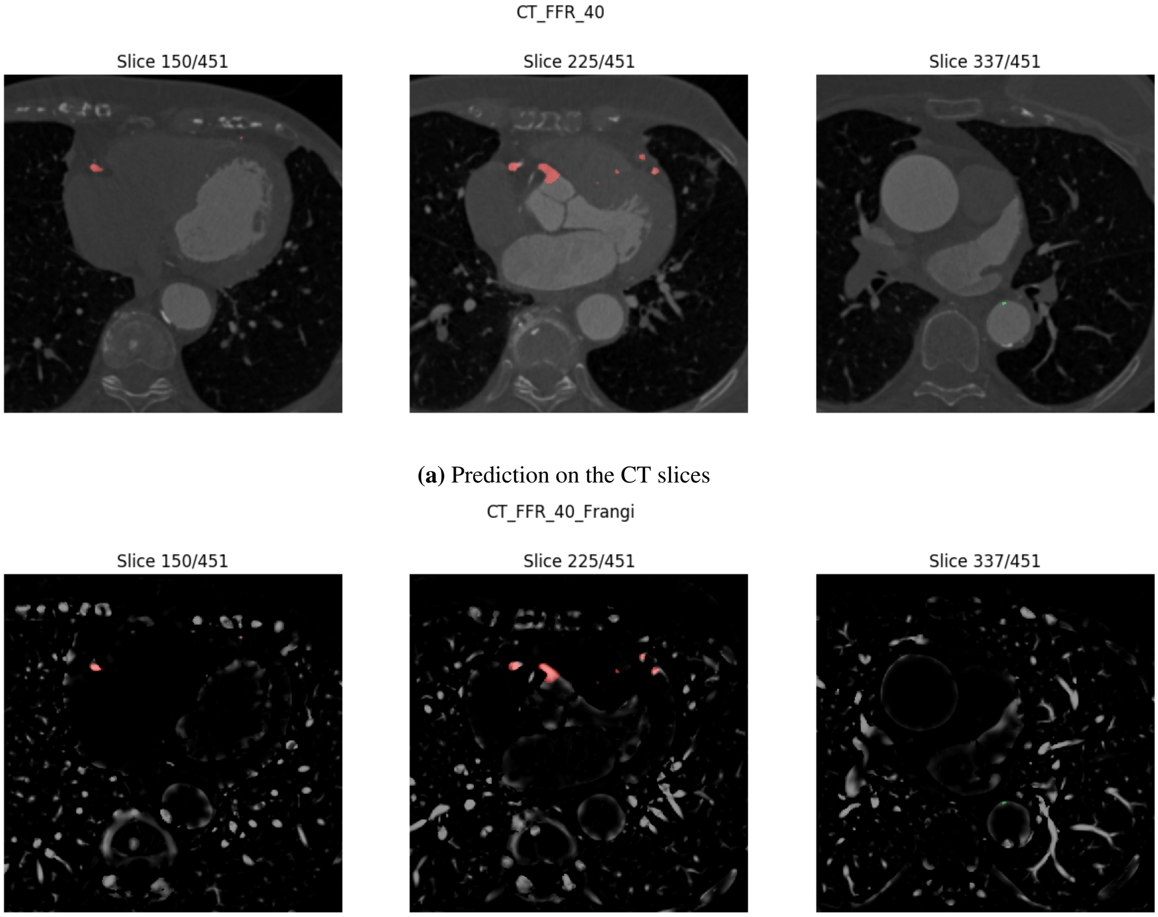
Figure 4.40: A visualization of the prediction of the coronary artery with the worst raw dice score for the BVNet model with Frangi filtered CT images as input, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

Figure 4.41: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's best prediction of the coronary artery splits with Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

Figure 4.42: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's worst prediction of the coronary artery splits with Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

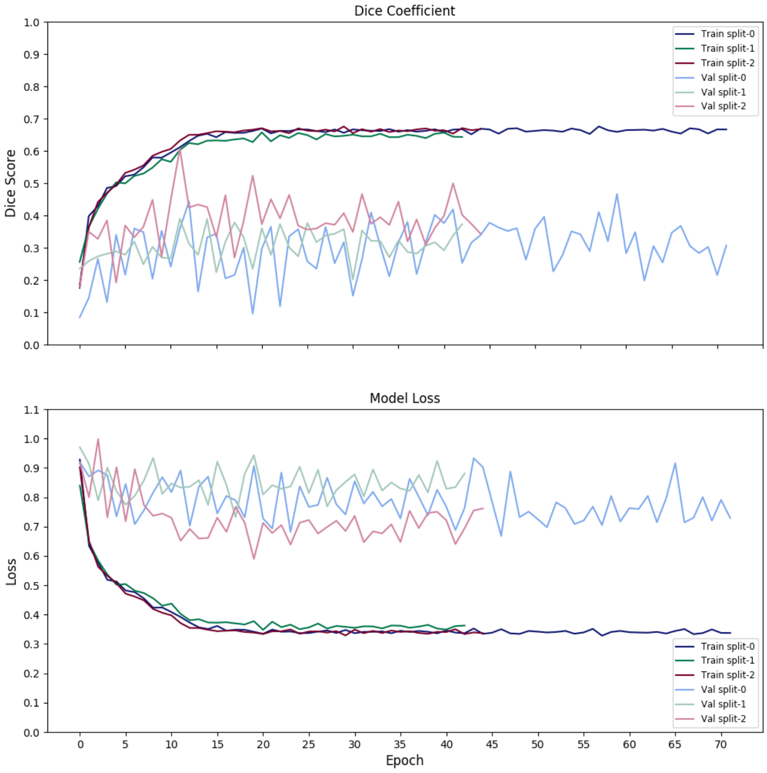


Figure 4.43: A graphs of the dice score and loss over the epochs for the different splits of the coronary artery on the BVNet model with Frangi filtered CT input

4.5.2 Combined CT and Frangi Input

Table 4.12 consists of a summary of the dice score of the BVNet model with combined CT and Frangi input for each of the splits. The tables in section E.2 consist of the individual test image results of each of the splits, which is used to create Table 4.12.

	Split-0		Split-1		Split-2	
	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$	Dice_{raw}	$\text{Dice}_{\text{post}}$
Min	0.332	0.324	0.518	0.529	0.521	0.52
Max	0.821	0.85	0.809	0.806	0.759	0.753
Mean	0.555	0.558	0.646	0.667	0.645	0.654

Table 4.12: The dice score of the different splits of the coronary artery using the BVNet model with combined input of CT and Frangi filtered CT images

Figure 4.44 shows the best raw prediction for each of the splits. Split-0 and split-1 are finding most of the ground truth except for the ones in the smallest arteries. This makes most of the TP connected in both the RCA and LM. These two splits have few FP, but the existing ones are either connected to or around the coronary artery.

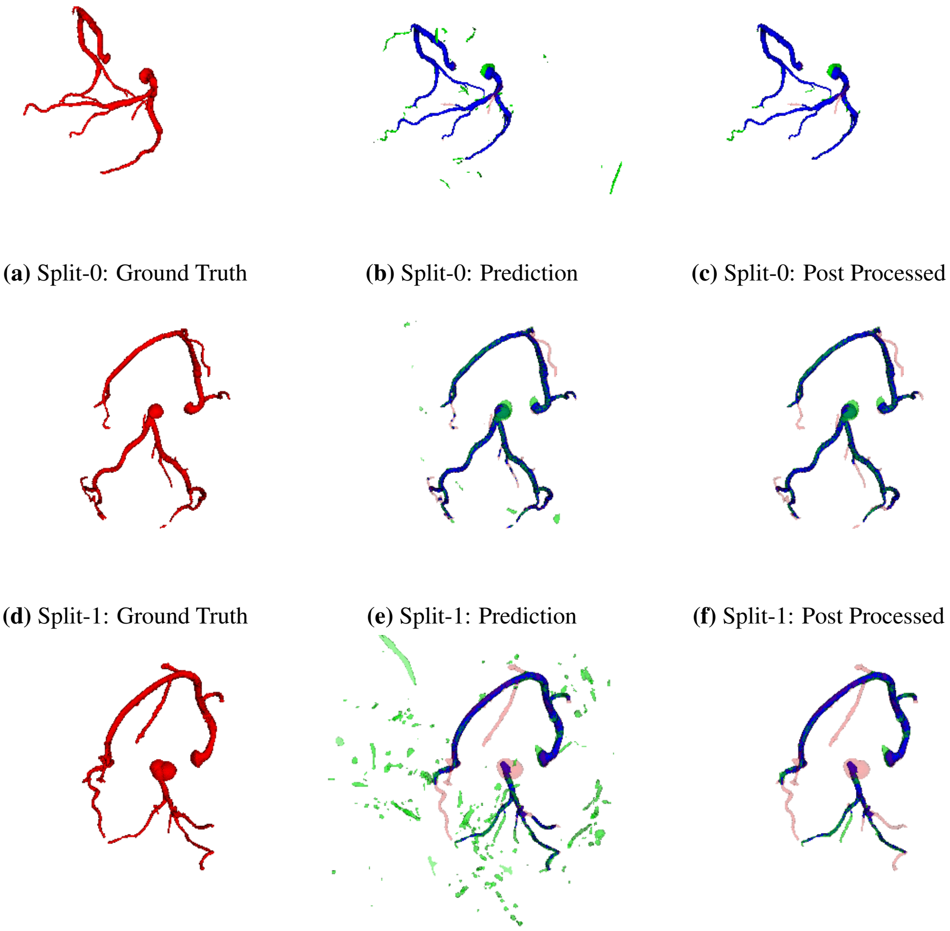
The split-2 prediction contains more FP and FN than the other two splits. The RCA ground truth is only predicted in the main branch, and LM have FN in the smaller arteries and in the connection to the Aorta. There are some FP connected to LM, creating a smaller artery on the left main branch. Most of the FP are around the coronary artery. The post processing removes some of the TP in the smaller arteries.

Figure 4.45 is showing the worst raw predictions of the splits. The split-0's worst prediction have found most of the RCA. A majority of the FN is in the starting point of the RCA for this prediction. The predicted LM has a lot of small connected TP, which is creating some gaps. The FP in this prediction consists mainly of tubular structures close to the RCA. The worst raw prediction of split-1 is finding most of the LM, but very few of the smaller arteries. The RCA is missing some parts, which is creating some large gaps. Most of the FP is connected with the TP.

The split-2 worst raw prediction distinguishes from the other to with having a lot of FP around the coronary artery. The TP is mainly in the main branches, and the FN is mostly in the last part of the smaller arteries in LM. There are also some gaps, where the LM branches out to smaller arteries.

Both in Figure 4.46 and Figure 4.47 shows that the majority of the NP is in areas with low intensity in the Frangi filtered CT image. This is a general pattern in the rest of the best and worst predictions found in section E.2.

The training performance is shown in Figure 4.48. Both the dice score and the loss for the splits have a stable evaluation. The validation loss for the splits contains some spikes and has values around 0.65 after epoch 15. The split-2 has its lowest validation loss around epoch 15, which makes it stops the training earlier than the other two. The validation dice score for split-0 has smaller spikes than the other two, which makes the evaluation more stable. The split-1 have a little bit lower validation dice score. There is a gap between the training and validation results during the training.

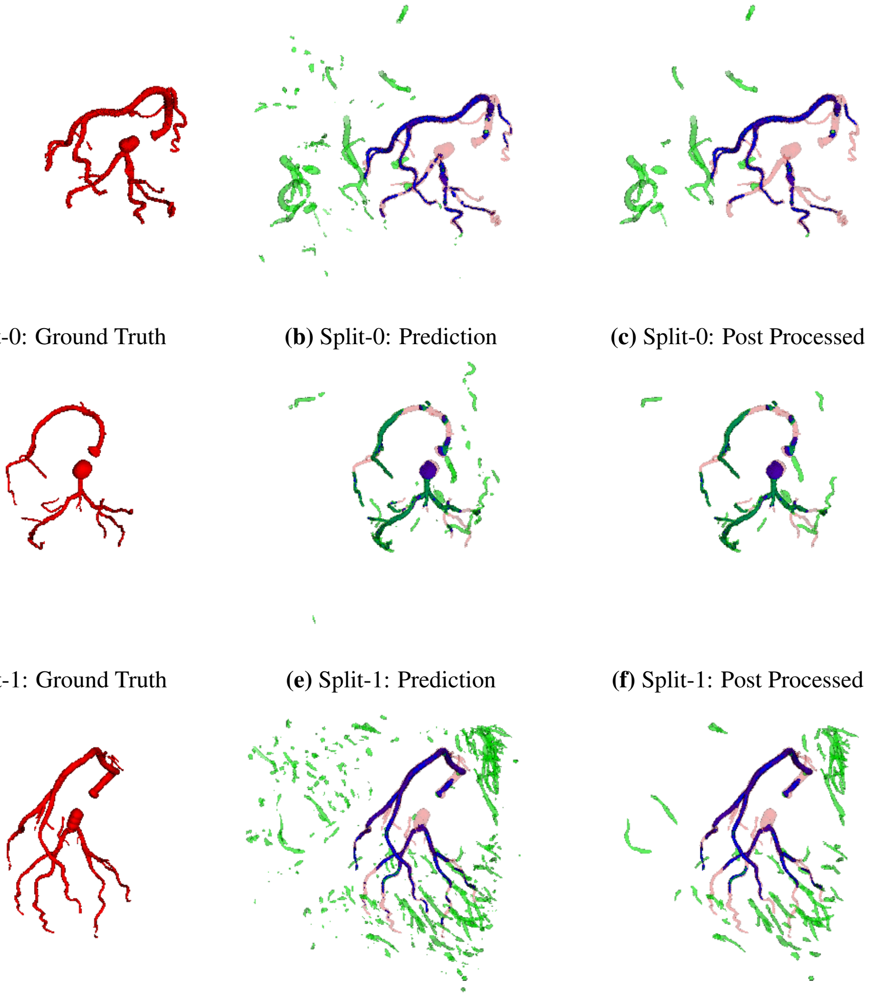


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

Figure 4.44: A visualization of the prediction of the coronary artery with the best raw dice score for the BVNet model with combined input of CT and Frangi filtered CT images, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

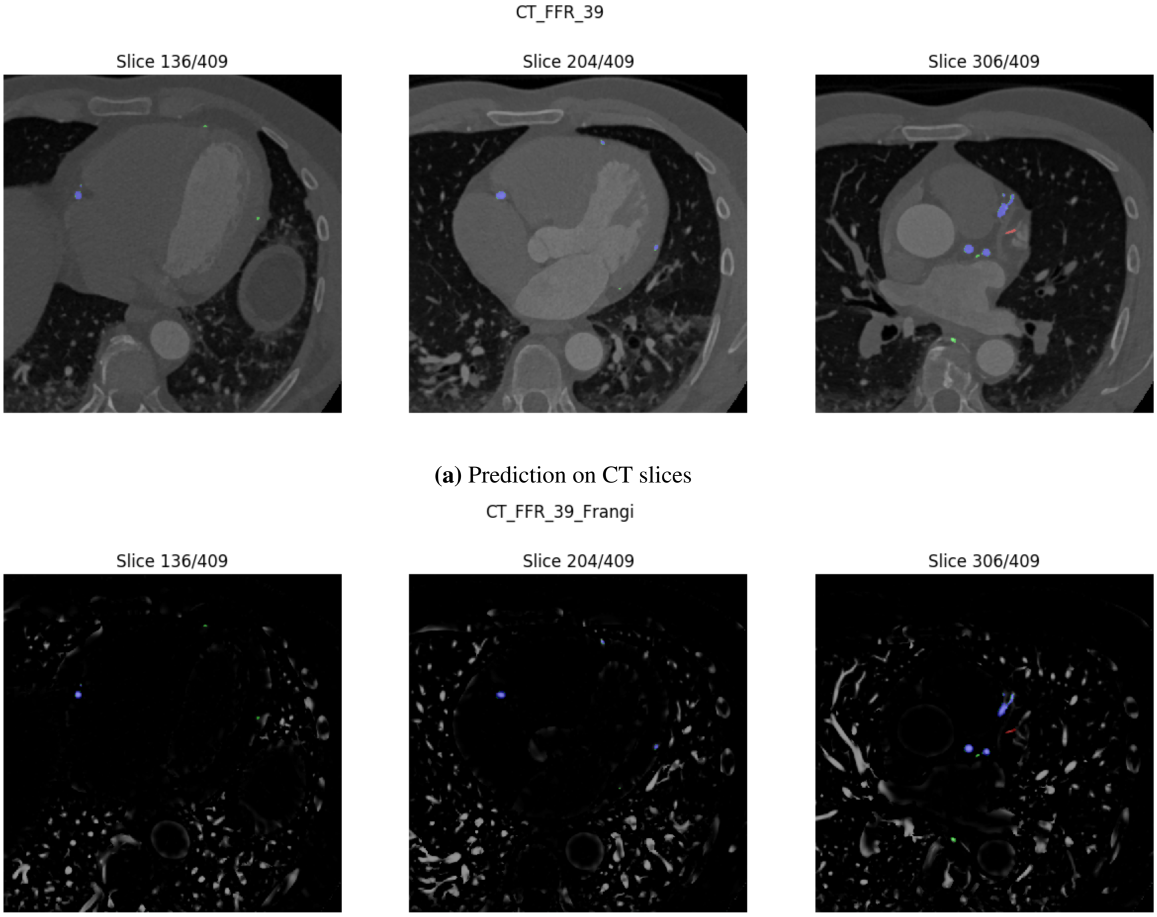


(a) Split-0: Ground Truth **(b)** Split-0: Prediction **(c)** Split-0: Post Processed

(d) Split-1: Ground Truth **(e)** Split-1: Prediction **(f)** Split-1: Post Processed

(g) Split-2: Ground Truth **(h)** Split-2: Prediction **(i)** Split-2: Post Processed

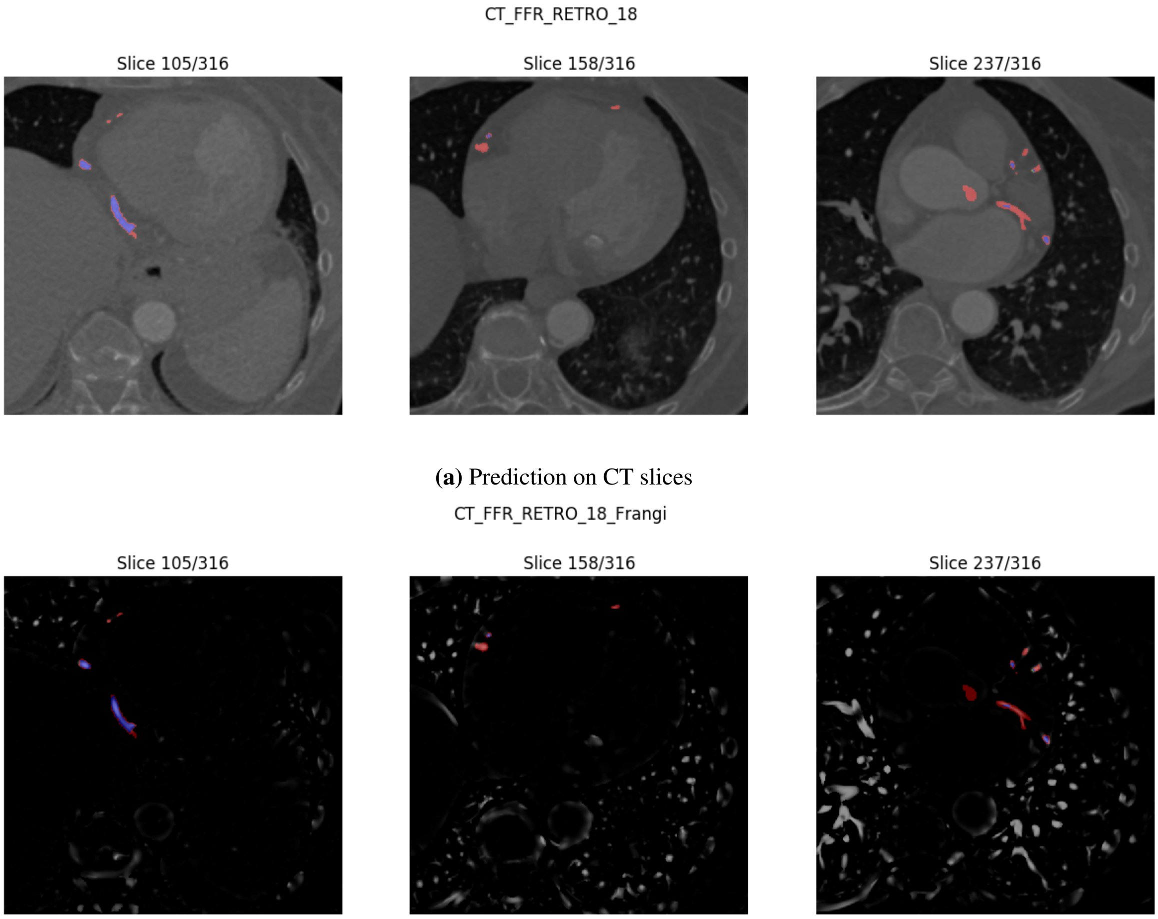
Figure 4.45: A visualization of the prediction of the coronary artery with the worst raw dice score for the BVNet model with combined input of CT and Frangi filtered CT images, where the row is the different splits. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on CT slices

(b) Prediction on Frangi filtered CT slices

Figure 4.46: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's best prediction of the coronary artery splits with combined CT and Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on CT slices

(b) Prediction on Frangi filtered CT slices

Figure 4.47: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's worst prediction of the coronary artery splits with combined CT and Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

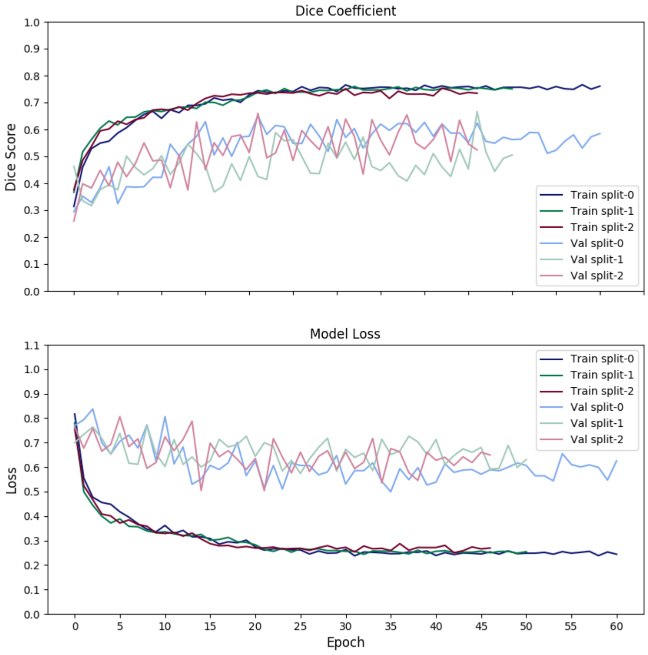


Figure 4.48: A graphs of the dice score and loss over the epochs for the different splits of the coronary artery on the BVNet model with combined CT and Frangi filtered CT input

Chapter 5

Discussion

This chapter discusses the result from chapter 4, and attempts to answer the research questions. It begins with the *Channel and Stride* experiment, which includes a discussion of what characteristics of the predictions are most important for the FFR-project. Next, the benefits and disadvantages of the models BVNet, BVNet3D, and SegCaps will be discussed in distinct sections. The following section contains a discussion of the usage of Frangi filter with the BVNet model. The chapter finishes with a reflection about what could have been done differently in this thesis.

5.1 Channels and Stride

Before suggesting an answer to RQ1 with the optimal number of channels and stride for improving the BVNet model understanding of the predicted slices surroundings, there needs to be a discussion of what the optimal prediction is. As explained in the motivation, the prediction results need to be good enough to be used with fluid dynamics in order to measure the FFR automatically. To be able to do that, the predictions of the coronary artery need to be connected without gaps and without FP on the inner radius of the arteries. The small connected components of FP around the coronary artery are not important. Because they can be removed by using a more strict post processing, which only keeps the two largest components. Most of the worst predictions from experiment 1 have few TP, which makes the models in this experiment unsuitable for the FFR project without any further development.

The huge difference in the minimum and maximum dice score may be a result of overfitting. Another reason could be that the image with the minimum dice score is an edge case, but to be sure the experiment should have been tried out with more splits. The difference between the minimum and maximum dice score can probably be reduced by applying data augmentation or adding more images to the dataset. The data augmentation should include minor intensity and contract changes or other characteristics, which would make the model more invariant against small changes in the dataset.

The usage of stride 2 increases the dice score for some of the models. The reason why this is not true for all models may be that these models may have gotten stuck in a local minimum during training. This may be an effect of some of the random events during training. The random events consist of shuffling at the beginning of each epoch or the random weights it is initialized with. A way to help the model to generalize is to test different learning rates, and perhaps add a learning scheduler.

The results of experiment 1 has few patterns, but one pattern is that the models with more channels have a better connection between the TP components and sometimes contain more noise. The results show that an increment in the number of channels sometimes result in a smaller difference between the minimum and maximum dice score if the model with 9 channels and stride 2 is not included. The result of 9 channels and stride 2 is most likely a result of the model getting stuck in a local minimum, which is probably caused by some of the random events mentioned earlier. This could also be the case of 5 channel and stride 2. Because it does not have the same pattern as the 3 and 7 channels with stride 2, where the dice mean dice score is a little better with stride 2 than stride 1 and 5. However, to be certain more testing is needed.

There are some minor differences in the number of FP and TP in the worst predictions. The model with 7 channels and stride 1 is the only one that finds almost the entire main branch of RCA and most of the LM with some gaps. The model with 3 channels and stride 5 and the model with 7 channels and stride 2 have a similar prediction of LM, which finds a lot of the ground truth with a lot of small gaps. This may indicate that a combination of increasing the channel size and stride may reduce the gaps. Because this pattern does not fit all cases, more testing needs to be done to be sure.

From this experiment, the model with 7 channels and stride 2 has the predictions closest to the optimal prediction because it has the second highest mean Dice$_{raw}$ score and the highest mean Dice$_{post}$ score with smaller gaps between the TP. The model with 5 channels and stride 1 has the highest mean Dice$_{raw}$ score but contains larger gaps between some of the TP. The model with 5 channels and stride 1 has used in the other experiments because it has a lower computational cost than the model with 7 channels and stride 1.

To summarize the result from section 4.1 shows that the mean dice score will increase if more than 1 channel is used. By increasing the channel size and stride will the gaps between the TP become smaller in some cases. This is most likely caused by the model getting more context of the predicted slice surroundings. None of these models are good enough to be used with fluid dynamics to measure the FFR automatically without further development, because there is a lot of gaps between the TP.

5.2 BVNet

This section will explore RQ2 by discussing the result from section 4.2. The result shows a big difference between the minimum and maximum dice score for all of the splits. As mention earlier, this could be a result of some image characteristics being underrepresented in the training set. Contrast variation is such a characteristic, which could make some of the smaller arteries harder to separate from its surrounding. The smaller arteries have some variation in the form of location and structure as well, which is another reason why they are a difficult target to predict.

A lot of the predictions contained some gaps in the larger arteries, which is unexpected because of the size but could be caused by a stenosis. The gaps before the out-branching may be an effect of these splits taking various forms, and the contrast could make it hard to find. The contrast problem may be solved by changing the Hounsfield window and level, which will involve finding the general intensity range of the coronary artery.

Another problem with this model is that it almost always has problems finding the right amount of ground truth where the connection to the Aorta starts for the coronary arteries. This may be a result of the labeling not being consistent in this area. In general, this model's best prediction is quite good, as it does not contain many FP and having a small amount of FN. The model is best at detecting the RCA of the coronary arteries. This may be a result of the LM having more variation in the number of smaller arteries and the shape of the arteries. Another interesting observation is that the split-0 Dice$_{raw}$ score is a bit lower than the result of experiment 1 with 5 channels and stride 1. The random events explained earlier, the learning scheduler or the batch size increment may have resulted in the decrement of the dice score. Because of these instabilities, more testing is needed to be sure that the model is not just unlucky with some of the random events.

The coronary artery segmentation predictions contain few FN and FP. The worst predictions from the different splits have more FN and FP than the best predictions, which indicates that the model is not very robust against changes. This could be fixed by analyzing the dataset further and applying some data augmentation to the underrepresented changes in the dataset. The contrast change which is done in the portal vein segmentation will most likely help the st.Olav dataset by centralizing the training, which may make removing of the only background slices unnecessary.

The minimum and maximum dice score of the Aorta segmentation is quite good for all the splits except the minimum of split-0, which is missing most of the bottom part. The majority of the best result has almost all of the FP connected to Aorta, and the ones not connected is removed by the post processing step. Most of the FN and FP connected to the Aorta is in the lower end or the back. These results may be caused by the Aorta not having many variations. One variation is that the surface can in some areas have a bulge. Another variation is at the end of the Aorta, which sometimes has a clean cut. This may be one of the reasons why the model has problems detecting these areas.

The predictions on the corresponding CT slices give us some information on the different contrasts and what is wrongly classified. From these figures is it possible to observe that many of the FP are on the slices not containing any Aorta masks. This means that the model wrongly predicts unseen slices, which is to be expected.

The result of the worst prediction of split-2 had some strange placed half ball ground truth close to the Aorta.

The predictions are probably not too affected by these ground truth because they are only in a very small portion of the dataset and that none of the seen predictions share these characteristics. The worst prediction of the split-0 may be a result of this, but it is more likely that it is a result of some uncommon structure, intensity, or contrast in that image.

The segmentation of the portal vein has some variation in the splits. The 3Dircadb dataset only contains 20 CT scans as mention earlier in chapter 3, which make this dataset more likely to overfit or underfit. The result is that the validation and test split only contain two images each, which makes it hard to see general trends in the results. This is most likely the reason why split-1 has very close maximum and minimum dice score.

Both the best and worst predictions show that the model has some problems with finding some of the lower parts of the portal vein. The CT scans with the corresponding prediction slice may indicate these NP's is in the areas with low contrast or intensity. This may be caused by not choosing the best Hounsfield level and window, which can be found by analyzing the CT images further and testing them with the model. This is probably the reason why the worst prediction in split-0 has many FP looking like smaller veins only in the top area. Another reason may be that the training images have too different characteristics from the validation and testing images.

From the training graphs of the coronary artery, Aorta, and portal vein, there is a difference between the validation and training result, as the training graphs is much better than the validation graphs. This is an indication of overfitting. The reason why there is some variation between the splits may be that some splits have easier validation images, or by some of the random events mention earlier.

5.3 BVNet3D

The following discussion of the results from experiment 3 will provide an answer to RQ3.

The results of the portal vein segmentation have less FP than the coronary artery and Aorta segmentation. This is probably because the ground truth of the portal vein is taking up more space in the CT images than the other two, which is resulting in less removed patches for the portal vein. Both the result of the coronary artery and portal vein have almost none of the smaller arteries/veins, which is a disadvantage for the BVNet model.

There is a large variation in the portal vein structure, which can be seen in the ground truth of the test images from the different splits in chapter 4. Most of them have in common a larger vein in the middle, which may be why the model leans this part best and have problem with the smaller veins. This problem can be solved by getting more images or adding data argumentation.

The BVNet3D model is better than the BVNet model at finding more of the ground truth masks and making fewer gaps in general. The BVNet3D model contains more FP than the BVNet model, which is most likely why it has a lower dice score. Most of the FP is located next to the artery with some looking like ribs or another tubular structure. The BVNet3D model has the same problem as the BVNet model with predicting the correct amount of the connection to the Aorta, which could be the result of some inconsistency labeling in this area. The worst prediction in split-0 is more or less the same for BVNet and BVNet3D, which indicates that this image is an edge case of the dataset.

The training performance graphs show that the model is overfitting because there is a huge difference between training and validation loss. The validation loss has values very close to 1 in all cases with some minor variation. This means that most of the validation samples are very different from the ones used for training. This is correct because all of the patches only containing background are removed in the training set.

In the case of Aorta, the dice score of the different splits become lower for BVNet3D than the BVNet model, which may be the result of it having fewer patches samples to train on. The Aorta is a relative huge artery, but it is only in a minor part of the CT images. This is probably the reason why the model has started to predict another relative huge artery/vessel close to the spine, as it has a structure similar to the Aorta. The relative huge artery may be predicted because the patches of this tubular structure look a lot like the other Aorta patches.

The major part of the FP is around the Aorta, and it is to be expected that the model wrongly predicts samples it has not seen before. The partly cubic shaped predictions may be caused by the patch size of $(64, 64, 64)$ not being optimal. The reason may be that some of the Aorta ground truth is in the endpoints of the patches.

A way to improve the model is to use overlapping patches for the training set. This will create more patches of the different connections areas and greater variations on where the ground truth is located in the patches. Another way is to add some previous and next patches as channels like done with slices for experiment 1, which helps the model by giving more context to the predicted slice but with patches in this case. Another disadvantage with this method will be the high computational cost.

5.4 SegCaps

In this section is some information on the SegCaps model discussed in order to answer the RQ4. CapsuleNet was first used for segmentation last year, and it is a relatively new field. The experimentation done by laLonde explained in subsection 2.3.3 was done by segmenting lungs of CT slices. The ground truth of the lungs will in most cases be in over 50% of the slice, which makes it easier to find than arteries/vessels, which sometimes are in less than 10% of the given slice.

The training performance gaps show that all of the SegCaps models have a low generalization gap compared to BVNet and BVNet3D. This is good when the optimal prediction should have a low number of NP for all the cases. The result of experiment 4 has a large difference in the dice score between the segmentation on the different labels. The coronary artery segmentation is founding a majority of the ground truth with some FP and the Aorta and portal vein containing a lot of FP and a few FP.

The worst prediction of the Aorta and portal vein have a lot of FP consisting of most of the tissue/organs close to the given label. The capsule has a vector instead of scalar containing some information about some features like intensity, rotation, etc., which is supposed to make the model more robust against changes. In this case, the capsules are making the difference between the maximum and minimum dice score lower than the BVNet and BVNet3D model, but the capsules may have confused the SegCaps model trained on Aorta. In some cases is another large tubular structure predicted with only few TP of the Aorta, where the post processing step removes the TP. The artery/vessel prediction and the Aorta prediction are having a large tubular structure, which may have led them to have a very similar capsule representation.

The prediction of the other tubular structure with few TP is most likely a result of low training performance because it predicts most of the spine instead of the Aorta in split-2. .

The Aorta and portal veins contain less training slices than the coronary artery because the Aorta ground truth are in fewer slices in the CT image and the portal vein is collected from only 16 CT images. This is probably one of the reasons why the Aorta and portal vein is harder to generalize than the coronary artery.

The SegCaps model does almost not learning anything when segmenting the Aorta and portal vein. This could be the effect of Sigmoid in the last layer, Adam, and dice loss not being the best combination when learning to segment these labels. Another reason could be that the routing between the capsules is not optimal or that they are hard to generalize.

The best portal vein prediction contains few vessel structures and a majority of the liver. This may be a result of the few samples and not the most optimal Hounsfield level and window. The BVNet and BVNet3D models nevertheless did not have trouble with this, but the SegCaps model can be sensitive to it.

The training performance graphs of the portal vein show that the model some minor improvement in the loss value during the training, which means that it almost did not learn anything. This fits with the visualizations of the results, which almost did not remove anything from the CT image.

The results of the SegCaps model is not as good as the BVNet and BVNet3D model, which could be caused by the SegCaps model only uses one channel of predicting the results. The coronary artery prediction had a similar result to the BVNet model with channel 1 but had a lower variance between the minimum and maximum dice score. The experiment 1 showed that the BVNet model increased the dice score by increasing the channel size. This is why the SegCaps model may increase the dice score by increasing the channel size like in experiment 1. The BVNet with channel 1 had a similar result but higher variance between the minimum and maximum dice score. Because this model was not tested on the Aorta and portal vein, it is hard to compare.

5.5 Frangi Filter

The result in section 4.5 has gathered information about the result of using the Frangi filter with BVNet model in order to answer RQ5. As mention in the section 5.1, the result needs to have connected TP without FP in the inner radius of the arteries to be used in the FFR project.

The worst predicted result of split-0 and split-1 did almost predict an empty image, which may be the result of not using the optimal hyperparameters to create the Frangi filtered images. This may explain why there is a huge difference in the results of the splits and the validation results. The best predictions are not on the same test image as the BVNet model without the Frangi filter. The Frangi input prediction image has fewer gaps in the TP and more FN than the BVNet with CT input prediction.

Some of the images have low intensity in some of the coronary artery locations, which could confuse the model. This is probably one of the reasons why the usage of the Frangi filter did not improve the mean dice score.

The combination of both CT and Frangi input have increased the minimum and maximum dice score for split-0 and split-1 compared to the BVNet with regular CT images input. In this case, the worst prediction has fewer gaps between the TP in the LM. The predictions of split-2 have more FP, which is located at the lung vessels shown by the slices in section E.1.

A way to improve the predictions could be to centralize the input image around that image intensity of the coronary artery for being able to use the same Frangi hyperparameters in general. In this case, the Frangi filter may be a lot less confused about which tubular structures it is supposed to find. Both the Frangi input and the combination model sometimes needs to guess there are parts of the coronary artery that are not visible in the Frangi filtered CT image.

The combination model has less FP in the worst case than the BVNet3D model. Both these model contain fewer gaps in the main branches, but the BVNet3D model is better at finding smaller arteries. This may be caused by a combination of the smaller arteries having less intensity in the Frangi filtered images and the BVNet3D model getting more information about the surroundings.

The combination input model of split-1 prediction has more FP connected to the outer radius than the other models. This may be caused by the model assuming some of the background around the higher intensity areas is part of the coronary artery. Another reason may be that some part of the ground truth incorrectly labeled as background.

To summarize the Frangi filter can in some cases improving the dice score in general. The TP is more connected as the predictions in BVNet3D, but in most cases contain less FP around the coronary artery. More testing is needed for finding the most optimal Frangi filter hyperparameters in order to improve the model even further.

5.6 Reflection

These models are computationally expensive, which is resulting in the calculations of one epoch is taking around an hour and others more than an hour. Most of the models train for around 45 epochs, which means that these models used around two days for training. This thesis had a lot of experimenting with the different model architecture and different label to predict because of the training time where has not much room for experimenting with different hyperparameters.

In hindsight, the focus should not have been to experiment with the Aorta and portal vein labels, but only focusing on the coronary artery. On the other side, these experiments provided a greater understanding of the benefits and disadvantages of the models when segmenting tubular structures in general. For improving the coronary artery predictions, more analysis on the dataset should have been done. This could have made it easier to know which of the preprocessing techniques to use on the CT images. Due to the time limitations of this thesis, there was only time to analyze a selected number of the CT images.

The post processing in this thesis did not provide most different in the dice score. This is mainly because of the large difference between the minimum and maximum dice score.

By using a more strict post processing would in some cases give a significant better dice score, but in most cases would have removed a lot of the TP. In the case of the BVNet3D and SegCaps, the FP sometimes got larger than the TP, which made it difficult to make a post processing method that worked in most of the cases. This was not prioritized in this thesis.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has experimented with different approaches for segmenting the coronary artery from CT images by using deep learning models. The wanted predictions from the models need to focus on finding the ground truth without FP in the inner radius of the arteries for all the CT images. The model should not mainly focus on removing the FP unconnected to the coronary artery because it can be removed by using a better post processing step.

Experiment 1 tested out the BVNet model with different channels and stride, which shows that increasing the number of channels will, in some cases, result in more of the wanted prediction characteristics. Because it gets a better understanding of the surroundings by the 3D effect, some other result indicates that using stride 2 may increase the dice score with some exceptions. The optimal channel size and stride depends on whether it is more beneficial to have fewer correct prediction with a lower computation cost or more correct predictions with higher computation cost.

The one with most connected TP and best mean Dice$_{post}$ score was BVNet with 7 channels and stride 2, and BVNet with 5 channels and stride 1 had the best Dice$_{raw}$ score.

The BVNet with 5 channels and stride 1 were used for the other experiments because it has lower computational cost than the model with 7 channels and stride 2. For both the coronary artery and portal vein, the BVNet model had problems with finding small arteries/veins and had in general many unconnected TP. The BVNet model was best at predicting the Aorta, which means it is good at finding large tubular structures.

The BVNet3D model is the model with most TP for the coronary artery and portal vein, and a lower prediction of TP in Aorta than the BVNet model. The increment of the FP may be caused by removing patches with only containing background.

BVNet3D did find most of the coronary artery but had a lot of FP around. Removing the input patch only containing background did most likely be the reason for the high validation loss and the FP around. This model found most of the portal vein, which means that it can probably be used on other artery or vessels as well with some further development, as explained below. The patches of the Aorta were too few after the removing of the only background patches, which resulted in making the BVNet model better than the BVNet3D model. The BVNet3D model predicted less TP and most more FP than the BVNet model.

The SegCaps model is the worst of the three models, but it shows potential. This is because it uses 1 channel and has a smaller gap between the training and validation values than the BVNet and BVNet3D model. This means it probably does not need much data augmentation for getting better results, but fine-tuning the hyperparameters. One large disadvantage of the SegCaps model is that it does not learn much when used on the Aorta and portal vein label. This is most likely caused by using too few training samples, which makes them harder to generalize. Another reason may be that not the optimal optimizer, loss function, or hyperparameters were chosen.

The use of Frangi filtered input does not increase the mean dice score, but sometimes it reduces the gaps between the TP, especially when using combined CT and Frangi filtered CT images as input. In many of the predictions on some the corresponding CT slices shows that some the ground truth is in very dark areas, which make them hard to find. By finding the optimal hyperparameters for the Frangi filter, the dice score will probably increase.

None of the models used in this thesis are good enough to be used together with fluid dynamics for automatically measure the FFR without further development.

6.2 Future Work

As mention in the discussion, there is a lot of improvement that can be done with the different models, which may result in good enough predictions for the FFR-project. The first step is to find the optimal Hounsfield level and window for the coronary artery in order to make it easier for the model to separate it from the other tissue. For finding the correct values, more analyzation of the dataset is needed.

As shown in the result is the combination of CT and Frangi filtered CT images as input helping the BVNet containing less gaps in the TP, which shows great potential. The next step is to find the optimal values of the hyperparameters of the Frangi filter for getting the intensity at the coronary artery higher. A way to further develop the BVNet3D model and SegCaps model is to experiment with the combined CT and Frangi filtered CT images as input.

The next step for the SegCaps model is to add some previous and next slices in order to get a 3D effect, as experiment 1 did for the BVNet model. The same can be done for the BVNet3D only with patches instead of slices. Another improvement regarding the SegCaps model is to find the optimal loss function, optimizer, learning rate, and other hyperparameter values.

Another improvement that can be done in general is to either add more data or apply data augmentation to decrease the overfitting problem. Experimenting with not removing the only background slices or patches needs to be done for testing whether it will decrease the number of FP without decreasing the TP. For helping the training, transfers learning should be tested by transferring weights of a large dataset of arteries/vessels for centralizing the training. This could also be done by first training the coronary artery on FrangiNet explained in subsection 2.3.1 and use its weights as a starting point for the other models.

Bibliography

[1] J Ed Barnes. Characteristics and control of contrast in ct. *Radiographics*, 12(4):825–837, 1992.

[2] Mahua Bhattacharya, GK Sharma, et al. Optimized coronary artery segmentation using frangi filter and anisotropic diffusion filtering. In *2013 International Symposium on Computational and Business Intelligence*, pages 261–264. IEEE, 2013.

[3] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.

[4] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[5] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[6] Fredrik E Fossan, Jacob Sturdy, Lucas O Müller, Andreas Strand, Anders T Bråten, Arve Jørgensen, Rune Wiseth, and Leif R Hellevik. Uncertainty quantification and sensitivity analysis for computational ffr estimation in stable coronary artery disease. *Cardiovascular engineering and technology*, 9(4):597–622, 2018.

[7] Weilin Fu, Katharina Breininger, Roman Schaffert, Nishant Ravikumar, Tobias Würfl, Jim Fujimoto, Eric Moult, and Andreas Maier. Frangi-net. In *Bildverarbeitung für die Medizin 2018*, pages 341–346. Springer, 2018.

[8] Woods R. Gonzalez, R. *Digital Image Processing*. Pearson Education International, 2010.

[9] Bengio Y. Goodfellow, I. and A. Courville. *Deep Learning*. MIT Press, 2016.

[10] Göran K Hansson. Inflammation, atherosclerosis, and coronary artery disease. *New England Journal of Medicine*, 352(16):1685–1695, 2005.

[11] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[12] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018.

[13] Frode Jacobsen. Fully-automatic coronary artery segmentation using deep convolutional autoencoders. Master's thesis, NTNU, 2018.

[14] Daniel Jimenez-Carretero, Andres Santos, Sjoerd Kerkstra, Rina Dewi Rudyanto, and Maria J Ledesma-Carbayo. 3d frangi-based lung vessel enhancement filter penalizing airways. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 926–929. IEEE, 2013.

[15] Trude Jostad. Coronary artery segmentation using deep learning. `https://drive.google.com/file/d/1Fiexm9NzatmSGY1qUIvELQ--_F8KuzHt/view?usp=sharing/`. (Accessed on 06/09/2019).

[16] Øyvind Kjerland. Segmentation of coronary arteries from ct-scans of the heart using deep learning. Master's thesis, NTNU, 2017.

[17] Rodney LaLonde and Ulas Bagci. Capsules for object segmentation. *arXiv preprint arXiv:1804.04241*, 2018.

[18] Ha Manh Luu, Camiel Klink, Adriaan Moelker, Wiro Niessen, and Theo Van Walsum. Quantitative evaluation of noise reduction and vesselness filters for liver vessel segmentation on abdominal cta images. *Physics in Medicine & Biology*, 60(10):3905, 2015.

[19] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[20] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press USA, 2015.

[21] World Health Organization. The top 10 causes of death. `http://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death`. (Accessed on 06/11/2019).

[22] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[24] Holger R Roth, Hirohisa Oda, Xiangrong Zhou, Natsuki Shimizu, Ying Yang, Yuichiro Hayashi, Masahiro Oda, Michitaka Fujiwara, Kazunari Misawa, and Kensaku Mori. An application of cascaded 3d fully convolutional networks for medical image segmentation. *Computerized Medical Imaging and Graphics*, 66:90–99, 2018.

[25] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[26] Taraldsen T. Sjaastad, I. *Anatomi og fysiologi, NA*. Medisinsk basislitteratur, 2001.

[27] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

Appendix A

Experiment 1: Channels and Stride

In this appendix is all the results of the models with different channel size and stride from experiment 1. The tables below were used to create the Table 4.1 and to find the prediction with the minimum and maximum dice score for each of the models.

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.251	0.246	0.257	0.238	0.245	0.255
CT_FFR_19	0.296	0.276	0.21	0.186	0.504	0.537
CT_FFR_28	0.611	0.608	0.463	0.455	0.896	0.915
CT_FFR_32	0.355	0.322	0.232	0.201	0.761	0.807
CT_FFR_39	0.674	0.641	0.588	0.511	0.789	0.859
CT_FFR_RETRO_14	0.113	0.075	0.06	0.039	0.891	0.949
CT_FFR_50	0.037	0.029	0.019	0.015	0.671	0.63
CT_FFR_RETRO_29	0.373	0.393	0.528	0.515	0.288	0.318
CT_FFR_RETRO_64	0.205	0.145	0.126	0.085	0.551	0.502
Average Scores	0.324	0.304	0.276	0.249	0.622	0.641

Table A.1: Result of split-0 of the BVNet with 1 channels and 1 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.247	0.24	0.161	0.145	0.526	0.689
CT_FFR_19	0.587	0.56	0.462	0.421	0.804	0.835
CT_FFR_28	0.649	0.65	0.5	0.495	0.925	0.947
CT_FFR_32	0.67	0.686	0.62	0.587	0.728	0.825
CT_FFR_39	0.809	0.819	0.827	0.826	0.791	0.813
CT_FFR_RETRO_14	0.292	0.253	0.175	0.148	0.871	0.889
CT_FFR_50	0.374	0.364	0.258	0.245	0.681	0.715
CT_FFR_RETRO_29	0.675	0.681	0.616	0.601	0.747	0.786
CT_FFR_RETRO_64	0.424	0.394	0.292	0.26	0.775	0.806
Average Scores	0.525	0.516	0.435	0.414	0.761	0.812

Table A.2: Result of split-0 of the BVNet with 3 channels and 1 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.234	0.22	0.177	0.16	0.344	0.351
CT_FFR_19	0.677	0.66	0.54	0.511	0.906	0.933
CT_FFR_28	0.743	0.744	0.625	0.623	0.916	0.924
CT_FFR_32	0.658	0.671	0.62	0.618	0.702	0.734
CT_FFR_39	0.825	0.832	0.847	0.844	0.805	0.82
CT_FFR_RETRO_14	0.242	0.179	0.144	0.101	0.773	0.778
CT_FFR_50	0.562	0.563	0.506	0.498	0.631	0.648
CT_FFR_RETRO_29	0.619	0.634	0.669	0.653	0.576	0.617
CT_FFR_RETRO_64	0.412	0.383	0.295	0.26	0.682	0.727
Average Scores	0.553	0.543	0.491	0.474	0.704	0.726

Table A.3: Result of split-0 of the BVNet with 3 channels and 2 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.248	0.245	0.204	0.189	0.315	0.349
CT_FFR_19	0.506	0.464	0.375	0.326	0.78	0.808
CT_FFR_28	0.513	0.505	0.353	0.343	0.935	0.961
CT_FFR_32	0.68	0.693	0.641	0.633	0.723	0.765
CT_FFR_39	0.757	0.761	0.673	0.665	0.866	0.888
CT_FFR_RETRO_14	0.344	0.314	0.215	0.191	0.866	0.876
CT_FFR_50	0.554	0.56	0.461	0.459	0.693	0.717
CT_FFR_RETRO_29	0.53	0.559	0.473	0.452	0.602	0.732
CT_FFR_RETRO_64	0.456	0.437	0.314	0.292	0.836	0.875
Average Scores	0.51	0.504	0.412	0.394	0.735	0.775

Table A.4: Result of split-0 of the BVNet with 3 channels and 5 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.361	0.359	0.269	0.258	0.55	0.589
CT_FFR_19	0.728	0.72	0.614	0.595	0.895	0.91
CT_FFR_28	0.782	0.78	0.687	0.681	0.908	0.914
CT_FFR_32	0.544	0.553	0.505	0.5	0.588	0.618
CT_FFR_39	0.794	0.8	0.868	0.865	0.731	0.744
CT_FFR_RETRO_14	0.455	0.441	0.318	0.301	0.798	0.824
CT_FFR_50	0.499	0.493	0.428	0.415	0.598	0.607
CT_FFR_RETRO_29	0.647	0.668	0.672	0.662	0.624	0.673
CT_FFR_RETRO_64	0.525	0.495	0.412	0.374	0.725	0.734
Average Scores	0.593	0.59	0.53	0.517	0.713	0.735

Table A.5: Result of split-0 of the BVNet with 5 channels and 1 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.239	0.213	0.156	0.132	0.511	0.544
CT_FFR_19	0.641	0.616	0.49	0.458	0.927	0.937
CT_FFR_28	0.728	0.727	0.592	0.587	0.945	0.954
CT_FFR_32	0.729	0.764	0.755	0.755	0.705	0.774
CT_FFR_39	0.777	0.781	0.71	0.708	0.858	0.871
CT_FFR_RETRO_14	0.34	0.284	0.208	0.167	0.929	0.937
CT_FFR_50	0.629	0.635	0.594	0.584	0.669	0.694
CT_FFR_RETRO_29	0.684	0.678	0.593	0.579	0.809	0.819
CT_FFR_RETRO_64	0.223	0.173	0.127	0.095	0.904	0.964
Average Scores	0.555	0.541	0.469	0.452	0.806	0.833

Table A.6: Result of split-0 of the BVNet with 5 channels and 2 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.279	0.258	0.171	0.153	0.755	0.819
CT_FFR_19	0.658	0.642	0.537	0.507	0.852	0.874
CT_FFR_28	0.704	0.697	0.573	0.562	0.911	0.918
CT_FFR_32	0.753	0.759	0.806	0.766	0.706	0.752
CT_FFR_39	0.801	0.807	0.782	0.78	0.821	0.837
CT_FFR_RETRO_14	0.287	0.249	0.173	0.146	0.843	0.854
CT_FFR_50	0.634	0.634	0.579	0.574	0.701	0.708
CT_FFR_RETRO_29	0.601	0.6	0.585	0.572	0.618	0.631
CT_FFR_RETRO_64	0.349	0.316	0.215	0.189	0.933	0.955
Average Scores	0.563	0.551	0.491	0.472	0.793	0.816

Table A.7: Result of split-0 of the BVNet with 5 channels and 5 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.288	0.294	0.248	0.239	0.343	0.385
CT_FFR_19	0.581	0.573	0.437	0.419	0.865	0.906
CT_FFR_28	0.638	0.637	0.489	0.482	0.92	0.935
CT_FFR_32	0.754	0.771	0.755	0.753	0.754	0.79
CT_FFR_39	0.783	0.792	0.741	0.738	0.831	0.854
CT_FFR_RETRO_14	0.321	0.288	0.196	0.169	0.875	0.963
CT_FFR_50	0.665	0.678	0.621	0.615	0.715	0.755
CT_FFR_RETRO_29	0.516	0.541	0.617	0.604	0.443	0.489
CT_FFR_RETRO_64	0.348	0.268	0.228	0.162	0.732	0.768
Average Scores	0.544	0.538	0.481	0.465	0.72	0.761

Table A.8: Result of split-0 of the BVNet with 7 channels and 1 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.29	0.293	0.258	0.252	0.332	0.35
CT_FFR_19	0.678	0.661	0.569	0.542	0.839	0.847
CT_FFR_28	0.764	0.754	0.664	0.64	0.9	0.918
CT_FFR_32	0.704	0.728	0.765	0.765	0.652	0.695
CT_FFR_39	0.829	0.836	0.843	0.84	0.816	0.833
CT_FFR_RETRO_14	0.39	0.375	0.252	0.238	0.863	0.879
CT_FFR_50	0.671	0.677	0.746	0.739	0.61	0.624
CT_FFR_RETRO_29	0.57	0.579	0.643	0.637	0.511	0.531
CT_FFR_RETRO_64	0.415	0.403	0.297	0.279	0.686	0.726
Average Scores	0.59	0.59	0.56	0.548	0.69	0.711

Table A.9: Result of split-0 of the BVNet with 7 channels and 2 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.291	0.287	0.207	0.194	0.492	0.546
CT_FFR_19	0.674	0.627	0.565	0.486	0.834	0.885
CT_FFR_28	0.741	0.742	0.622	0.621	0.916	0.922
CT_FFR_32	0.736	0.752	0.736	0.734	0.737	0.771
CT_FFR_39	0.841	0.854	0.84	0.84	0.842	0.869
CT_FFR_RETRO_14	0.303	0.277	0.189	0.168	0.763	0.792
CT_FFR_50	0.604	0.603	0.518	0.508	0.724	0.741
CT_FFR_RETRO_29	0.634	0.649	0.648	0.643	0.62	0.656
CT_FFR_RETRO_64	0.425	0.399	0.291	0.264	0.789	0.821
Average Scores	0.583	0.577	0.513	0.495	0.746	0.778

Table A.10: Result of split-0 of the BVNet with 7 channels and 5 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.231	0.219	0.175	0.158	0.342	0.356
CT_FFR_19	0.644	0.623	0.517	0.489	0.855	0.858
CT_FFR_28	0.703	0.698	0.571	0.561	0.915	0.923
CT_FFR_32	0.755	0.768	0.768	0.73	0.743	0.809
CT_FFR_39	0.815	0.831	0.84	0.827	0.792	0.836
CT_FFR_RETRO_14	0.238	0.189	0.15	0.116	0.574	0.526
CT_FFR_50	0.631	0.635	0.651	0.647	0.612	0.623
CT_FFR_RETRO_29	0.536	0.542	0.662	0.647	0.451	0.467
CT_FFR_RETRO_64	0.373	0.356	0.26	0.239	0.658	0.701
Average Scores	0.547	0.54	0.51	0.49	0.66	0.678

Table A.11: Result of split-0 of the BVNet with 9 channels and 1 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.171	0.144	0.118	0.096	0.306	0.292
CT_FFR_19	0.478	0.446	0.326	0.296	0.896	0.906
CT_FFR_28	0.525	0.519	0.367	0.36	0.924	0.932
CT_FFR_32	0.705	0.703	0.629	0.602	0.802	0.845
CT_FFR_39	0.828	0.833	0.773	0.758	0.892	0.925
CT_FFR_RETRO_14	0.099	0.054	0.052	0.028	0.871	0.986
CT_FFR_50	0.641	0.641	0.537	0.534	0.794	0.801
CT_FFR_RETRO_29	0.583	0.6	0.556	0.552	0.613	0.656
CT_FFR_RETRO_64	0.207	0.105	0.122	0.058	0.679	0.544
Average Scores	0.471	0.449	0.387	0.365	0.753	0.765

Table A.12: Result of split-0 of the BVNet with 9 channels and 2 in stride

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.198	0.19	0.168	0.151	0.241	0.257
CT_FFR_19	0.545	0.52	0.431	0.396	0.743	0.757
CT_FFR_28	0.654	0.657	0.524	0.522	0.868	0.886
CT_FFR_32	0.575	0.599	0.566	0.566	0.585	0.637
CT_FFR_39	0.777	0.845	0.821	0.819	0.736	0.873
CT_FFR_RETRO_14	0.255	0.213	0.152	0.122	0.778	0.844
CT_FFR_50	0.613	0.619	0.544	0.54	0.702	0.726
CT_FFR_RETRO_29	0.417	0.435	0.491	0.477	0.362	0.4
CT_FFR_RETRO_64	0.409	0.4	0.323	0.299	0.557	0.607
Average Scores	0.494	0.498	0.447	0.432	0.619	0.665

Table A.13: Result of split-0 of the BVNet with 9 channels and 5 in stride

Appendix B

Experiment 2: BVNet

In this appendix is the results of the segmentation of the coronary artery, Aorta, and portal vein using BVNet model, where each subsection contain tables of the dice score, recall, and precision of the predictions on the test splits used for calculating the Table 4.2 for the coronary arteries, Table 4.3 for Aorta, and Table 4.4 for the portal vein. Each of the subsections has as well a figure of the best and worst prediction slices on the corresponding CT slice, which was used as input.

B.1 Coronary Artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.233	0.223	0.215	0.19	0.255	0.272
CT_FFR_19	0.574	0.564	0.479	0.453	0.714	0.748
CT_FFR_28	0.703	0.705	0.603	0.601	0.842	0.854
CT_FFR_32	0.718	0.752	0.732	0.73	0.704	0.776
CT_FFR_39	0.774	0.795	0.816	0.813	0.737	0.779
CT_FFR_RETRO_14	0.278	0.255	0.177	0.153	0.655	0.778
CT_FFR_50	0.643	0.648	0.63	0.624	0.657	0.673
CT_FFR_RETRO_29	0.559	0.601	0.659	0.647	0.485	0.56
CT_FFR_RETRO_64	0.377	0.357	0.276	0.244	0.594	0.668
Average Scores	0.54	0.545	0.51	0.495	0.627	0.679

Table B.1: Result of split-0 of the BVNet for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.445	0.444	0.371	0.346	0.556	0.62
CT_FFR_37	0.616	0.654	0.774	0.751	0.511	0.58
CT_FFR_18	0.658	0.635	0.516	0.465	0.909	1.0
CT_FFR_40	0.769	0.834	0.865	0.864	0.692	0.807
CT_FFR_8	0.721	0.73	0.599	0.591	0.903	0.955
CT_FFR_RETRO_86	0.781	0.784	0.666	0.663	0.945	0.96
CT_FFR_RETRO_89	0.539	0.525	0.515	0.363	0.566	0.949
CT_FFR_Pilot_7	0.622	0.642	0.533	0.525	0.745	0.824
Average Scores	0.644	0.656	0.605	0.571	0.728	0.837

Table B.2: Result of split-1 of the BVNet for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.693	0.704	0.697	0.694	0.69	0.715
CT_FFR_25	0.627	0.651	0.801	0.799	0.515	0.549
CT_FFR_4	0.803	0.81	0.806	0.774	0.8	0.85
CT_FFR_RETRO_23	0.547	0.548	0.39	0.383	0.912	0.963
CT_FFR_RETRO_83	0.664	0.664	0.658	0.592	0.67	0.756
CT_FFR_16	0.69	0.734	0.681	0.649	0.699	0.844
CT_FFR_Pilot_3	0.631	0.606	0.476	0.441	0.937	0.97
CT_FFR_33	0.686	0.694	0.67	0.666	0.703	0.725
Average Scores	0.668	0.676	0.647	0.625	0.741	0.796

Table B.3: Result of split-2 of the BVNet for the coronary artery

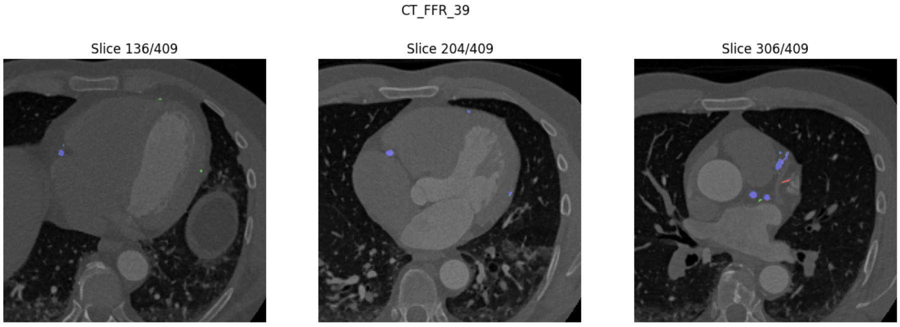
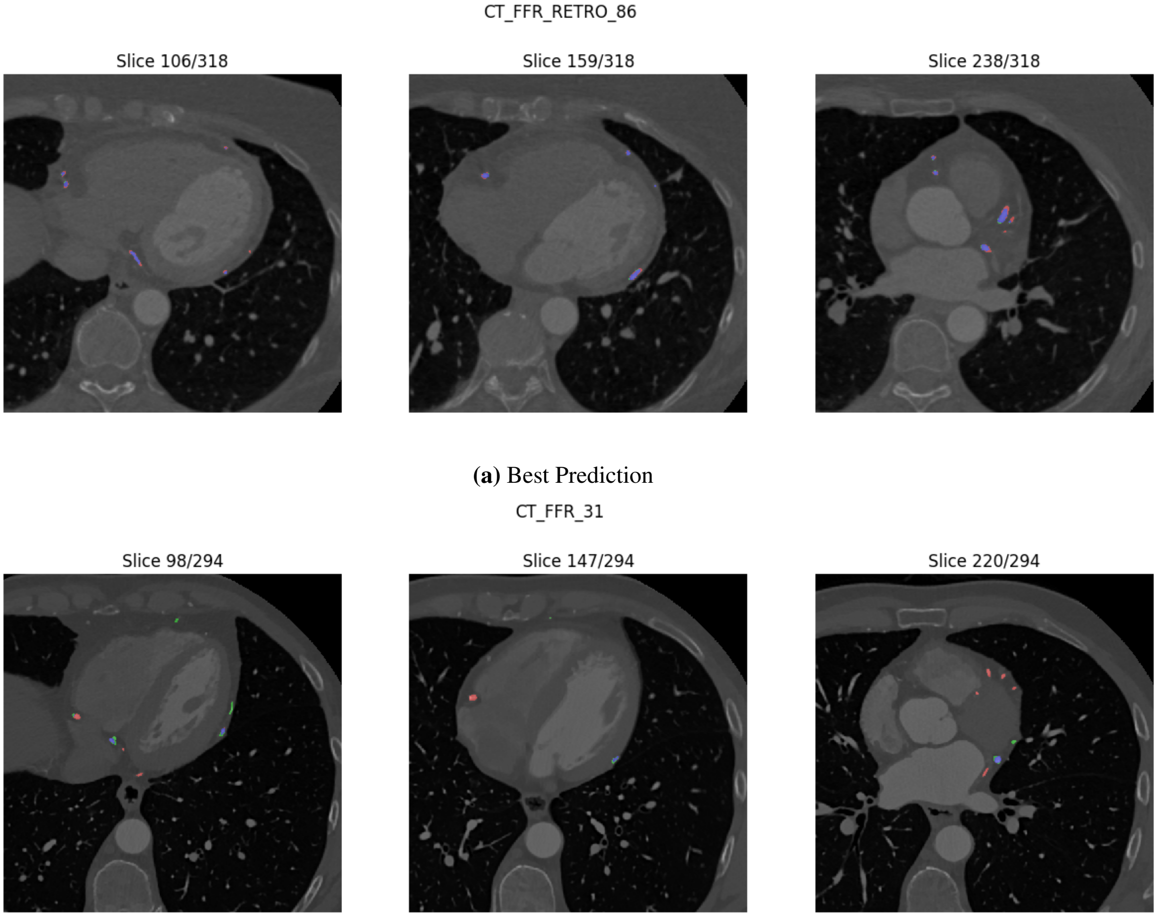


Figure B.1: A selection of some of the CT slices of the BVNet model's best prediction of the coronary artery split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best Prediction

(b) Worst Prediction

Figure B.2: A selection of some of the CT slices of the BVNet model's best and worst prediction of the coronary artery split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

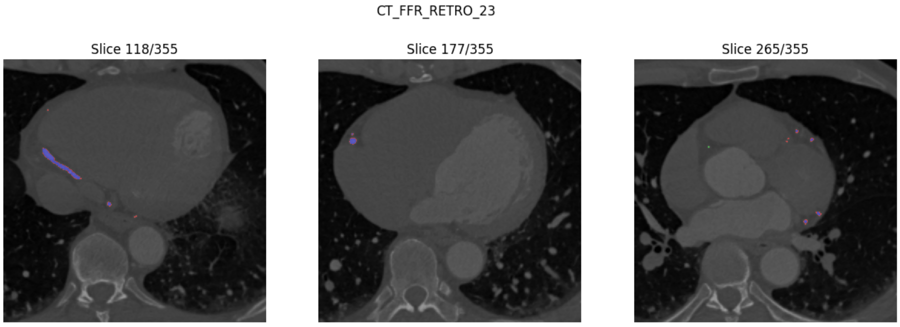


Figure B.3: A selection of some of the CT slices of the BVNet model's worst prediction of the coronary artery split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

B.2 Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.682	0.728	0.577	0.577	0.833	0.986
CT_FFR_19	0.951	0.966	0.964	0.964	0.939	0.968
CT_FFR_28	0.918	0.923	0.934	0.934	0.901	0.912
CT_FFR_32	0.945	0.947	0.976	0.976	0.916	0.919
CT_FFR_39	0.958	0.963	0.988	0.988	0.929	0.938
CT_FFR_RETRO_14	0.925	0.936	0.888	0.887	0.966	0.99
CT_FFR_50	0.959	0.963	0.981	0.981	0.939	0.946
CT_FFR_RETRO_29	0.908	0.909	0.889	0.889	0.929	0.929
CT_FFR_RETRO_64	0.933	0.935	0.879	0.879	0.994	0.999
Average Scores	0.909	0.919	0.897	0.897	0.927	0.954

Table B.4: Result of split-0 of the BVNet for Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.815	0.878	0.855	0.855	0.779	0.901
CT_FFR_37	0.939	0.946	0.911	0.911	0.969	0.983
CT_FFR_18	0.875	0.972	0.973	0.973	0.795	0.971
CT_FFR_40	0.912	0.968	0.958	0.958	0.87	0.978
CT_FFR_8	0.917	0.967	0.978	0.978	0.863	0.957
CT_FFR_RETRO_86	0.925	0.944	0.926	0.926	0.923	0.963
CT_FFR_RETRO_89	0.798	0.8	0.94	0.94	0.694	0.696
CT_FFR_Pilot_7	0.899	0.935	0.968	0.968	0.839	0.905
Average Scores	0.885	0.926	0.939	0.939	0.841	0.919

Table B.5: Result of split-1 of the BVNet for Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.939	0.971	0.978	0.978	0.903	0.964
CT_FFR_25	0.964	0.967	0.994	0.994	0.936	0.941
CT_FFR_4	0.91	0.923	0.935	0.935	0.886	0.912
CT_FFR_RETRO_23	0.934	0.936	0.894	0.893	0.978	0.983
CT_FFR_RETRO_83	0.959	0.968	0.97	0.97	0.948	0.967
CT_FFR_16	0.969	0.985	0.994	0.994	0.945	0.976
CT_FFR_Pilot_3	0.935	0.94	0.943	0.943	0.926	0.936
CT_FFR_33	0.947	0.986	0.976	0.976	0.919	0.996
Average Scores	0.944	0.959	0.96	0.96	0.93	0.959

Table B.6: Result of split-2 of the BVNet for Aorta

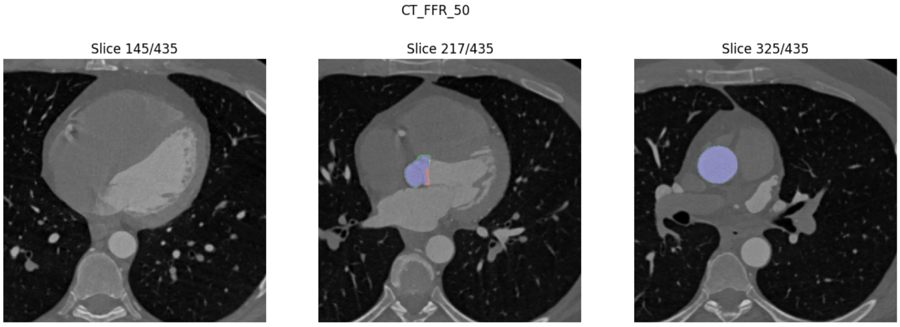
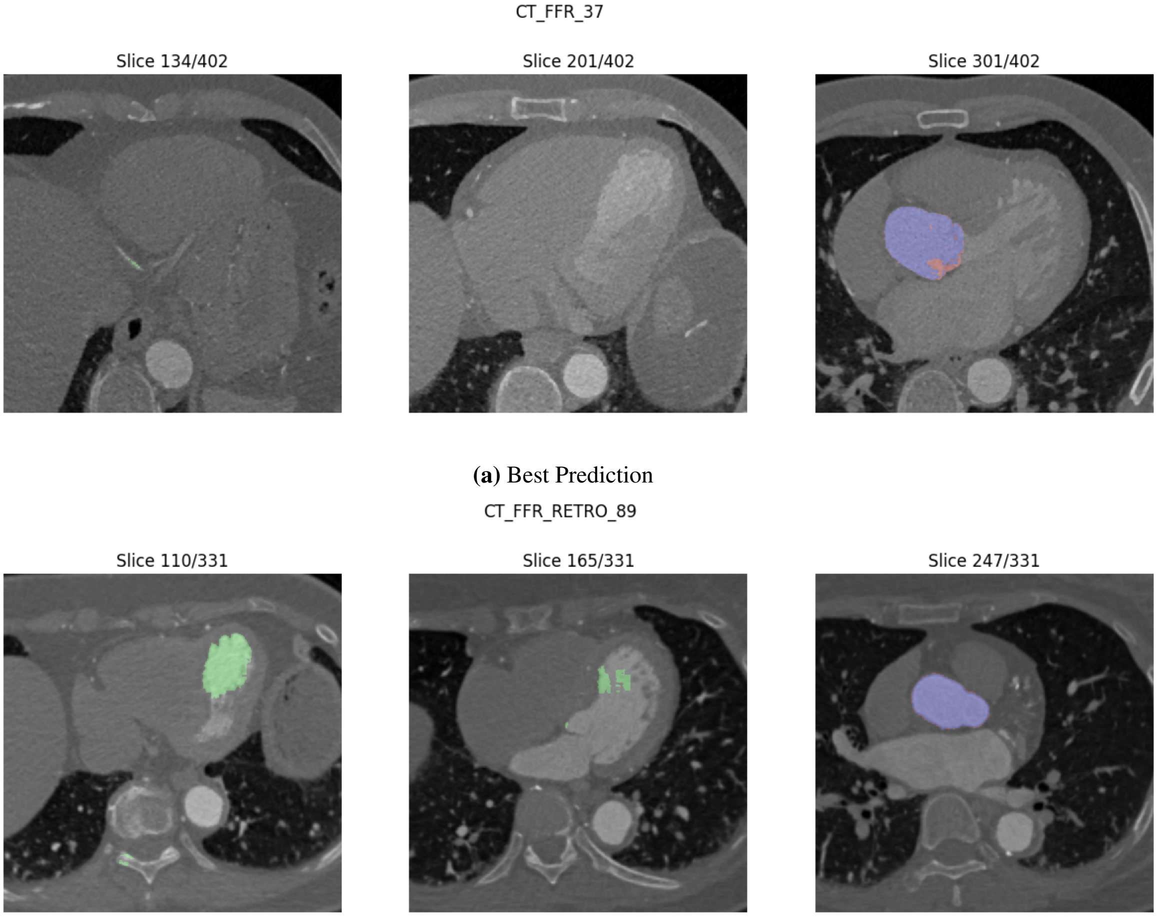


Figure B.4: A selection of some of the CT slices of the BVNet model's best prediction of the Aorta split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best Prediction

(b) Worst Prediction

Figure B.5: A selection of some of the CT slices of the BVNet model's best and worst prediction of the Aorta split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

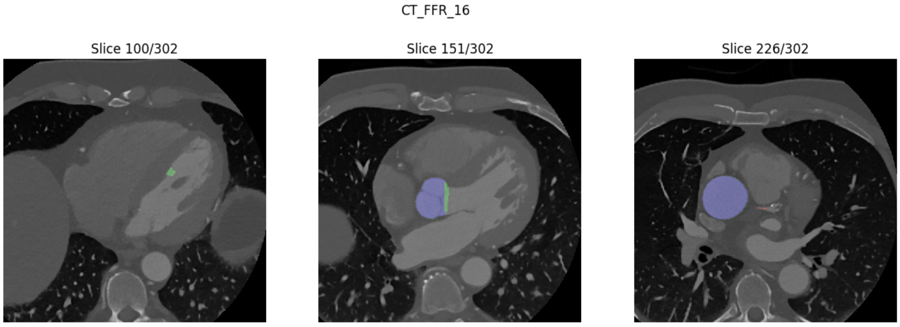


Figure B.6: A selection of some of the CT slices of the BVNet model's worst prediction of the Aorta split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

B.3 Portalvein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.20	0.244	0.261	0.462	0.402	0.166	0.193
3Dircadb1.7	0.323	0.332	0.451	0.44	0.252	0.266
Average Scores	0.284	0.296	0.457	0.421	0.209	0.23

Table B.7: Result of split-0 of the BVNet for the portal vein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.9	0.141	0.128	0.104	0.088	0.22	0.234
3Dircadb1.17	0.406	0.423	0.406	0.32	0.407	0.624
Average Scores	0.274	0.275	0.255	0.204	0.314	0.429

Table B.8: Result of split-1 of the BVNet for the portal vein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.6	0.362	0.368	0.292	0.28	0.477	0.536
3Dircadb1.18	0.356	0.371	0.597	0.589	0.254	0.271
Average Scores	0.359	0.369	0.445	0.434	0.365	0.403

Table B.9: Result of split-2 of the BVNet for the portal vein

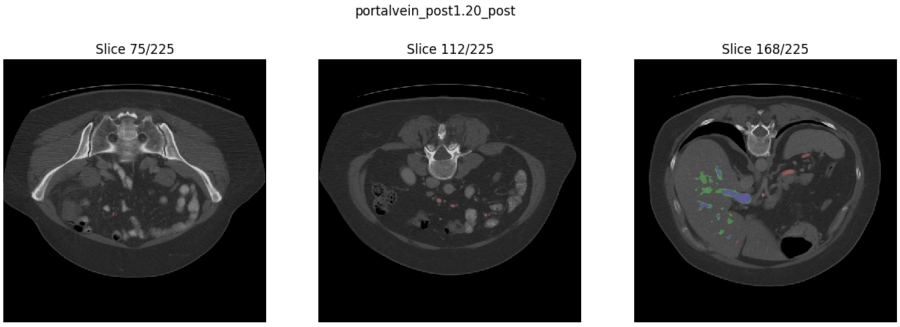


Figure B.7: A selection of some of the CT slices of the BVNet model's worst prediction of the portal vein split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

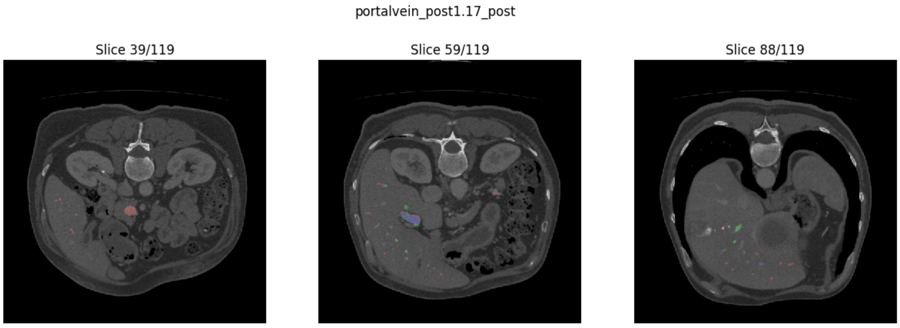
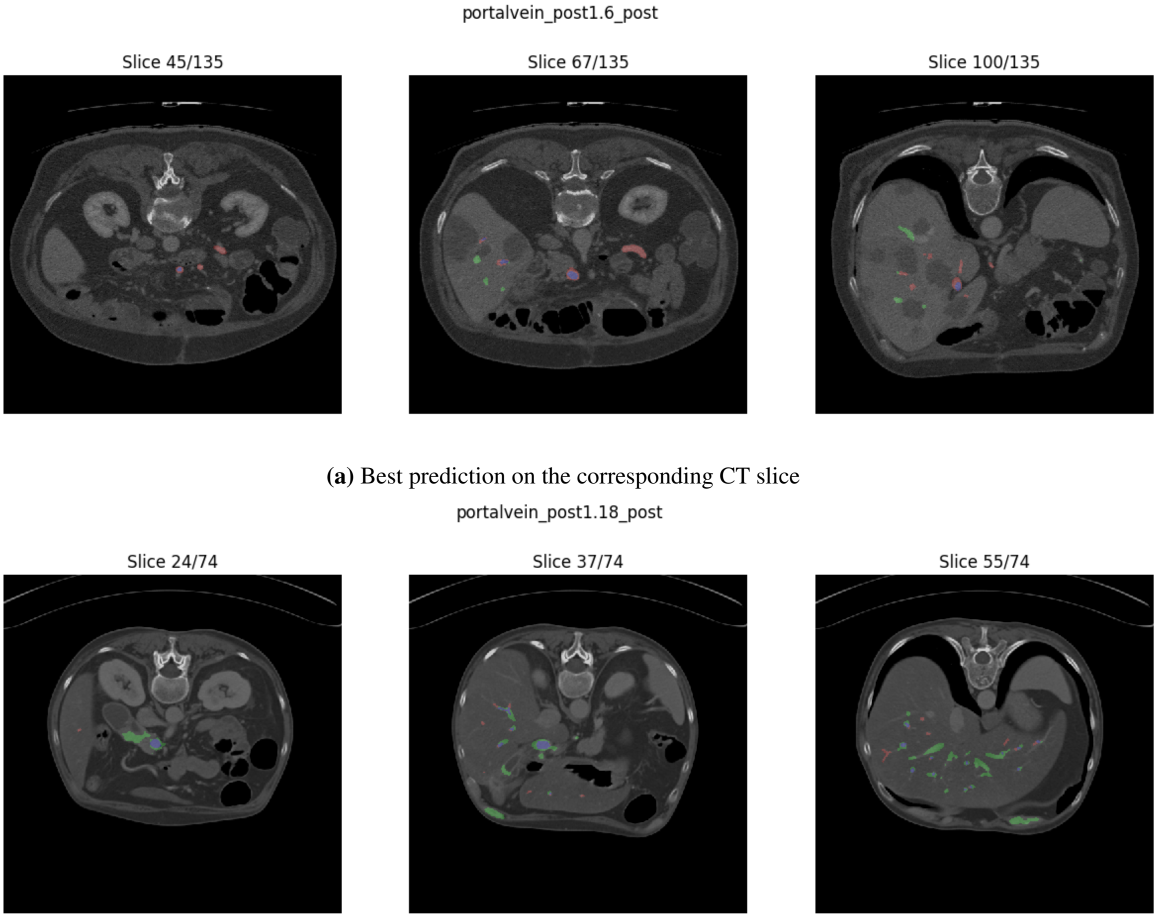


Figure B.8: A selection of some of the CT slices of the BVNet model's best prediction of the portal vein split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best prediction on the corresponding CT slice

(b) Worst prediction on the corresponding CT slice

Figure B.9: A selection of some of the CT slices of the BVNet model's best and worst prediction of the portal vein split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

Appendix C

Experiment 3: BVNet3D

In this appendix is all the individual test image result for each of the splits for the segmentation of the coronary artery, Aorta, and portal vein using the BVNet3D model.

C.1 Coronary Artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.185	0.182	0.192	0.175	0.178	0.19
CT_FFR_19	0.548	0.582	0.547	0.536	0.548	0.637
CT_FFR_28	0.561	0.584	0.524	0.521	0.603	0.664
CT_FFR_32	0.647	0.797	0.835	0.792	0.528	0.802
CT_FFR_39	0.667	0.713	0.893	0.888	0.532	0.596
CT_FFR_RETRO_14	0.235	0.211	0.181	0.143	0.335	0.399
CT_FFR_50	0.615	0.63	0.707	0.705	0.544	0.57
CT_FFR_RETRO_29	0.31	0.334	0.721	0.69	0.198	0.22
CT_FFR_RETRO_64	0.232	0.216	0.212	0.168	0.257	0.304
Average Scores	0.444	0.472	0.535	0.513	0.414	0.487

Table C.1: Result of split-0 of the BVNet3D for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.56	0.586	0.735	0.725	0.452	0.492
CT_FFR_37	0.367	0.408	0.799	0.799	0.238	0.274
CT_FFR_18	0.388	0.411	0.597	0.578	0.287	0.319
CT_FFR_40	0.474	0.514	0.909	0.909	0.32	0.359
CT_FFR_8	0.73	0.814	0.718	0.695	0.742	0.982
CT_FFR_RETRO_86	0.621	0.678	0.671	0.66	0.578	0.696
CT_FFR_RETRO_89	0.302	0.32	0.639	0.629	0.198	0.215
CT_FFR_Pilot_7	0.396	0.432	0.619	0.618	0.291	0.331
Average Scores	0.48	0.52	0.711	0.702	0.388	0.459

Table C.2: Result of split-1 of the BVNet3D for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.413	0.448	0.715	0.715	0.29	0.326
CT_FFR_25	0.434	0.479	0.901	0.901	0.286	0.326
CT_FFR_4	0.7	0.723	0.826	0.826	0.608	0.643
CT_FFR_RETRO_23	0.48	0.492	0.418	0.408	0.563	0.62
CT_FFR_RETRO_83	0.495	0.535	0.732	0.709	0.374	0.429
CT_FFR_16	0.655	0.691	0.792	0.785	0.559	0.618
CT_FFR_Pilot_3	0.517	0.542	0.553	0.542	0.486	0.542
CT_FFR_33	0.456	0.488	0.771	0.763	0.323	0.358
Average Scores	0.519	0.55	0.713	0.706	0.436	0.483

Table C.3: Result of split-2 of the BVNet3D for the coronary artery

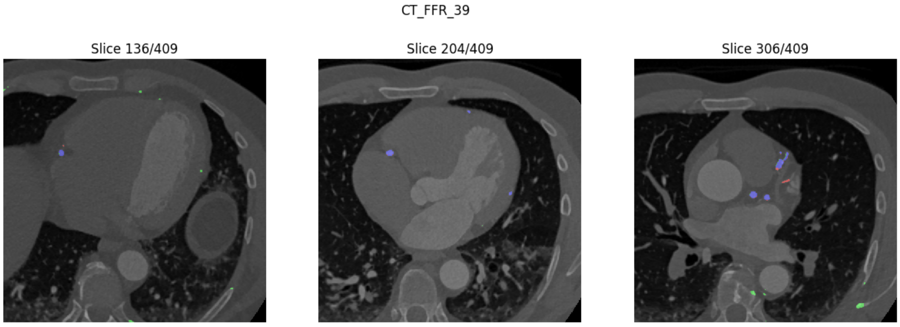


Figure C.1: A selection of some of the CT slices of the BVNet3D model's best prediction of the coronary artery split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

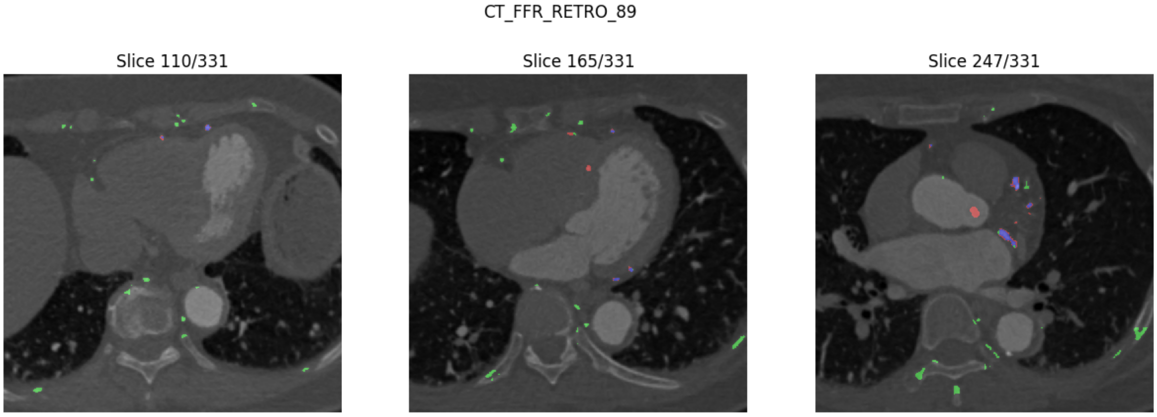
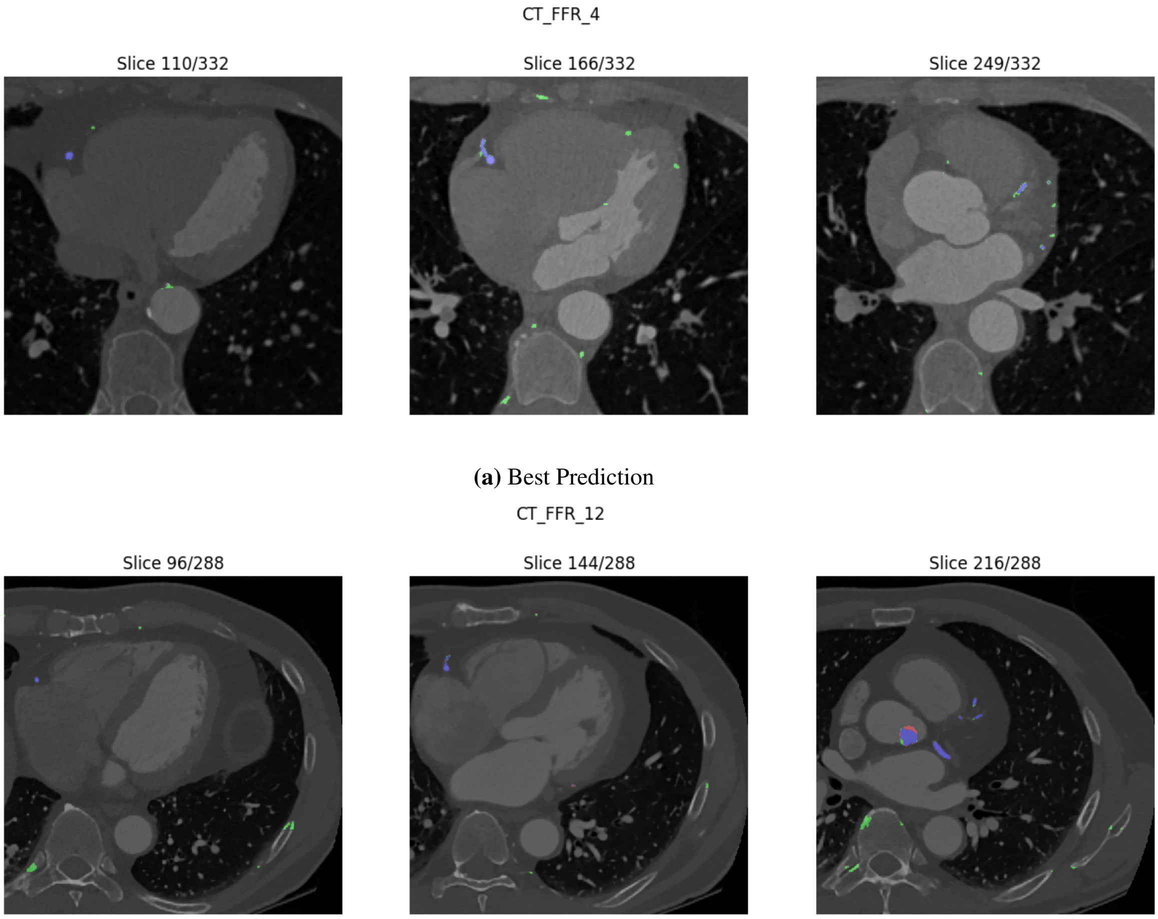


Figure C.2: A selection of some of the CT slices of the BVNet3D model's worst prediction of the coronary artery split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best Prediction

(b) Worst Prediction

Figure C.3: A selection of some of the CT slices of the BVNet3D model's best and worst prediction of the coronary artery split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

C.2 Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.395	0.396	0.411	0.41	0.38	0.383
CT_FFR_19	0.52	0.521	0.918	0.918	0.363	0.364
CT_FFR_28	0.377	0.377	0.938	0.938	0.236	0.236
CT_FFR_32	0.368	0.369	0.922	0.922	0.23	0.231
CT_FFR_39	0.618	0.62	0.994	0.994	0.449	0.451
CT_FFR_RETRO_14	0.474	0.475	0.734	0.734	0.35	0.351
CT_FFR_50	0.42	0.421	0.98	0.98	0.267	0.268
CT_FFR_RETRO_29	0.517	0.581	0.887	0.886	0.365	0.433
CT_FFR_RETRO_64	0.742	0.781	0.748	0.746	0.736	0.821
Average Scores	0.492	0.505	0.837	0.836	0.375	0.393

Table C.4: Result of split-0 of the BVNet3D for Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.367	0.367	0.993	0.993	0.225	0.225
CT_FFR_37	0.653	0.661	0.96	0.96	0.495	0.504
CT_FFR_18	0.45	0.451	0.981	0.981	0.292	0.293
CT_FFR_40	0.566	0.566	0.986	0.986	0.396	0.397
CT_FFR_8	0.592	0.63	0.944	0.944	0.431	0.473
CT_FFR_RETRO_86	0.584	0.586	0.943	0.943	0.423	0.425
CT_FFR_RETRO_89	0.431	0.432	0.929	0.929	0.281	0.281
CT_FFR_Pilot_7	0.474	0.476	0.97	0.97	0.314	0.316
Average Scores	0.515	0.521	0.963	0.963	0.357	0.364

Table C.5: Result of split-1 of the BVNet3D for Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.44	0.443	0.99	0.99	0.283	0.285
CT_FFR_25	0.468	0.47	0.993	0.993	0.306	0.308
CT_FFR_4	0.56	0.561	0.943	0.943	0.398	0.399
CT_FFR_RETRO_23	0.624	0.778	0.64	0.64	0.608	0.993
CT_FFR_RETRO_83	0.624	0.626	0.972	0.972	0.46	0.462
CT_FFR_16	0.659	0.661	0.992	0.992	0.493	0.496
CT_FFR_Pilot_3	0.631	0.633	0.941	0.941	0.474	0.477
CT_FFR_33	0.572	0.574	0.974	0.974	0.405	0.407
Average Scores	0.572	0.593	0.931	0.931	0.428	0.478

Table C.6: Result of split-2 of the BVNet3D for Aorta

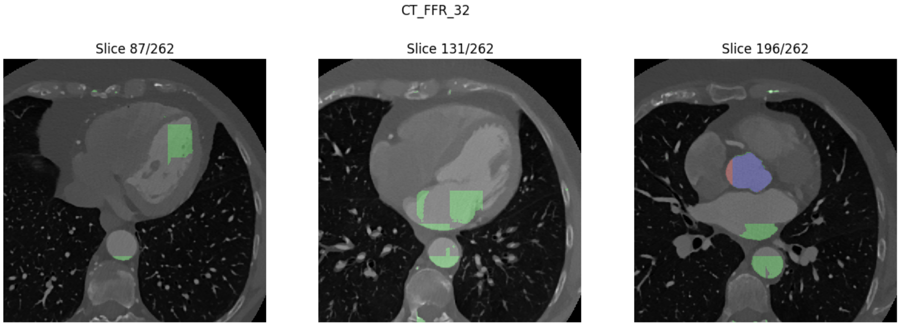


Figure C.4: A selection of some of the CT slices of the BVNet model's worst prediction of the Aorta split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

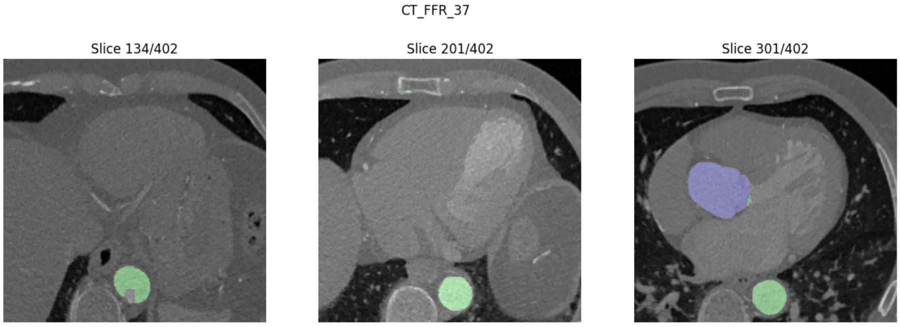
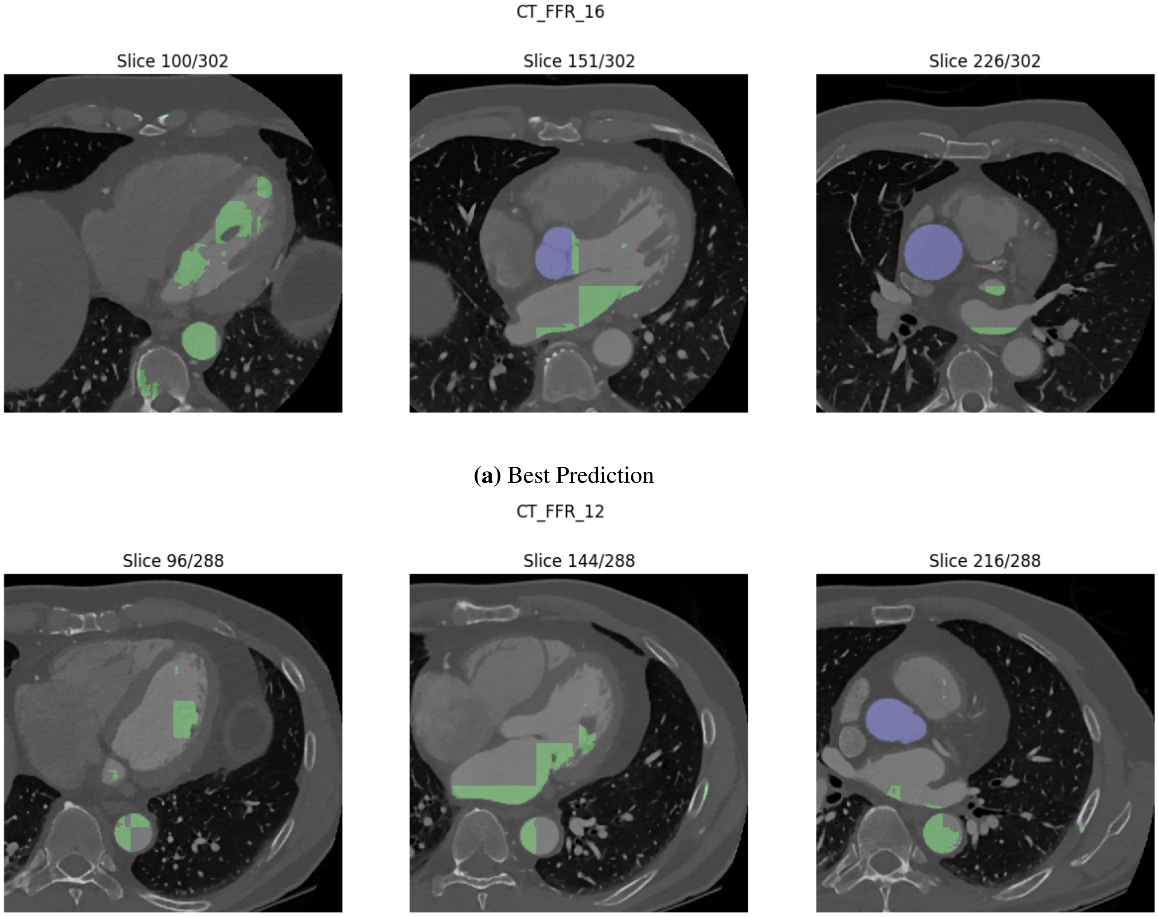


Figure C.5: A selection of some of the CT slices of the BVNet3D model's best prediction of the Aorta split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best Prediction

(b) Worst Prediction

Figure C.6: A selection of some of the CT slices of the BVNet3D model's best and worst prediction of the Aorta split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

C.3 Portal Vein

Test Image	Dice		Recall		Precision	
3Dircadb1.20	0.477	0.517	0.468	0.4	0.486	0.731
3Dircadb1.7	0.476	0.485	0.503	0.479	0.451	0.491
Average Scores	0.476	0.501	0.486	0.44	0.469	0.611

Table C.7: Result of split-0 of the BVNet3D for the portal vein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.9	0.533	0.529	0.445	0.371	0.665	0.923
3Dircadb1.17	0.678	0.663	0.66	0.578	0.697	0.777
Average Scores	0.606	0.596	0.553	0.474	0.681	0.85

Table C.8: Result of split-1 of the BVNet3D with 5 channels and 1 in stride for the portal vein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.6	0.481	0.494	0.387	0.381	0.635	0.703
3Dircadb1.18	0.365	0.359	0.35	0.312	0.381	0.423
Average Scores	0.423	0.427	0.369	0.346	0.508	0.563

Table C.9: Result of split-2 of the BVNet3D for the portal vein

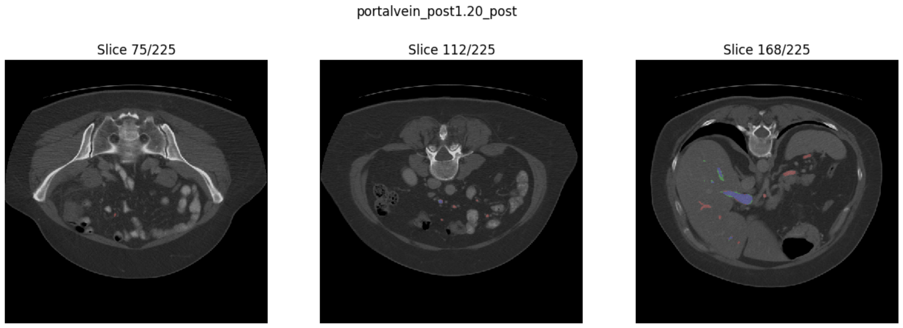


Figure C.7: A selection of some of the CT slices of the BVNet3D model's best prediction of the portal vein split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

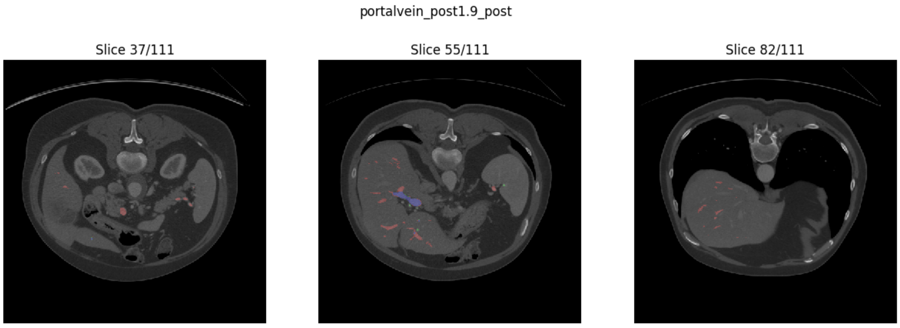
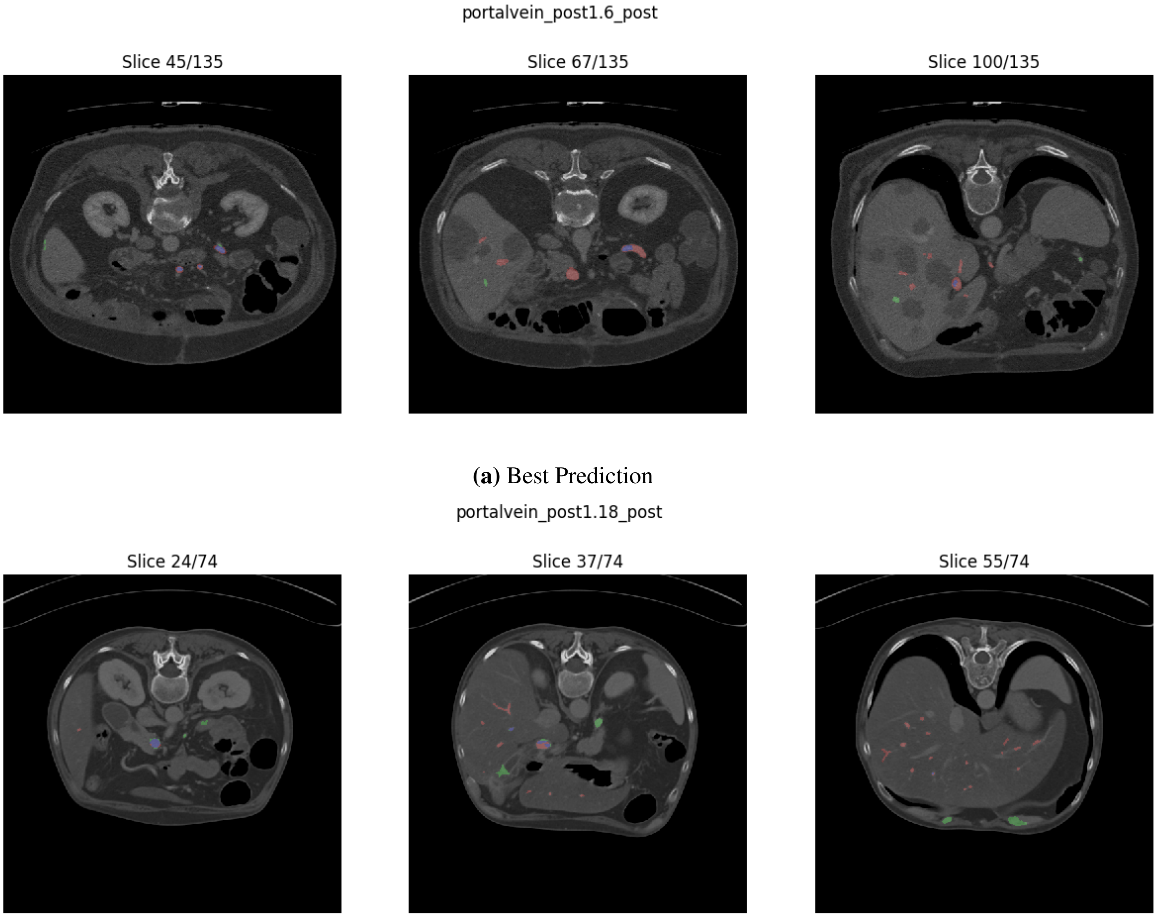


Figure C.8: A selection of some of the CT slices of the BVNet3D model's worst prediction of the portal vein split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best Prediction

(b) Worst Prediction

Figure C.9: A selection of some of the CT slices of the BVNet3D model's best and worst prediction of the portal vein split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

Appendix D

Experiment 4: SegCaps

In this appendix is the result of the different splits containing each test image for the coronary artery, Aorta, and portal vein using the SegCaps model. The tables below are used to create the tables in section 4.4.

D.1 Coronary Artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.46	0.48	0.437	0.422	0.486	0.558
CT_FFR_19	0.612	0.632	0.492	0.478	0.811	0.93
CT_FFR_28	0.705	0.716	0.6	0.597	0.853	0.893
CT_FFR_32	0.683	0.717	0.627	0.621	0.751	0.848
CT_FFR_39	0.77	0.801	0.826	0.821	0.722	0.782
CT_FFR_RETRO_14	0.472	0.466	0.342	0.321	0.76	0.848
CT_FFR_50	0.66	0.668	0.611	0.602	0.717	0.75
CT_FFR_RETRO_29	0.515	0.57	0.768	0.727	0.387	0.469
CT_FFR_RETRO_64	0.518	0.502	0.434	0.391	0.642	0.704
Average Scores	0.599	0.617	0.571	0.553	0.681	0.754

Table D.1: Result of split-0 of the SegCaps for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.29	0.271	0.192	0.163	0.595	0.8
CT_FFR_37	0.286	0.366	0.404	0.359	0.222	0.374
CT_FFR_18	0.424	0.486	0.388	0.359	0.467	0.749
CT_FFR_40	0.363	0.414	0.349	0.32	0.378	0.587
CT_FFR_8	0.443	0.434	0.346	0.297	0.615	0.806
CT_FFR_RETRO_86	0.486	0.501	0.421	0.363	0.576	0.812
CT_FFR_RETRO_89	0.262	0.294	0.35	0.288	0.21	0.301
CT_FFR_Pilot_7	0.361	0.371	0.343	0.303	0.382	0.479
Average Scores	0.365	0.392	0.349	0.306	0.43	0.613

Table D.2: Result of split-1 of SegCaps for the cornary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.439	0.449	0.491	0.475	0.396	0.426
CT_FFR_25	0.579	0.621	0.76	0.758	0.467	0.527
CT_FFR_4	0.667	0.679	0.625	0.61	0.716	0.766
CT_FFR_RETRO_23	0.596	0.606	0.475	0.457	0.802	0.9
CT_FFR_RETRO_83	0.574	0.594	0.494	0.456	0.686	0.851
CT_FFR_16	0.62	0.626	0.519	0.494	0.769	0.855
CT_FFR_Pilot_3	0.638	0.64	0.501	0.473	0.877	0.99
CT_FFR_33	0.513	0.528	0.506	0.468	0.521	0.606
Average Scores	0.578	0.593	0.546	0.524	0.655	0.74

Table D.3: Result of split-2 of SegCaps for the coronary artery

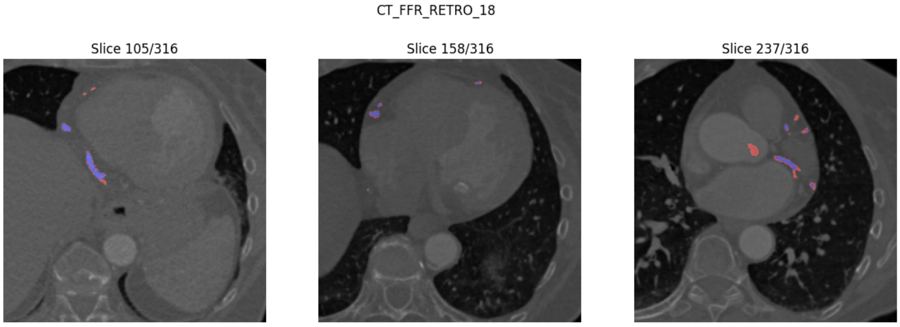


Figure D.1: A selection of some of the CT slices of the SegCaps model's worst prediction of the coronary artery split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

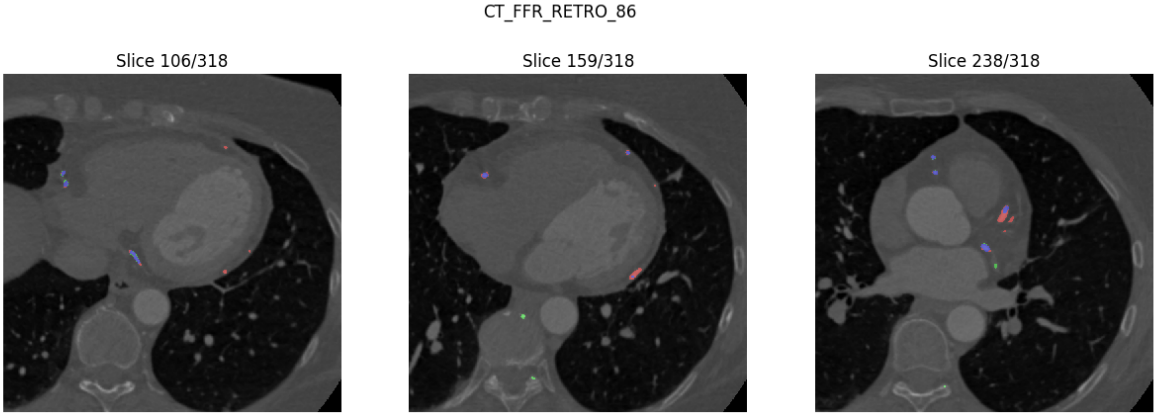
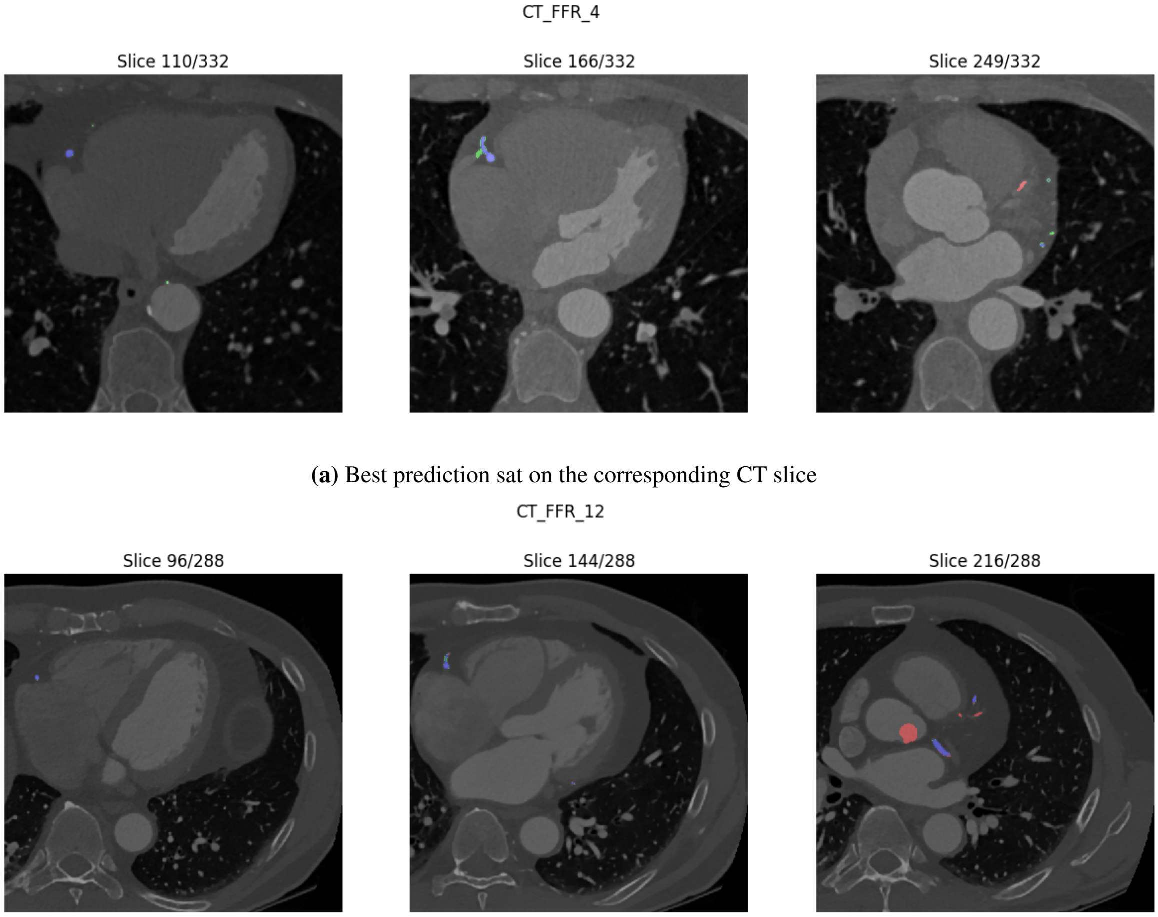


Figure D.2: A selection of some of the CT slices of the SegCaps model's best prediction of the coronary artery split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best prediction sat on the corresponding CT slice

(b) Worst prediction sat on the corresponding CT slice

Figure D.3: A selection of some of the CT slices of the SegCaps model's best and worst prediction of the coronary artery split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

D.2 Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.102	0.0	0.098	0.0	0.107	0.0
CT_FFR_19	0.291	0.302	0.828	0.828	0.176	0.184
CT_FFR_28	0.211	0.212	0.865	0.865	0.12	0.121
CT_FFR_32	0.154	0.162	0.937	0.937	0.084	0.089
CT_FFR_39	0.353	0.387	0.881	0.881	0.221	0.248
CT_FFR_RETRO_14	0.178	0.179	0.586	0.585	0.105	0.105
CT_FFR_50	0.215	0.224	0.976	0.976	0.121	0.126
CT_FFR_RETRO_29	0.256	0.257	0.879	0.879	0.15	0.151
CT_FFR_RETRO_64	0.146	0.145	0.235	0.231	0.106	0.106
Average Scores	0.212	0.207	0.698	0.687	0.132	0.126

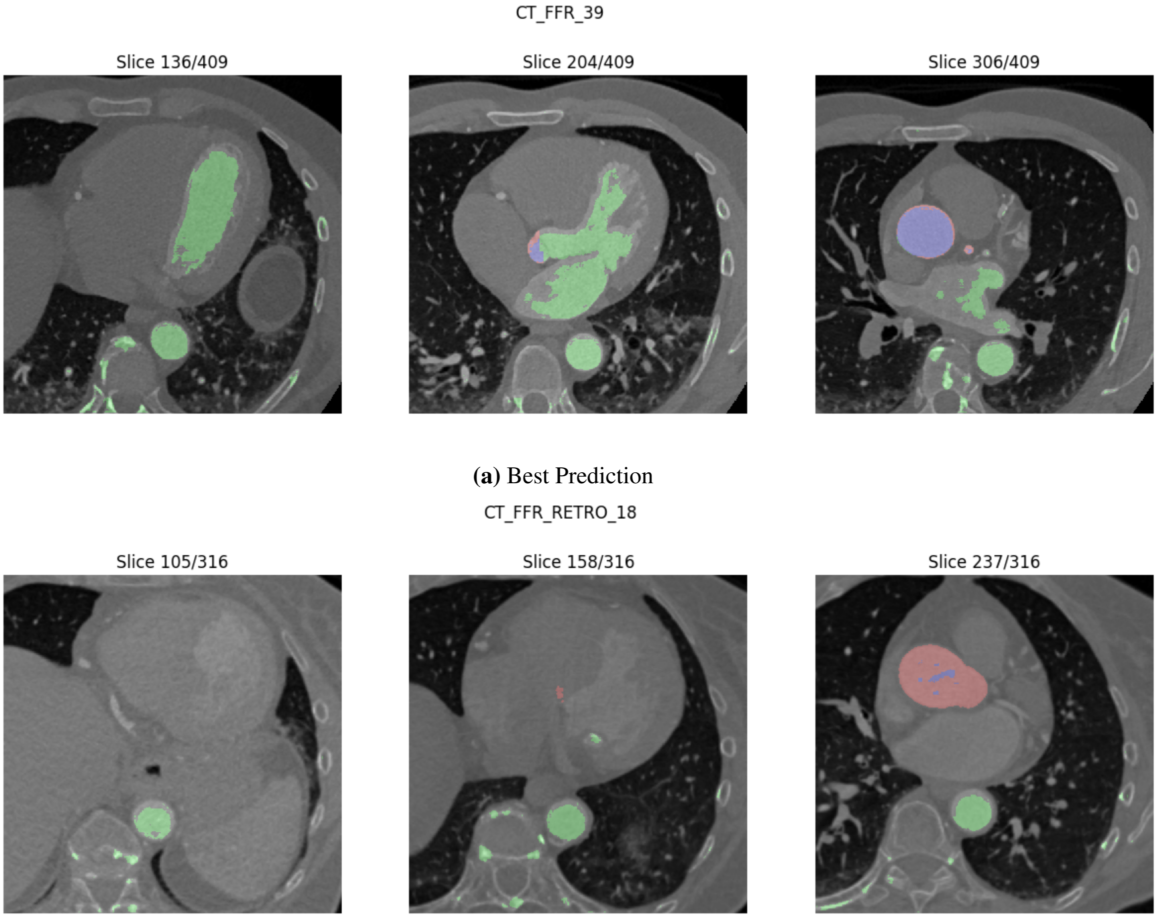
Table D.4: Result of split-0 of SegCaps for Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.196	0.217	0.84	0.84	0.111	0.125
CT_FFR_37	0.668	0.688	0.769	0.768	0.591	0.623
CT_FFR_18	0.441	0.449	0.643	0.643	0.335	0.345
CT_FFR_40	0.467	0.468	0.906	0.906	0.315	0.315
CT_FFR_8	0.587	0.644	0.815	0.815	0.458	0.532
CT_FFR_RETRO_86	0.471	0.484	0.795	0.794	0.334	0.348
CT_FFR_RETRO_89	0.328	0.329	0.821	0.821	0.205	0.206
CT_FFR_Pilot_7	0.282	0.286	0.732	0.731	0.174	0.178
Average Scores	0.43	0.446	0.79	0.79	0.315	0.334

Table D.5: Result of split-1 of SegCaps for Aorta

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.155	0.155	0.84	0.84	0.085	0.086
CT_FFR_25	0.22	0.221	0.959	0.959	0.124	0.125
CT_FFR_4	0.335	0.353	0.942	0.942	0.203	0.217
CT_FFR_RETRO_23	0.047	0.0	0.052	0.0	0.043	0.0
CT_FFR_RETRO_83	0.332	0.333	0.892	0.892	0.204	0.205
CT_FFR_16	0.319	0.32	0.907	0.907	0.194	0.194
CT_FFR_Pilot_3	0.265	0.271	0.699	0.699	0.164	0.168
CT_FFR_33	0.23	0.23	0.834	0.834	0.133	0.133
Average Scores	0.238	0.235	0.766	0.759	0.144	0.141

Table D.6: Result of split-2 of SegCaps for Aorta



(a) Best Prediction

(b) Worst Prediction

Figure D.4: A selection of some of the CT slices of the SegCaps model's best and worst prediction of the Aorta split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

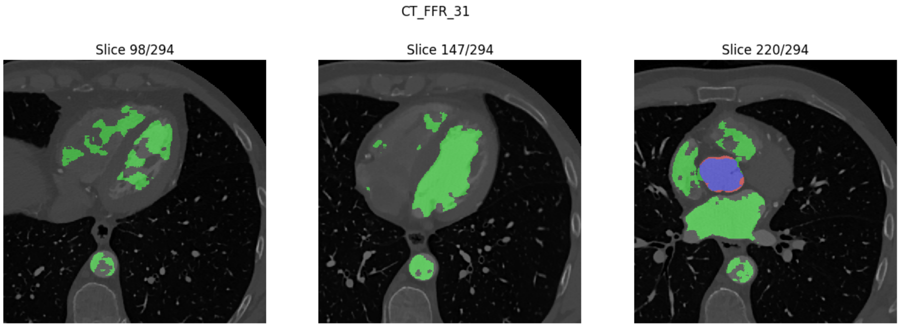


Figure D.5: A selection of some of the CT slices of the SegCaps model's worst prediction of the Aorta split-1, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

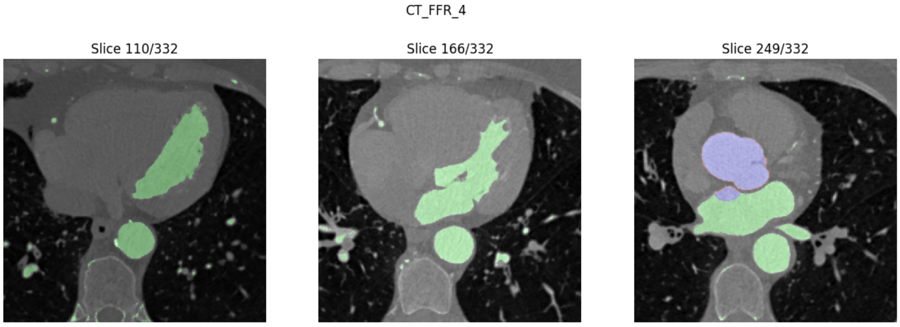


Figure D.6: A selection of some of the CT slices of the SegCaps model's best prediction of the Aorta split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

D.3 Portal Vein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.20	0.038	0.05	0.492	0.486	0.02	0.026
3Dircadb1.7	0.049	0.053	0.933	0.933	0.025	0.027
Average Scores	0.044	0.052	0.712	0.71	0.023	0.027

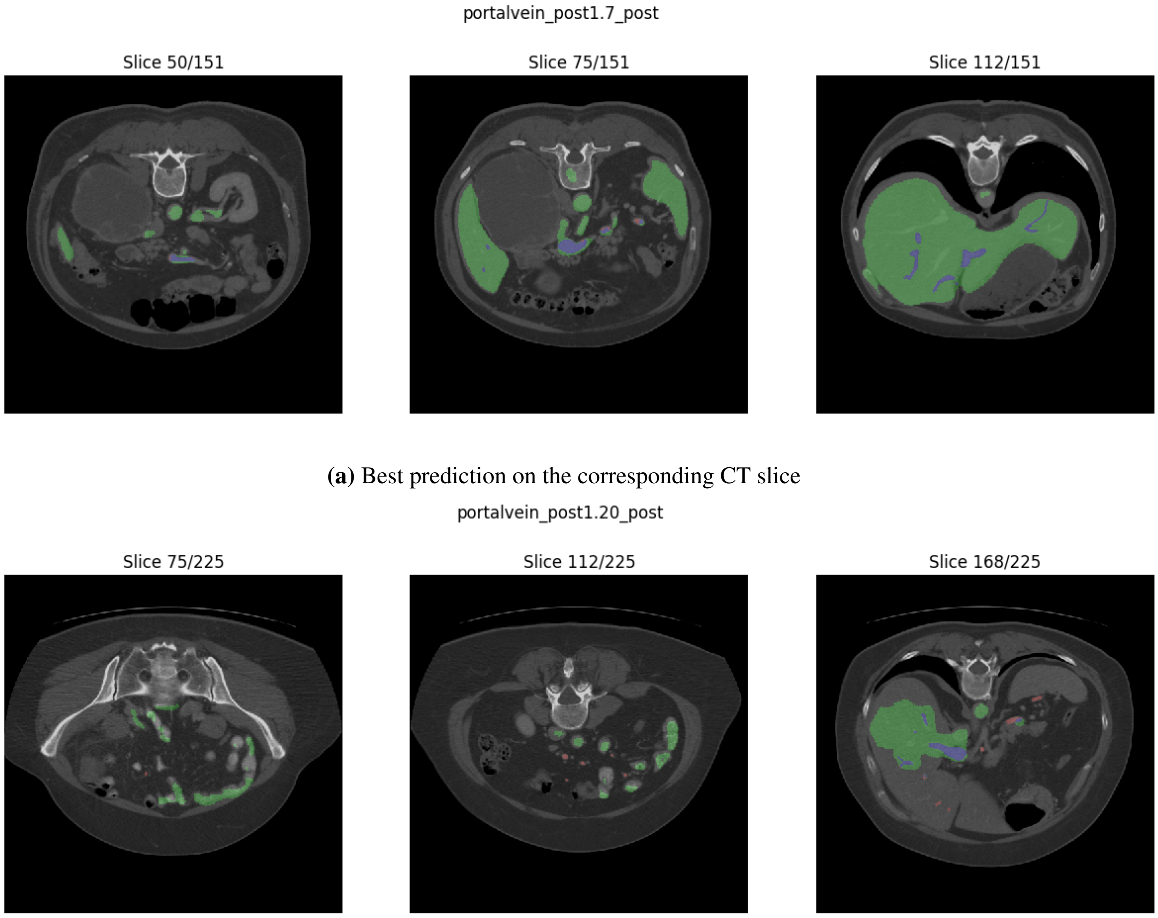
Table D.7: Result of split-0 of SegCaps for the portal vein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.9	0.021	0.018	0.095	0.069	0.012	0.01
3Dircadb1.17	0.114	0.053	0.506	0.172	0.064	0.031
Average Scores	0.067	0.035	0.301	0.12	0.038	0.021

Table D.8: Result of split-1 of SegCaps for the portal vein

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
3Dircadb1.6	0.077	0.083	0.565	0.56	0.041	0.045
3Dircadb1.18	0.043	0.045	0.84	0.84	0.022	0.023
Average Scores	0.06	0.064	0.702	0.7	0.032	0.034

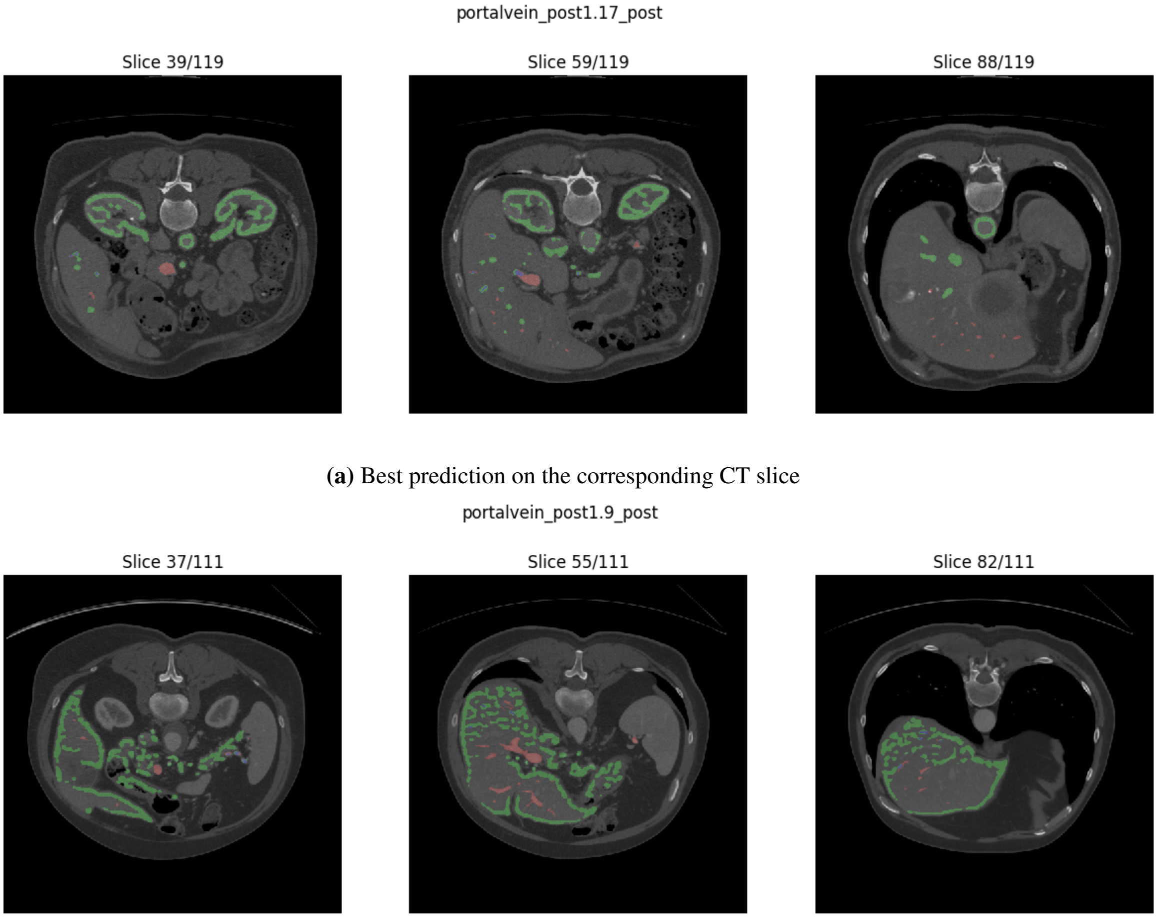
Table D.9: Result of split-2 of SegCaps for the portal vein



(a) Best prediction on the corresponding CT slice

(b) Worst prediction on the corresponding CT slice

Figure D.7: A selection of some of the CT slices of the SegCaps model's best and worst prediction of the portal vein split-0, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN



(a) Best prediction on the corresponding CT slice

(b) Worst prediction on the corresponding CT slice

Figure D.8: A selection of some of the CT slices of the SegCaps model's best and worst prediction of the portal vein split-2, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN

Appendix E

Experiment 5: Frangi Filter

In this appendix is the result of all the different test images of the different splits from experiment 5, which consisted of experimenting with the Frangi filter. The tables below are used to create the summary tables in section 4.5 and the CT and Frangi filtered CT slices with the corresponding prediction overlaid is used to finding the general trends of which tissue is wrongly classified.

E.1 Frangi Input

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.049	0.041	0.044	0.033	0.055	0.055
CT_FFR_19	0.153	0.171	0.222	0.204	0.117	0.147
CT_FFR_28	0.333	0.313	0.204	0.188	0.914	0.923
CT_FFR_32	0.057	0.057	0.03	0.03	0.884	0.956
CT_FFR_39	0.554	0.605	0.636	0.636	0.49	0.577
CT_FFR_RETRO_14	0.106	0.111	0.16	0.134	0.079	0.095
CT_FFR_50	0.636	0.648	0.533	0.524	0.787	0.849
CT_FFR_RETRO_29	0.401	0.429	0.747	0.734	0.274	0.304
CT_FFR_RETRO_64	0.033	0.031	0.024	0.02	0.052	0.065
Average Scores	0.258	0.267	0.289	0.278	0.406	0.441

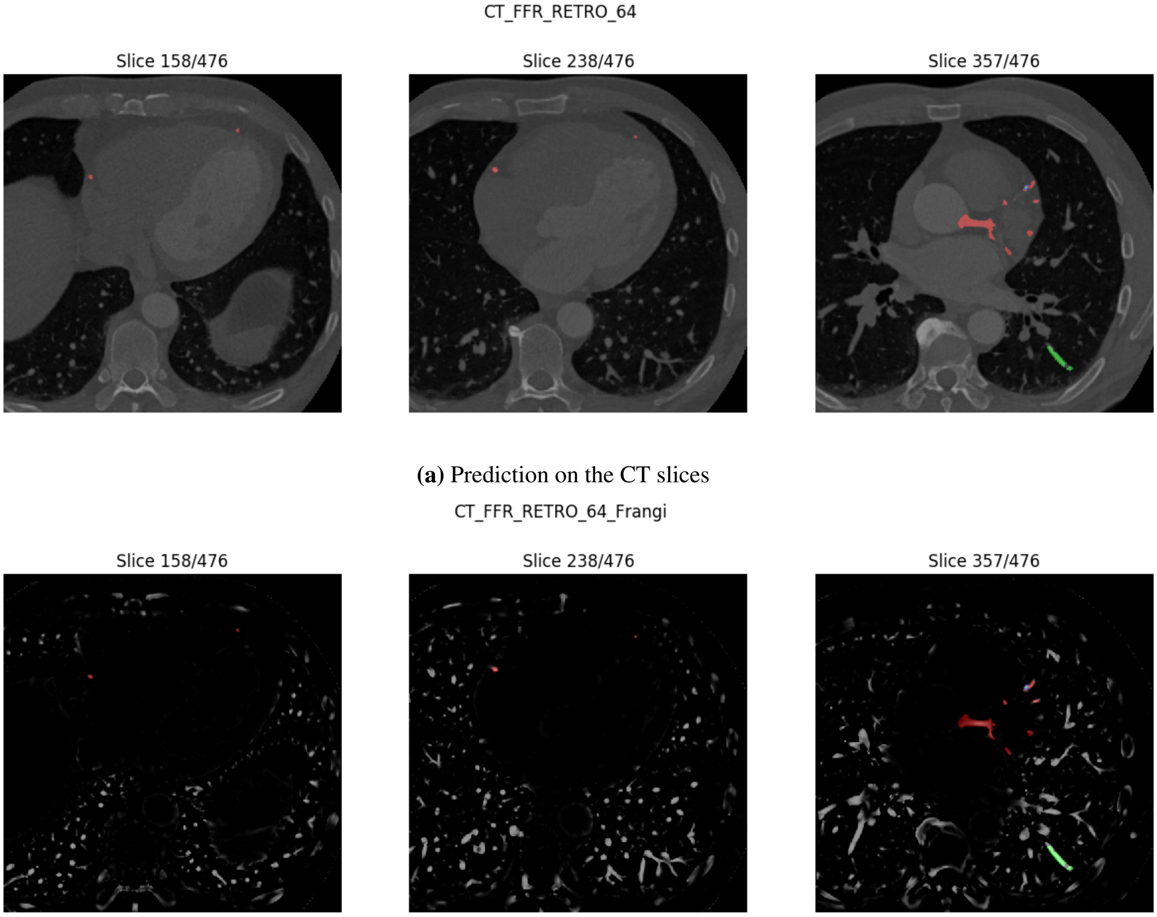
Table E.1: Result of split-0 of the BVNet with Frangi filtered images for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.458	0.455	0.339	0.317	0.706	0.806
CT_FFR_37	0.315	0.345	0.736	0.729	0.201	0.226
CT_FFR_18	0.287	0.343	0.637	0.63	0.185	0.236
CT_FFR_40	0.014	0.009	0.007	0.005	0.272	0.405
CT_FFR_8	0.397	0.441	0.57	0.563	0.305	0.363
CT_FFR_RETRO_86	0.093	0.098	0.584	0.569	0.051	0.054
CT_FFR_RETRO_89	0.208	0.195	0.135	0.119	0.459	0.524
CT_FFR_Pilot_7	0.493	0.49	0.388	0.359	0.679	0.769
Average Scores	0.297	0.312	0.442	0.429	0.37	0.446

Table E.2: Result of split-1 of the BVNet with Frangi filtered images for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.519	0.528	0.552	0.549	0.49	0.507
CT_FFR_25	0.251	0.293	0.783	0.779	0.149	0.181
CT_FFR_4	0.559	0.675	0.602	0.584	0.521	0.799
CT_FFR_RETRO_23	0.285	0.318	0.344	0.331	0.244	0.307
CT_FFR_RETRO_83	0.611	0.609	0.522	0.498	0.737	0.785
CT_FFR_16	0.527	0.54	0.554	0.546	0.502	0.534
CT_FFR_Pilot_3	0.225	0.232	0.221	0.189	0.23	0.3
CT_FFR_33	0.567	0.587	0.629	0.621	0.516	0.557
Average Scores	0.443	0.473	0.526	0.512	0.424	0.496

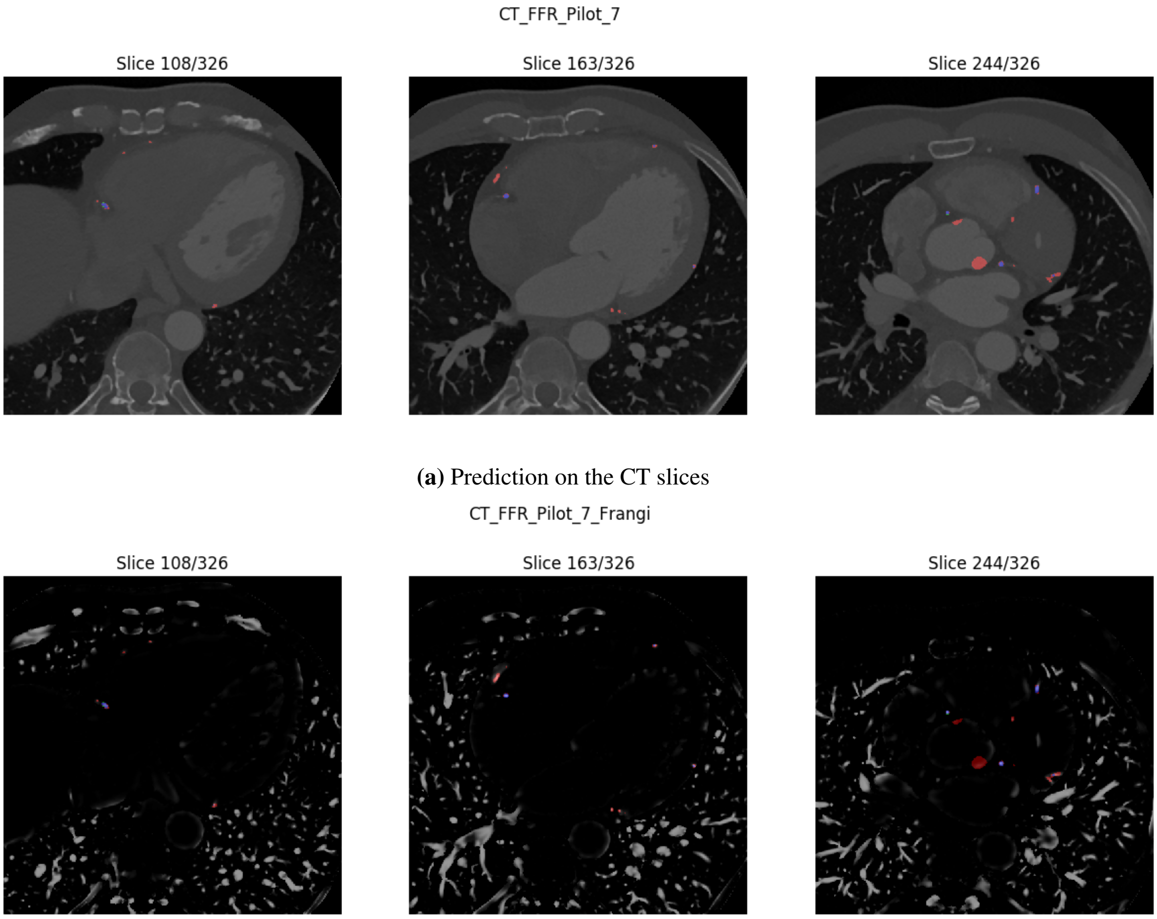
Table E.3: Result of split-2 of the BVNet with Frangi filtered images for the coronary arteryl



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

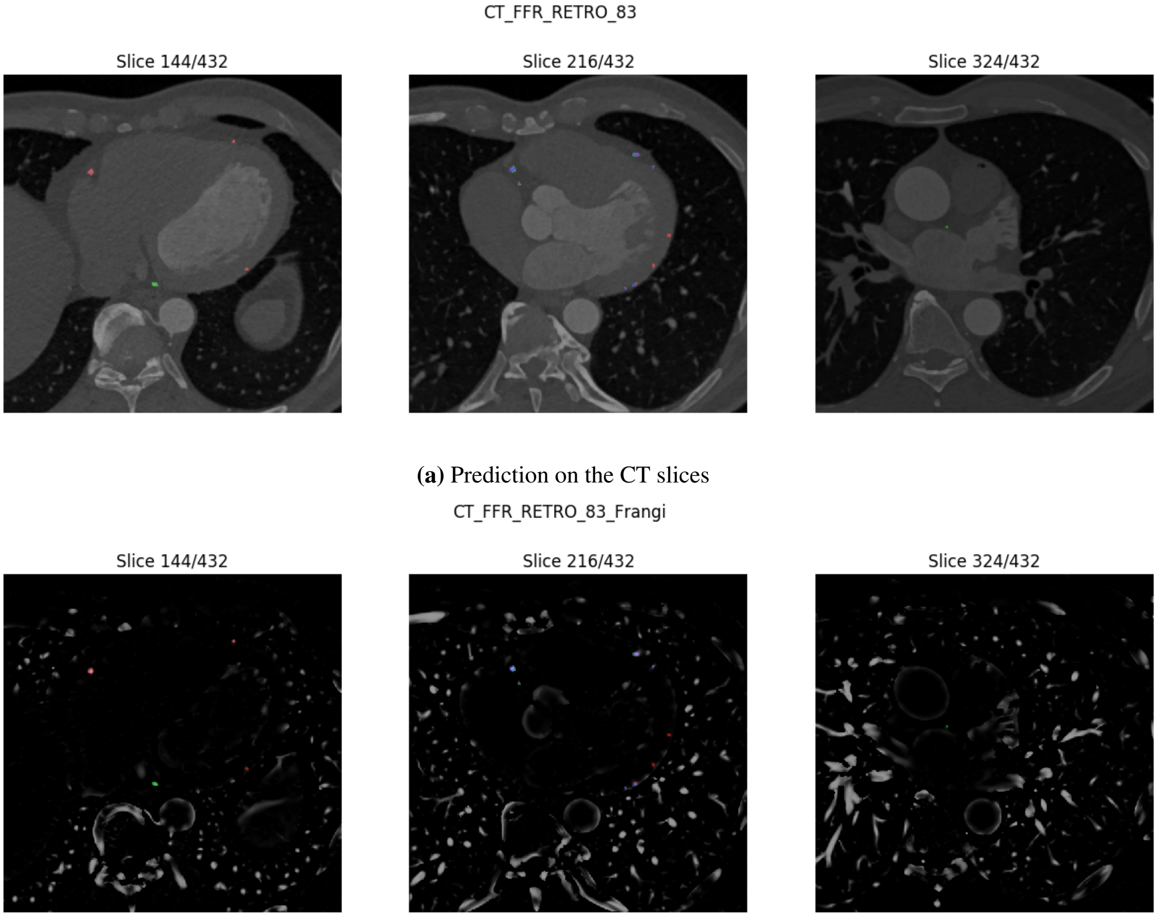
Figure E.1: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's worst prediction of the coronary artery split-0 with Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

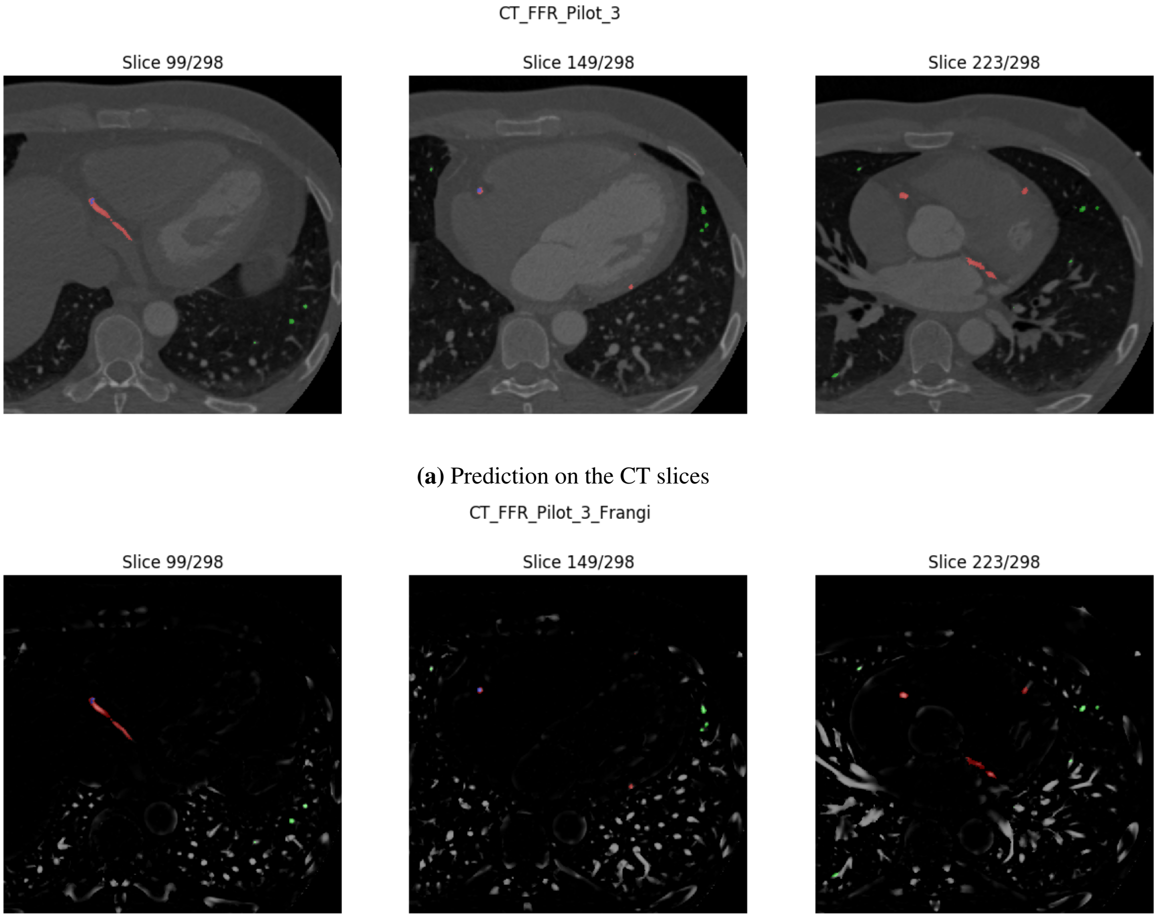
Figure E.2: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's best prediction of the coronary artery split-1 with Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

Figure E.3: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's best prediction of the coronary artery split-2 with Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

Figure E.4: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's worst prediction of the coronary artery split-2 with Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

E.2 Combined CT and Frangi Input

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_RETRO_18	0.332	0.324	0.278	0.258	0.414	0.436
CT_FFR_19	0.668	0.674	0.552	0.545	0.846	0.884
CT_FFR_28	0.778	0.779	0.687	0.684	0.898	0.905
CT_FFR_32	0.613	0.628	0.597	0.597	0.63	0.663
CT_FFR_39	0.821	0.85	0.899	0.899	0.756	0.806
CT_FFR_RETRO_14	0.346	0.321	0.215	0.195	0.894	0.906
CT_FFR_50	0.54	0.541	0.453	0.449	0.67	0.679
CT_FFR_RETRO_29	0.455	0.473	0.697	0.678	0.337	0.363
CT_FFR_RETRO_64	0.443	0.431	0.351	0.326	0.6	0.639
Average Scores	0.555	0.558	0.525	0.515	0.672	0.698

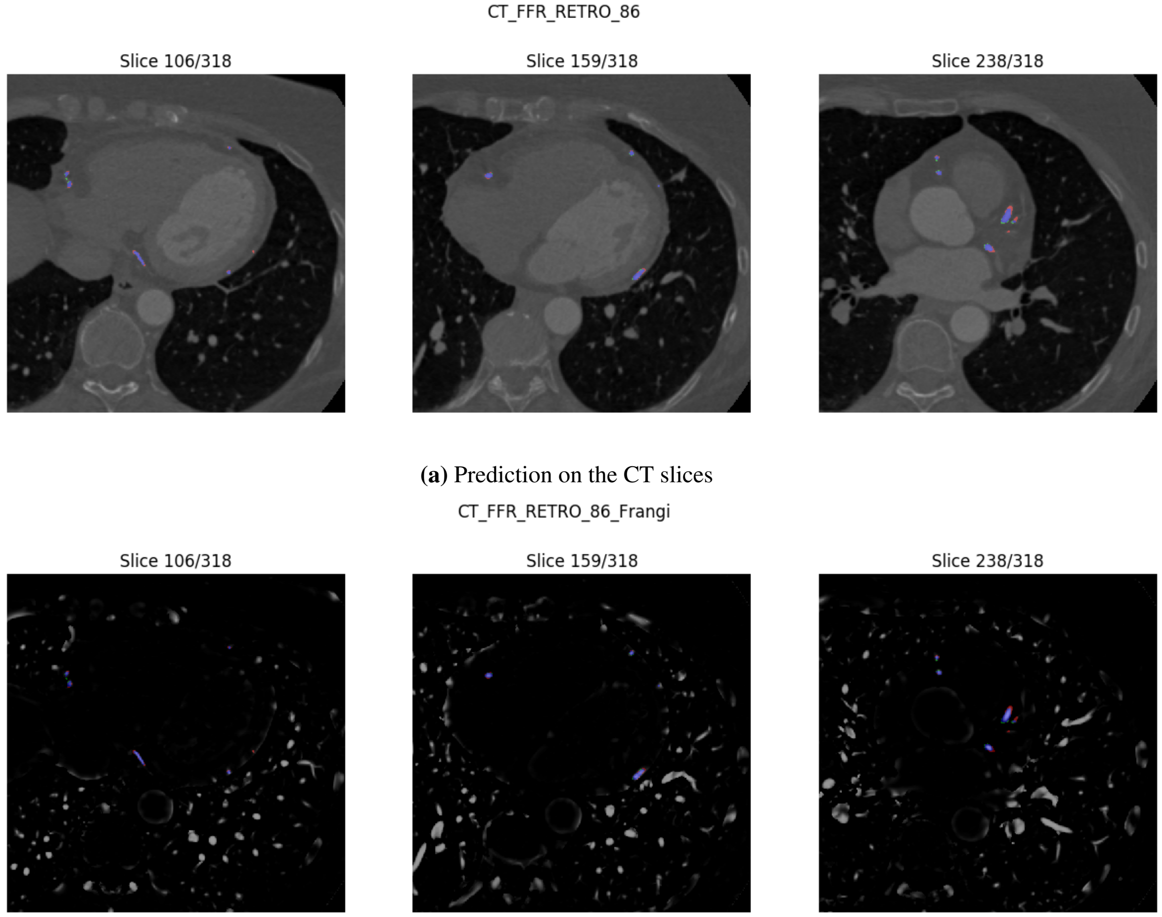
Table E.4: Result of split-0 of the BVNet with both CT and Frangi filtered images as input for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_31	0.518	0.529	0.516	0.511	0.519	0.549
CT_FFR_37	0.6	0.647	0.757	0.741	0.498	0.574
CT_FFR_18	0.697	0.695	0.565	0.552	0.908	0.938
CT_FFR_40	0.725	0.738	0.895	0.892	0.61	0.63
CT_FFR_8	0.735	0.735	0.677	0.646	0.804	0.853
CT_FFR_RETRO_86	0.809	0.806	0.768	0.757	0.855	0.863
CT_FFR_RETRO_89	0.532	0.536	0.585	0.462	0.488	0.637
CT_FFR_Pilot_7	0.646	0.667	0.599	0.594	0.7	0.76
Average Scores	0.658	0.669	0.67	0.644	0.673	0.725

Table E.5: Result of split-1 of the BVNet with both CT and Frangi filtered images as input for the coronary artery

Test Image	Dice		Recall		Precision	
	Raw	Post	Raw	Post	Raw	Post
CT_FFR_12	0.627	0.636	0.646	0.645	0.609	0.627
CT_FFR_25	0.63	0.662	0.877	0.877	0.491	0.531
CT_FFR_4	0.759	0.753	0.736	0.707	0.783	0.805
CT_FFR_RETRO_23	0.521	0.52	0.37	0.361	0.881	0.929
CT_FFR_RETRO_83	0.694	0.698	0.633	0.573	0.768	0.892
CT_FFR_16	0.698	0.716	0.623	0.594	0.794	0.901
CT_FFR_Pilot_3	0.597	0.592	0.434	0.426	0.956	0.969
CT_FFR_33	0.638	0.655	0.611	0.606	0.668	0.714
Average Scores	0.645	0.654	0.616	0.599	0.744	0.796

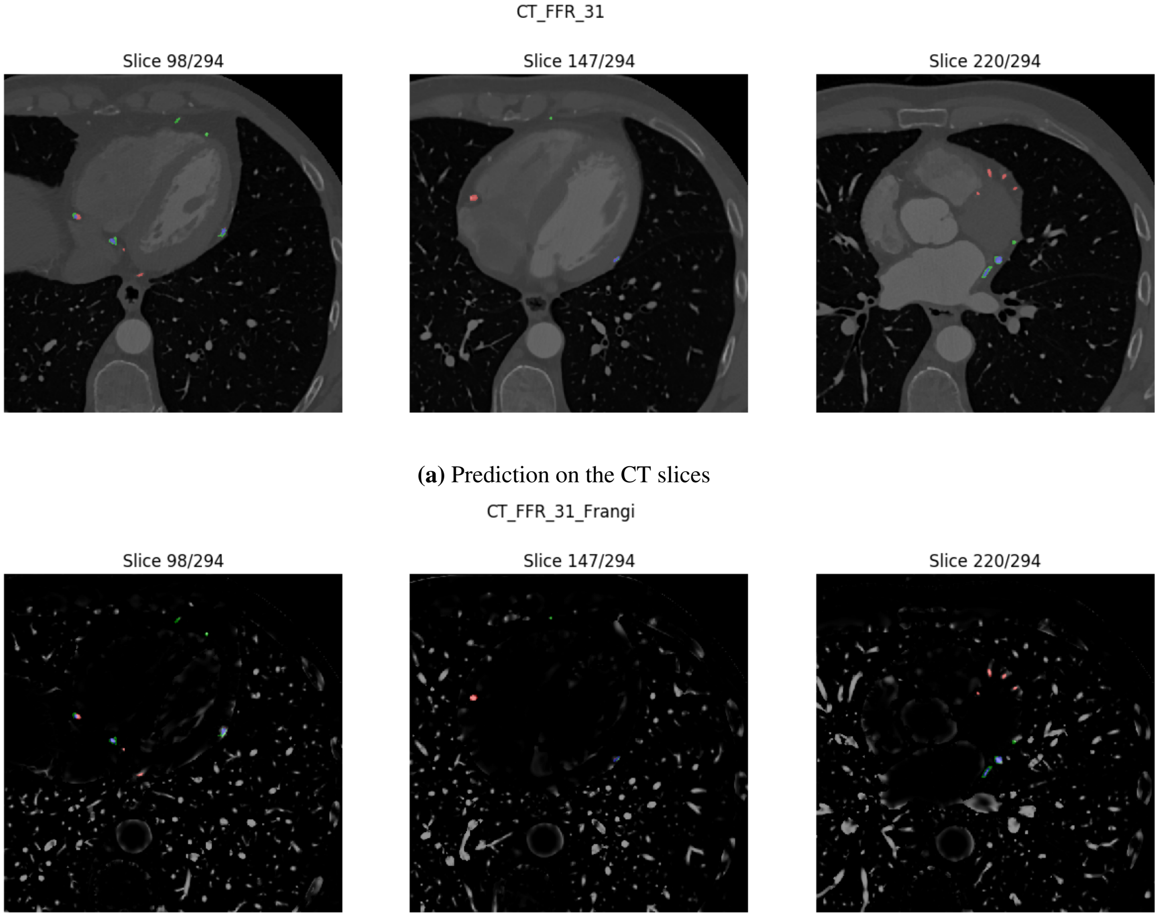
Table E.6: Result of split-2 of the BVNet with both CT and Frangi filtered images as input for the coronary artery



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

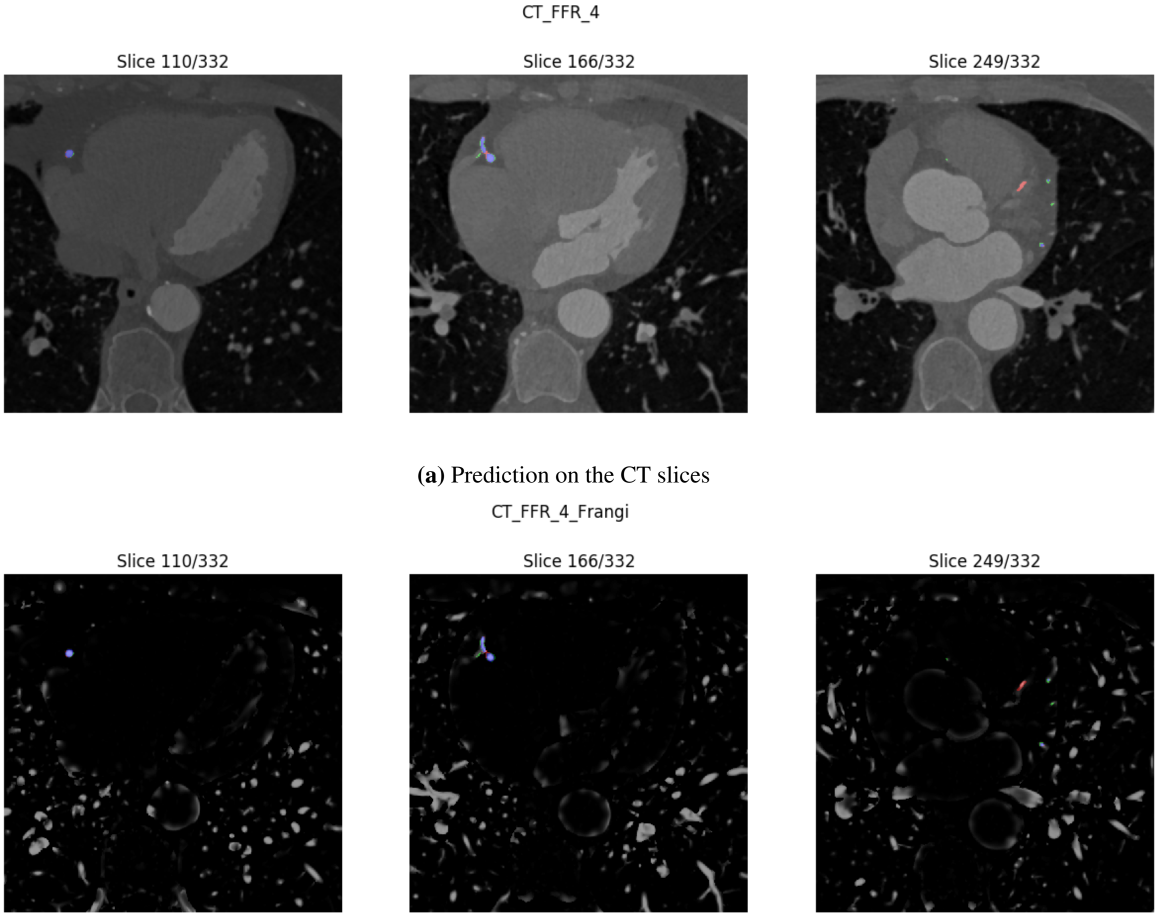
Figure E.5: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's best prediction of the coronary artery split-1 with combined CT and Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

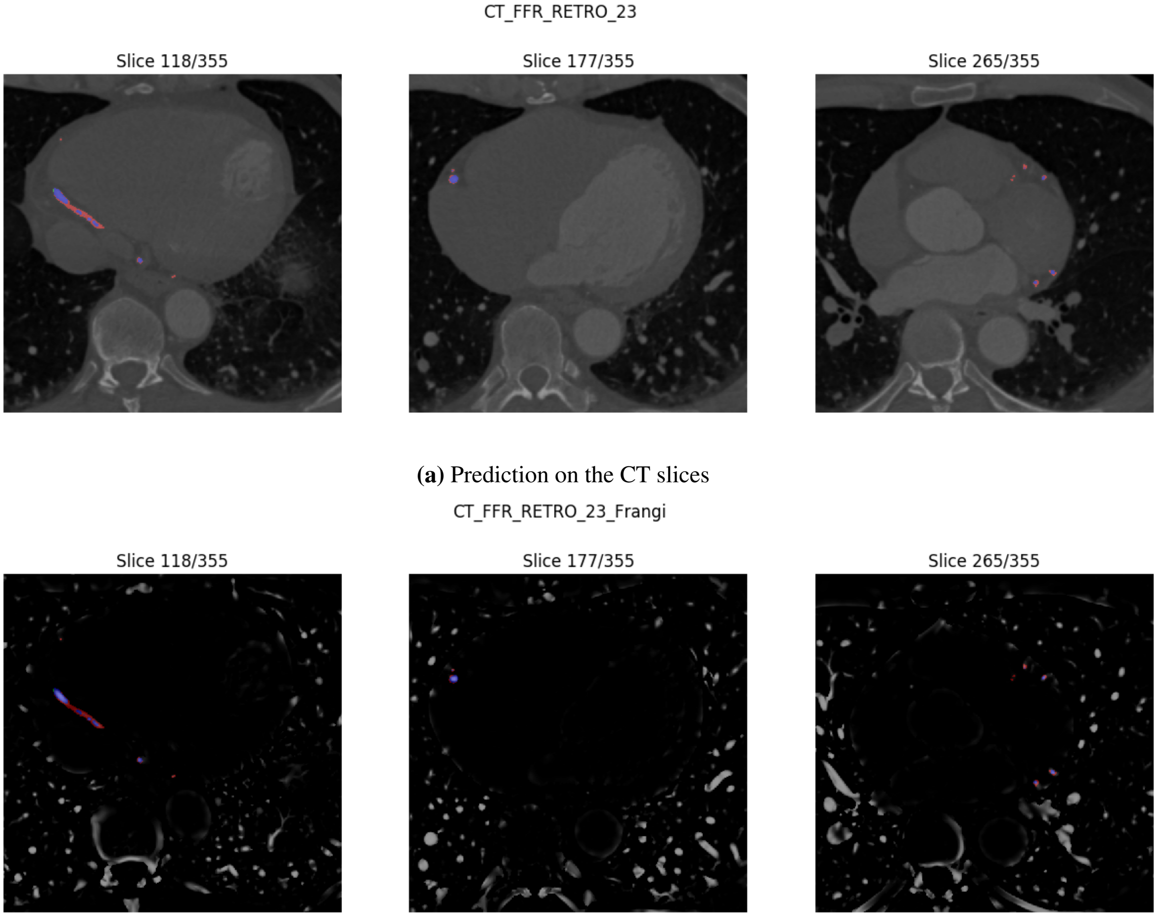
Figure E.6: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's worst prediction of the coronary artery split-1 with combined CT and Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

Figure E.7: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's best prediction of the coronary artery split-2 with combined CT and Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.



(a) Prediction on the CT slices

(b) Prediction on the Frangi filtered CT slices

Figure E.8: A selection of some of the CT and Frangi filtered CT slices of the BVNet model's worst prediction of the coronary artery split-2 with combined CT and Frangi filtered CT images as input, where the corresponding prediction is overlaid. The prediction is color-coded with blue=TP, green=FP, and red with low opacity=FN.

Trude Jostad

Coronary Artery Segmentation using Deep Learning

NTNU
Norwegian University of
Science and Technology