Master's thesis

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

Sigvard Johansen Seljelv

# Deep Learning for Deformation Analysis in Echocardiography

Master's thesis in Electronic Systems Design and Innovation
Supervisor: Lasse Løvstakken
June 2021

NTNU
Norwegian University of
Science and Technology

Sigvard Johansen Seljelv

# Deep Learning for Deformation Analysis in Echocardiography

Master's thesis in Electronic Systems Design and Innovation
Supervisor: Lasse Løvstakken
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

# Abstract

Heart disease is one of the leading causes of death worldwide. Early detection of risk factors is crucial for effective and accurate treatment. Today, screenings of patients are most commonly performed using cardiac ultrasound, a non-invasive procedure that creates an image of the heart. The analysis of the output image requires years of training and experience to master, and even then, the task is not trivial. Automation of parts of this assessment can help the analysis process and enable less experienced personnel to perform the procedure. This thesis proposes a deep convolutional neural network tasked to automatically segment out the mitral valve in echocardiographic images. The idea is that automatic segmentations can help clinicians detect irregularities in the behavior of the valve. Two variations based on the U-Net architecture are proposed; one trained using segmentations of the valve apparatus (U-Net OV-C) and one trained with the addition of segmentations of the left ventricle, myocardium and left atrium in addition to the valve (U-Net Auto-R). Clinicians manually create the valve segmentations, and the chamber segmentations are auto-generated by a pre-trained neural network. The models are trained using a data set of 824 echocardiographic images composed of mostly sick patients with differing degrees of valve deformation. Two features from the predicted segmentations are extracted, the center annulus points and an estimation of the leaflet angles. The accuracy of the models is measured using the DICE score. The U-Net OV-C model produces segmentations with an accuracy of 0.691, 0.696, 0.423, and 0.548 for the posterior leaflet, anterior leaflet, posterior annulus, and anterior annulus, respectively. The accuracy of the U-Net Auto-R model given in the same order is 0.693, 0.700, 0.398, and 0.438. The annulus center points are extracted from the segmentations and compared to the center point of the ground truth segmentations. The segmentations produced by the U-Net OV-C network result in a median error 3.39 mm for the posterior annulus and 2.73 mm for the anterior annulus. The U-Net Auto-R produces a median error of 3.78 mm and 2.97 mm for the posterior and anterior annulus, respectively. The same procedure is performed for the angle estimation, where the U-Net OV-C has a median error of 6.73 and 10.58 degrees for the posterior and anterior leaflet angles. The U-Net Auto-R has a median error of 7.43 and 10.30 degrees. Whether or not these feature extractions are reliable enough to be used in a clinical setting is not explored in this thesis and needs further investigation in future work. The overall results indicate that the inclusion of context does not have any obvious advantages in terms of performance for the U-Net model. On the contrary, it shows that it in some cases worsens the performance. However, artificial addition of more data through augmentation indicates that context does not hurt the performance noticeably if enough data is provided during training. Thus, more data is required to properly assess the possible gains of context addition for the model.

# Samandrag

Hjartesjukdom er ei av dei leiande dødsårsakene i verda. Tidleg deteksjon av risikofaktorar er avgjerande for effektiv og nøyaktig behandling. I dag blir undersøkingar av pasientar oftast utført ved hjelp av hjarte-ultralyd, ein noninvasiv prosedyre som produserer eit bilde av hjartet. Analysen av bildet krev mange år med trening og erfaring, men sjølv då er ikkje oppgåva triviell. Automatisering av delar av analyseprosessen kan hjelpe klinikarane til å ta betre slutningar, samtidig som det mindre erfarne personell kan utføre undersøkinga. Denne oppgåva foreslår eit djupt, konvolusjonelt nevralt nettverk som kan utføre automatisk segmentering av mitralklaffen frå ekkokardiografiske bilde. Tanken er at desse segmenteringane kan hjelpe klinikarane med å oppdage uregelmessigheiter i klaffen sin oppførsel. To variantar basert på U-Net-arkitekturen er foreslått; ein trent ved å bruke segmenteringar av klaffeapparatet (U-Net OV-C) og ein trent med tillegg av segmenteringar av venstre ventrikkel, myokard og venstre atrium, i tillegg til klaffeapparatet (U-Net Auto-R). Segmenteringane av klaffen er manuelt laga av klinikarar og kammersegmenteringane er automatisk generert av eit nevralt nettverk trent på førehand. Modellane er trent med eit datasett som inneheld 824 ekkokardiografiske bilde, og er samansett av stort sett sjuke pasientar med forskjellige gradar av klaffedeformasjon. To komponentar frå dei predikerte segmenteringane er henta ut, midtpunktet til for annulus segmenteringane og ein estimering av klaffevinklane. Nøyaktigheita til modellane målast med 'DICE score'. U-Net OV-C-modellen produserer segmenteringar med ei nøyaktigheit på høvesvis 0.691, 0.696, 0.423 og 0.548 for det bakre seglet, det fremre seglet, bakre annulus og fremre annulus. Nøyaktigheita til U-Net Auto-R modellen er 0.693, 0.700, 0.398 og 0.438 gitt i same rekkefølge. Midtpunktet til annulus prediksjonane blir samanlikna med midtpunktet til fasitsegmenteringane. Segmenteringane produsert av U-Net OV-C-nettverket, resulterer i ein medianfeil 3.39 mm for bakre annulus, og 2.73 mm for fremre annulus. U-Net Auto-R produserer ein medianfeil på høvesvis 3.78 mm og 2.97 mm for bakre og fremre annulus. Den same prosedyren blir gjort for vinkelestimering, der U-Net OV-C har ein medianfeil på 6.73 og 10.58 grader for bakre og fremre segl. U-Net Auto-R har ein medianfeil på 7.43 og 10.30 grader. I kva grad desse komponentane er pålitelege nok til å bli brukt i kliniske oppgåver blir ikkje utforska i denne oppgåva. Dette er noko som treng ytterlegare undersøking i framtidig arbeid. Dei samla resultanta indikerer at kontekstilsetning ikkje har noko openberre fordelar når det gjeld ytinga til U-Net-modellen. Resultata viser tvert imot at det i nokre tilfelle forverrar ytinga. Kunstig tilsetting av meir data gjennom augmentering viser derimot at kontekst ikkje skader ytinga merkbart viss modellen får nok data under trening. Dermed treng ein meir data for å kunne fastslå moglege gevinstar av kontekstilsetning for modellen.

# Foreword

In the process of working with this project I have received help from many talented people. First and foremost, I would like to thank Sigurd Vangen Wifstad for his invaluable assistance and for always taking the time for lengthy walks and discussions regarding the technical solutions. I also want to thank my fellow co-students studying Electronic Systems Design and Innovation for rewarding discussions and encouragement. Furthermore, I would like to thank Ståle Wågen Hauge, Sigbjørn Sæbø and Håvard Dalen for their efforts related to annotating the echocardiographic images and for all the information and answers they have provide regarding the data material. Lastly, I want to express my gratitude to my main supervisor Lasse Løvstakken for his counseling and facilitation of the whole project.

# Contents

# 1 Introduction

## 1.1 Motivation

Cardiovascular disease (CVD) is a collective term used for all diseases related to the heart or blood vessels. A 2015 study by Wang et al. [1] showed that outside of Africa, CVD is the leading cause of death in the world, with a steady increase over the past years. They attribute this to the fact that the occurrence of CVD increases as the overall life expectancy increases, which is the case worldwide, especially outside of Africa. Studies by McGill, McMahan, and Gidding [2] and O'Donnell et al. [3] suggest that up to approximately 90% of deaths caused by CVD could be prevented. Early detection of risk factors is an important step to increase the effectiveness of treatment. Some common risk factors include deviations from normal heart function, high cholesterol levels, and blood pressure. Most risk factors can be reduced or eliminated through surgery or medical treatment.

Mitral valvular heart disease is a sub-class of CVD, which are diseases affecting the mitral valve in the heart. The valve separates the left ventricle and the left atrium. Several valvular heart diseases involve deformation or damage of the valve, which may lead to dangerous irregularities in the cardiac cycle. Diagnosing valvular heart disease is done by cardiologists using ultrasound technology to inspect the heart, called echocardiography. An echocardiogram is created from acoustic waves transmitted from a probe into the cardiovascular system. The same probe captures echoes reflected by the anatomy of the heart and can be used to visualize the heart and the blood flow within. Quantifying the relations between the image of the valve and the different valvular diseases is not trivial; even a clinician with years of experience may struggle. If quantization of the valve could be done automatically, this would help both experienced and inexperienced cardiologists to detect faulty valves.

When one faces these kinds of problems in the current era of technology and research, it is common to apply so-called Artificial Intelligence (AI) to solve the problem. The term AI is loosely defined but usually refers to algorithms that can make decisions without explicit programming. Over the past years, these algorithms have outperformed several state-of-the-art classical methods in many fields of science. Some people compare this progress to that of the industrial revolution in the 19th century, naming this trend the AI revolution [4]. In particular, Machine Learning (ML), a sub-category of AI, has proven to be very useful for problems where large amounts of data exist. ML algorithms are based on the creation of models with a vast number of parameters, which are optimized to a training set.

Choosing what type of machine learning algorithm one should use heavily depends on the problem at hand. When it comes to image processing tasks, Convolution Neural Networks (CNNs) have become the norm. CNNs are a class of machine learning algorithms utilizing convolutional operations. When these operations are applied to images, the operations function as image filters. One can draw similarities between CNNs and the human cortex [5], because both extract general features from visual input to determine the contents of the input.

A recent review study by Chen et al. [6] compared the frequency and results of algorithms used for several medical image-related problems. Their findings suggest that the U-Net architecture is the most popular and effective for echocardiographic

segmentation tasks. Costa et al. [7] demonstrated that segmentation of the mitral valve using U-net is possible using annotations of the leaflets of the valve. They reported difficulties separating the valve from other structures in the heart, such as the walls of the heart chambers. The mitral valve is a small component compared to the other structures in the heart, and it often occupies a small portion of the whole image, making it hard to detect. One possible solution to this could be to give the network more context, i.e., annotations of the other structures in the image.

## 1.2 Problem Statement

This thesis explores the potential of using deep learning methods to segment out the mitral valve and its annulus fibrosus points in echocardiographic images. The algorithms are trained and tested on data produced by clinicians, who have traced the valve and marked its annulus points. This data is then used to train a deep neural network that aims to make the same annotations as the clinicians on similar images. The potential benefits or disadvantages of adding segmentations of other structures of the heart is also explored.

# 2 Background

*The following section is taken from [8]* and adapted to the context of this report.

## 2.1 The Heart

### 2.1.1 Composition

The human heart is an intricate system that consists of several parts. There are four chambers inside the heart, two chambers at the top named the left and right atria and two chambers at the bottom named left and right ventricles. The chambers are made of different muscle tissue and can be activated by the nervous system. The chamber walls are composed of several layers. The myocardium (MY) is the thickest layer and often used when referring to the chamber walls as a whole. All the chambers have valves separating them from each other and valves separating them from the blood vessels [9]. One of these valves is the mitral valve (MV), which separates the left atrium and the left ventricle. The valve is a complicated structure that includes a fibrous annulus, an anterior and posterior leaflet with two commissures, and multiple chordae tendineae that are attached to the papillary muscles of the left ventricle (LV) [10]. The points where the leaflets are connected to the fibrous annulus are referred to as annulus points. The anterior mitral valve leaflet (AMVL) is connected to the anterior annulus (AA) point, and the posterior mitral valve leaflet (PMVL) is connected to the posterior annulus (PA) point. The different parts of the mitral valve ensure that the apparatus functions optimally during the cardiac cycle [9].

### 2.1.2 The Cardiac Cycle

The heart functions as a pump that transports deoxygenated blood into the lungs, which oxygenates it and then transports it back into the body. This is a continuous cycle illustrated in figure 1. It is natural to start at the point where all chambers are relaxed. After this, during atrial systole, the atria push blood into the ventricles by contracting. When the blood has moved into the ventricles, the atria relax again in a step called atria diastole. At approximately the same time, the ventricular systole begins, where the ventricles start to contract. This reduces the volume of the ventricles and pushes the blood into the blood vessels that transport the blood in the body. After the blood has left the ventricles, the ventricular diastole begins, and the ventricles start to relax. The first step of the ventricular diastole is isovolumetric relaxation, where the ventricular muscles relax. This causes the pressure to drop. To prevent backflow from the atria, the semilunar valves between the ventricles and atria close. The second step is ventricular filling, where all the valves open and blood from the major veins flow into the atria and ventricles as all the cambers relax, thus completing the cardiac cycle [11]. The key events of the cycle can be outlined for any of the chambers as consisting of diastole (relaxation) and systole (contraction).

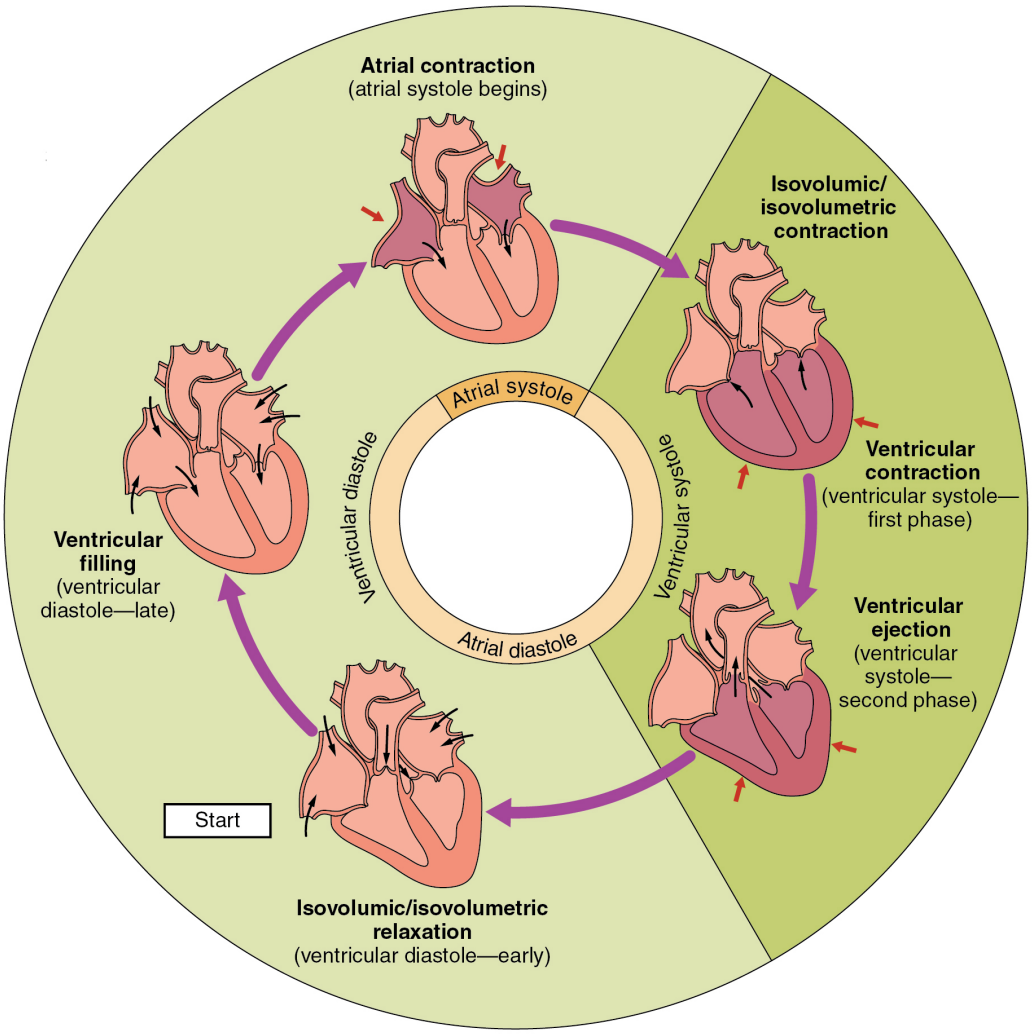



Figure 1: "Overview of the Cardiac Cycle: The cardiac cycle begins with atrial systole and progresses to ventricular systole, atrial diastole, and ventricular diastole, when the cycle begins again. Correlations to the ECG are highlighted." - The illustration is taken and adapted from OpenStax College, CC BY 3.0 <https://creativecommons.org/licenses/by/3.0>, via Wikimedia Commons.

### 2.1.3 Mitral Valvular Heart Disease

One of the central functions of the mitral valve is preventing backflow between the left ventricle (LV) and the left atria (LA). Diseases causing damage or deformation of the valve could disturb the balance of the heart and ultimately lead to death. Deformed valves could lead to leakage between the left atrium and the left ventricle, commonly referred to as mitral regurgitation or mitral insufficiency [12].

Several different diseases can damage the mitral valve. One of these is rheumatic heart disease (RHD), a condition where the heart valves have been permanently damaged. The damage occurs as a result of rheumatic fever, an inflammatory illness affecting various connective tissues, particularly the heart, joints, skin, and brain

[13].

Mitral valve prolapse (MVP) occurs when the mitral valve's two leaflets do not seal smoothly or uniformly and instead bulge (prolapse) upward into the left atrium. Click-murmur syndrome, Barlow's syndrome, and floppy valve syndrome are all names for mitral valve prolapse. MVP is not dangerous in all cases, but sometimes it can lead to mitral stenosis [14].

A narrowing of the mitral valve opening is known as mitral stenosis and restricts the blood flow from the left atrium to the left ventricle. This disturbs the balance in the cardiac cycle. In some cases, the volume and pressure of blood that remains in the left atrium rises, causing the left atrium to expand and fluid to accumulate in the lungs [15].

## 2.2 Echocardiography

Diagnostic ultrasound is a non-invasive method utilized in several medical examinations. Echocardiography is diagnostic ultrasound performed on the heart. The procedure is performed with an ultrasound probe connected to a computer that processes the data the probe captures. The probe, also called a transducer, can both emit ultrasound waves and detect the echoes reflected by different elements inside the body. The different elements inside the body reflect the waves at different rates, which can be used to form an image of the heart in the case of echocardiography.

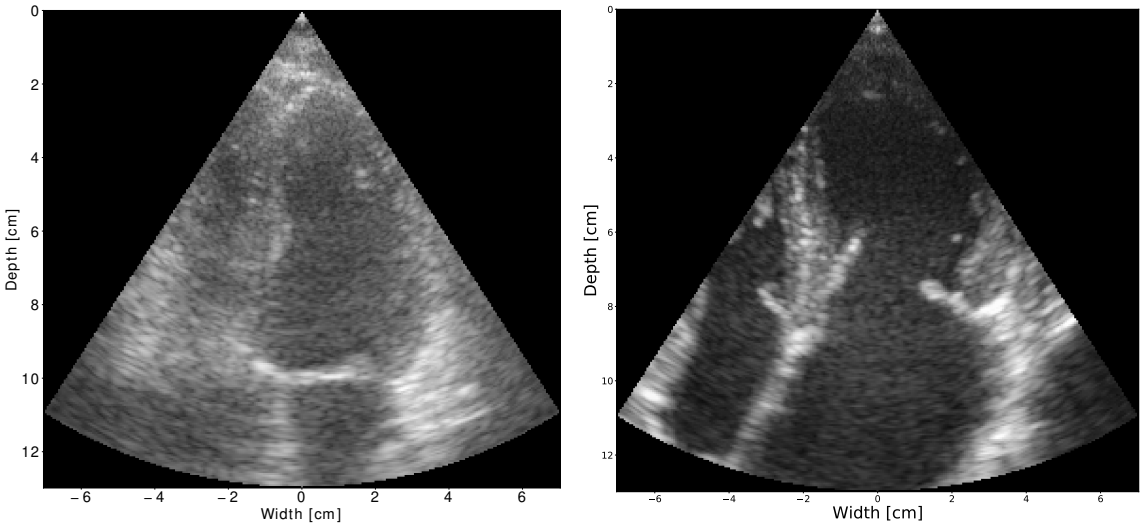
The frequency of the waves used by the probe affects the quality of the image. Higher frequency provides images with improved resolution. However, higher frequency causes weaker penetration of tissue, resulting in higher attenuation of the signal. The key is to find a balance between frequency and resolution. In some cases, there is also a possibility to place a probe inside the body via the gastrointestinal tract in the case of echocardiography. This reduces the amount of tissue between the probe and the heart, which means that a higher frequency can be used. However, this method is more invasive and more complex, requiring more resources.

There are several different ways to visualize the measurements. For 2D echocardiographic imaging, brightness mode (B-mode) is widespread. This method draws the amplitude of the measured reflection along an axis representing time or distance. This generates a single scan-line of an image. Multiple B-mode lines are generated by sweeping the wavefront along an axis direction. These lines are then placed next to each other, resulting in a two-dimensional image with time/depth along one axis and lateral position along the other axis [16].

An echocardiographic examination often involves images of the heart from various views. These views are defined by the position and orientation of the transducer relative to the heart, resulting in an image showing different parts of the heart. The most common positions are parasternal and apical, which can be combined with different probe orientations. Different orientations result in differing cuts of the tomographic plane through the heart such as long axis, short axis, four-chamber, and two-chamber, to mention a few variations [16]. Two examples of B-mode images with different views are shown in figure 2.

During screenings, a clinician will look for these views and modify them to some extent, depending on the purpose of the screening. If the focus of the screening is

the left ventricle, the clinician will, for example, look for the A2C with a ventricular focus. This may result in subpar imaging of the mitral valve or the left atrium.




(a) Apical two-chamber (A2C) (b) Apical four-chamber (A4C)

Figure 2: Examples of B-mode still images of (a) apical two-chamber and (b) apical four-chamber views.

## 2.3 Machine Learning

The concept of machine learning dates back to the middle of the 20th century [17], but due to limitations related to hardware it remained a niche field of computer science for a long time. However, given recent advancements in hardware, the field has seen a rapid rise in popularity. Machine learning is a term used to describe a wide range of algorithms that use empirical data to learn decision-making. In recent years, these algorithms have proven to be helpful in various applications, often outperforming traditional methods. All of the techniques are dependent on learning from a training data set, referred to as "training" or "model fitting". There are two main types of training, supervised and unsupervised. During supervised training, the model uses training data with labels. The model then attempts to understand the relationship between the labels and the data to predict the labels of data samples not used during training. Unsupervised training does not have training data with labels. Since there are no labels, the model tries to learn general features of the input data and then tries to make decisions for unseen data based on the general features.

Deep learning describes a form of machine learning technique where the models consist of several different nonlinear mapping nodes. The nodes are connected and form a network. Because of their apparent resemblance to the organization of neural networks in the human brain, these nodes are commonly referred to as neurons. As a result of this, the whole network is often called a neural network. The terms Deep learning, neural networks, and deep neural networks are frequently interchanged. In this context, "deep" is used for a network built up of multiple layers of neurons in a cascade. More layers result in a deeper network. The number of learnable model parameters increases as the depth of the network grows, allowing for more sophisticated models that can perform more complex tasks.

### 2.3.1 Convolutional Neural Network

In recent years, methods using convolutional neural networks (CNNs) have become the state-of-the-art solution used for many image processing problems. A CNN is based on a cascade of linear convolutions [18, Chap. 9]. At each layer of the cascade, there is an added bias and a nonlinear activation function. Mathematically we can describe the behavior of one such convolutional layer with the forward pass equation as shown in equation (1).

$$\mathbf{a}^{l+1} = h(\mathbf{b}^l + \mathbf{w}^l * \mathbf{a}^l). \tag{1}$$

$\mathbf{a}^l$ is the activation at layer $l$. A layer $l$ can have multiple activations with different parameters running in parallel. If this is the case, it is common to call the activations feature maps because the activations represent features of the input data. $\mathbf{b}^l$ and $\mathbf{w}^l$ are the bias and weights at layer $l$, respectively. h(x) is the activation function. $\mathbf{b}^l$ and $\mathbf{w}^l$ are learnt parameters, which are optimized during training. This is accomplished by using a loss function L [18, Chap. 4.3]. A common loss function is mean-squared-error (MSE), which is mostly used for regression problems. The MSE is computed with the predicted output tensor $\hat{\mathbf{y}}$ and the target tensor $\mathbf{y}$ as shown in equation (2).

$$L = \frac{1}{N} \sum_{x_0, x_1, \ldots, x_{n-1}} (\hat{\mathbf{y}} - \mathbf{y})^2, \tag{2}$$

where L is a scalar value produced by the summation over all $n$ variables $(x_0, x_1, ..., x_{n-1})$ of the tensor with $N$ points across n dimensions [18, page 105]. Another common loss function is log loss (also called cross-entropy (CE) loss), shown in equation (3).

$$L = -\sum_{i=1}^{N} t_i \log(p_i), \tag{3}$$

where $n$ is the number of classes, $t_i$ is the truth label, and $p_i$ is the probability of class $i$. Naturally, the loss should be minimized. This is done with gradient descent using backpropagation equations [18, Chap. 6], outlined for two dimensional CNNs in equation (4).

$$\begin{aligned} \delta^l_{x,y} &= \frac{\partial L}{\partial z^l_{x,y}}, \\ \frac{\partial L}{\partial \omega^l_k} &= \boldsymbol{\delta}^l_k * \text{rot180}(\boldsymbol{a}^{l-1}), \\ \frac{\partial L}{\partial b^l} &= \sum_{x,y} \delta^l_{x,y}, \end{aligned} \tag{4}$$

where $\delta^l_{x,y}$ is the backward propagated loss at layer l for point $(x, y)$ in the input matrix. The feature map index at layer l is represented by the index k. The term rot180 refers to a 180-degree rotation.

A neural network is trained by adjusting the weights of the neurons to give the best possible outcome. The backpropagation equations calculate the loss function and update the model weights through gradient descent. The elementary approach to gradient descent is known as batch gradient descent, shown in equation (5).

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_{\boldsymbol{w}_t} L(\boldsymbol{w}_t). \tag{5}$$

Here, $\nabla_{\omega_t} L(\omega_t)$ is calculated first and then the weights $\omega$ are updated with a factor $\eta$, referred to as the learning rate. This approach could be problematic given a large data set. In order to circumvent this problem, it is common to use a modified variation, namely the Stochastic Gradient Descent (SGD) optimizer shown in equation (6).

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_{\boldsymbol{w}_t} L(\boldsymbol{w}_t; x^{(i)}, y^{(i)}). \tag{6}$$

In the SGD optimizer equation $x^{(i)}$ and $y^{(i)}$ are sample $i$ of the training data input and target data, respectively. The optimizer updates the weights for each data sample in the data set, resulting in a considerable simplification of the gradient computation. Another approach is to divide the data set into several smaller batches and calculate the gradient for each mini-batch. This method is called Mini-batch gradient descent. One can improve the updating of the weights by adding a moment aspect accompanying the gradient, as described in equation (7).

$$\begin{aligned} \boldsymbol{v}_t &= \gamma \boldsymbol{v}_{t-1} + \eta \nabla_{\boldsymbol{w}_t} L(\boldsymbol{w}_t), \\ \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \boldsymbol{v}_t, \end{aligned} \tag{7}$$

where $\mathbf{v}_t$ is the moment at time $t$ and $\gamma \in [0, 1)$. The addition of moment gives the optimizer momentum, which reduces oscillations around local minima, resulting in improved convergence. This method is further improved by using adaptive moments. The adaptive approach changes the hyperparameters to the data so that sparse data samples or mini-batches have a greater influence on the update. This means that the optimizer will prioritize unusual data over common data, highlighting the data's diversity. There are several different optimizers with different implementations of adaptive moment, but the most popular one is the Adaptive Moment Estimation (ADAM) shown in equation (8).

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon} \hat{\boldsymbol{m}}_t. \tag{8}$$

$\hat{\boldsymbol{m}}_t$ and $\hat{\boldsymbol{v}}_t$ are the unbiased estimates of the first and second order moments of the gradient, respectively. $\epsilon$ is a tiny number added to prevent division by zero. ADAM has been demonstrated to outperform most other optimizers empirically, making it a popular choice [19].

The activation function, $h(x)$, can be any nonlinear function. The optimal function will vary based on model choice and data set and ultimately boils down to a design choice because the activation function alters the model's behavior. Some common activation functions are:

- **Rectified Linear Unit (ReLU)**, shown in equation (9). The ReLU is a popular choice that is more robust against the vanishing gradient problem than many other activation functions [18, Chap. 6].
- **Sigmoid**, shown in equation (10). The sigmoid transforms the input into output in the range [0, 1], and it has been a popular choice for a long time because its behavior is similar to that of biological neurons. However, deep networks often have a big problem with saturating and vanishing gradients [18, Chap. 3].
- **Hyperbolic tangent (tanh)**, shown in equation (11). The tanh is similar to the sigmoid; however, it returns numbers in the [-1, 1] range. The zero-centered outputs suppress oscillations during gradient update, allowing for a more stable learning phase [18, Chap. 6].
- **Softmax**, shown in equation (12). The softmax is an activation function often used for classification tasks. The function outputs normalized confidence for the predicted output classes, K, based on Luce's choice axiom [18, Chap. 6].

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \tag{9}$$

$$\text{Sigmoid}(x) = \frac{1}{1 + exp(-x)}, \tag{10}$$

$$\tanh(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)}. \tag{11}$$

$$\text{softmax}(\boldsymbol{x})_i = \frac{exp(x_i)}{\sum_{j=1}^{K} exp(x_j)} \tag{12}$$

CNNs have traditionally been sequential structures. Over the last years, residual structured CNNs have emerged by introducing skip-connections between layers. In a residual network, there are several different paths with different lengths between the input and output. This combines the advantages of shallow connections being easy to train with the advantages of deep connections being able to attain high accuracy [20]. The phrase "hidden layer" is frequently used to denote any network layer that is not the input or output layer. The hidden layers can include a variety of additional layer types that serve diverse goals in addition to convolutional layers. The following are a few that are worth mentioning:

- **Pooling layers** reduce the dimensionality of the input. A pooling function with stridden steps is applied across the input data. The most common functions include max pooling, which outputs the maximum value of the input values at each stride, and average pooling, which outputs the average value of the values at each stride [18, Chap. 9.3].
- **Transposed convolutional layers** increase the dimensionality of the input data. A stridden convolution is applied, resulting in the output having more data points compared to the input [21].

- **Dropout layers** are used to regulate the parameters where the goal is to reduce overfitting during training. The layer randomly selects some activations and sets them to zero in a given layer [18, Chap. 7.12].
- **Concatenation layers** are used to merge different layers by concatenating outputs from two layers, commonly used in Residual Networks [22].

A common problem one can face during training of a deep neural network is the exploding gradient problem. If the input data has values larger than 1 there is a possibility that the gradient can grow exponentially as it is backpropagated through the network and can result in unstable learning [18, page 282]. A possible solution for the problem is to normalize the data. Normalized data have values in the range $[0, 1]$ and can be accomplished by dividing every value in a sample by the maximum value for the given sample. If the sample includes negative values, the values need to be converted to positive, either by taking the absolute value or shifting the range of the sample into a range containing only positive values. The optimal method depends on the data at hand.

### 2.3.2 U-Net

The U-Net architecture was first proposed in an article by Ronneberger, Fischer, and Brox [23] in 2015. The theoretical structure of the U-Net architecture is shown in figure 3. The idea behind the architecture is to construct a compression/decompression pipeline using pooling and transposed convolutional layers. The pooling layers create compressed versions of the input by gradually reducing the dimensionality. The compressed versions of the input capture more general, low-level features of the inputs. Next, the compressed inputs are decompressed by up-convolutional layers, which try to reconstruct the target images using the low-level features. Furthermore, there are skip-connections between each compression level. This allows information from each compression level to contribute to the final reconstruction of the output image. This method enables the architecture to encode data with a complex feature space and reconstruct them while keeping the loss of information low. Concatenation layers are used to merge the skip-connection and the transposed convolution.

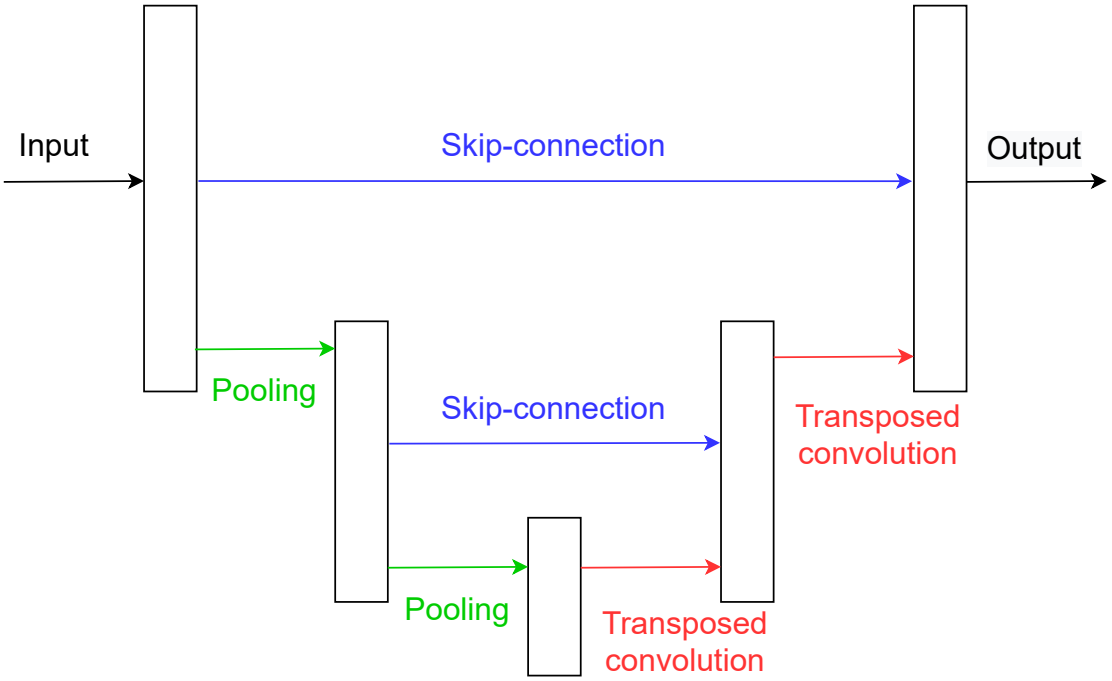



Figure 3: A theoretical example of the U-Net architecture. The convolutional layers are connected by pooling and transposed convolutional layers, resulting in compression/decompression pipeline. Each compression layer is connected using skip-connections. The illustration and caption are taken from [8].

### 2.3.3 Data Augmentation

In machine learning, one of the most common problems is the lack of data. This is especially apparent in segmentation tasks because the annotation process requires time and labor. Data augmentation is one possible way to combat this problem. By slightly altering the existing training input data and annotations, we can artificially increase training data.

There are several different types of augmentations. The best-suited augmentation methods greatly depend on the data. Gamma, Gaussian, rotation, and cropping are examples of relevant augmentation methods for echocardiographic images and segmentation.

Gamma augmentation utilizes the Power-Law (gamma) transformation shown in equation (13). For images, each pixel, $P_i$, in the input image are transformed using the two positive parameters, $g$ and $\gamma$, that control the transformation curve and results in an output pixel $P_o$ [24]. The gain, $g$, is a scaling constant often equal to 1. The gamma, $\gamma$, is a non-negative real number used to change the relative intensity between the pixels in the image. $\gamma > 1$ results in a darkened image, $\gamma = 1$ gives the same image, and $\gamma < 1$ brightens the image.

$$P_o = g \cdot P_i^{\gamma} \tag{13}$$

Gaussian augmentation is performed by applying a Gaussian filter to an image and can be viewed as a smoothing filter. A Gaussian filter is a filter with an impulse response that approximates a Gaussian function. The one-dimensional Gaussian filter has an impulse response as shown in equation (14). This filter can be used as

a kernel for the smoothing of data. The amount of smoothing correlates to the size of the standard deviation used for the Gaussian kernel [25, page 154].

$$g(x, \sigma) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \tag{14}$$

Cropping augmentation is a simple modification where a section of the image and its associated masks are zoomed in on and then resized to the original size of the image. Rotation augmentation changes the orientation of the image while keeping the original size. Cropping and rotation augmentation changes the location of all the values in the image. This results in activation of different neurons during training and therefore appears as new unique data samples.

Dealing with data augmentation, it is crucial to ensure that the resulting augmented data is different enough from the original data. If the augmentations are not sufficiently different from the original data, one risks overfitting during training.

## 2.4 Metrics

### 2.4.1 DICE score

There are several different methods one can use to evaluate the performance of a neural network. DICE score, also known as the F1 score, is a popular metric used to evaluate the performance of semantic segmentation networks. Equation (15) shows the mathematical definition and is a measurement of the similarity between two shapes $A$ and $B$. A DICE score of 1 is the highest value and indicates a perfect match between shape $A$ and $B$.

$$DICE = \frac{2 \cdot |A \cap B|}{|A| + |B|} \tag{15}$$

### 2.4.2 Manhattan Distance

The smaller an object is, the harder it is to detect and segment out correctly. This makes traditional comparing metrics, such as DICE score, less reliable when comparing the performance of two models. Another possible metric one can use is the Manhattan distance between two points in a matrix. For segmentation tasks, the distance between the center points of the ground truth segmentations and the predicted segmentations could be used. The Manhattan Distance, $d$, between two points, $x_0, y_0$ and $x_1, y_1$, in a 2D-matrix is defined by equation (16) [26].

$$d = |x_0 - x_1| + |y_0 - y_1| \tag{16}$$

### 2.4.3 Angle estimation

With the heart valve in mind, a possible metric could be to estimate the angle between each valve leaflet and the plane between the two annulus points. This can be done in several different ways. A popular feature extraction technique used in computer vision and digital image processing is the Hough transform. The technique aims to apply a voting mechanism to locate imperfect examples of objects inside a given class of forms. This voting mechanism is carried out in a parameter space,

from which object candidates are produced as local maxima in an accumulator space, which is generated explicitly using the Hough transform method.

Detection of straight lines uses the simplest case of the Hough transform. Straight lines can be expressed as $y = mx + b$ and can be represented as a point $(b, m)$ in the parameter space. This simple representation is not ideal as the slope parameter $m$ approaches unbounded values for vertical lines, resulting in a heavy computational burden. To combat this problem Duda and Hart [27] proposed the use of the Hesse normal form shown in equation (17) instead.

$$\rho = x \cos\theta + y \sin\theta, \tag{17}$$

where $\rho$ is the line perpendicular to the estimated detected line in the image, and $\theta$ is the angle between the estimated $\rho$ line and the x-axis. The angle we wish to estimate is the angle between the leaflet and the annulus plane. To get this angle, we first need the angle between the predicted valve line and the line going through the annulus point of the leaflet and is parallel to the y-axis. Using origo in the top left corner of a two-dimensional matrix we get the angle for the left leaflet expressed in equation (18) and for the right leaflet expressed in equation (19).

$$\phi_{left} = \frac{\pi}{2} - |\theta| \tag{18}$$

$$\phi_{right} = \frac{\pi}{2} - \theta \tag{19}$$

The angle $\phi$ can then be used to get the angle between the plane between the two annulus points and the leaflet. The angle, $\alpha$, between the two annulus points, $(x_0, y_0)$ and $(x_1, y_1)$, is calculated using equation (20).

$$\alpha = \arctan \frac{|y_1 - y_0|}{|x_1 - x_0|} \tag{20}$$

The resulting estimated angle, $\beta$, between the leaflet and the annulus plane is shown in equation (21). Figure 4 illustrates the layout for the transform performed on an example posterior leaflet.

$$\beta = \phi - \alpha \tag{21}$$

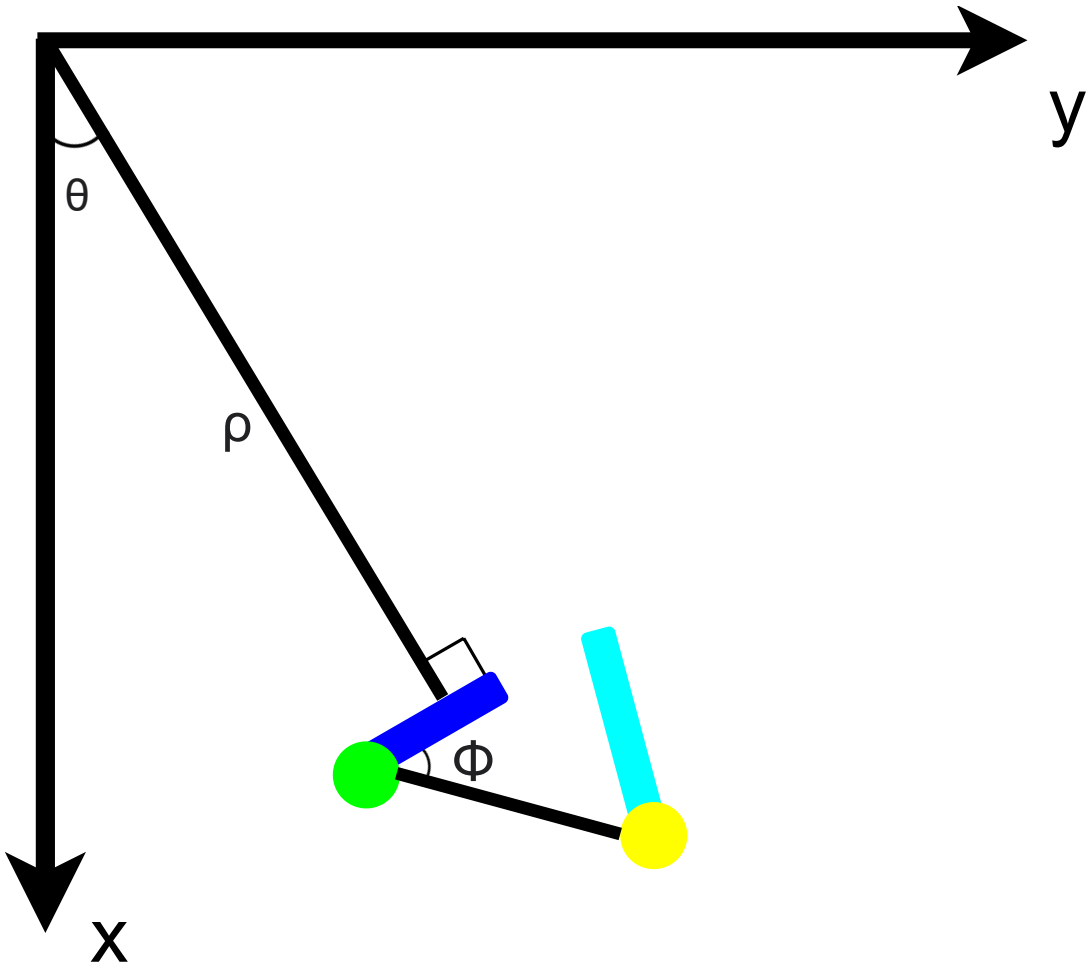



Figure 4: Layout used for angle estimation between a leaflet and its associated annulus plane.

# 3 Methodology

## 3.1 Overview

Two main configurations of a U-Net model are trained to segment the same valve apparatus as a trained clinician. One model is trained using only the ground truth segmentations performed by the clinician. The other uses auto-generated segmentations of the left ventricle, myocardium, and left atrium, created by another pre-trained network to the valve segmentations. B-mode images are extracted from a selection of recordings on the DICOM-file format, and the valve annotations are fetched from a database. The data then goes through a pre-processing stage where the data is tailored to fit the U-Net model. The auto-generated segmentations are added during this stage for the model using them. The pre-processing stage also includes data augmentation, which artificially increases the number of samples in the data set. The data is then fed into the network for training. After training, the output data is post-processed to clean the output. The networks are implemented in Python using the Keras framework inside the Tensorflow Python package [28]. The training and testing are performed on a NVIDIA Quadro P5000 GPU.

## 3.2 Data set

Since we want to detect faulty mitral valves, it is natural to use annotated ultrasound recordings with different deformations of the valve. The data samples used for training and testing are taken from surgical screenings done at Tikur Anbessa Specialized Hospital in Addis Ababa, Ethiopia. A Norwegian heart team did the screenings during five missions from March 2016 to November 2019. The patients were recruited from a waiting list of 6000 local patients with RHD waiting for surgery. Local staff selected 88 patients for surgical screenings. Valve annotations have been done on 253 recordings from 48 of these patients and form the basis of our data set. One patient had two sessions, one before operation and one after. Thus, the whole data set contains almost exclusively patients with varying extents of some form of heart disease.

All recordings are 2D recordings acquired using a GE Vivid E9 ultrasound scanner (GE Ultrasound, Horten, Norway). The recordings are in a DICOM file format. Each file contains a sequence of differing lengths and metadata. It is beneficial to be able to detect the valve at different stages of the cardiac cycle. To accomplice this, frames at the end of systole and end of diastole were annotated. In addition, some frames in between these two endpoints have been annotated to increase the variety of the data set.

The data set contains various echocardiographic views, namely PLAX, ALAX, A4C, and A2C. The images are $275 \times 256$ pixels each. The network we are using requires a constant size for all the images. In addition, the sides of the images need to be a power of two. 275 is not a power of two, which means that we need to add padding to the images. We add zero paddings to the images, so their shape is $512 \times 512$ when they go into the network. This is required for two main reasons. The U-Net architecture requires input data dividable by $2^n$, where $n$ is the number of compression layers, and because of floating-point arithmetic optimization. We solve this problem by zero-padding the images before they go into the network.

## 3.3 Manual Annotations

*The following section is taken from [8]* and modified to fit the context of this report.

We are using a neural network trained using supervised learning. We need labels of ground truths for the valve and annulus points we want the network to segment. Manual segmentation of the mitral valve requires experience and is not a task suited for anyone. The help of trained clinicians is needed to get reliable annotations. To create the annotations, the clinicians use an application tailored for valve annotations.

### 3.3.1 Annotation Process

The flowchart shown in figure 5 illustrates the functionality of the application. The user needs to choose an annotation task and data cohort the annotation task should be performed on. The application then fetches previous annotations done on the selected data cohort. Then, the application will display the first DICOM-file in the cohort or the first file with no annotations. In this way, the application will remember where the last user left off and allow the next user to continue the work effortlessly. Before any annotation can be performed, the clinician needs to evaluate if the quality of the recording is suitable for valve evaluation. An option to save the current DICOM-file as rejected is added to the pipeline. If the file is suited, the user can start annotating. The user first traces the valve using a free draw tool and then marks the annulus points with a circle for each point. These annotations are then stored in a database.

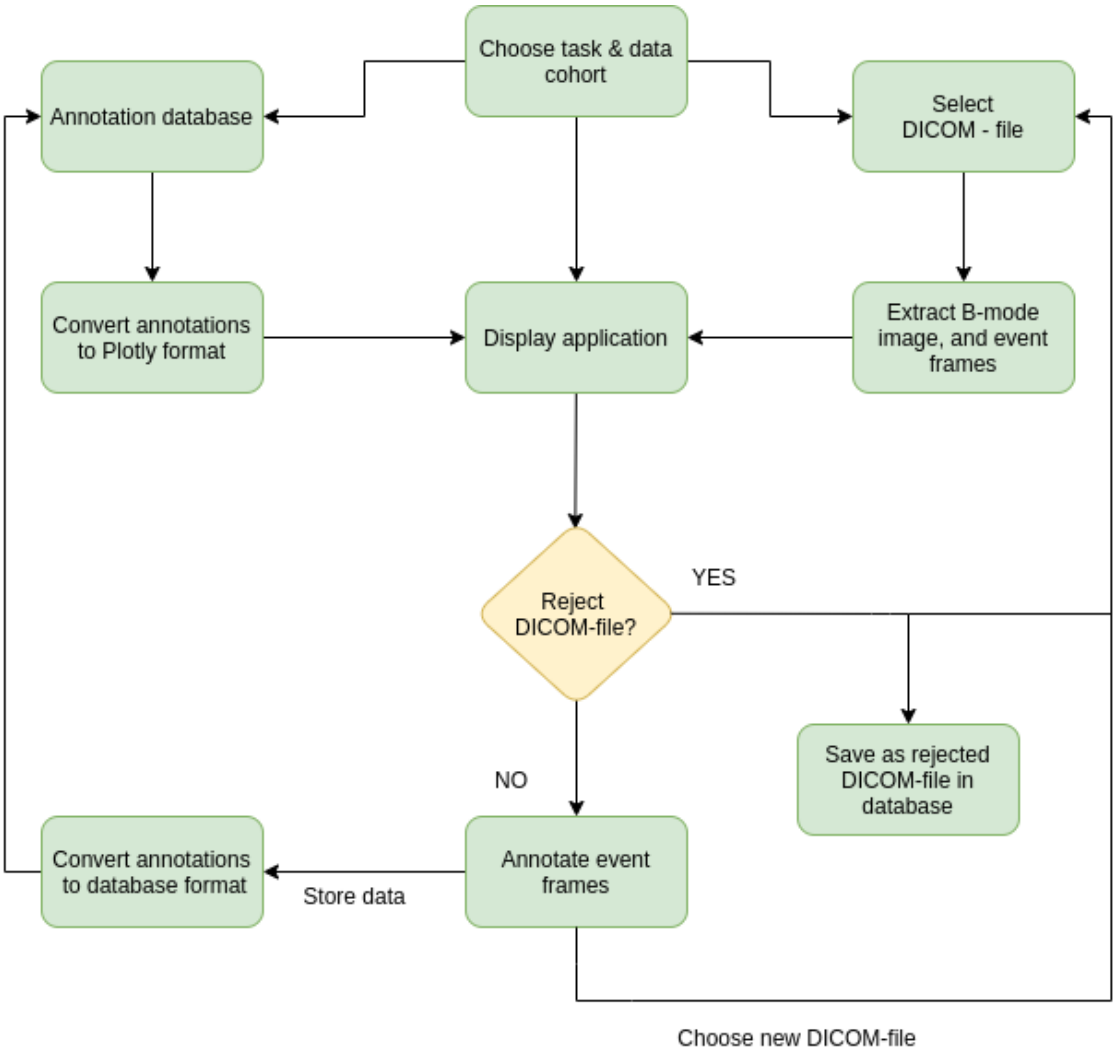
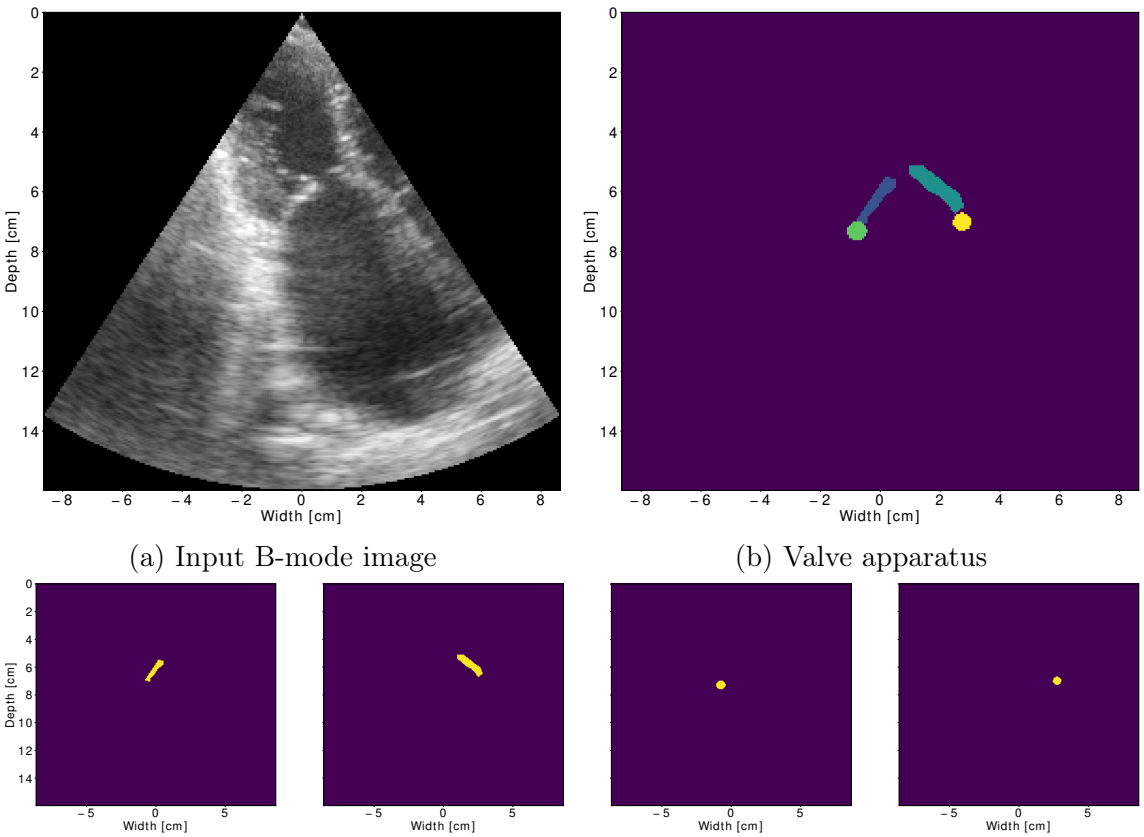



Figure 5: A flowchart showing the functionality of the annotation tool. It starts by choosing the annotation task and data cohort. Previous annotations are fetched from the database and converted to Plotly [29] format. At the same time, the first DICOM-file without annotations in the cohort is selected, and the B-mode image sequence and event frames are fetched. These are displayed with settings determined by the chosen task. The user then rejects the DICOM-file if it is not suited for the task, and this file is saved as rejected. If this is not the case, the user annotates the event frames, which are converted to the database format and saved. When all the desired frames have been annotated, the user chooses the next DICOM-file or exits the application. The figure and caption are taken from [8].

There are two different types of annotations, circles and paths. The circles consist of two x- and two y-coordinates, and the paths are series of x- and y-coordinates. This format needs to be converted to a binary annotation mask to be compatible with the neural network.

The labels we put into the network need 2D arrays with identical dimensions to the input image. This array must contain the pixel values of the image and a mask with information about what class each pixel is assigned. When we feed the data into the network, we one-hot encode the data, resulting in one layer for each class. Figure 6 shows a B-mode image of the heart and the resulting valve annotations in one layer and the resulting one-hot encoded layers used during training.




(a) Input B-mode image

(b) Valve apparatus

(c) One-hot encoded valve apparatus

Figure 6: Example B-mode image (a) and segmentation of the posterior leaflet (blue), posterior annulus (green), anterior leaflet (teal), and anterior annulus (yellow) produced by a trained clinician (b). The resulting one-hot encoded valve segmentations are shown in (c).

## 3.4 Auto-generated annotations

The annotations of the mitral valve and the annulus points make up a small portion of the image. This could pose a problem for the neural network, as small objects are harder to detect. One possible solution could be to provide the network with more context by adding annotations of other objects in the image. It is logical to add annotations of the left ventricle, left atrium, and myocardium in the case of the mitral valve since they are present in all echocardiographic views where the mitral valve is present.

The annotations can be created manually by clinicians, but the annotation process requires much time. To save time, we automatically annotate these objects using an already existing neural network trained for this task. This can be accomplished by using the network proposed by Smistad et al. [30]. The segmentation network has been trained to segment out the left ventricle, myocardium, and left atrium. For future reference, we name this network SmistadLVLA. One drawback of using auto-generated annotations is that the resulting annotations heavily depend on the data the network has been trained with. The network proposed by Smistad et al. [30] has been trained on data of mostly healthy patients and of good quality, which is very different from the Ethiopian data we wish to annotate using the network. This will result in annotations of varying quality and will impact the accuracy of the final segmentations.

Figures 7 and 8 show good and bad examples, respectively, of auto-generated annotations by the SmistadLVLA network. In figure 8 we can see that the automatic annotation of the left atrium is virtually zero. To combat this problem, we use another pre-trained network. It uses the same network structure proposed by Smistad et al. [30], but trained on different data and only for segmentation of the left atrium. We name this network SmistadLA for future reference. The performance of the SmistadLA network is better for some images and worse for others compared to the SmistadLVLA network. In figure 9 an example where the SmistadLA network makes a better prediction for the LA is shown, and figure 10 illustrates an example where the SmistadLVLA network performs a better prediction. There are some instances where both the networks predict almost no segmentation of the left atrium, shown in figure 11.

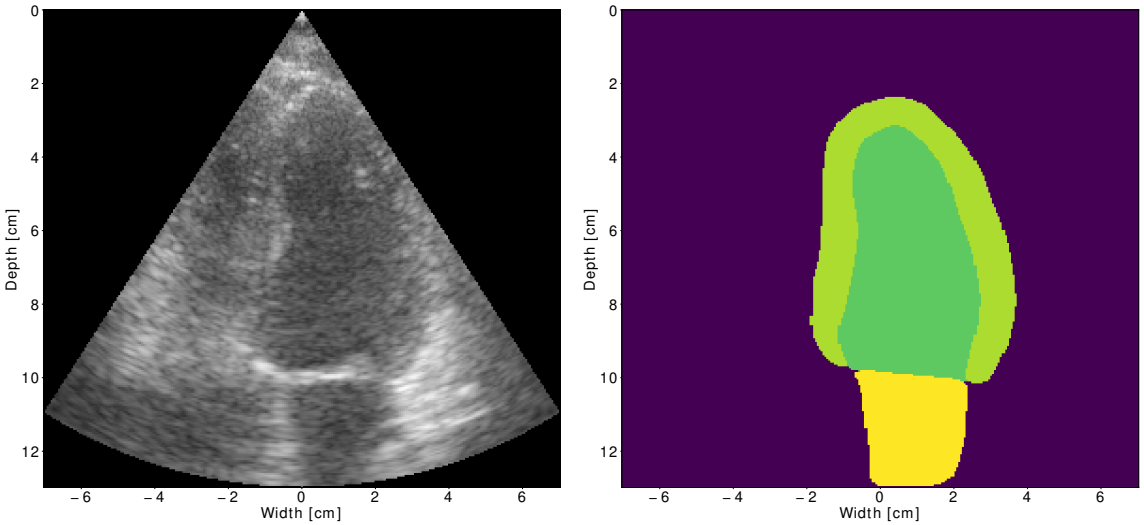
## 3.5 Annotation Modifications

Given the varying quality of the automatically generated annotations, it would be beneficial to modify them to potentially increase the networks' performance.

### 3.5.1 Reassignment

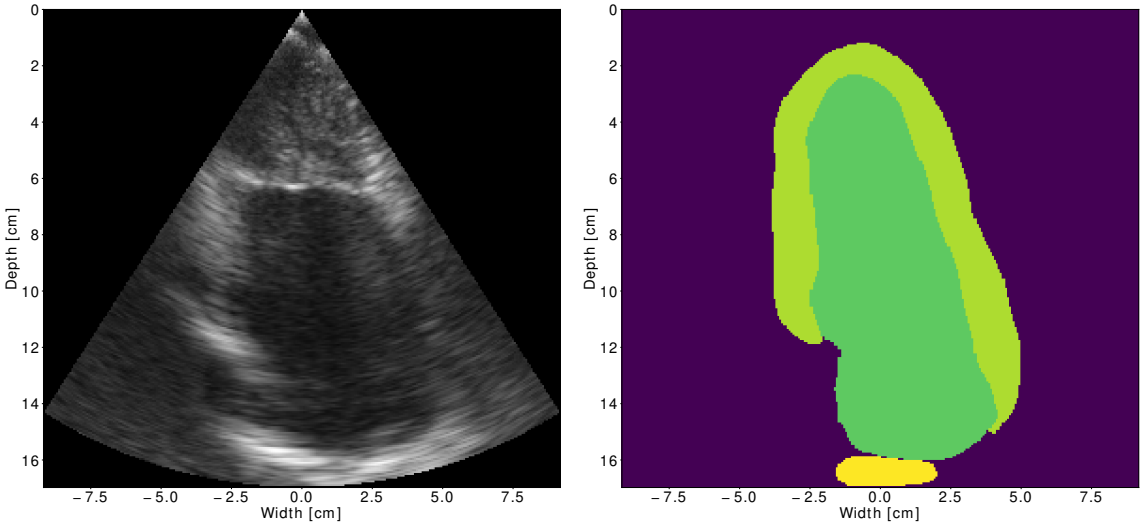
Given the unusual echocardiographic data, some of the auto-generated segmentations are not ideal, as shown in figure 8. Sometimes the generated left ventricle segmentation is actually a segmentation of the left atrium, and in some cases, the atrium walls are segmented as myocardium. Using incorrect segmentations during training is not optimal, but it is vital to keep as much data as possible given the small data set at hand.

We know that the left ventricle and myocardium are located above the line between the two annulus points and that the left atrium is located below. Since we have



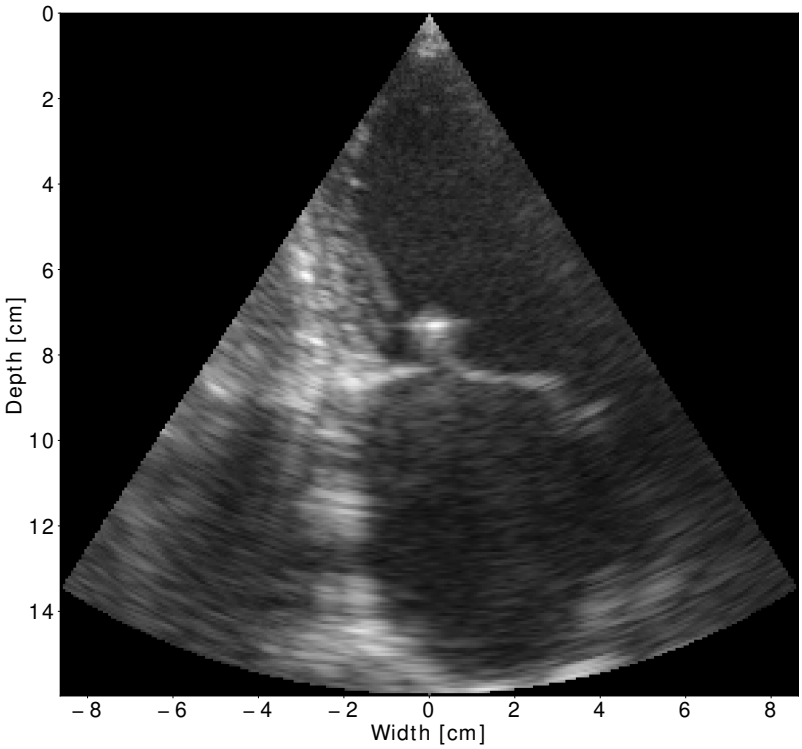

(a) Input B-mode image (b) Segmentation by SmistadLVLA

Figure 7: An example of good predictions of the left ventricle (dark green), myocardium (light green) and left atrium (yellow) performed by the SmistadLVLA network.

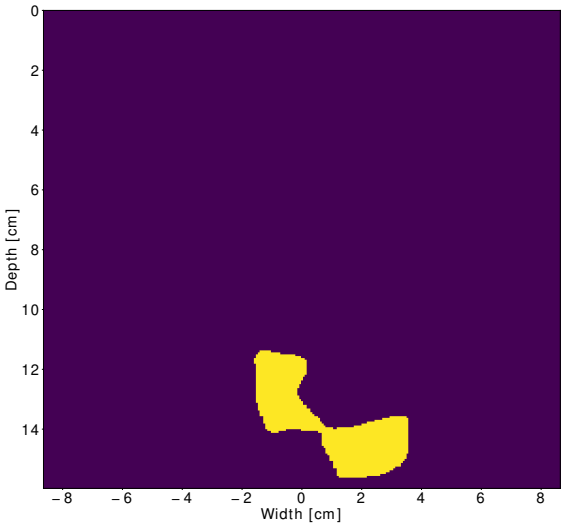



(a) Input B-mode image (b) Segmentations by SmistadLVLA

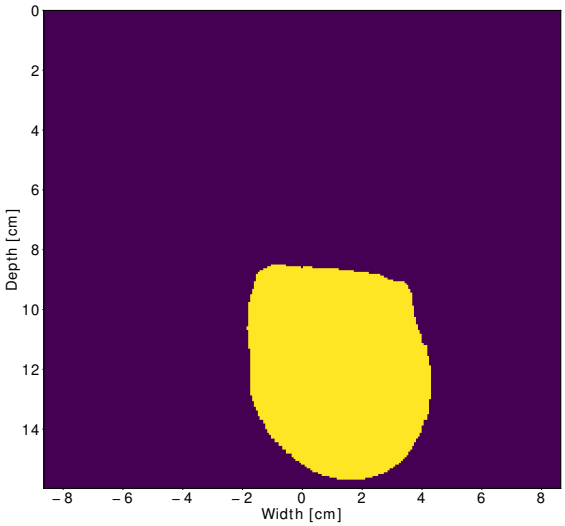
Figure 8: An example of bad predictions of the left ventricle, myocardium and left atrium performed by the SmistadLVLA network.




(a) Input B-mode image

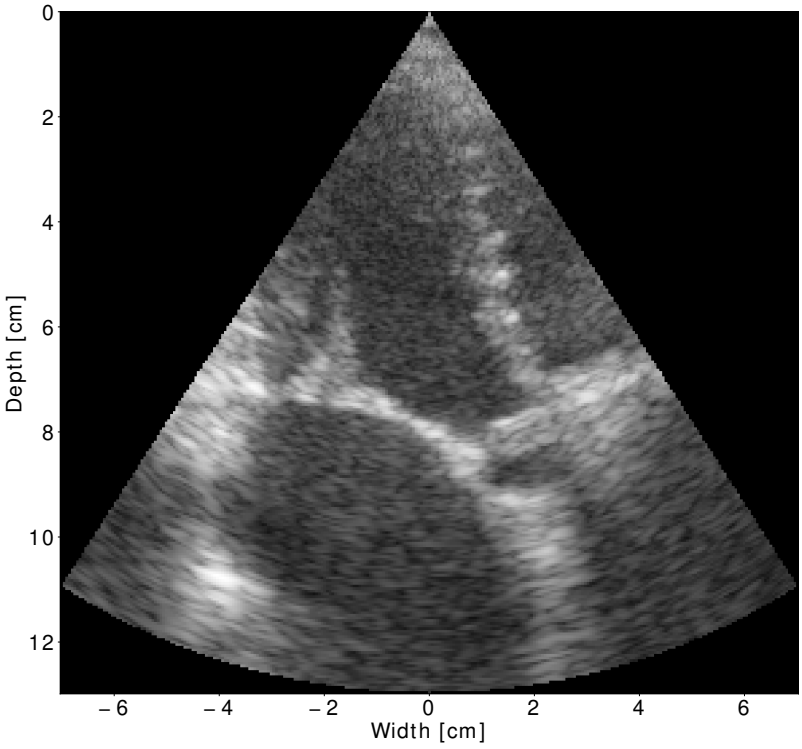



(b) SmistadLVLA

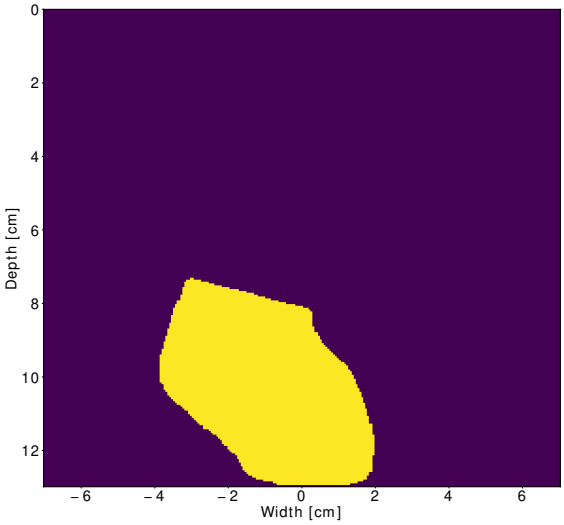



(c) SmistadLA

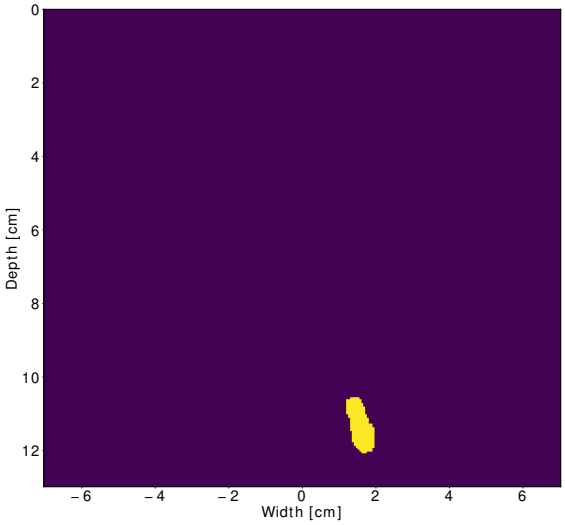
Figure 9: Prediction of the left atrium in a B-mode image where the prediction (b) by the SmistadLVLA network is visibly worse than the prediction (c) by the SmistadLA network.




(a) Input B-mode image

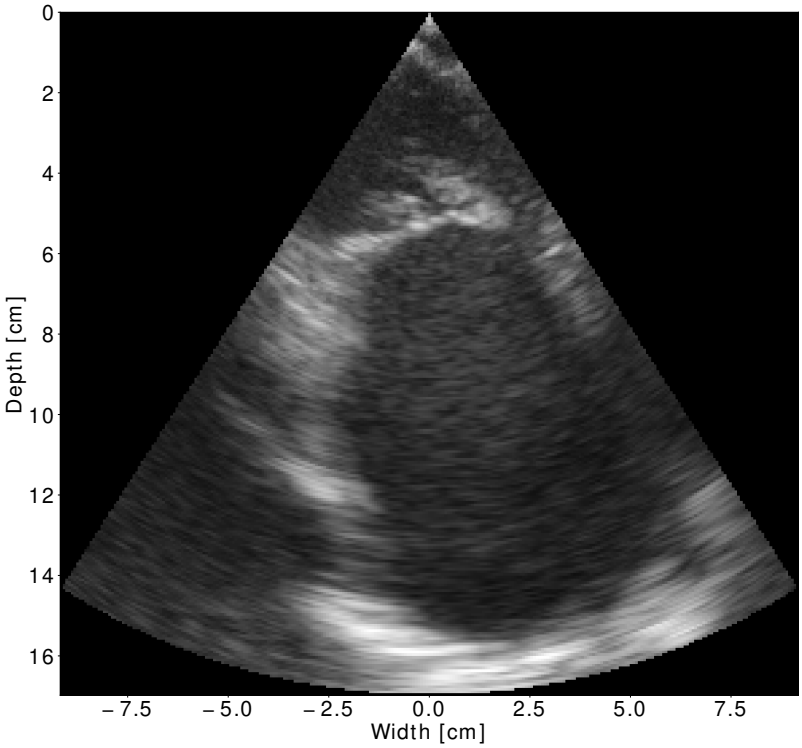



(b) SmistadLVLA

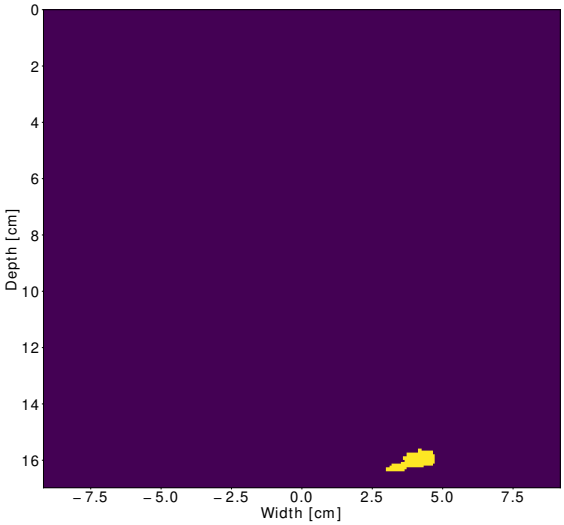



(c) SmistadLA

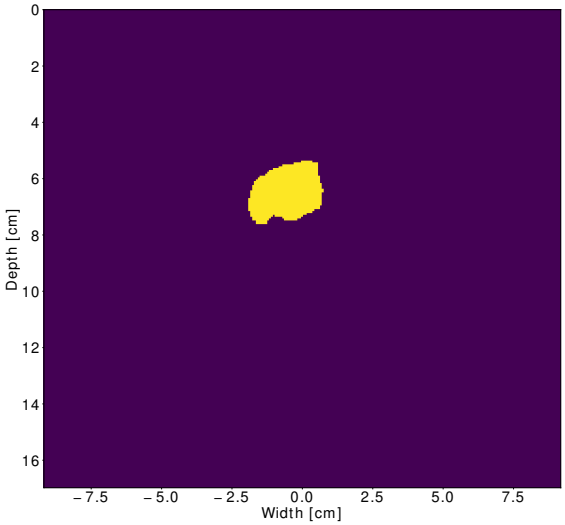
Figure 10: Prediction of the left atrium in a B-mode image where the prediction (b) by the SmistadLVLA network is visibly better than the prediction (c) by the SmistadLA network.




(a) Input B-mode image




(b) SmistadLVLA




(c) SmistadLA

Figure 11: Prediction of the left atrium in a B-mode image where both the prediction (b) by the SmistadLVLA and (c) by SmistadLA are subpar.

annotations of the annulus points, we can draw a line between the two points and draw a horizontal line from each point to the edge of the image. This can be used to create two boolean filters we can use on the auto-generated annotations. One with positive values above the annulus plane and one with positive below.

These filters can be used to reassign incorrect segmentations. In the cases where the auto-generated left ventricle segmentation by SmistadLVLA is incorrect, it often segments out the left atrium while the segmentation of the left atrium is almost empty. To combat this problem, we merge the left ventricle and atrium by SmistadLVLA and the segmentation of the left atrium by SmistadLA into one layer. Then we apply the annulus plane boolean filters and assign the segmentations above the annulus line as ventricle and annotation bellow it as atrium. For the myocardium, only the segmentation above the annulus plane is kept. The segmentations below are discarded. Figure 12 show an example B-mode image, original automatic segmentations, annotation of the valve apparatus, the resulting filters, and the new reassigned segmentations.

This leaves two considerable problems with the automatic annotations, which are hard to solve. Some samples result in segmentations where only part of the left atrium has been segmented after the reassignment, illustrated in the example in figure 13. The other problem is illustrated in figure 14, where there is no segmentation of the left ventricle and myocardium left after the reassignment process.

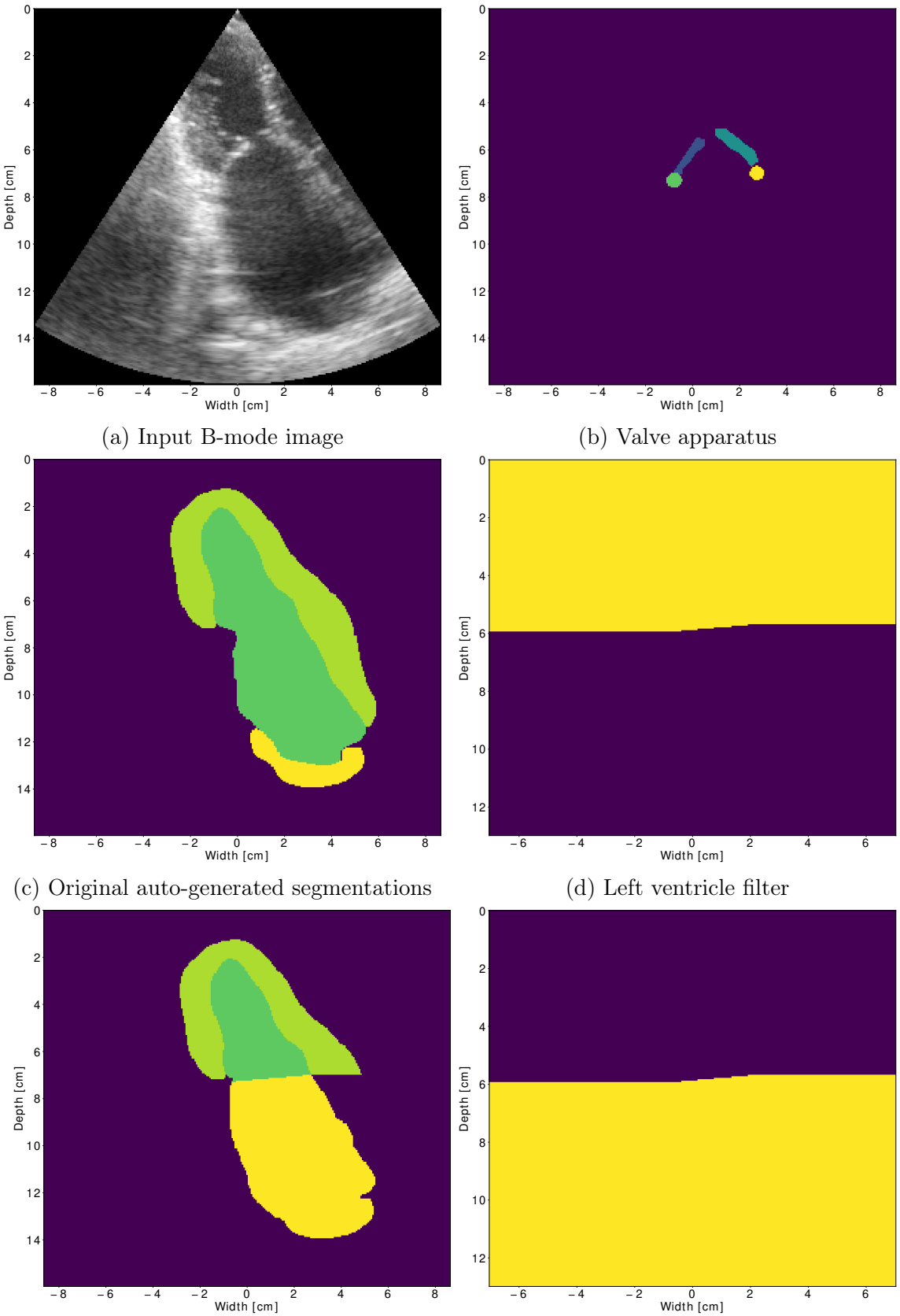
### 3.5.2 Remove shared pixels

The automatic segmentations often overlap the annotations of the valve and annulus points. This can pose a problem for convergence in the network and the network's confidence about the class of a given pixel. In the pre-processing of the data, we therefore remove pixels in the auto-generated annotations that share index values with the valve apparatus. Figure 15 shows an example of the original automatic segmentations, valve and annulus points annotations, and the resulting segmentations with cutouts.

### 3.5.3 Background class

In some echocardiographic views, such as A4C, the other chambers and valves are present in the picture. These structures can confuse the network and impact performance negatively. To reduce this impact, we make a background class by adding all annotations into one class using a logical or operation and then invert the mask. The addition of a background class is also beneficial with regard to convergence during training. Without a background class, some configurations of the data set and neural network may struggle to converge.

An example with all annotations for one image and its resulting background class is shown in figure 16. The background class created from only the valve apparatus is also shown in the same figure.




(a) Input B-mode image

(b) Valve apparatus

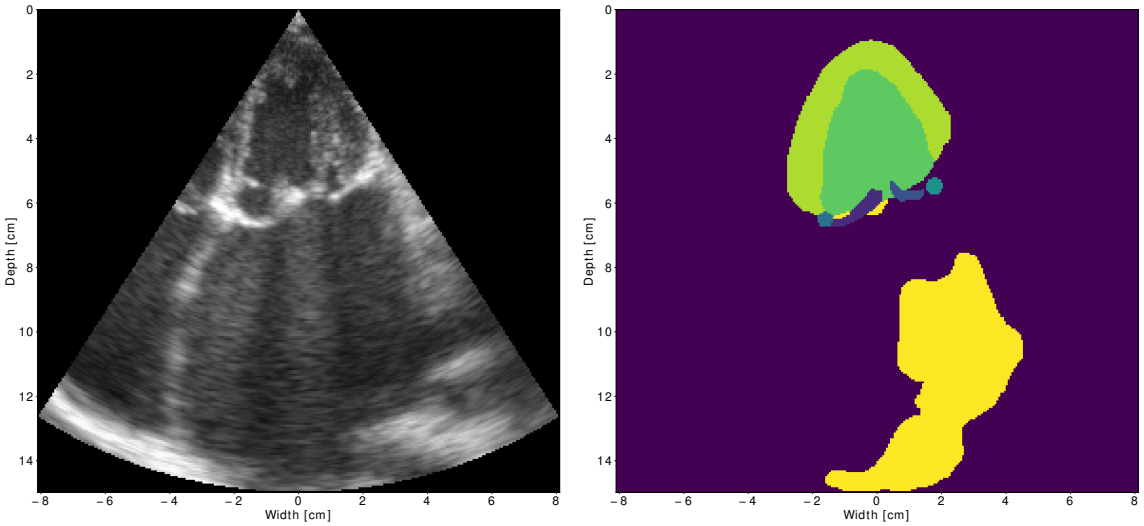
(c) Original auto-generated segmentations

(d) Left ventricle filter

(e) Filtered segmentations

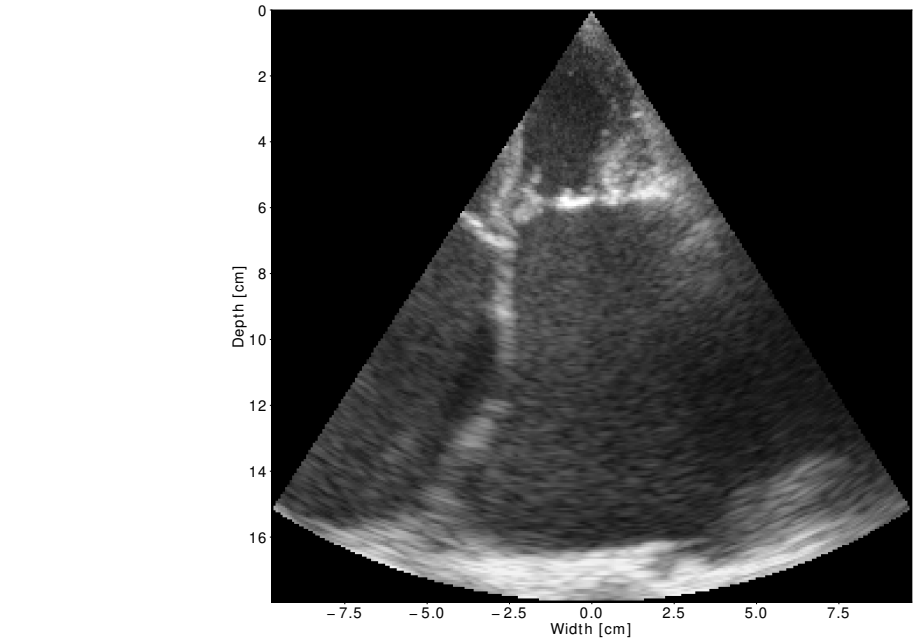
(f) Left atrium filter

Figure 12: B-mode image (a) with its (b) associated valve apparatus annotations. (d) and (f) show the resulting boolean ventricle filter and boolean left atrium filter respectively. (c) show the original automatic annotations of the left ventricle, left atrium, and myocardium. (e) show the resulting annotations after reassignment.

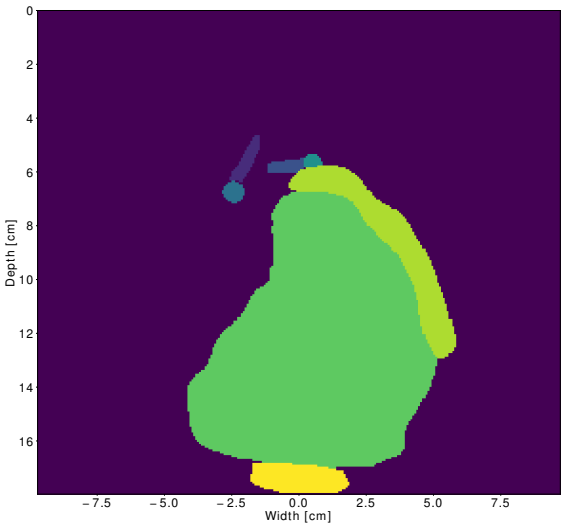



(a) Input B-mode image (b) Reassigned

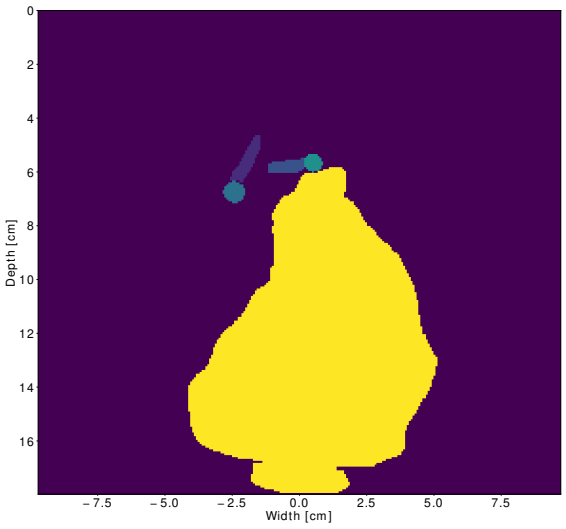
Figure 13: An example after reassignment have been performed, where the segmentation of the left atrium (yellow) only covers a small part of the actual volume of the chamber.




(a) Input B-mode image

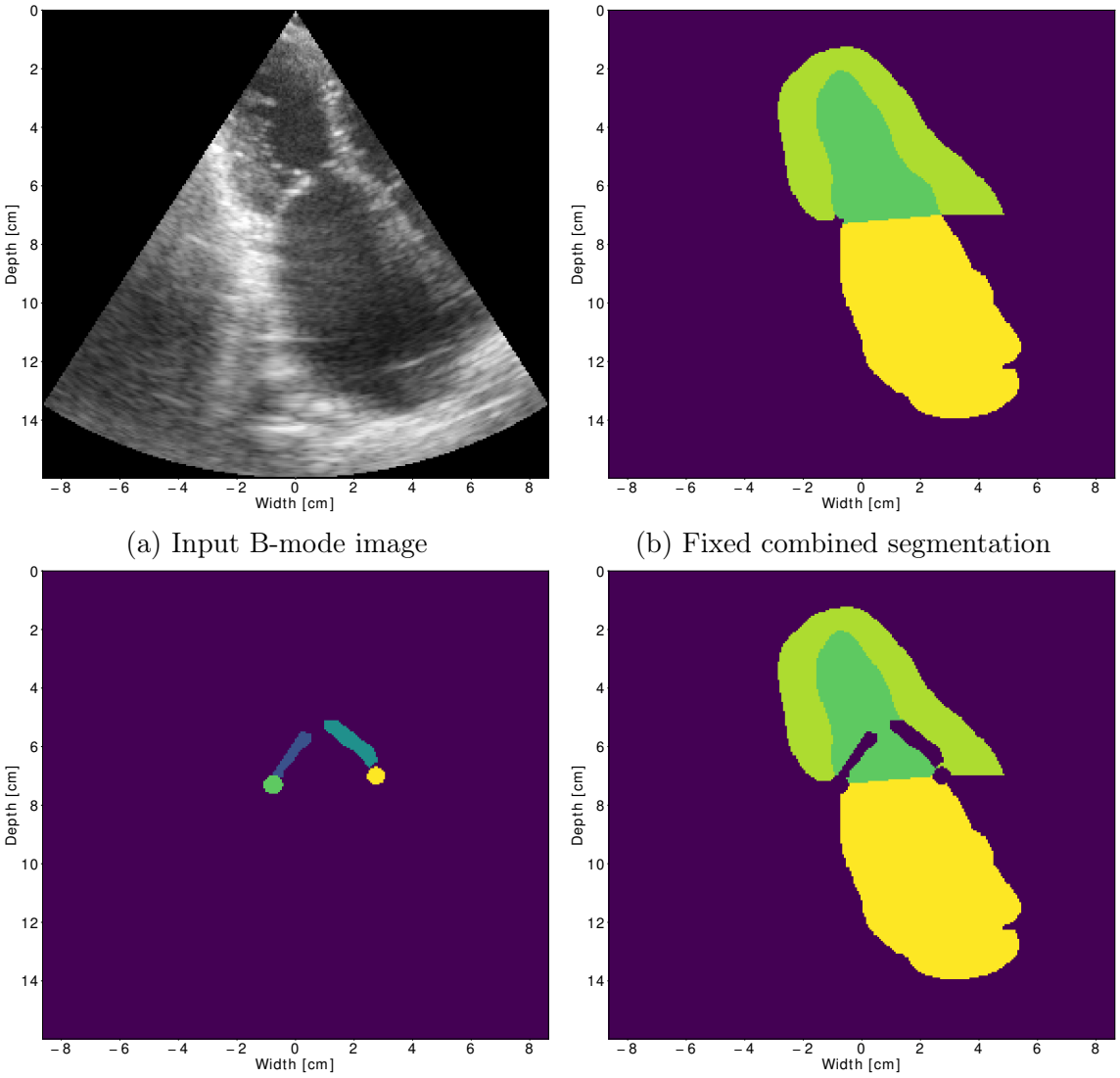



(b) SmistadLVLA




(c) Reassigned

Figure 14: One of the cases where the input B-mode image (a) results in a poor segmentation (b) by the SmistadLVLA network. Thus giving no segmentation (c) of the left ventricle or myocardium after reassignment.



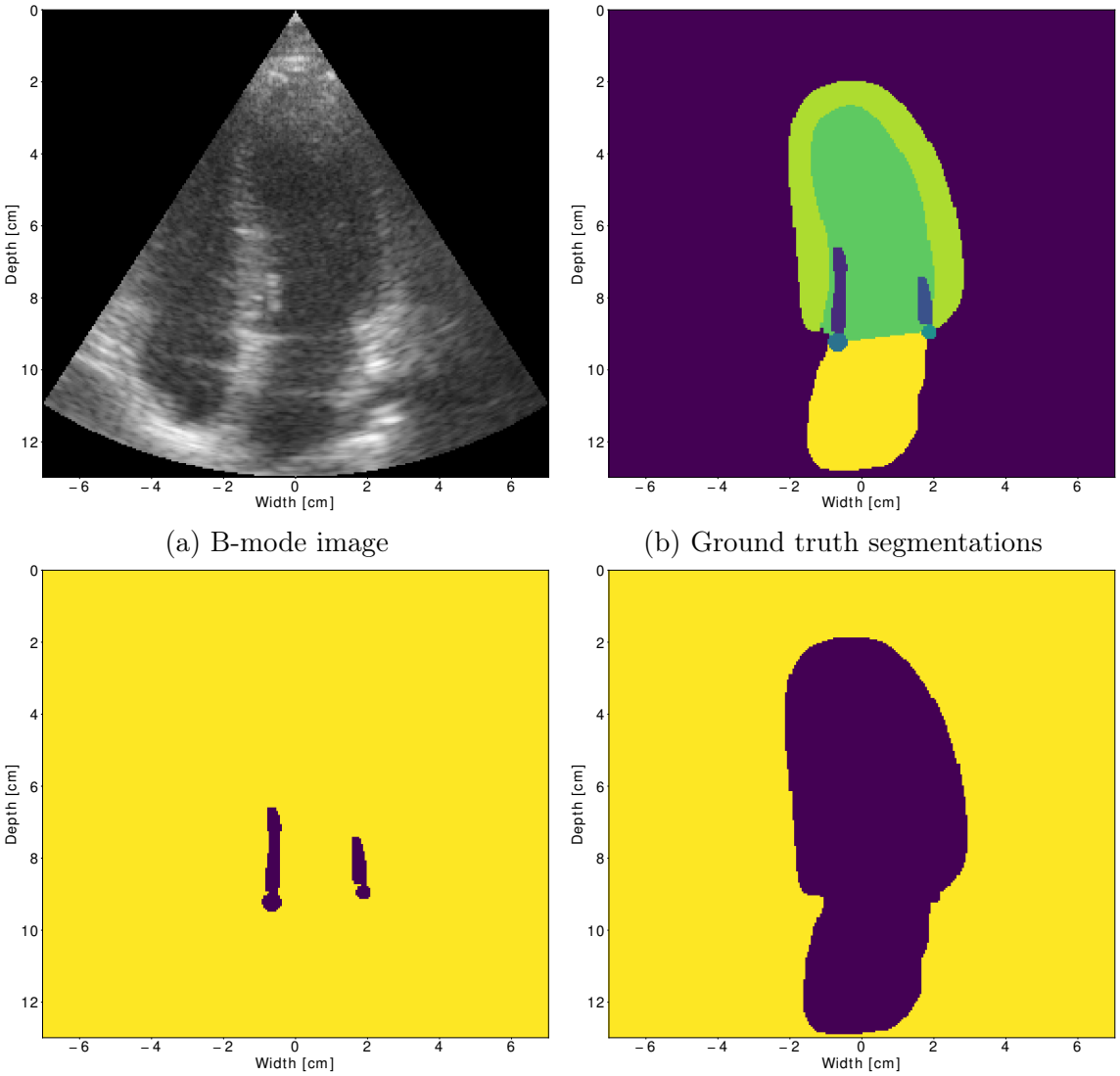

(a) Input B-mode image

(b) Fixed combined segmentation

(c) Mitral valve and annulus annotation.

(d) Resulting segmentation with cutout.

Figure 15: Example showing (a) B-mode input image, (b) the ground truth segmentation of the left ventricle, left atrium, and myocardium, (c) manual valve and annulus annotation, and the resulting segmentations of the left ventricle, left atrium and myocardium with cutouts.




(a) B-mode image

(b) Ground truth segmentations

(c) Resulting background, valve apparatus

(d) Resulting background, all classes

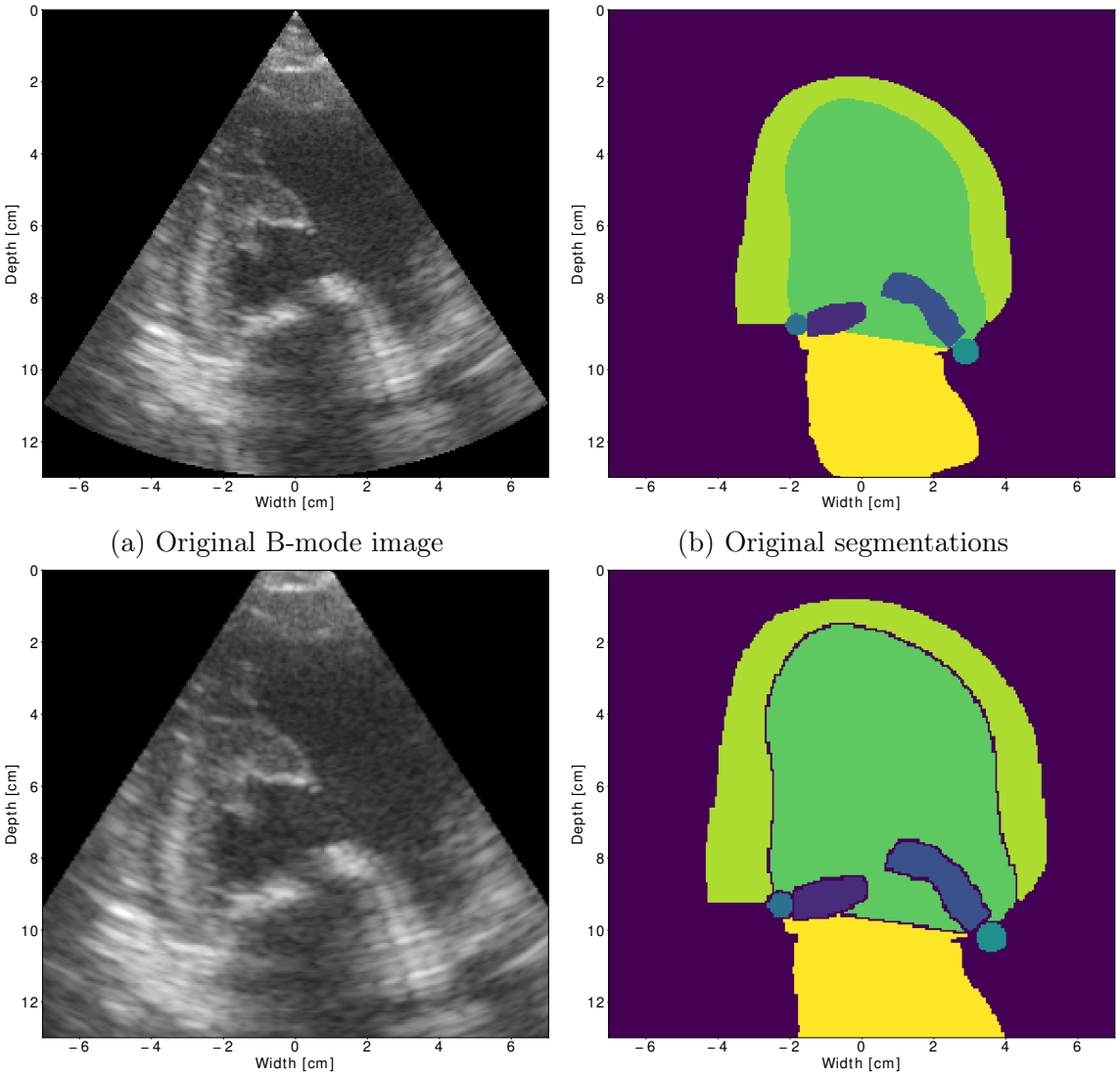
Figure 16: An original B-mode input image (a) and it associated ground truth annotations (b) and the resulting background class with (c) only valve apparatus, and with (d) all classes.

## 3.6 Data Augmentation

Given the small data set at hand, we use data augmentation to increase the number of samples. We perform cropping, gamma, rotation, and Gaussian augmentations. Gamma and Gaussian augmentations are performed only on the input B-mode image. A copy of the associated ground truth masks is appended alongside the augmented B-mode image. Rotation and cropping augmentation are performed on both the B-mode image and the ground truth masks.

Cropping and gamma augmentations are done using the TensorFlow.image module in Python. The cropping is done by zooming in on the picture with a random percentage in an empirically selected interval of $0.75 \pm 0.05$. This changes the dimensions of the image, so the image is resized back to the original dimensions. In figure 17 an example of an original image and mask, and a cropped version shown. Gamma augmentation is only applied to the input image and changes its brightness with a given constant. This constant is chosen at random in an empirically chosen interval of $0.7 \pm 0.2$ for brightening of images and $1.35 \pm 0.2$ for darkening. Figure 18 shows an example of an original image, its brightened and darkened counterparts.

Rotation and Gaussian augmentations are done using the Scipy.ndimage module in Python. The images are rotated with 10 and $-5$ degrees. Figure 19 shows an example of rotation 10 degrees. For the Gaussian augmentation a Gaussian filter, described in section 2.3.3, have been applied to the images with $\sigma = 0.5 \pm 0.2$. Figure 20 shows examples of Gaussian augmentation with $\sigma = 1$ and $\sigma = 3$. The example values are chosen to clearly show the impact of the different values.



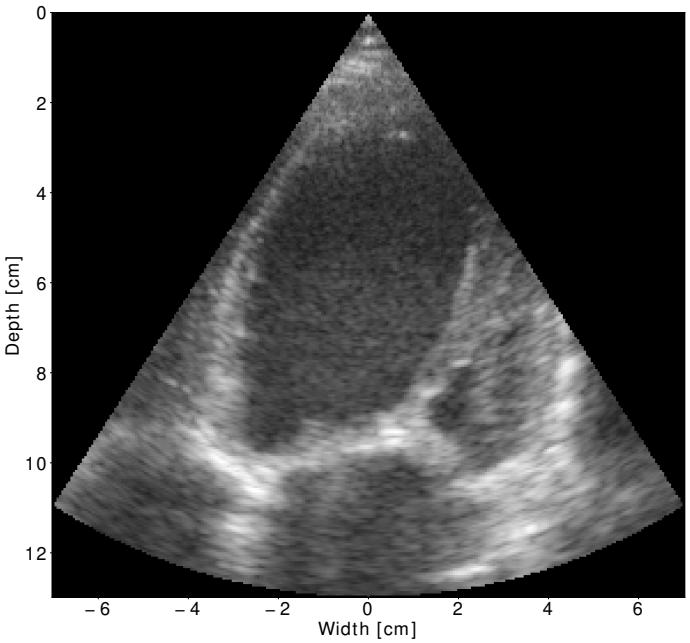

(a) Original B-mode image

(b) Original segmentations

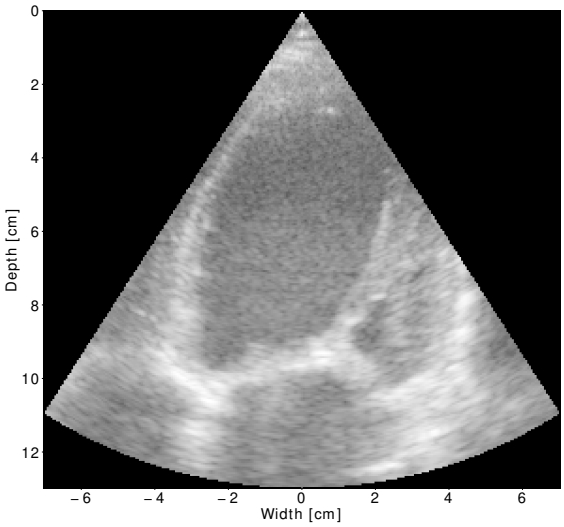
(c) Cropped B-mode image

(d) Cropped segmentations

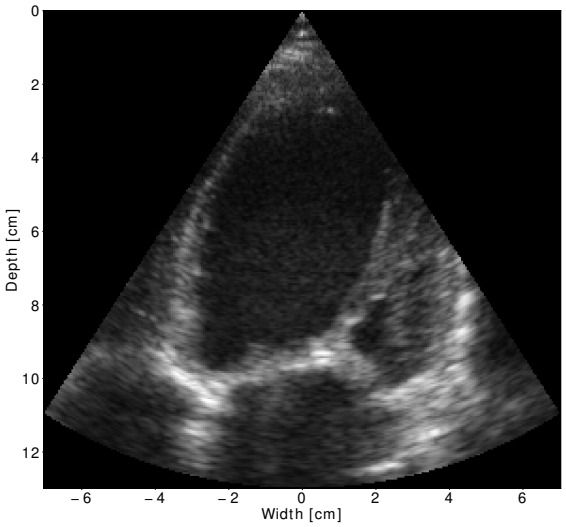
Figure 17: An original B-mode input image (a) and it associated ground truth segmentations (b) and the resulting cropped B-mode image (c) and segmentations. This example uses a center crop value of 0.8.




(a) Original image

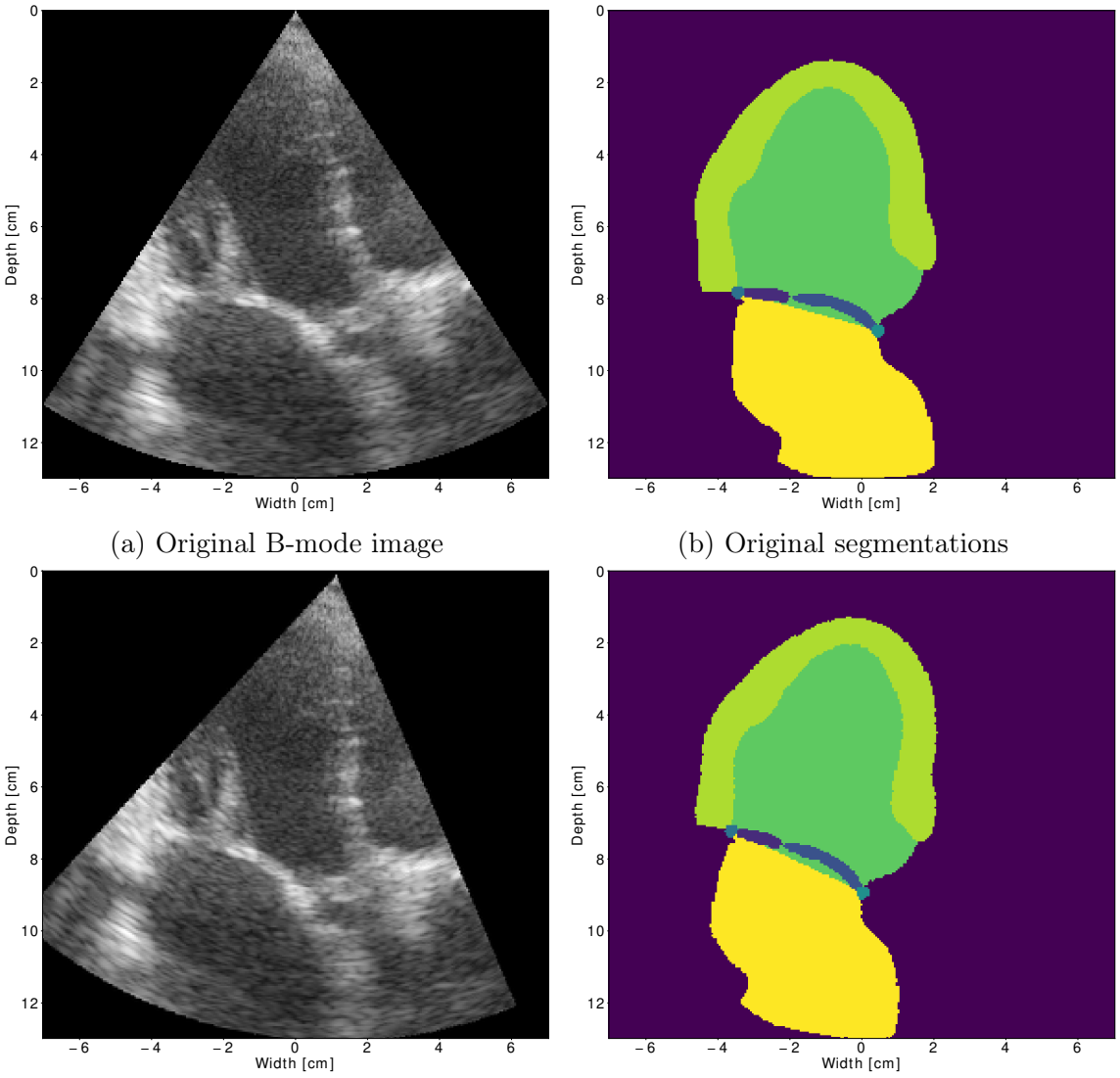



(b) $\gamma = 0.45$




(c) $\gamma = 1.65$

Figure 18: An original B-mode input image (a) with its corresponding gamma augmented counterparts with $\gamma = 0.45$ (b) and $\gamma = 1.65$ (c).



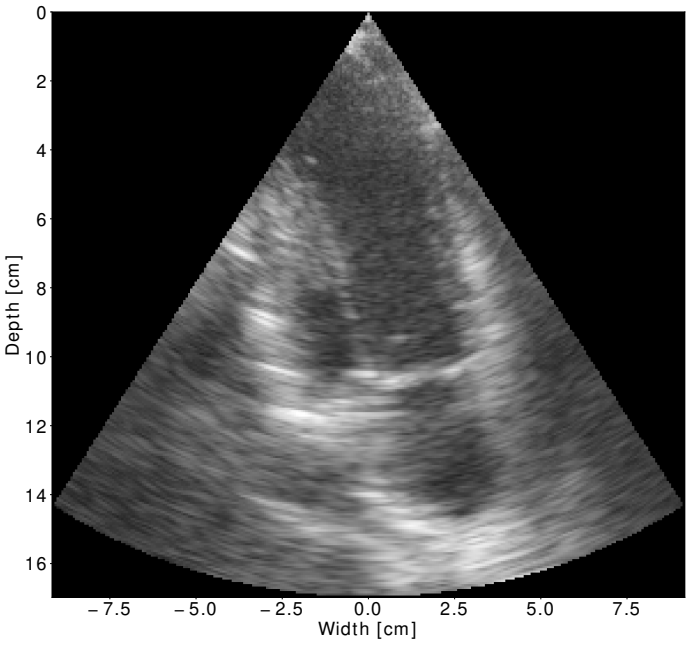

(a) Original B-mode image

(b) Original segmentations

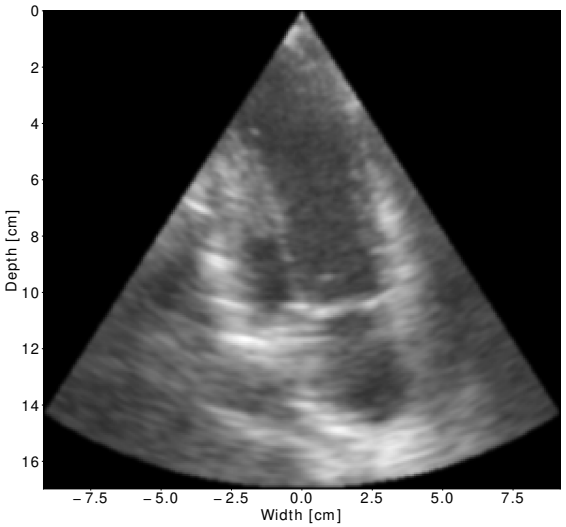
(c) Rotated B-mode image

(d) Rotated segmentations

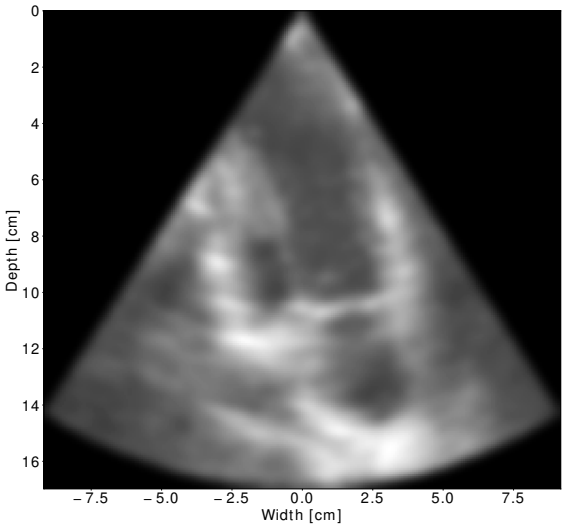
Figure 19: An original B-mode input image (a) and it associated ground truth segmentations (b) and the resulting rotated B-mode image (c) and segmentations. This example has been rotated 10 degrees relative to the vertical centerline of the image.




(a) Original image




(b) $\sigma = 1$




(c) $\sigma = 3$

Figure 20: An original B-mode input image (a) with its corresponding Gaussian augmented counterparts with $\sigma = 1$ (b) and $\sigma = 3$ (c).

## 3.7 U-Net

### 3.7.1 Model Architecture

The proposed structure we use for the network is the U-Net architecture described in section 2.3.2. Figure 21 shows the implemented U-Net architecture. The model is composed of 30 hidden layers, including four $2 \times 2$ max-pooling layer and four $2 \times 2$ transposed convolution layers, resulting in a total of $1,940,885$ model parameters. The number of parameters will have small variations depending on the number of classes in the ground truth masks. The final layer at each compression level has a skip-connection to the first layer of the corresponding decompression level. The skip-connection is implemented using a concatenation layer, meaning the data from the compression layer passing through the skip-connection is concatenated with the data on the receiving end of the connection. All the hidden layers use the ReLU activation function. Two variations of the final output convolution layer have been used. One where the sigmoid activation function is used, and one where the softmax activation function is used. These activation functions are described in section 2.3.1. The input layer takes in a 4D tensor $B \times W \times H \times C$, where $B$ is the batch size, $W$ and $H$ are the width and height of the image, respectively, and $C$ is the number of channels at the input. For the input ultrasound images, we use black and white images, i.e., one channel images. The ground truth masks have one channel for each class. The network will therefore have the same amount of output channels as the number of classes. The network has four compression layers. This means that the $2 \times 2$ max-pooing is applied four times to the input image. As a result, both the height and the width are compressed with a factor of 16 at the last compression layer. Thus, the chosen values for height and width need to be divisible by 16.

### 3.7.2 Training and Testing

The model is trained and tested with several different configurations of the input data. We train models using only the valve apparatus and models using the automatic annotations of the left ventricle, left atrium, and myocardium. In addition, some changes are made concerning output activation function and loss function. The different model variations are listed in table 1.

Table 1: Model variations. All the variations also use the background class.

| Name | Loss function | Classes | LV, MY, and LA | Reassigned |
|---|---|---|---|---|
| U-Net OV-B | Binary CE | 5 | No | – |
| U-Net OV-C | Categorical CE | 5 | No | – |
| U-Net Auto | Categorical CE | 8 | Yes | No |
| U-Net Auto-R | Categorical CE | 8 | Yes | Yes |

The data set contains 824 images, where 662 are used for training and 162 are used for testing. To mitigate overfitting, we use a portion of the training set as a validation set during training. A good indication of overfitting is a rising validation loss. The validation loss is calculated after each epoch. The validation loss is then compared to the previous smallest validation loss. The model is saved if the validation loss of the current epoch is smaller than the previous minimum. All models are trained for 50 epochs.

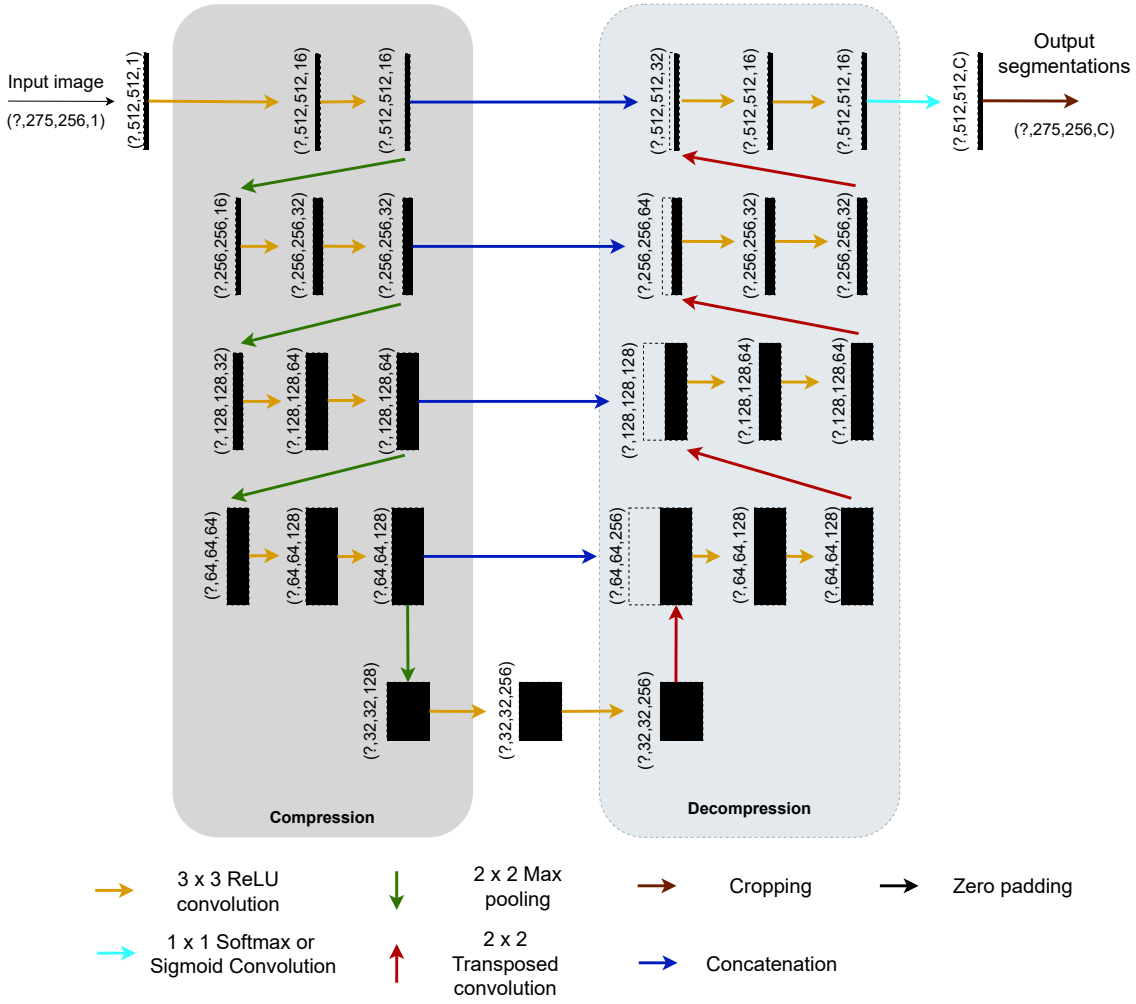



Figure 21: The implemented U-Net architecture. The tuple next to each layer denotes the shape of the tensor output from the given layer. The tensors have four dimensions. The first dimension is the batch size, an arbitrary integer denoted by "?". The second and third dimensions are the image height and width. The fourth dimension is the channels i.e. the feature maps. The channel dimension of the last output layer, $C$, will be equal to the number of classes the network is tasked to segment out.

The last 10% of the training set is used for validation during training. This gives a small validation set and is not ideal. With a small validation set, there is a high probability for a low variance in the set. A low variance could give a wrong indication of the validation, either too positive or negative. To decrease the probability of low variance in the validation set, we shuffle the training set before the validation data is split. This will give a different validation set for each session, thus lowering the chance for wrong indications from the validation set.

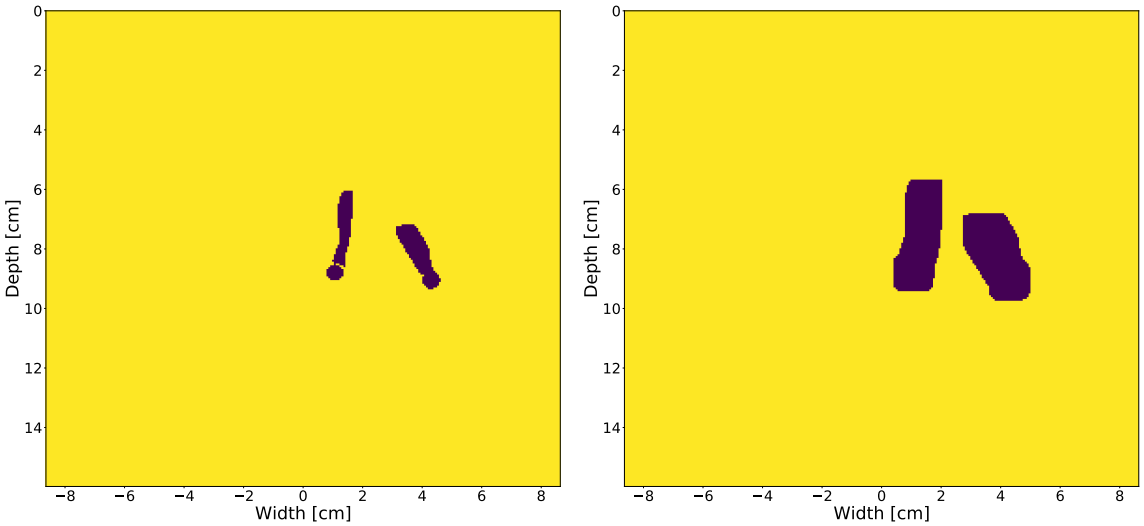
The loss function used during training is the cross-entropy described in section 2.3.1. Keras has three different variations of the cross-entropy function, namely: binary, categorical, and sparse categorical. They differ with regard to the format of the input data they require. Binary cross entropy requires one-hot encoded data where the masks have boolean values, True or False. Categorical cross-entropy also uses one-hot encoded masks where the values are integers, often 0 and 1. Sparse categorical cross-entropy, which assigns each pixel one class using integers, requires one number for each class, thus having only one layer compared to binary and categorical, which uses one layer for each class. As a result of this, the sparse variation takes up less space in exchange for less output information. For analytical purposes, it is beneficial to have all the possible information at hand. Therefore, we experiment with binary and categorical cross-entropy. However, the sparse categorical cross-entropy could be explored in the future when more domain knowledge is available. To optimize the loss functions, we use the ADAM optimizer with the standard hyperparameters set by Keras. These are a learning rate, $\eta$, of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\eta = 1e - 07$ [31]. We choose the following pairings of loss functions and activation functions for the final output layer: binary cross-entropy loss with sigmoid activation function and categorical cross-entropy with the softmax activation function.

The B-mode images have values with the data type uint-8, which means that the values in the images are in the interval $[0, 255]$. This could pose a problem during training because of the exploding gradient problem. To combat this, we normalize the data in the images resulting in values in the interval $[0, 1]$ by dividing each pixel value in the image by the maximum pixel value in each image. This also results in a change in datatype, from uint-8 to float-64.

Making the networks converge can sometimes be challenging, especially if the ground truths used are a small portion of the whole image. This is the case for both the U-Net OV-B and U-Net OV-C networks, which only use the mitral valve apparatus and background during training and therefore struggle to converge.

In the case of the U-Net OV-B network, the values of the ground truth masks need to be binary, True, or False. Thus, the only modification we can do to the segmentation is to change the size. We do not want to change the size of the individual valve annotations since they are made "perfect" by the clinicians, which leaves the background class. We, therefore, choose to enlarge the valve annotations before inverting when we create the background class. This enlargement is performed by combining the Gaussian kernel described in section 2.3.3 with boolean data, which outputs an enlarged version of the input masks. An example of the original background and the enlarged version, using a standard deviation $\sigma = 2$, is shown in figure 22.

With regards to the U-Net OV-C network, we have more freedom to change the input values. Using segmentation channels where 0 indicates false and 1 indicates




(a) Original background (b) Enlarged background

Figure 22: Example showing (a) the original background class and (b) resulting enlarged background class using $\sigma = 2$ for the Gaussian kernel.

true, the network does not converge. The U-Net OV-C network uses channels with the data type uint-8 with the range of $[0, 255]$. By changing the channel values to 1 to indicate false and 255 to indicate true, the network converges.

Training of the U-Net Auto and U-Net Auto-R do not have the same problem with convergence. Feeding the network more information helps to make convergence easier for both binary and categorical cross-entropy losses. To limit the scope of the project, we choose to focus only on the categorical cross-entropy loss for these models because it offers more flexibility concerning segmentation modifications.

As mentioned in section 3.5.1 some of the auto-generated ground truth segmentations are not ideal. Of the 828 samples in the whole data set, this is the case for 88 samples. To explore their impact on the valve segmentation a subset of the data set is created without these 88 samples, referred to as the cleaned data set. This leaves a total of 740 samples, 574 for training and validation, and 166 for testing.

When the networks have been trained, the test set images are run through the networks. The output channels of the network give each pixel in each channel a value in the range $[0, 1]$. This value represents the confidence that said pixel in the B-mode image belongs to the class of the channel. These predictions are then compared to the ground truth by calculation the DICE score, Manhattan distance for the annulus points, and estimation of the angle between the annulus plane and each leaflet. To calculate these values, one needs to set a threshold value used with the channels. Meaning that all values in a channel below the threshold are classified as not belonging to the channel class, and all values above are considered as a part of the channel class. The threshold should be chosen to give the highest possible score.

The Manhattan distance output the distance between the two center points of the annulus point ground truth and segmentation in terms of pixels, which is not the optimal unit to use. We therefore use the metadata in the DICOM files to get the physical measurements (max length and width) of each recording and convert the distance to centimeters.

### 3.7.3 Sequence test

Each image sample in the data set is taken from a DICOM-file containing one or more complete cardiac cycles. The files usually contain at least 100 frames. A tiny fraction of these has annotations we can use to evaluate the performance of the model quantitatively. Since the finished model is supposed to do segmentation for the whole cycle, it would be beneficial to explore the model's performance for all the frames. The results of this would need a qualitative analysis. The 162 images in the test set are taken from 40 different DICOM-files. For the sequence test, we extract the whole sequence of each file and run them through the network. We then do a visual inspection of the resulting segmentations.

## 3.8 Post-processing

A common problem with this kind of segmentation task is the occurrence of artifacts. The inclusion of the background class will reduce the number of artifacts but not eliminate them. We, therefore, need to run the raw output from the networks through a post-processing pipeline. A standard post-processing method is to fill in any holes in the segmentations and only keep the larges region of each segmented class.

# 4 Results

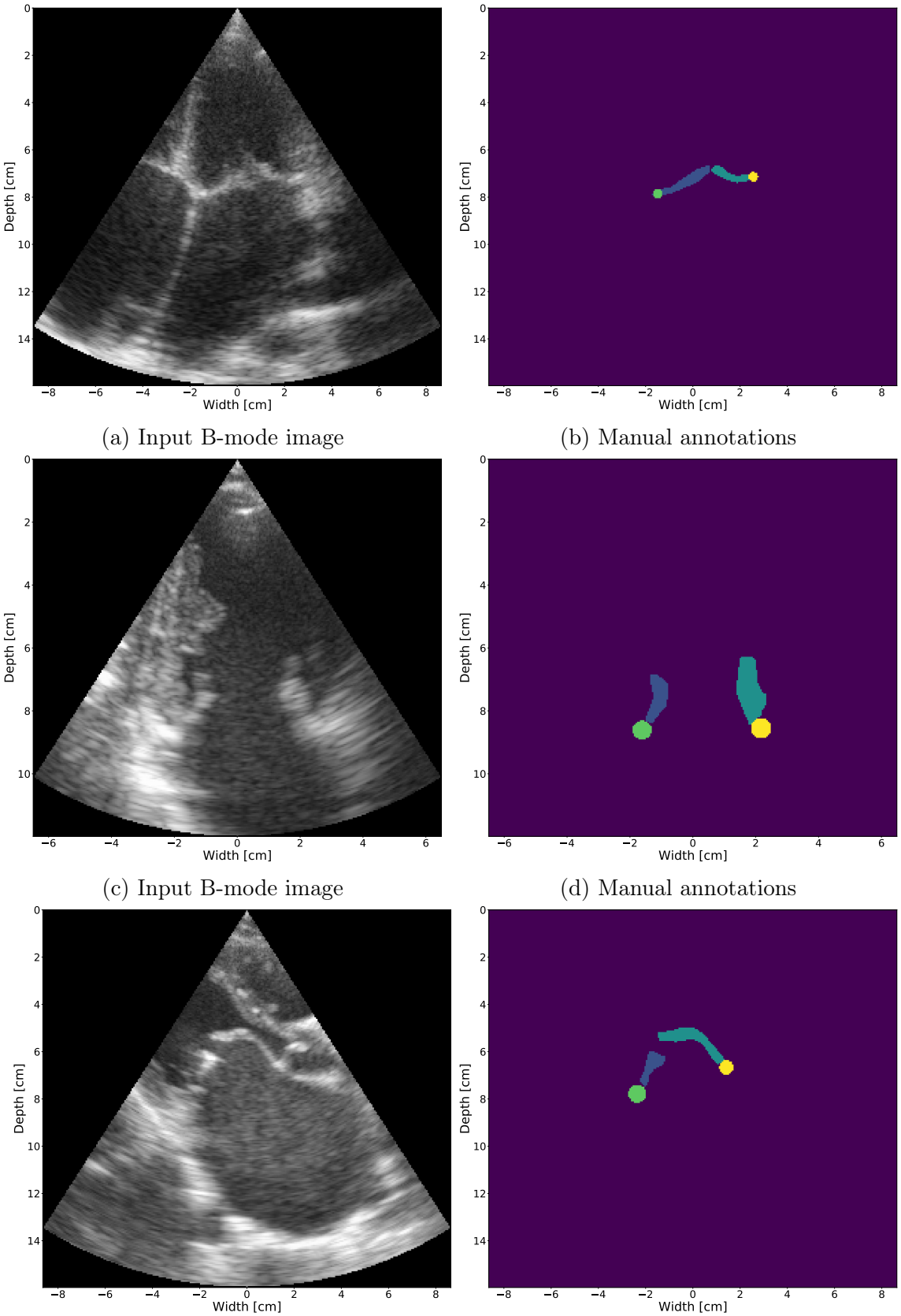
## 4.1 Data Set

### 4.1.1 Manual Annotations

Figure 23 shows three samples from the data set with their manual annotations created by a trained clinician. The samples are taken from three different DICOM-files.

### 4.1.2 Pre-processing

Figure 24 shows three B-mode images taken from the same DICOM-file and the auto-generated ground truth segmentations of the left ventricle, myocardium, and left atrium (after pre-processing). All the B-mode images are relatively similar, but the SmistadLVLA and SmistadLA networks fail to produce good segmentations for one of the examples. The lousy example contains no segmentation of the left ventricle and a subpar segmentation of the left atrium.




(a) Input B-mode image

(b) Manual annotations

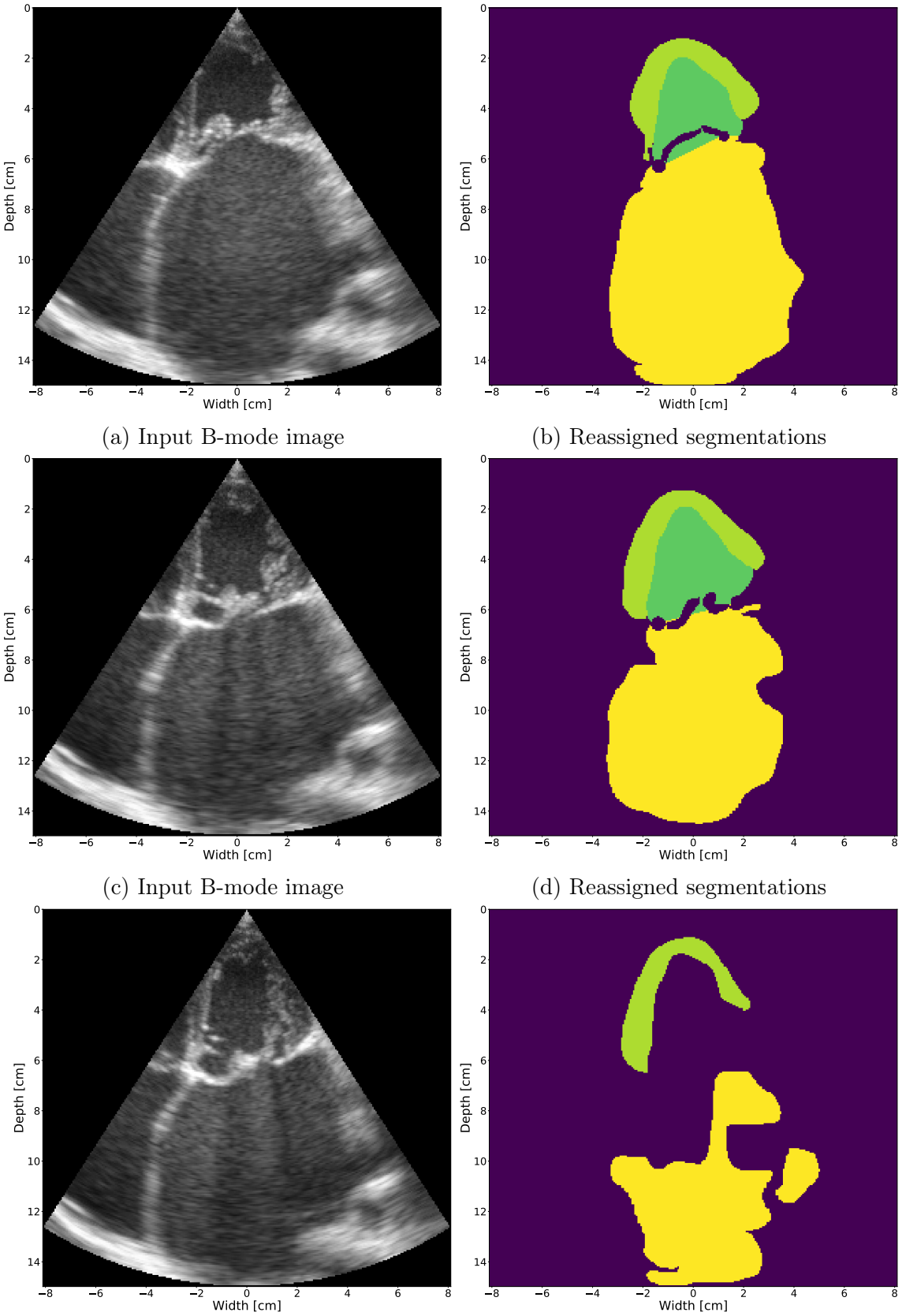
(c) Input B-mode image

(d) Manual annotations

(e) Input B-mode image

(f) Manual annotations

Figure 23: Three frames taken from three different DICOM-files and their manual annotations of the posterior leaflet (blue), posterior annulus (green), anterior leaflet (teal), and anterior annulus (yellow) produced by a trained clinician.




(a) Input B-mode image

(b) Reassigned segmentations

(c) Input B-mode image

(d) Reassigned segmentations

(e) Input B-mode image

(f) Reassigned segmentations

Figure 24: Three frames taken from the same DICOM-file and their reassigned auto-generated segmentations for the left ventricle, myocardium, and left atrium. The segmentations performed on the input images (a) and (c) are reasonable, but the segmentations on the input image (e) are subpar.

## 4.2 Only Valve Apparatus

This section contains the produced results using only the valve apparatus for training and testing the U-Net model. The data set contains 824 images, where 662 are used for training and 162 are used for testing. 10% of the training set is used for validation during training. The 162 images in the test set are taken from 40 DICOM-files.

### 4.2.1 Input Configuration

Table 2 shows the DICE scores from the U-Net OV-B and U-Net OV-C networks. The only difference in the training of these two networks is the difference in the loss function, output layer activation function, and input values of the ground truth masks. The raw output of the networks has been post-processed as described in section 3.8. Figures 25 - 26 show example with index 51 from the test set and the produced segmentations from the U-Net OV-B and U-Net OV-C networks, respectively. From table 2 it is apparent that the U-Net OV-C outperforms the U-Net OV-B. The U-Net OV-C network is therefore chosen for further investigation and improvement.

### 4.2.2 Augmentation Impact

Augmentation is performed before the training starts and is performed on every image in the training set. The samples in the test set are not augmented. To analyze the impact of each augmentation type, the model has been trained with each type separately. The augmentation adds 662 images for each augmentation applied to the data set. Thus, each run using only one method is trained and validated using 1324 images, and the variation using all augmentation methods is trained using 3972 images. The resulting DICE scores for each run are shown in table 3. The model using all the augmentations has the overall best performance and is therefore used henceforth when referring to the U-Net OV-C model. Figures 27 - 29 show three example segmentations and output channels produced by the U-Net OV-C network. Figure 30 shows predictions performed on the test sample with index 51 in the test set.

### 4.2.3 Feature Extraction Performance

Table 4 shows a summary of the difference in distance between the predicted center points for the annulus points and the ground truth center points. Figures 31 - 33 show three examples for the center point annulus predictions, one good, one average, and one subpar performance, respectively. Figure 34 shows an example where the difference is huge. Table 5 shows a summary of the difference of the estimated angles between the leaflets and the annulus plane for the predicted segmentations and the ground truth. Figures 35 - 37 show three examples for the estimated angles, one good, one average, and one subpar performance, respectively.

### 4.2.4 Sequence Test

Using the U-Net OV-C, we run the test set DICOM-files through the network and visually inspect the output for the whole sequence. Figure 38 shows a selection of frames from DICOM-file number 5. The sequence shown starts at index 26 with a closed valve and opens up as time moves. It reaches maximum opening at approximately index 32 and then starts to close again. This selection highlights a problem

observed in many of the test files. The model struggles to segment out the valve when the leaflets are open. The frame with index 34 from the same DICOM-file is shown in figure 39 to more clearly illustrate the problem.

Another problem is that the network sometimes connects the leaflets when the valve is open. This is illustrated in figure 40, which shows a selection of frames from DICOM-file number 12 from the test set. The first frame (index 26) of the selection shows a closed valve. The valve reaches maximum opening at approximately index 32 and then starts to close.

Some of the test DICOM-files yield reasonable results from the network. Figure 42 shows a selection of frames from DICOM-file 7 where the segmentations from the network are excellent.

Table 2: List of the DICE scores relating to the valve apparatus for the U-Net OV-B and U-Net OV-C networks. The highest score between the two models for each measurement of each class are highlighted.

| Class | U-Net OV-B | | | | U-Net OV-C | | | |
|---|---|---|---|---|---|---|---|---|
| | Min | Max | Mean | Median | Min | Max | Mean | Median |
| PMVL | 0.000 | 0.888 | 0.565 | 0.640 | 0.000 | **0.898** | **0.601** | **0.686** |
| AMVL | 0.000 | **0.906** | 0.615 | 0.644 | 0.000 | 0.898 | **0.646** | **0.685** |
| PA | 0.000 | 0.667 | 0.166 | 0.119 | 0.000 | **0.889** | **0.325** | **0.310** |
| AA | 0.000 | **0.871** | 0.309 | 0.310 | 0.000 | **0.871** | **0.362** | **0.391** |

Table 3: List of augmentation methods and their resulting DICE scores with the U-Net OV-C network. The highest score in each column is highlighted.

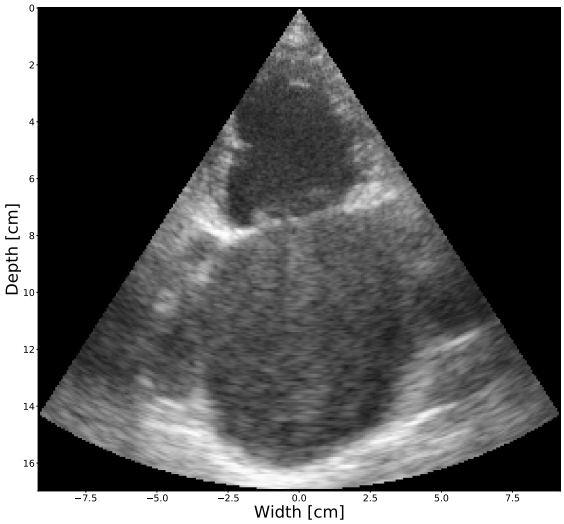
| Augmentation | PMVL | | AMVL | | PA | | AA | |
|---|---|---|---|---|---|---|---|---|
| | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| No aug | 0.601 | 0.686 | 0.646 | 0.685 | 0.325 | 0.310 | 0.362 | 0.391 |
| Cropping (0.75) | 0.571 | 0.683 | 0.602 | 0.661 | 0.393 | 0.412 | 0.387 | 0.466 |
| Gaus (0.5) | 0.595 | 0.667 | 0.629 | 0.673 | **0.440** | **0.518** | 0.397 | 0.393 |
| Gamma (0.7) | 0.586 | 0.656 | 0.636 | 0.690 | 0.404 | 0.421 | 0.419 | 0.438 |
| Gamma (1.3) | **0.610** | 0.683 | 0.615 | 0.662 | 0.411 | 0.426 | 0.457 | 0.490 |
| Rotation (10 °) | 0.590 | 0.685 | 0.632 | 0.696 | 0.347 | 0.375 | **0.497** | **0.569** |
| Rotation (-5 °) | 0.597 | 0.665 | **0.659** | **0.698** | 0.396 | 0.448 | 0.432 | 0.468 |
| All | 0.607 | **0.691** | 0.641 | 0.696 | 0.395 | 0.423 | 0.477 | 0.548 |

Table 4: A summary of the Manhattan distance metric calculated using the annulus points predictions performed by the U-Net OV-C network and the ground truth segmentations. The distances are given in centimeters.

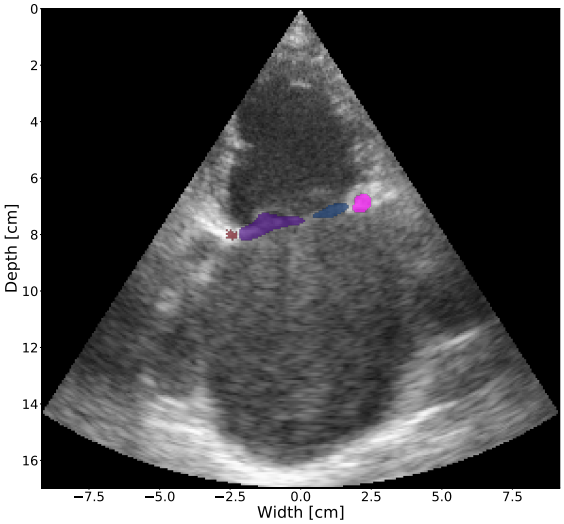
| **Class** | **Min** | **Max** | **Mean** | **Median** |
|---|---|---|---|---|
| Posterior annulus | 0.020 | 1.750 | 0.436 | 0.339 |
| Anterior annulus | 0.019 | 6.841 | 0.416 | 0.273 |

Table 5: A summary of the difference between the estimated angles using the predictions performed by the U-Net OV-C network and the ground truth segmentations. The angles are given in degrees.

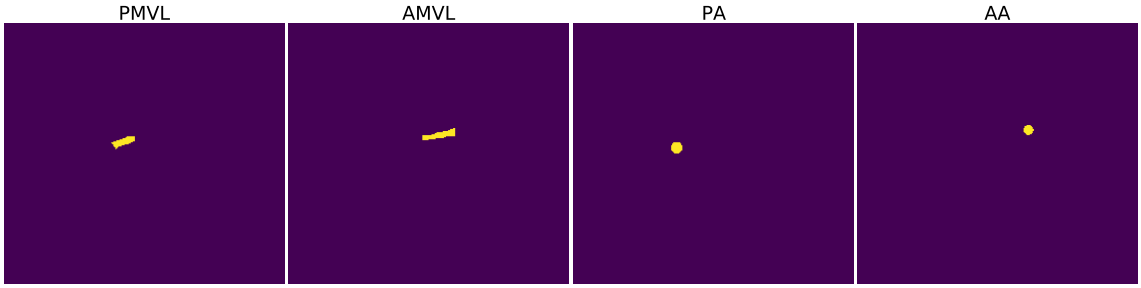
| **Class** | **Min** | **Max** | **Mean** | **Median** |
|---|---|---|---|---|
| Posterior leaflet | 0.050 | 88.24 | 15.12 | 6.73 |
| Anterior leaflet | 0.11 | 105.0054 | 20.42 | 10.58 |



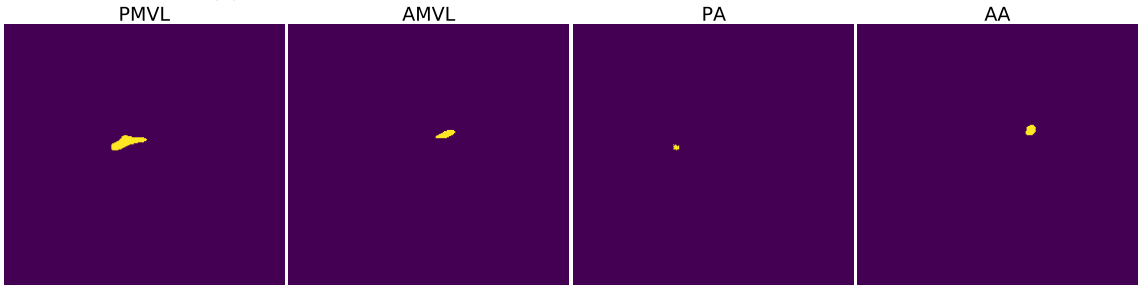

(a) Input B-mode image



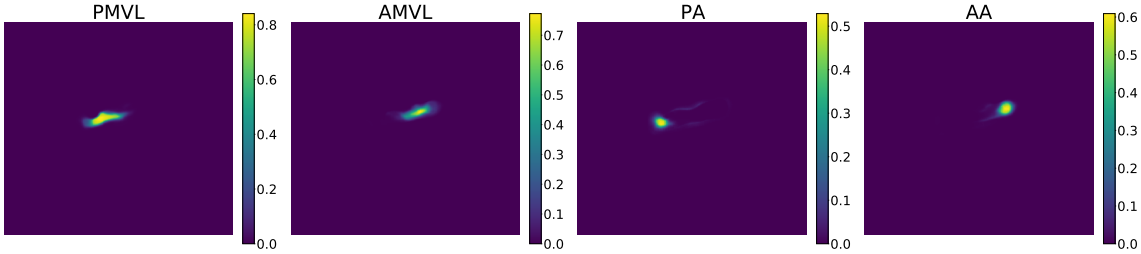

(b) Predicted Segmentations




(c) One-hot encoded ground truth channel segmentations

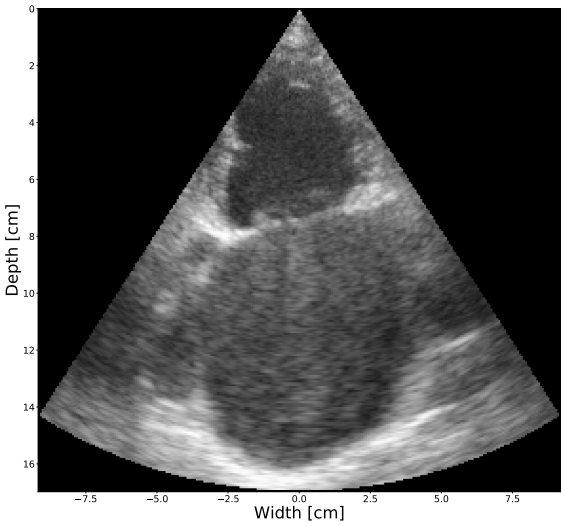



(d) One-hot encoded predicted channel segmentations

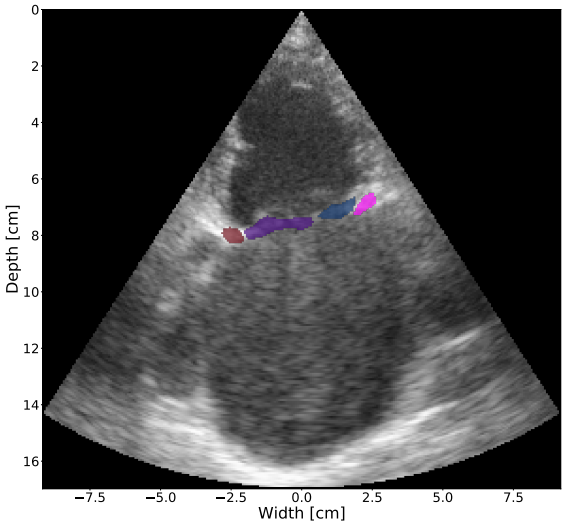



(e) Raw output channel segmentations

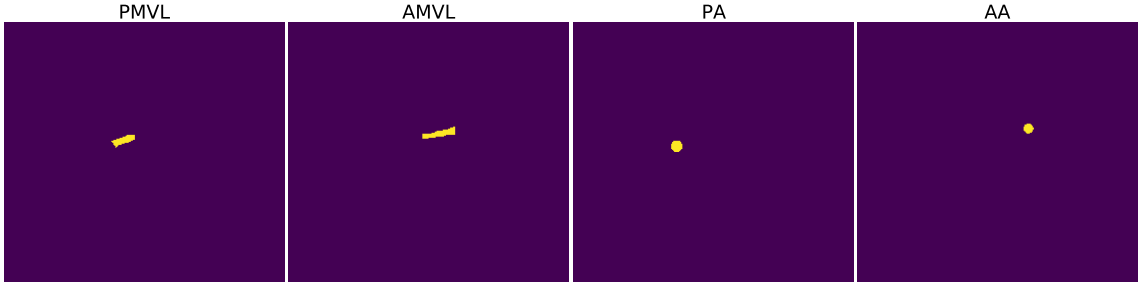
Figure 25: Test sample, index 51, produced by the U-Net OV-B network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), and AA (pink). (c) and (d) show the one-hot encoded channels of the ground truth and post-processed predicted segmentations, respectively. The raw output from the network is shown in (e).



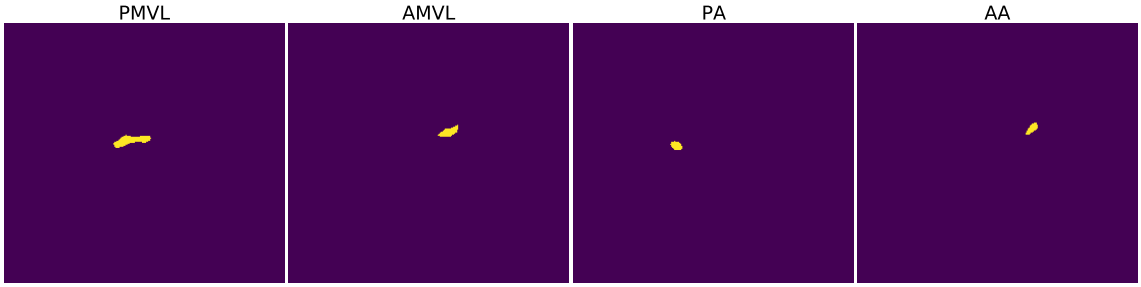

(a) Input B-mode image



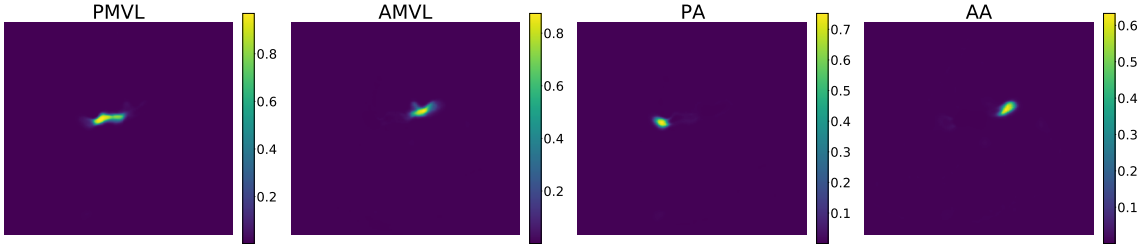

(b) Predicted Segmentations




(c) One-hot encoded ground truth channel segmentations

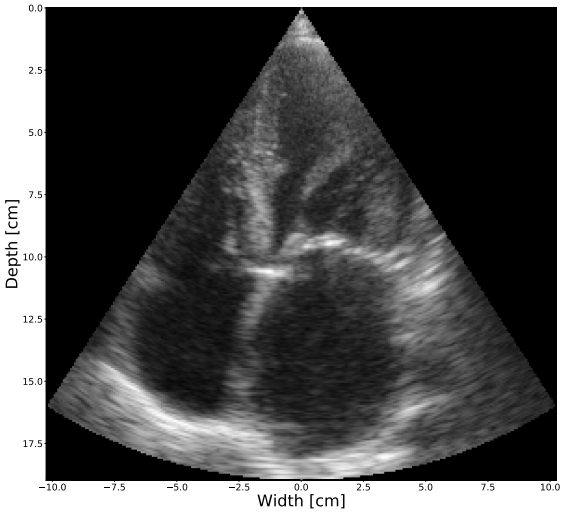



(d) One-hot encoded predicted channel segmentations

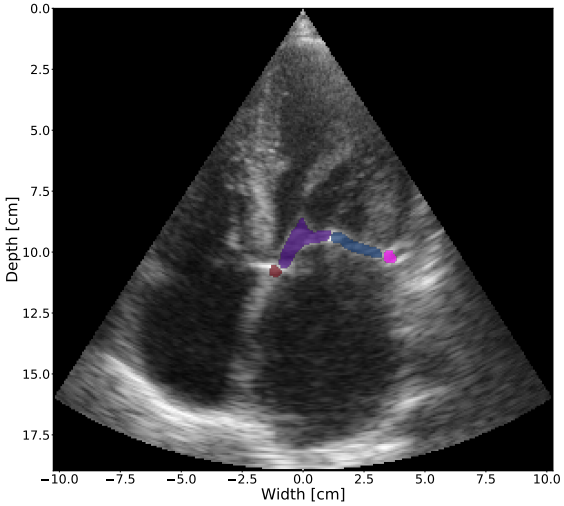



(e) Raw output channel segmentations

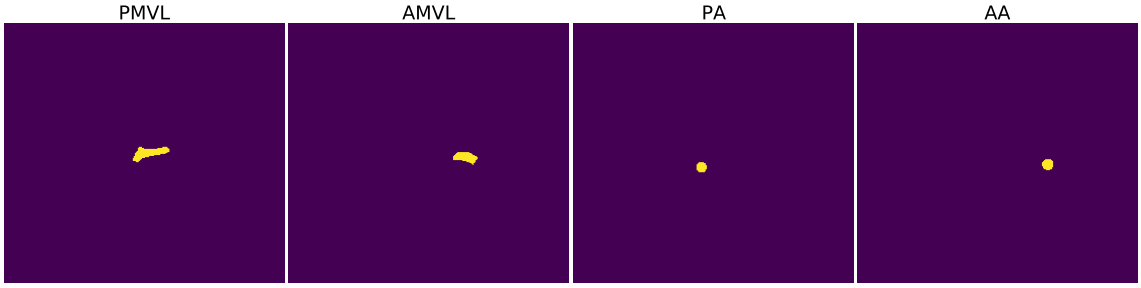
Figure 26: Test sample, index 51, produced by the U-Net OV-C network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), and AA (pink). (c) and (d) show the one-hot encoded channels of the ground truth and post-processed predicted segmentations, respectively. The raw output from the network is shown in (e).



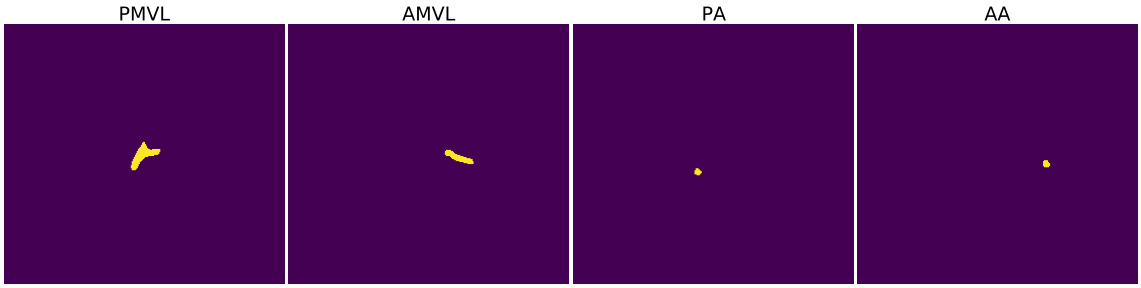

(a) Input B-mode image



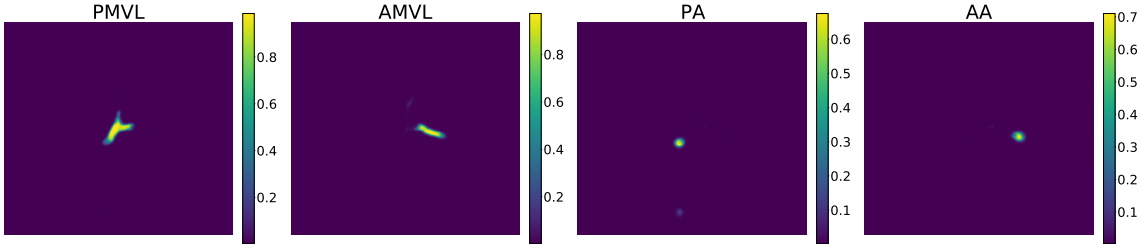

(b) Predicted Segmentations




(c) One-hot encoded ground truth channel segmentations

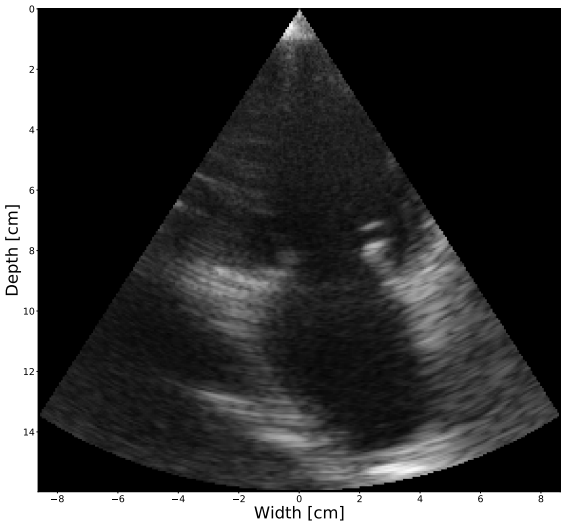



(d) One-hot encoded predicted channel segmentations

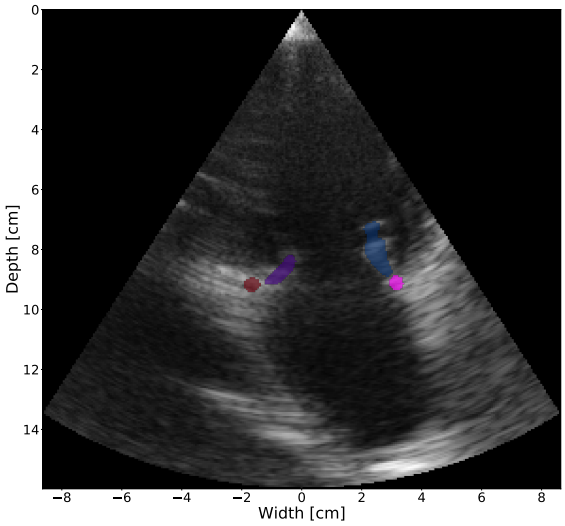



(e) Raw output channel segmentations

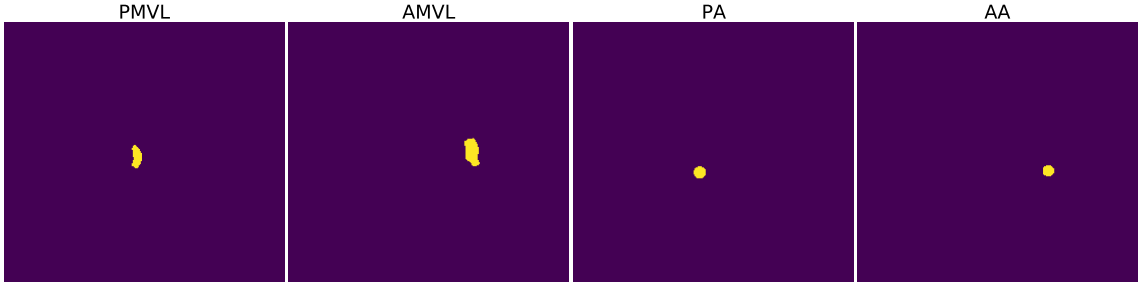
Figure 27: Test sample, index 3, produced by the U-Net OV-C network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), and AA (pink). (c) and (d) show the one-hot encoded channels of the ground truth and post-processed predicted segmentations, respectively. The raw output from the network is shown in (e).



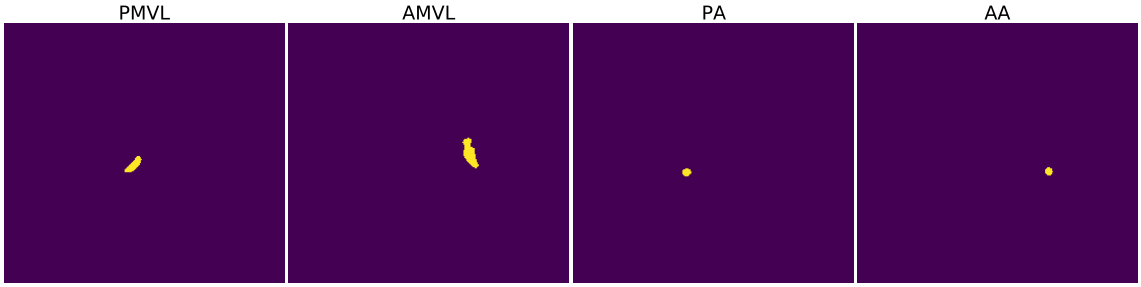

(a) Input B-mode image



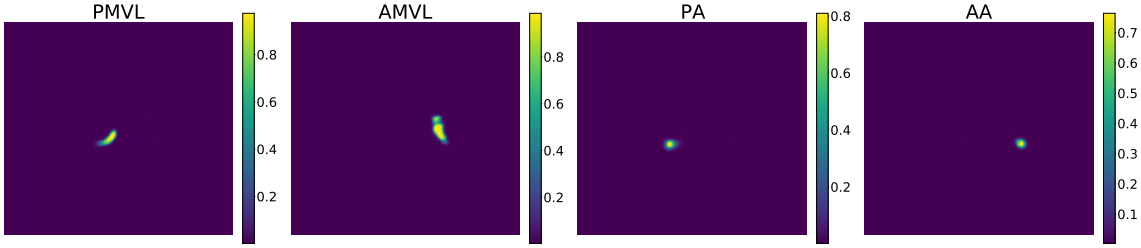

(b) Predicted Segmentations




(c) One-hot encoded ground truth channel segmentations

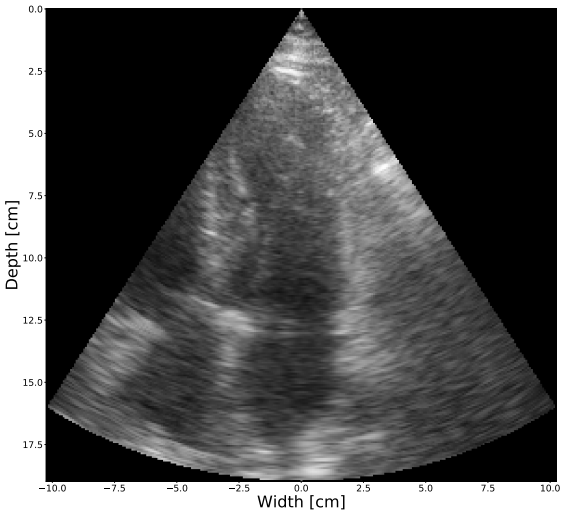



(d) One-hot encoded predicted channel segmentations

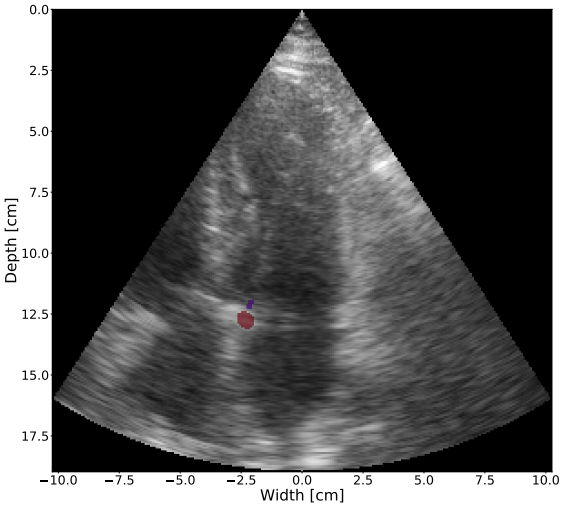



(e) Raw output channel segmentations

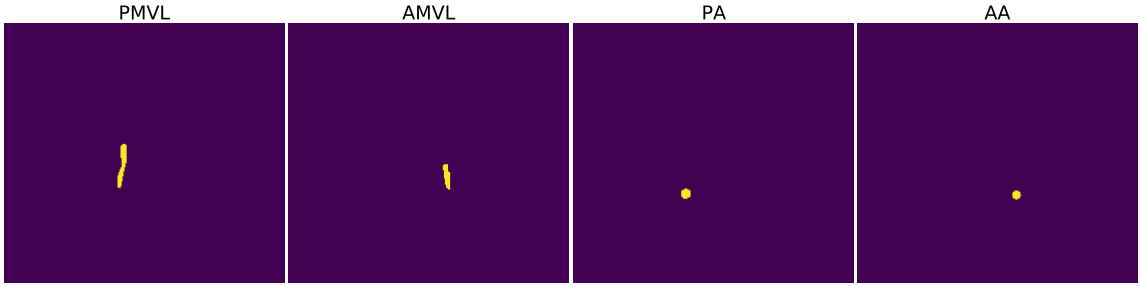
Figure 28: Test sample, index 28, produced by the U-Net OV-C network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), and AA (pink). (c) and (d) show the one-hot encoded channels of the ground truth and post-processed predicted segmentations, respectively. The raw output from the network is shown in (e).



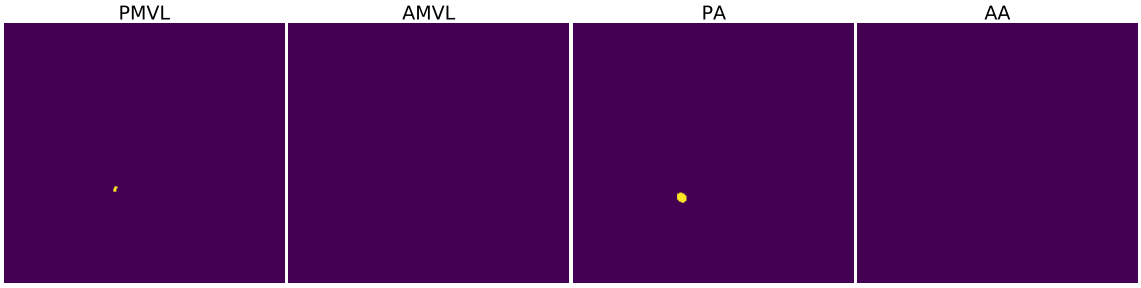

(a) Input B-mode image



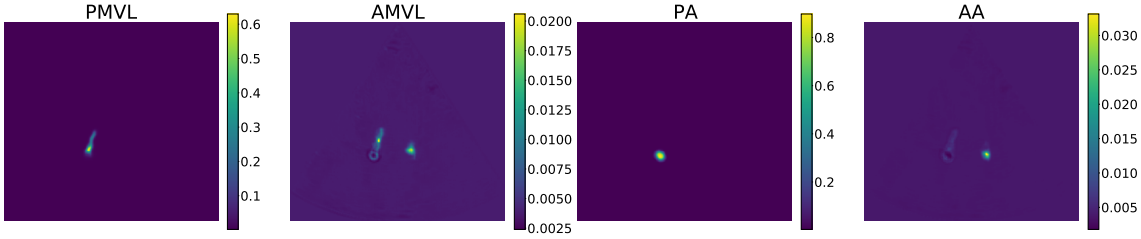

(b) Predicted Segmentations




(c) One-hot encoded ground truth channel segmentations

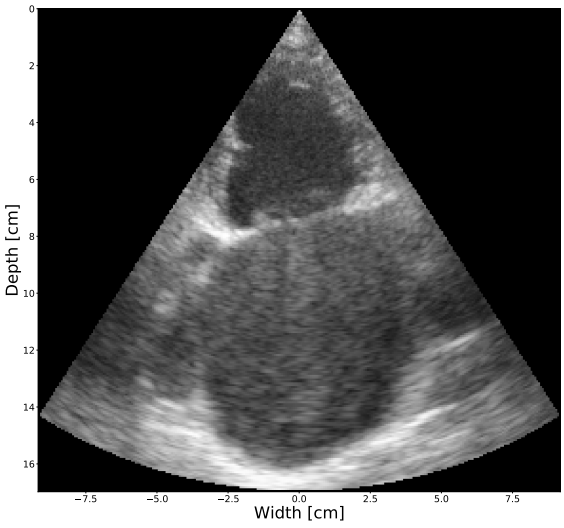



(d) One-hot encoded predicted channel segmentations

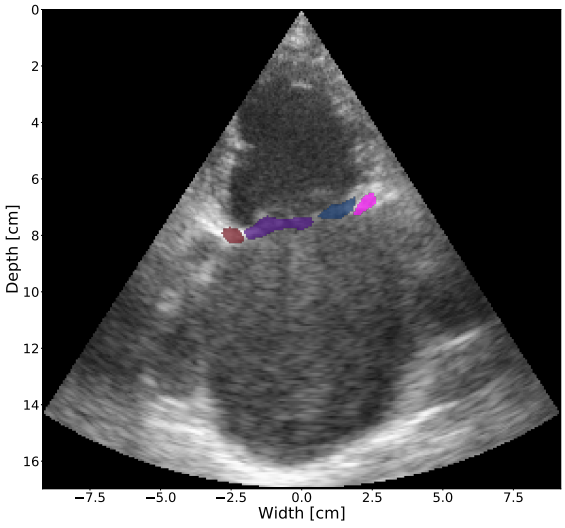



(e) Raw output channel segmentations

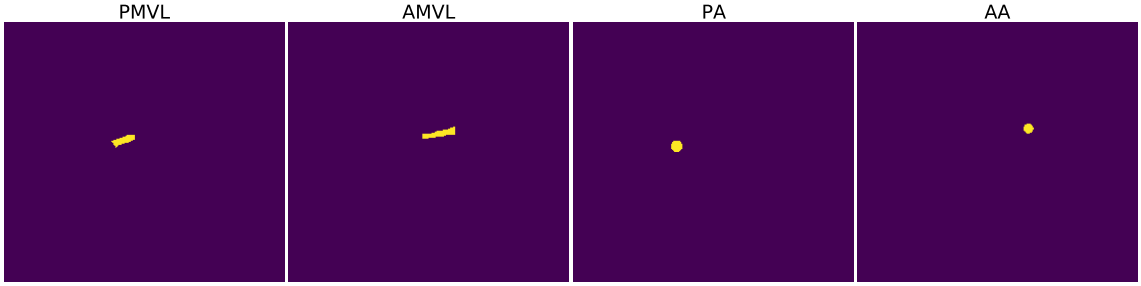
Figure 29: Test sample, index 81, produced by the U-Net OV-C network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), and AA (pink). (c) and (d) show the one-hot encoded channels of the ground truth and post-processed predicted segmentations, respectively. The raw output from the network is shown in (e).



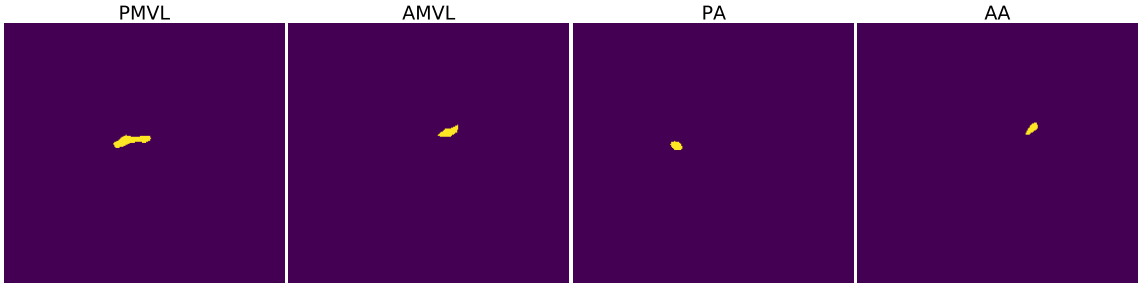

(a) Input B-mode image



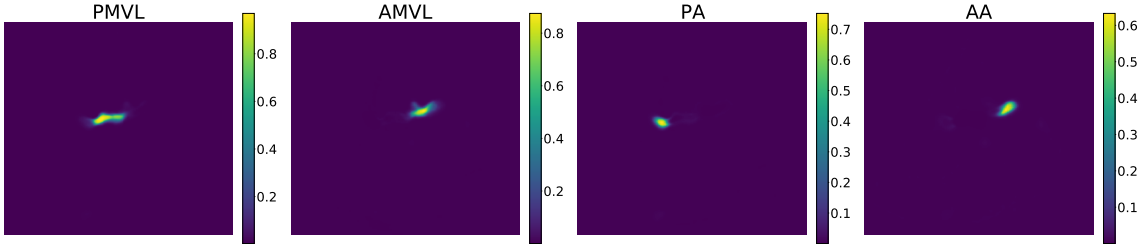

(b) Predicted Segmentations




(c) One-hot encoded ground truth channel segmentations

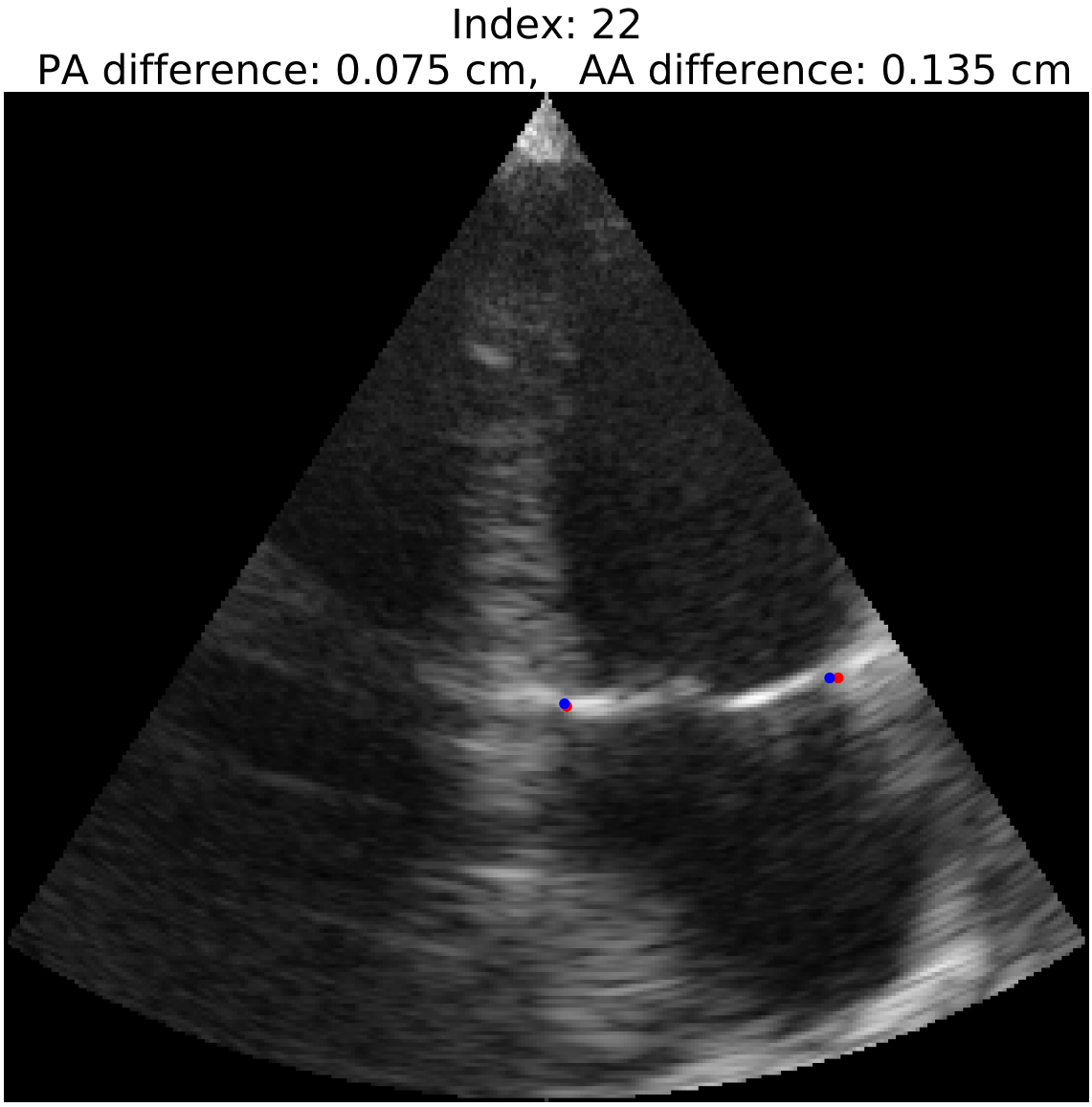



(d) One-hot encoded predicted channel segmentations

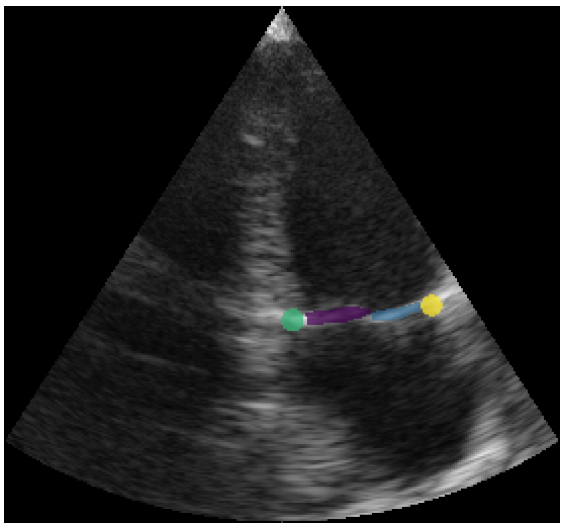



(e) Raw output channel segmentations

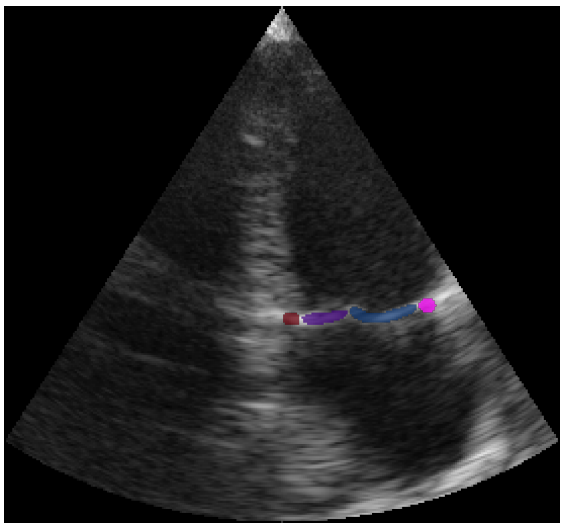
Figure 30: Test sample, index 51, produced by the U-Net OV-C network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), and AA (pink). (c) and (d) show the one-hot encoded channels of the ground truth and post-processed predicted segmentations, respectively. The raw output from the network is shown in (e).




(a) Input B-mode image

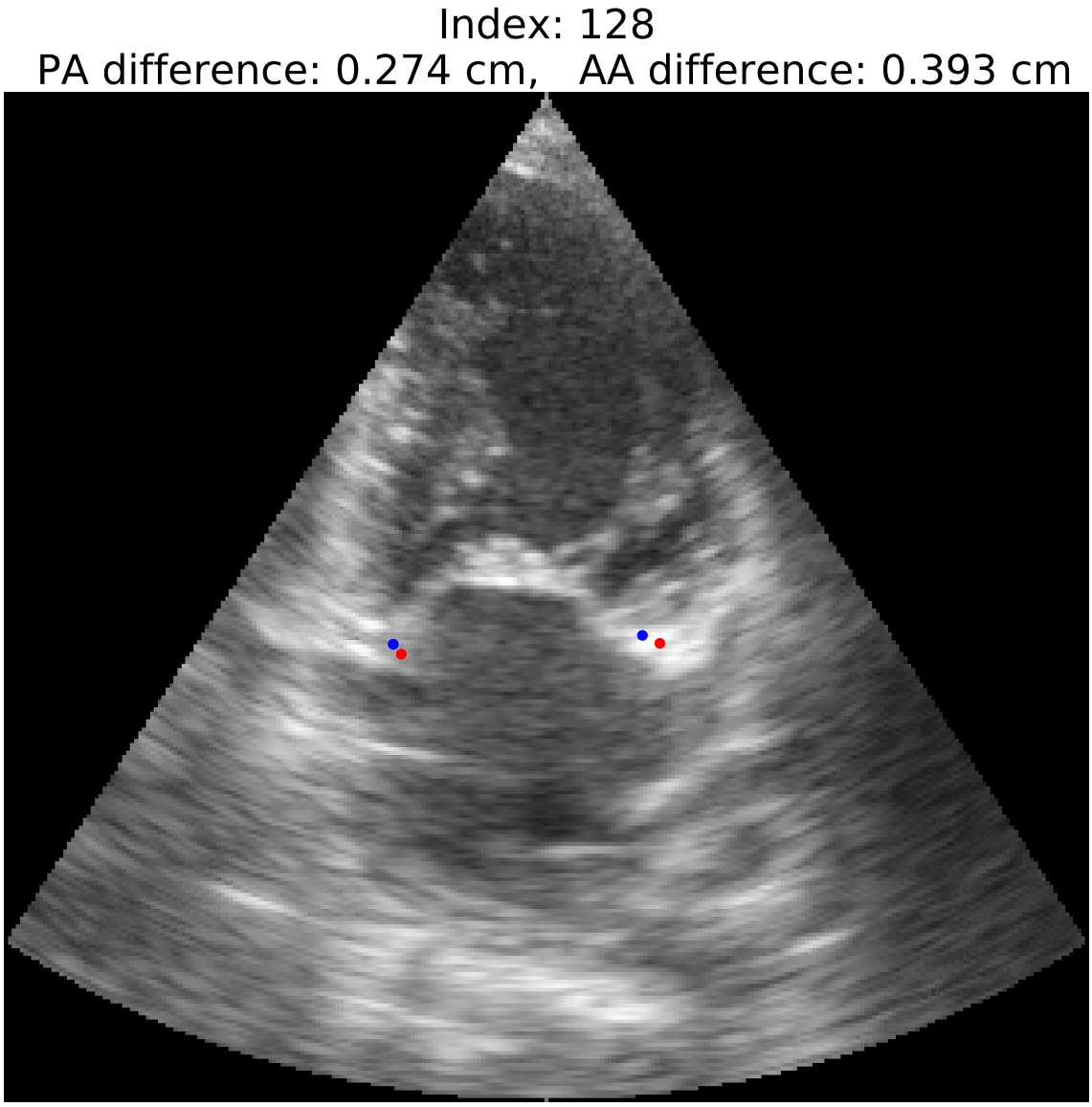


(b) Ground truth

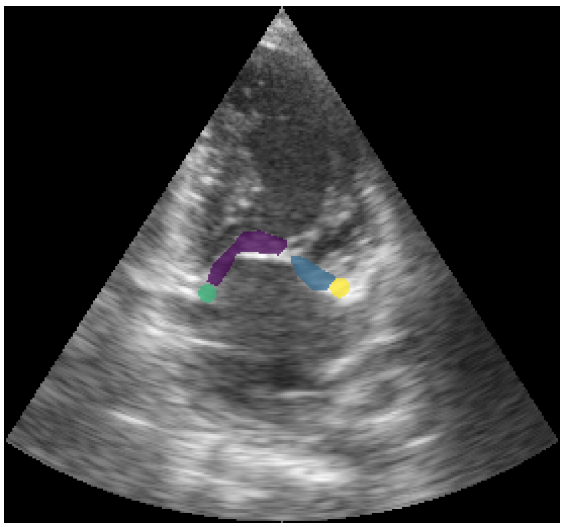


(c) Prediction

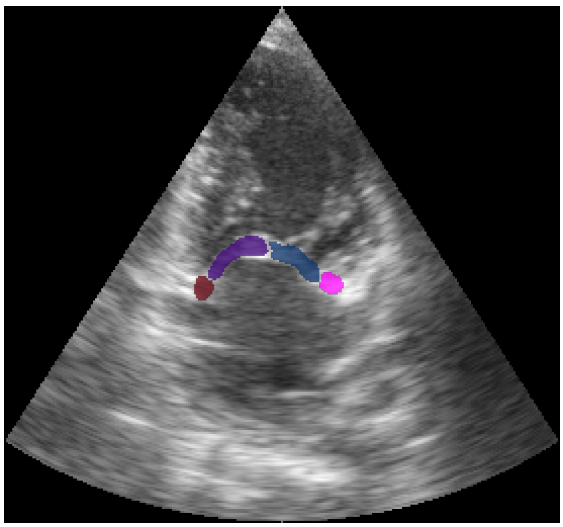
Figure 31: Sample with index 22 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net OV-C model.




(a) Input B-mode image

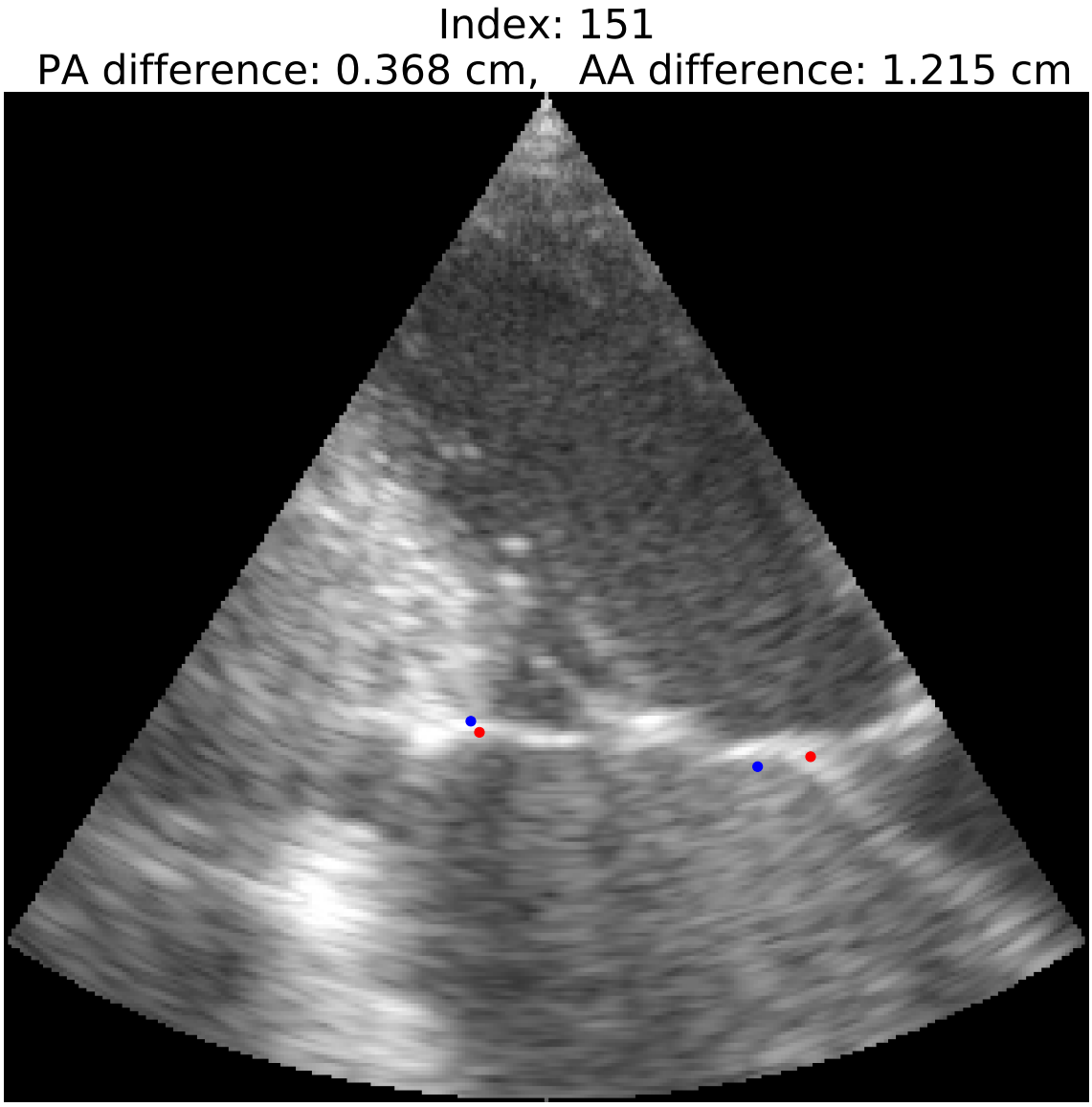


(b) Ground truth

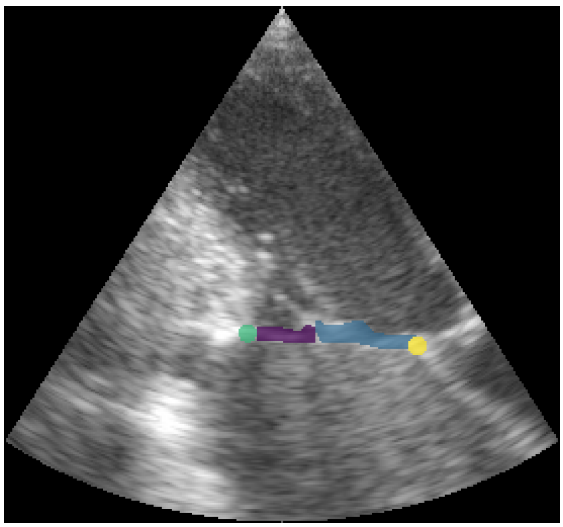


(c) Prediction

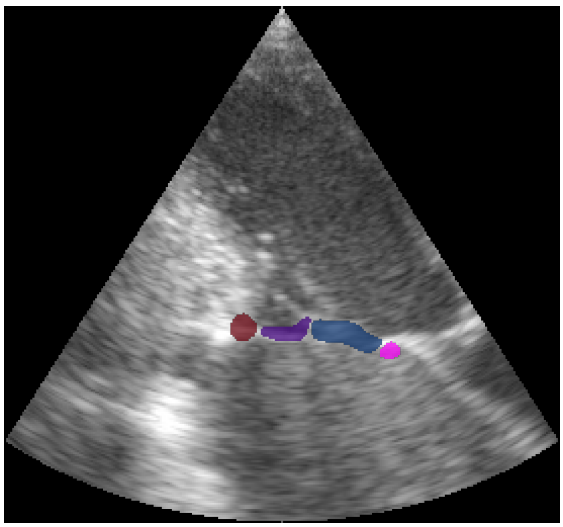
Figure 32: Sample with index 128 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net OV-C model.




(a) Input B-mode image

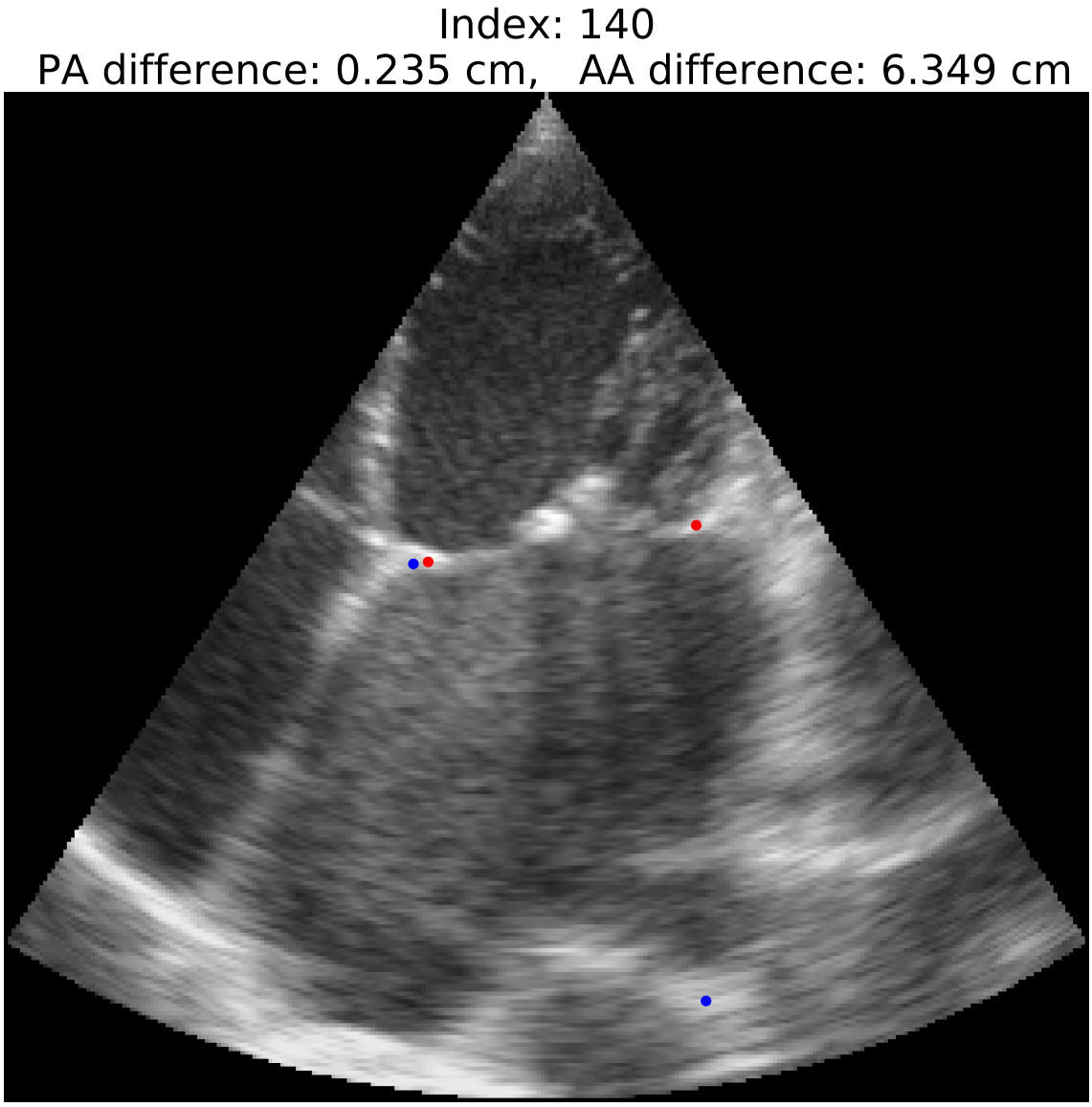


(b) Ground truth

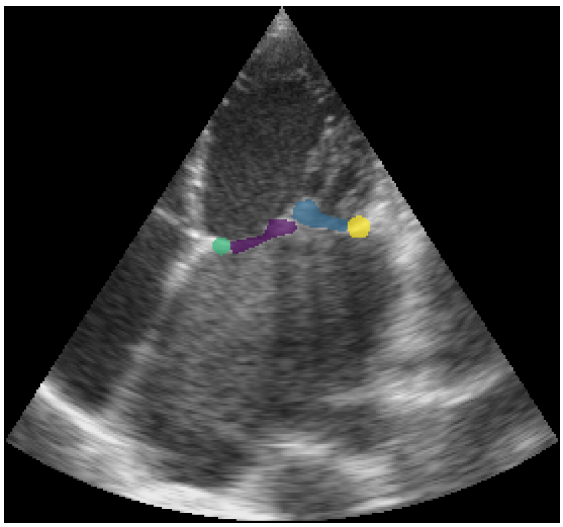


(c) Prediction

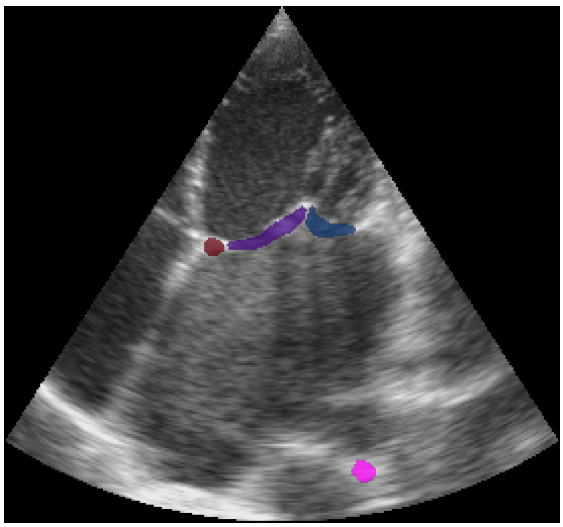
Figure 33: Sample with index 22 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net OV-C model.




(a) Input B-mode image

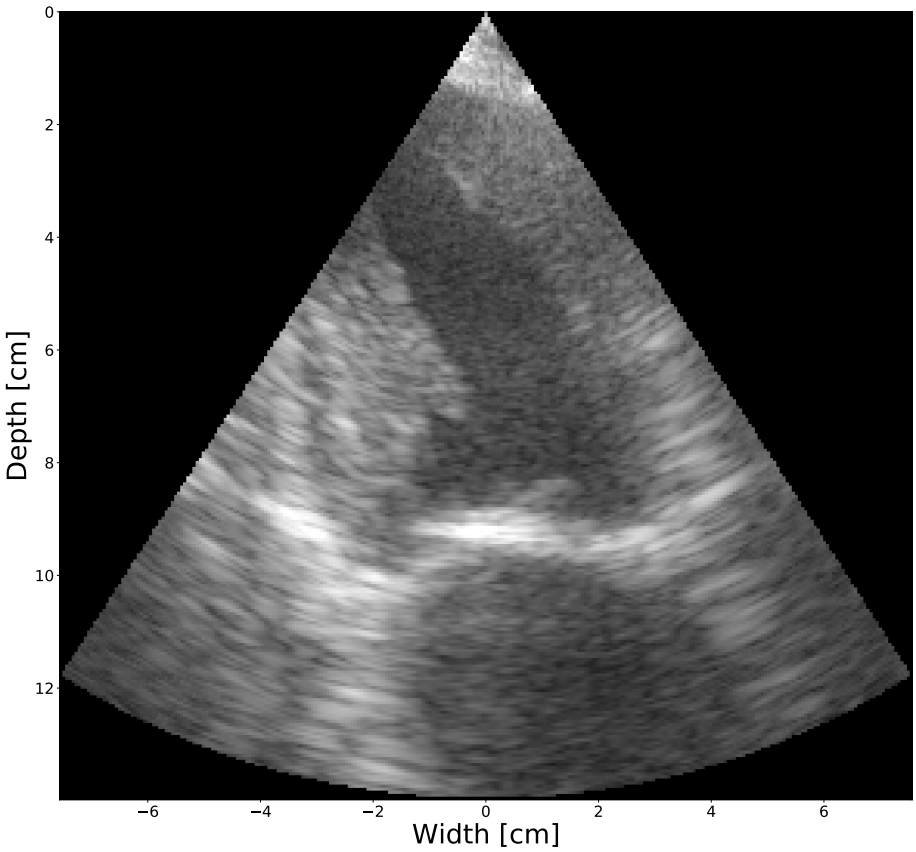


(b) Ground truth

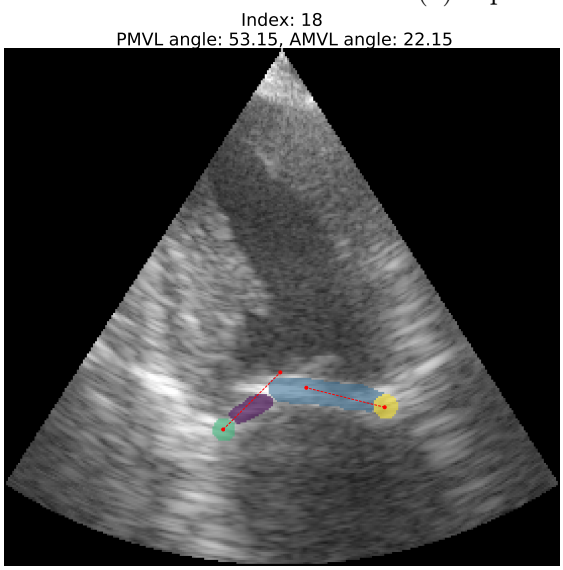


(c) Prediction

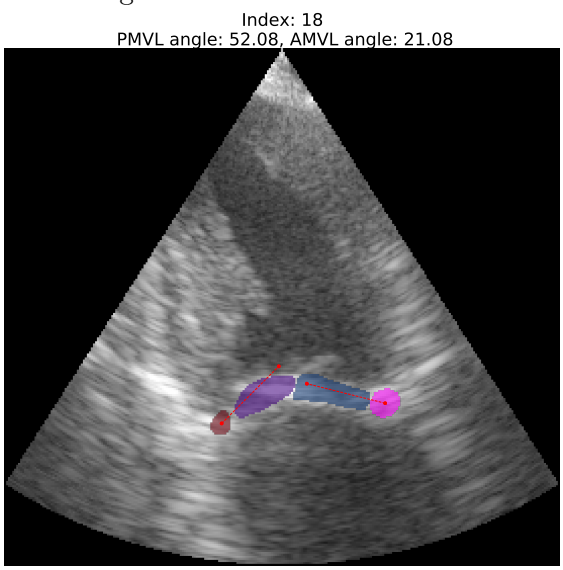
Figure 34: Sample with index 140 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net OV-C model.




(a) Input B-mode image

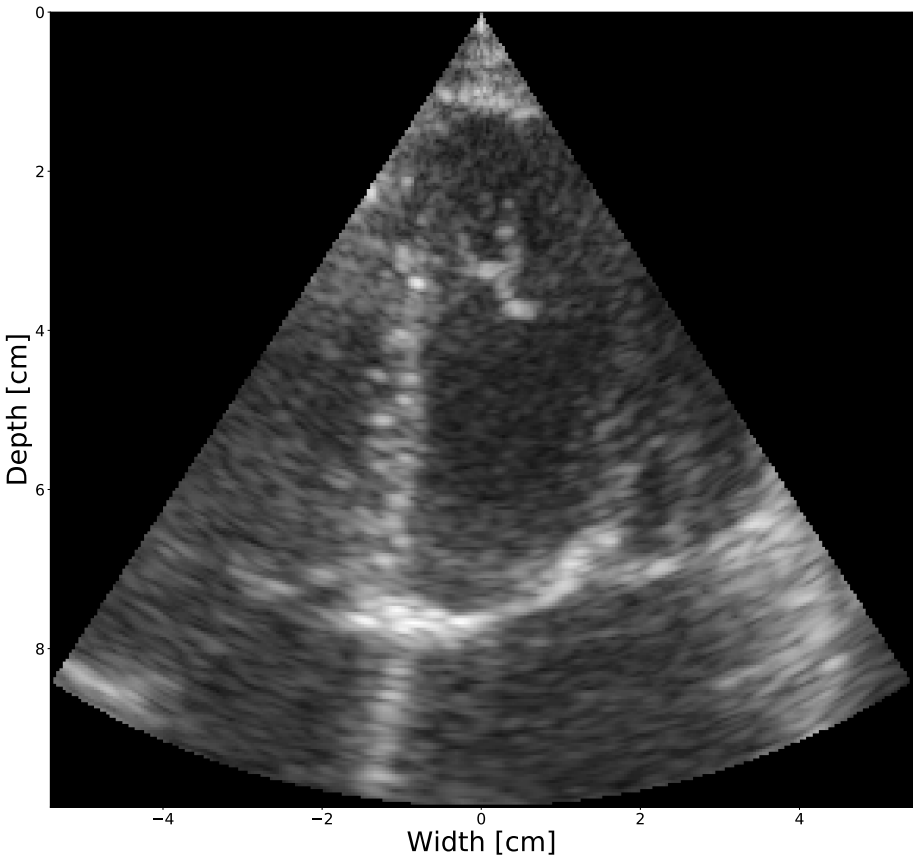



(b) Ground truth

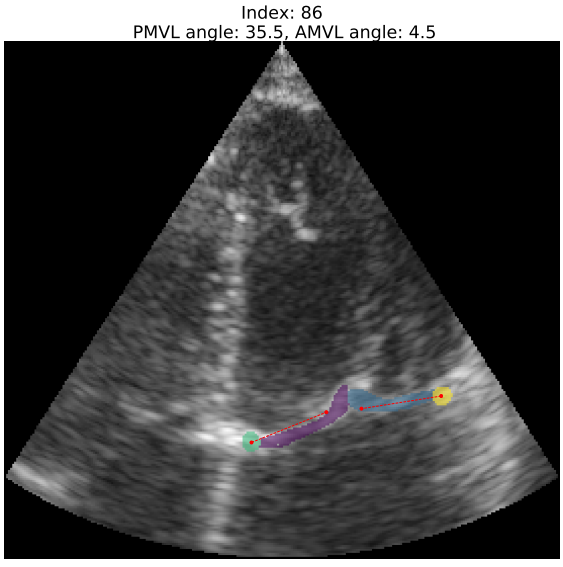



(c) Prediction

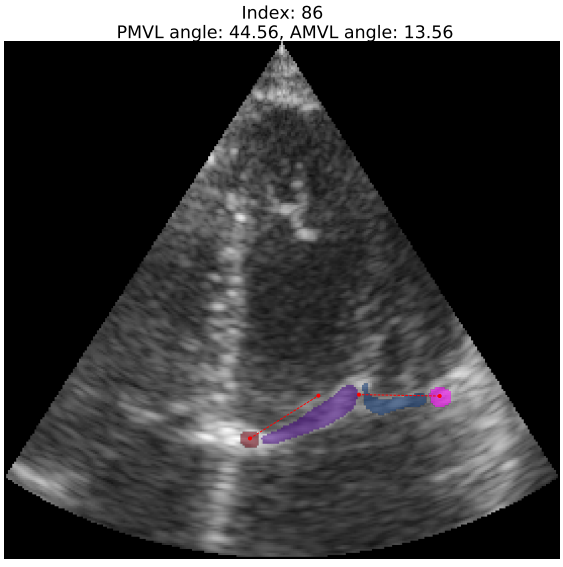
Figure 35: Angle estimation performed on the segmentations produced by the U-Net OV-C network, sample with index 18 from the test set (a), where the estimated angles of the ground truth segmentations (b) and predicted segmentations (c) are similar.




(a) Input B-mode image

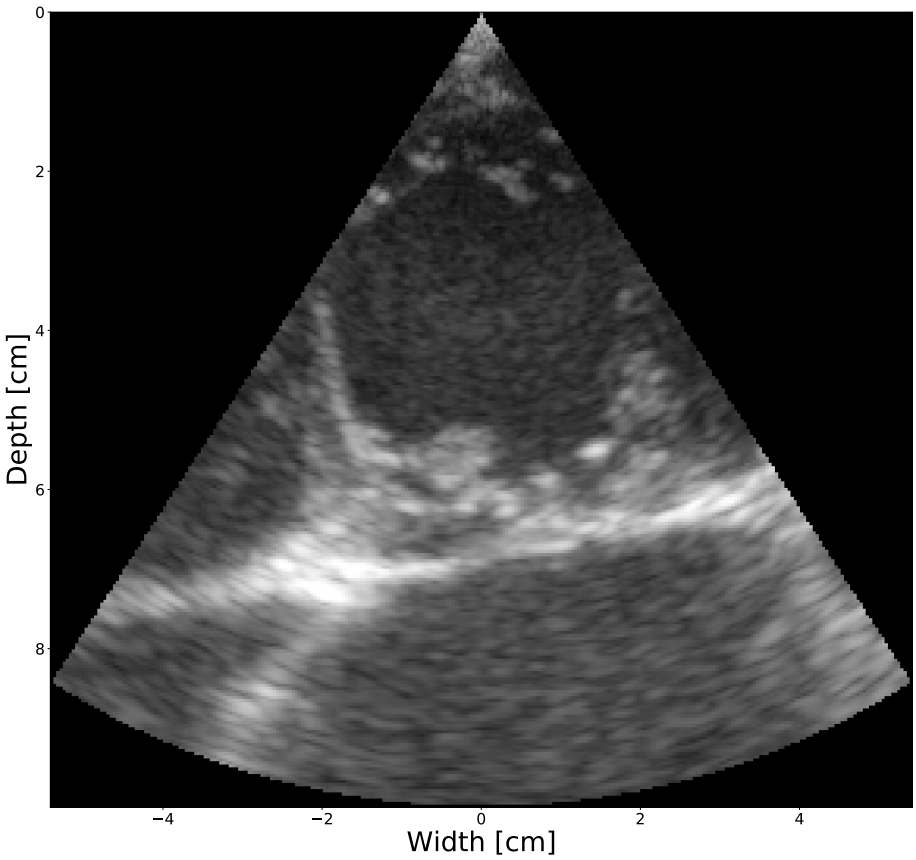



(b) Ground truth

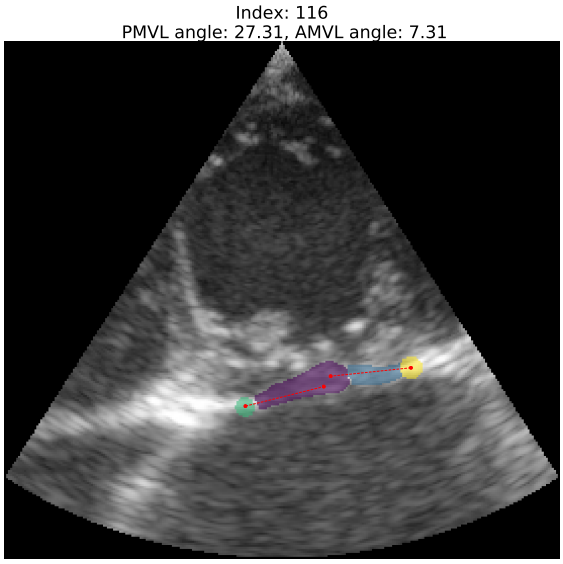



(c) Prediction

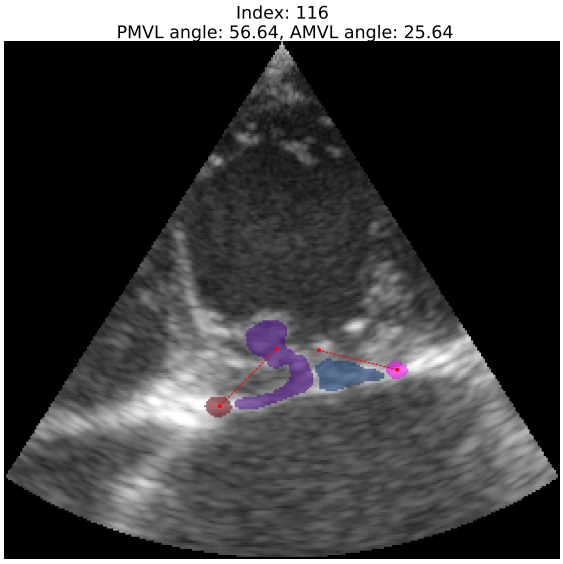
Figure 36: Angle estimation performed on the segmentations produced by the U-Net OV-C network, sample with index 86 from the test set (a) with a median difference between the estimated angles of the ground truth segmentations (b) and predicted segmentations (c).




(a) Input B-mode image




(b) Ground truth




(c) Prediction

Figure 37: Angle estimation performed on the segmentations produced by the U-Net OV-C network, sample with index 116 from the test set (a), where the estimated angles of the ground truth segmentations (b) and predicted segmentations (c) are far from each other.

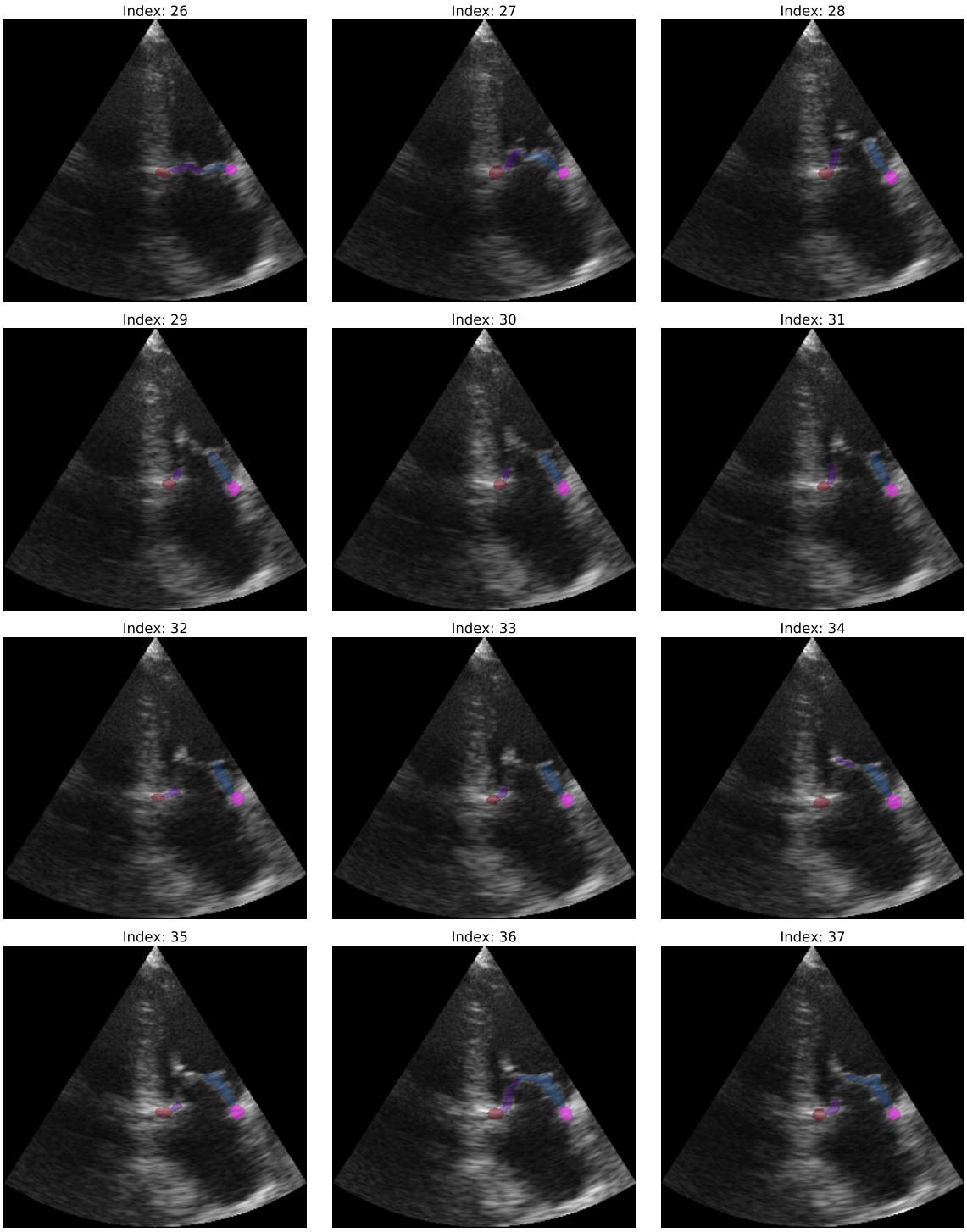
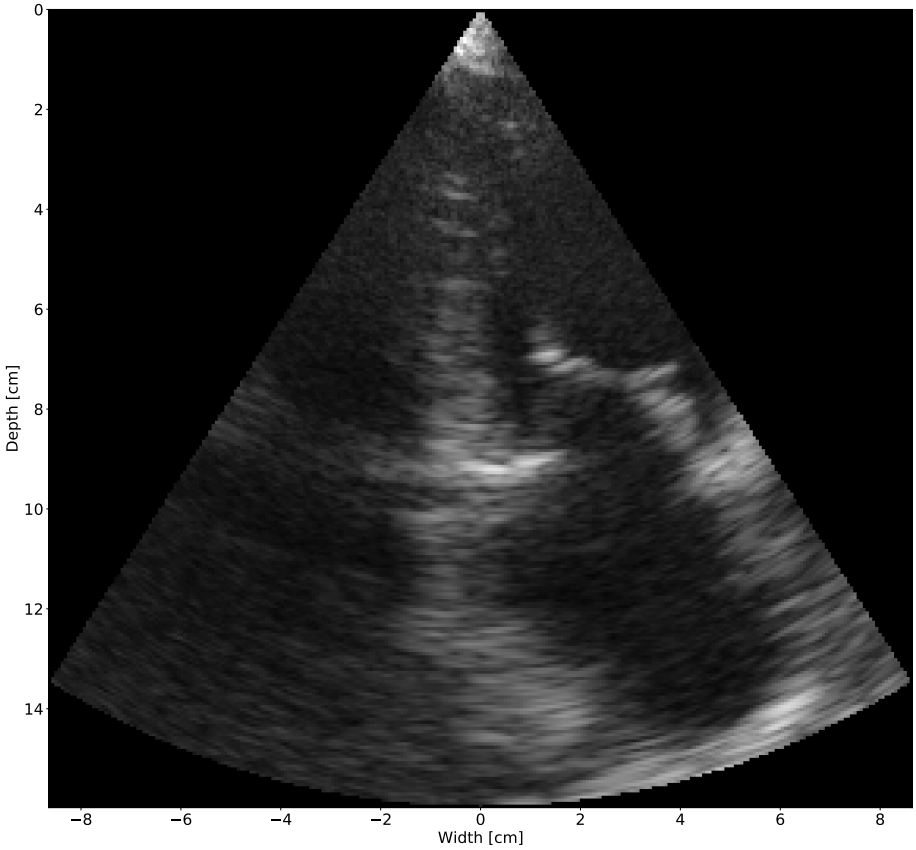
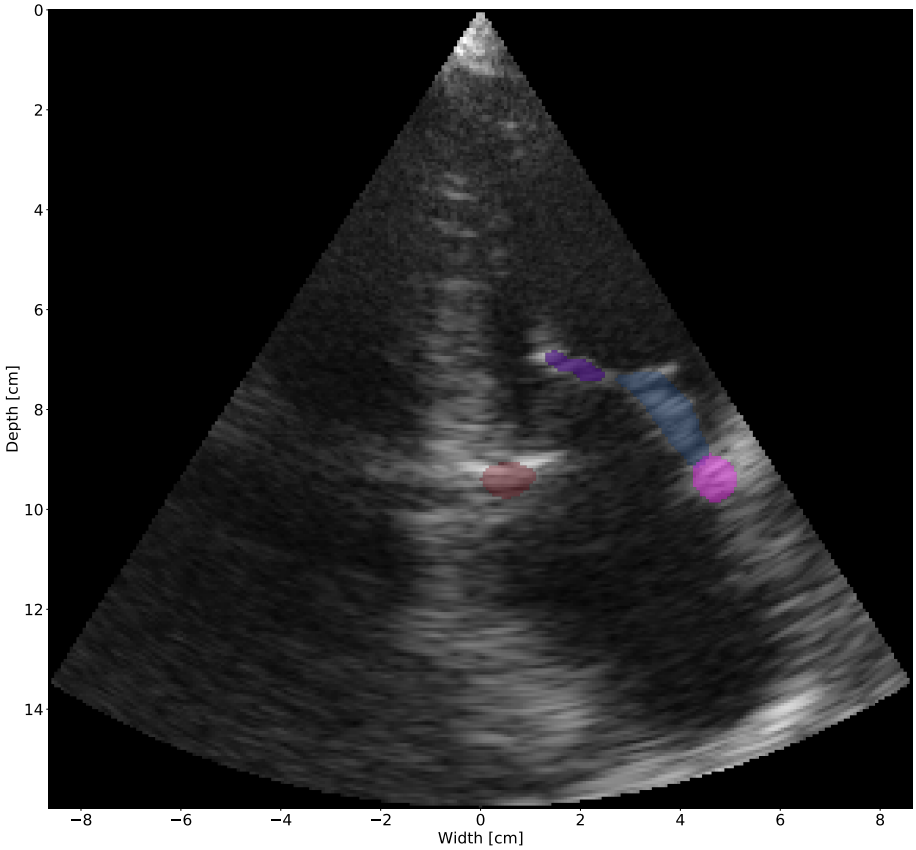



Figure 38: A selection of frames from DICOM-file number 5 in the test set. Each image is a combination of the input B-mode image with the predicted segmentations of the valve apparatus by U-Net OV-C layered on top. In this selection, the valve starts closed (index 26), reaching maximum opening at approximately index 32, and then starts to close.




(a) Input B-mode image, index 34




(b) B-mode image with segmentations layered on top

Figure 39: Outtake from figure 38, frame 34, showing a subpar segmentation by the network U-Net OV-C. The valve is open, and the network only segments out a small part of the posterior leaflet.

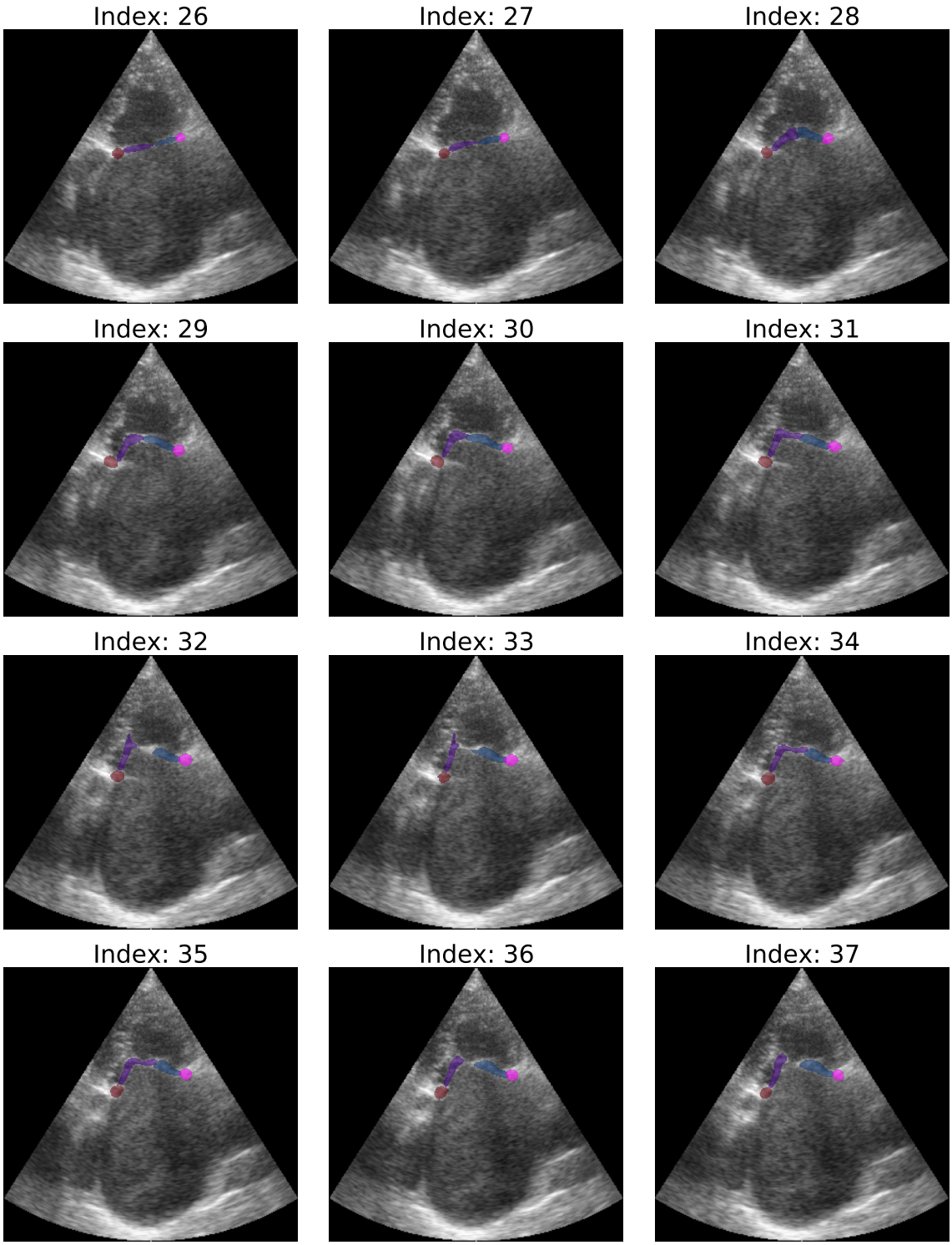
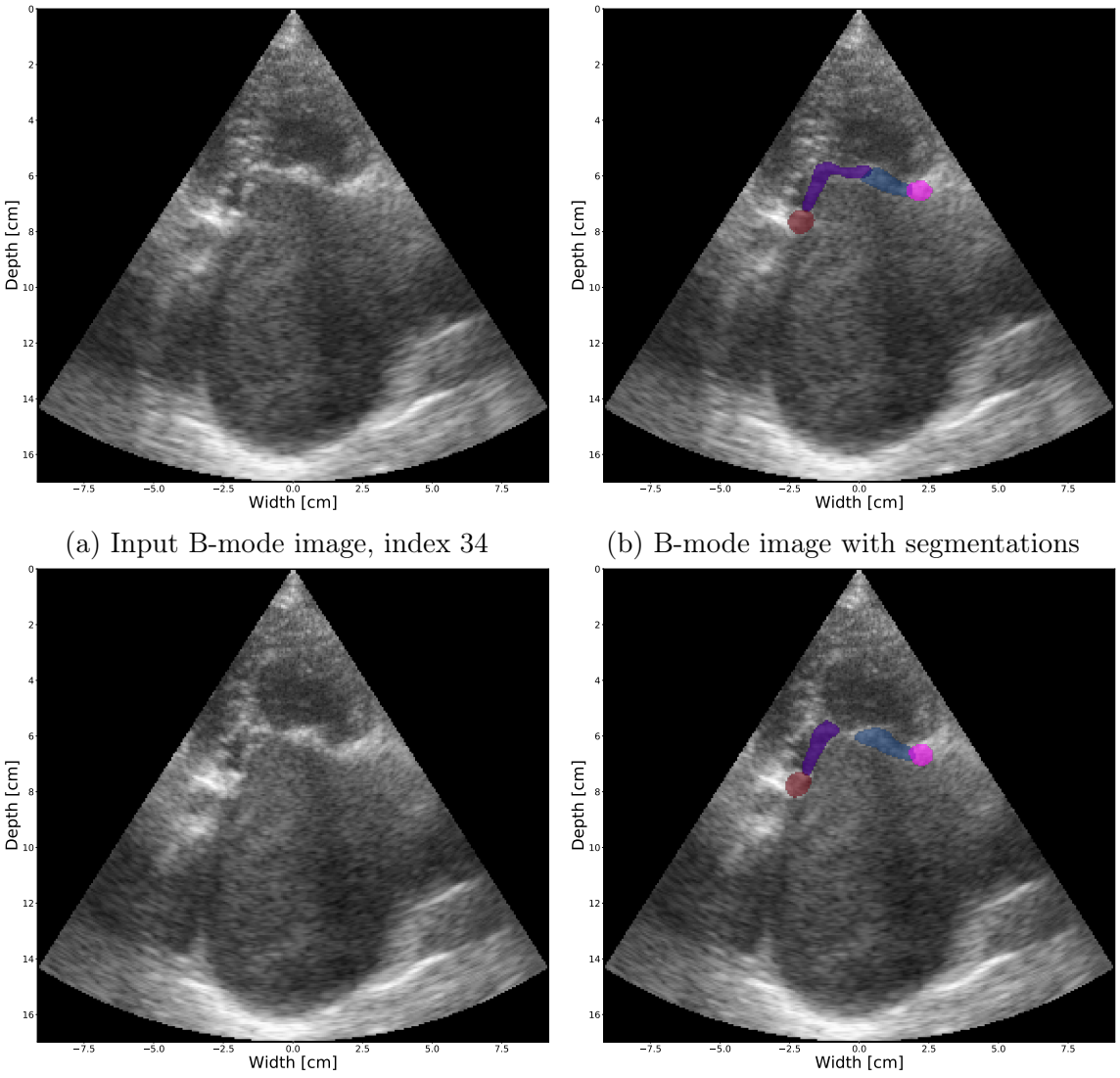



Figure 40: A selection of frames from DICOM-file number 12 in the test set. Each image is a combination of the input B-mode image with the predicted segmentations of the valve apparatus by U-Net OV-C layered on top. In this selection, the valve starts closed (index 26), reaching maximum opening at approximately index 32, and then starts to close.




(a) Input B-mode image, index 34

(b) B-mode image with segmentations

(c) Input B-mode image. index 36

(d) B-mode image with segmentations

Figure 41: Outtake from figure 40, frame 34 and 36, showing a subpar segmentation by the network U-Net OV-C. The valve segmentation of the frame with index 34 connects the two leaflets when they are supposed to be opened. The segmentation of the frame with index 36 shows a reasonable segmentation.

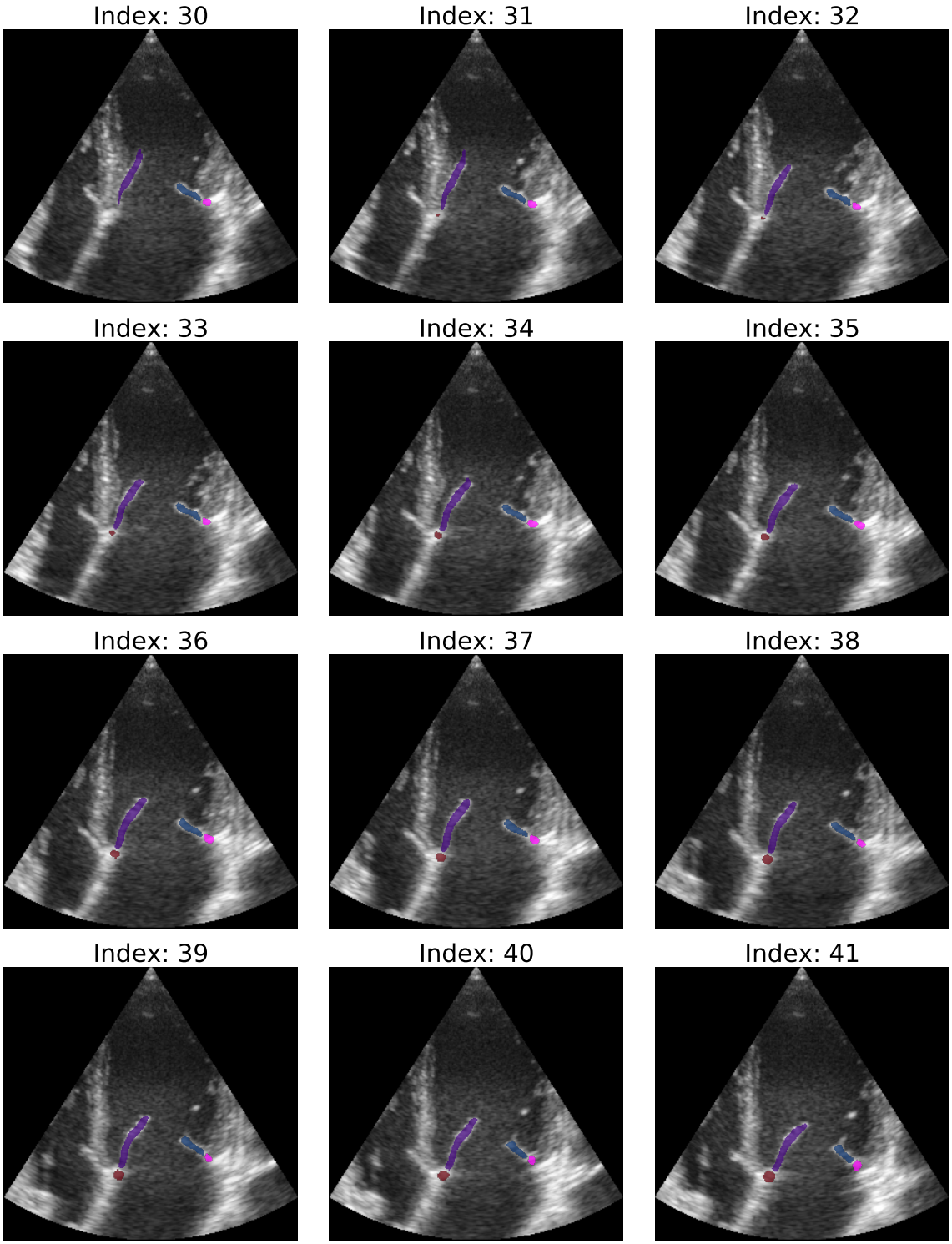



Figure 42: A selection of frames from DICOM-file number 8 in the test set. Each image is a combination of the input B-mode image with the predicted segmentations of the valve apparatus by U-Net OV-C layered on top. In this selection, the valve starts opened (index 30) and starts to close as time moves.

## 4.3 Auto-generated Segmentations

This section contains the results produced by the networks trained with both the valve annotations and the automatically generated segmentations of the left ventricle, left atrium, and myocardium. It uses the same data set split for training and testing as described in section 4.2.

### 4.3.1 Impact of Reassignment

The two models, U-Net Auto and U-Net Auto-R, have been trained with identical hyperparameters. The only difference between them is the inclusion of the reassignment stage in the pre-processing pipeline for the U-Net Auto-R model.

Figure 43 and 44 show predicted segmentations, input ground truth segmentation channels, and post-processed output channels on the same test example by the U-Net Auto and U-Net Auto-R networks, respectively. The input ground truth segmentation and raw output channels for the same test sample are shown in figure 45 from the U-Net Auto, and figure 46 from the U-Net Auto-R. Another example with the same structure is shown in figure 47 - 50.

The examples illustrate that the reassignment is a necessary predicament for the general performance of the model, with regards to the segmentation of the left ventricle, myocardium, and left atrium. The U-Net Auto model is therefore not developed further.

### 4.3.2 Augmentation Impact

The augmentation pipeline for the U-Net Auto-R is identical to the pipeline described in section 4.2.2. The resulting DICE scores for each run are shown in table 3. The model using all the augmentations have the overall best performance and is therefore used henceforth when referring to the U-Net Auto-R model. Figures 51, 53, and 55 show three example segmentations and output channels produced by the U-Net Auto-R network. Figures 52, 54, and 56 show the raw network output from the same three examples, respectively.

### 4.3.3 Cleaned Data Set

To investigate the impact of the subpar automatically generated segmentations on the valve predictions, the U-Net Auto-R model has been trained with the original data set and with the cleaned version. The creation of the cleaned data set is described in section 3.7.2. The DICE scores for the valve apparatus are shown in table 8, and for the auto-generated segmentations in table 9. The input data of each variation have been augmented so that they have approximately the same amount of data samples, around 4000 samples in total for both variations.

### 4.3.4 Feature Extraction Performance

Table 10 shows a summary of metrics for the difference in distance between the predicted center points for the annulus points and the ground truth center points. Figures 57 - 59 show three examples for the center point annulus predictions, one good, one average, and one subpar performance, respectively. Figure 60 shows an example where the difference is huge. Table 11 shows a summary of measurements for the difference between the estimated angles of the leaflets and the annulus plane

for the predicted segmentations and the ground truth. Figures 61 - 63 show three examples for the estimated angles, one good, one average, and one bad performance, respectively.

### 4.3.5 Sequence Test

Using the U-Net Auto-R, we run the test set DICOM-files through the network and visually inspect the output. Figure 64 shows a selection of frames from DICOM-file number 5. The model struggles to segment out the valve when the leaflets are open and sometimes connects the leaflets when the valve is open. Frame 36 from this sequence is shown in figure 65 to highlight the problem.

Figure 66 shows a selection of frames from DICOM-file 35 and illustrates a problem with the post-processing method. During post-processing, the largest region in the channel is chosen as the correct segmentation, and in some cases, the artifact is larger than the correct segmentation. Figure 67 shows frame 39 from the same sequence.

For some of the test DICOM-files, the network performs reasonably well. Figure 68 shows a selection of frames from DICOM-file 7 where the segmentations from the network are excellent.

Table 6: List of augmentation methods and their resulting DICE scores for the valve apparatus with the U-Net Auto-R network. The highest score in each column is highlighted.

| Augmentation | PMVL | | AMVL | | PA | | AA | |
|---|---|---|---|---|---|---|---|---|
| | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| No aug | 0.455 | 0.514 | 0.446 | 0.521 | 0.173 | 0.042 | 0.214 | 0.040 |
| Cropping (0.75) | 0.458 | 0.572 | 0.487 | 0.550 | 0.201 | 0.136 | 0.312 | 0.342 |
| Gaus (0.5) | 0.536 | 0.586 | 0.571 | 0.621 | 0.275 | 0.253 | **0.407** | **0.490** |
| Gamma (0.7) | 0.546 | 0.630 | 0.605 | 0.652 | 0.295 | 0.318 | 0.389 | 0.434 |
| Gamma (1.3) | 0.557 | 0.614 | 0.577 | 0.605 | 0.174 | 0.068 | 0.309 | 0.281 |
| Rotation (10 °) | 0.461 | 0.527 | 0.526 | 0.612 | 0.248 | 0.165 | 0.260 | 0.236 |
| Rotation (-5 °) | 0.505 | 0.602 | 0.538 | 0.613 | 0.193 | 0.090 | 0.341 | 0.358 |
| All | **0.589** | **0.693** | **0.631** | **0.700** | **0.375** | **0.398** | 0.399 | 0.438 |

Table 7: List of augmentation methods and their resulting DICE scores for LV, MY, and LA with the U-Net Auto-R network. The highest score in each column is highlighted.

| Augmentation | Background | | LV | | MY | | LA | |
|---|---|---|---|---|---|---|---|---|
| | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| No aug | 0.963 | 0.974 | 0.770 | 0.894 | 0.740 | 0.769 | 0.858 | 0.915 |
| Cropping (0.75) | 0.961 | 0.975 | 0.749 | 0.892 | 0.759 | 0.797 | 0.823 | 0.913 |
| Gaus (0.5) | 0.967 | 0.977 | 0.784 | 0.921 | 0.816 | 0.851 | 0.879 | 0.929 |
| Gamma (0.7) | 0.968 | 0.977 | 0.785 | 0.909 | 0.809 | 0.851 | **0.900** | 0.937 |
| Gamma (1.3) | **0.970** | **0.978** | 0.794 | 0.917 | 0.816 | 0.836 | 0.893 | 0.939 |
| Rotation (10 °) | 0.964 | 0.973 | 0.787 | 0.916 | 0.805 | 0.847 | 0.870 | 0.928 |
| Rotation (-5 °) | 0.965 | 0.974 | 0.796 | 0.918 | 0.790 | 0.847 | 0.874 | 0.932 |
| All | 0.969 | **0.978** | **0.809** | **0.929** | **0.838** | **0.870** | 0.899 | **0.944** |

Table 8: List of the DICE scores, for the valve apparatus, produced with the U-Net Auto-R trained with the original data set and with the cleaned data set. The highest score between the two models for each measurement of each class are highlighted.

| Class | Original data set | | | | Cleaned data set | | | |
|---|---|---|---|---|---|---|---|---|
| | Min | Max | Mean | Median | Min | Max | Mean | Median |
| PMVL | 0.000 | 0.898 | 0.589 | **0.693** | 0.000 | **0.900** | **0.618** | 0.677 |
| AMVL | 0.000 | **0.898** | **0.631** | **0.700** | 0.000 | 0.886 | 0.614 | 0.663 |
| PA | 0.000 | **0.909** | **0.375** | **0.398** | 0.000 | 0.889 | 0.276 | 0.248 |
| AA | 0.000 | **0.879** | **0.399** | **0.438** | 0.000 | 0.857 | 0.377 | 0.406 |

Table 9: List of the DICE scores, for the auto-generated classes, produced with the U-Net Auto-R trained with the original data set and with the cleaned data set. The highest score between the two models for each measurement of each class are highlighted.

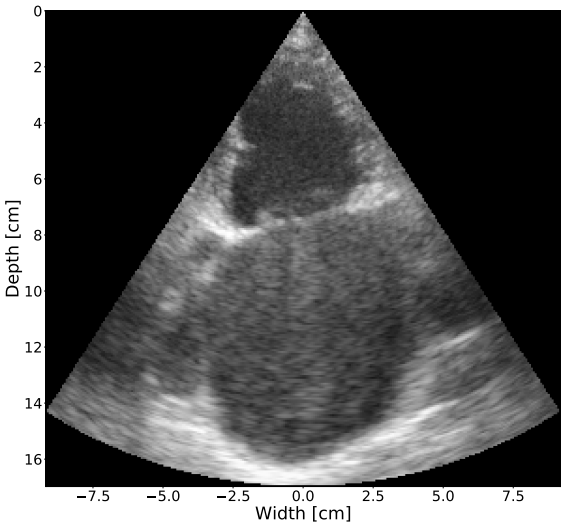
| Class | Original data set | | | | Cleaned data set | | | |
|---|---|---|---|---|---|---|---|---|
| | Min | Max | Mean | Median | Min | Max | Mean | Median |
| Background | 0.893 | **0.994** | 0.969 | 0.978 | **0.897** | 0.992 | **0.972** | **0.979** |
| LV | 0.000 | **0.982** | 0.809 | 0.929 | 0.000 | 0.977 | **0.877** | **0.930** |
| MY | 0.021 | 0.949 | 0.838 | 0.870 | **0.587** | **0.954** | **0.858** | **0.882** |
| LA | 0.187 | **0.988** | 0.899 | 0.944 | **0.410** | **0.988** | **0.910** | **0.946** |

Table 10: A summary of manhattan distance metric for the annulus points predictions performed by the U-Net Auto-R network. The distances are given in centimeters.

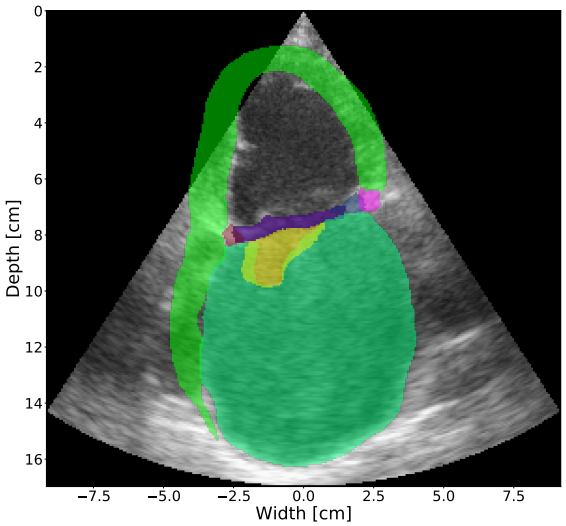
| **Class** | **Min** | **Max** | **Mean** | **Median** |
|---|---|---|---|---|
| Posterior annulus | 0.007 | 3.181 | 0.492 | 0.378 |
| Anterior annulus | 0.037 | 6.804 | 0.481 | 0.297 |

Table 11: A summary of the difference between the estimated angles of the valve predictions performed by the U-Net Auto-R network and the ground truth segmentations. The angles are given in degrees.

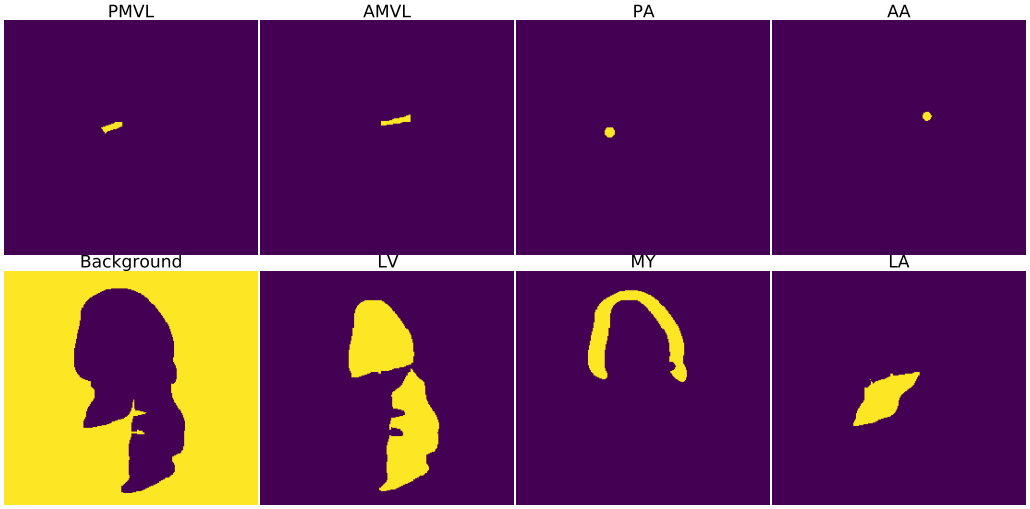
| **Class** | **Min** | **Max** | **Mean** | **Median** |
|---|---|---|---|---|
| Posterior leaflet | 0.010 | 94.00 | 16.74 | 7.43 |
| Anterior leaflet | 0.012 | 137.16 | 20.25 | 10.30 |



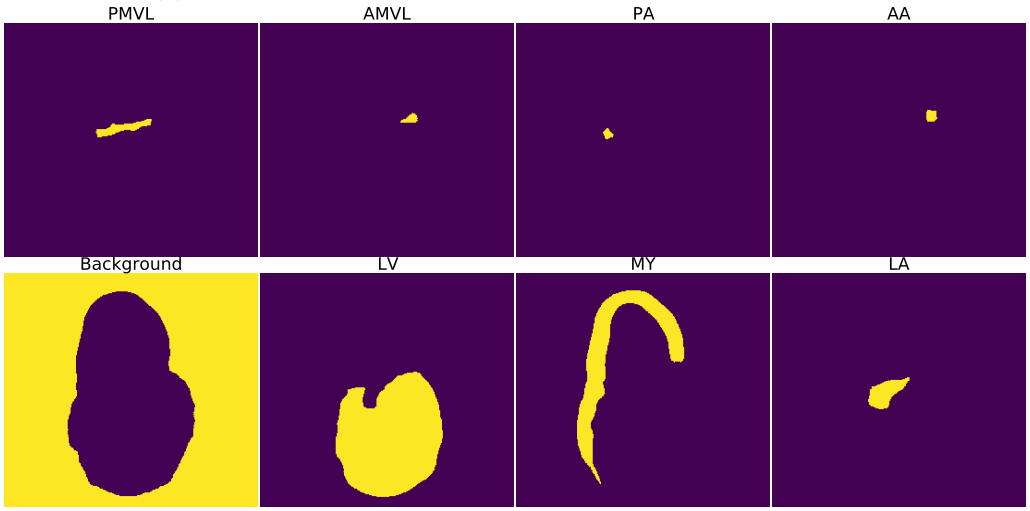

(a) Input B-mode image




(b) Predicted Segmentations

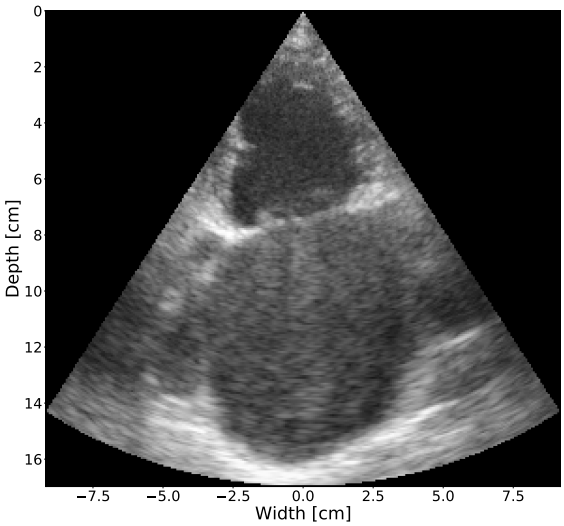



(c) One-hot encoded ground truth channel segmentations

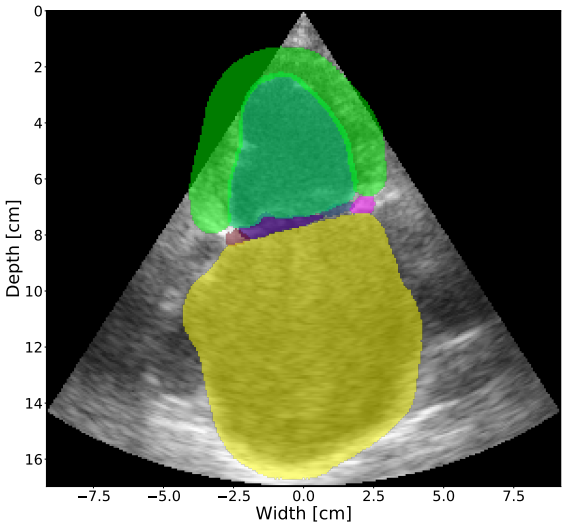



(d) One-hot encoded predicted channel segmentations

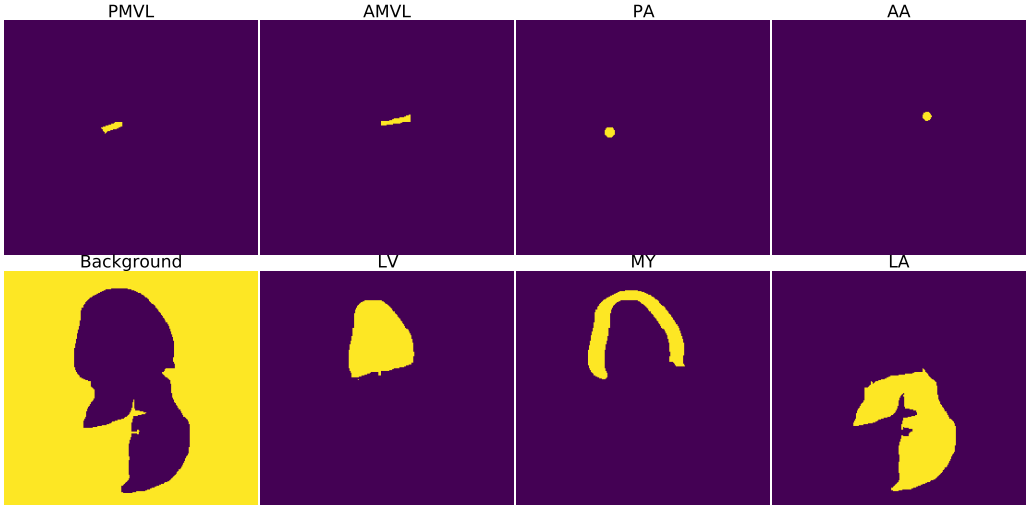
Figure 43: Test sample, index 51, produced by the U-Net Auto network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), AA (pink), LV (dark green), MY (light green), and LA (yellow). (c) and (d) show the one-hot encoded channels of the ground truth and predicted segmentations, respectively. The presented output segmentations have been post-processed.



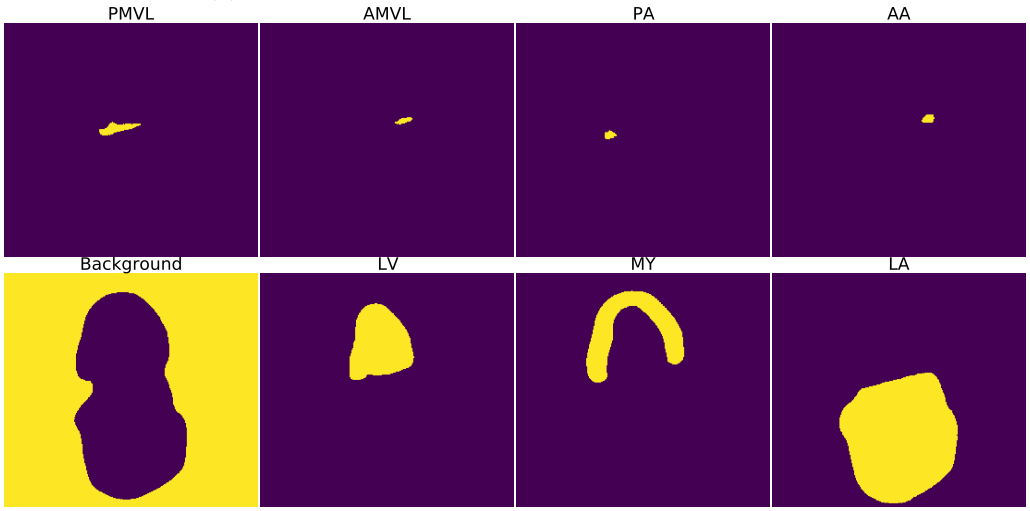

(a) Input B-mode image




(b) Predicted segmentations

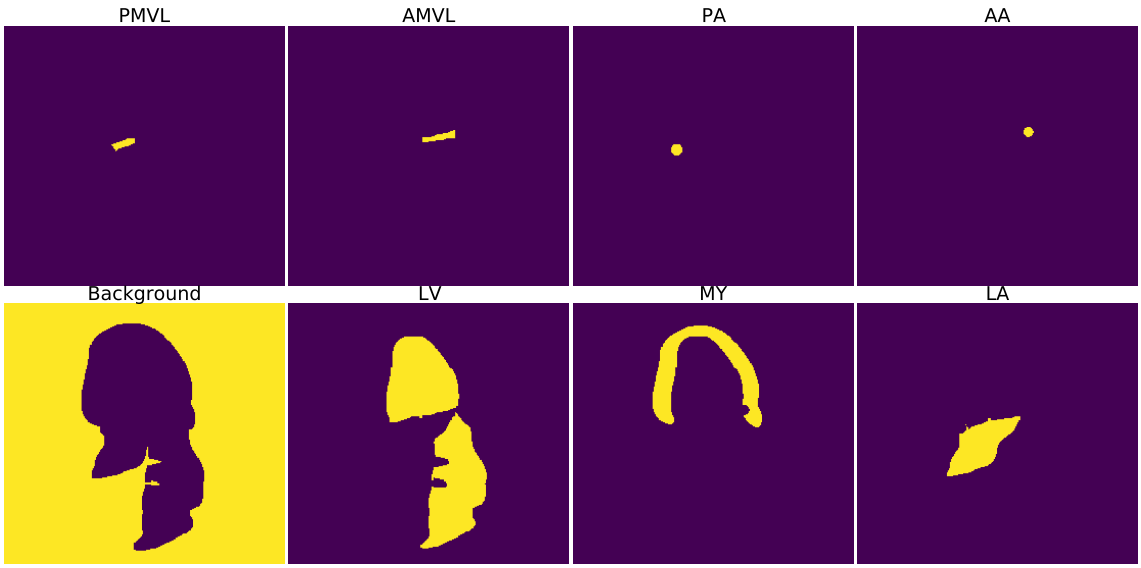



(c) One-hot encoded input channel segmentation

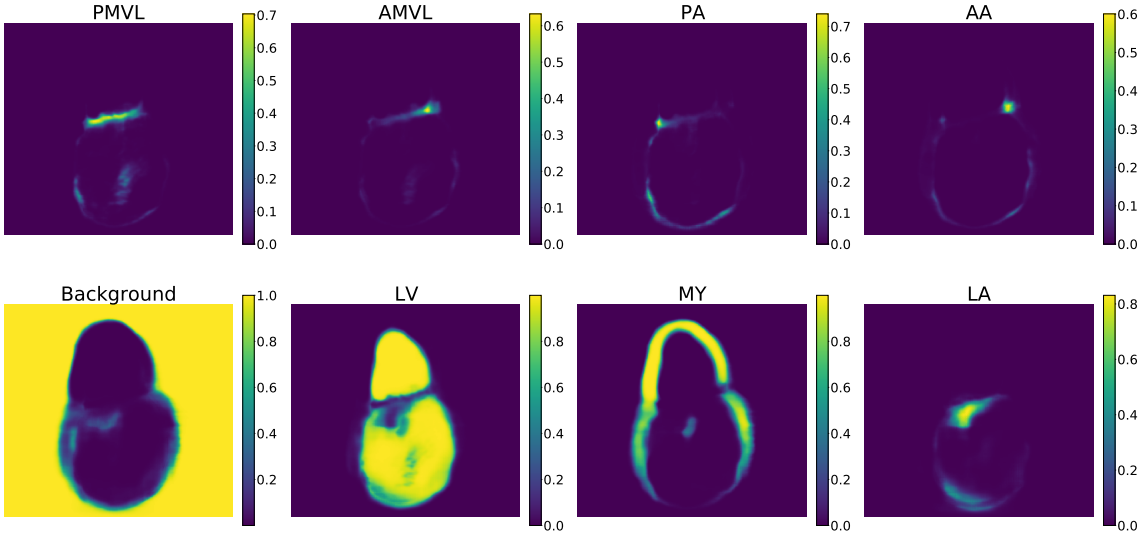



(d) One-hot encoded output channel segmentations

Figure 44: Test sample, index 51, produced by the U-Net Auto-R network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), AA (pink), LV (dark green), MY (light green), and LA (yellow). (c) and (d) show the one-hot encoded channels of the ground truth and predicted segmentations, respectively. The presented output segmentations have been post-processed.

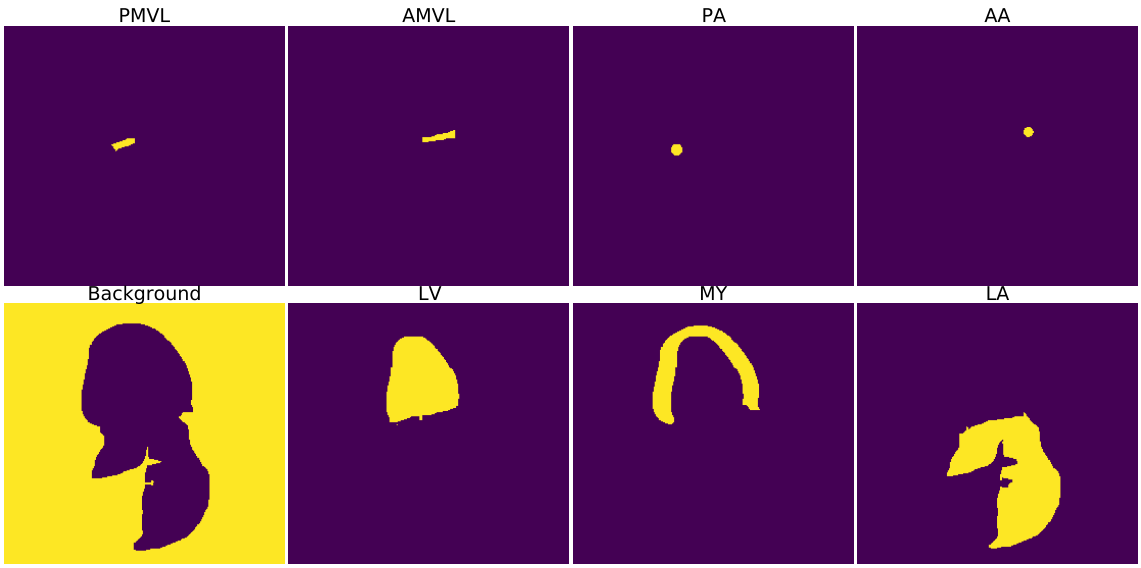



(a) One-hot encoded input channel segmentation

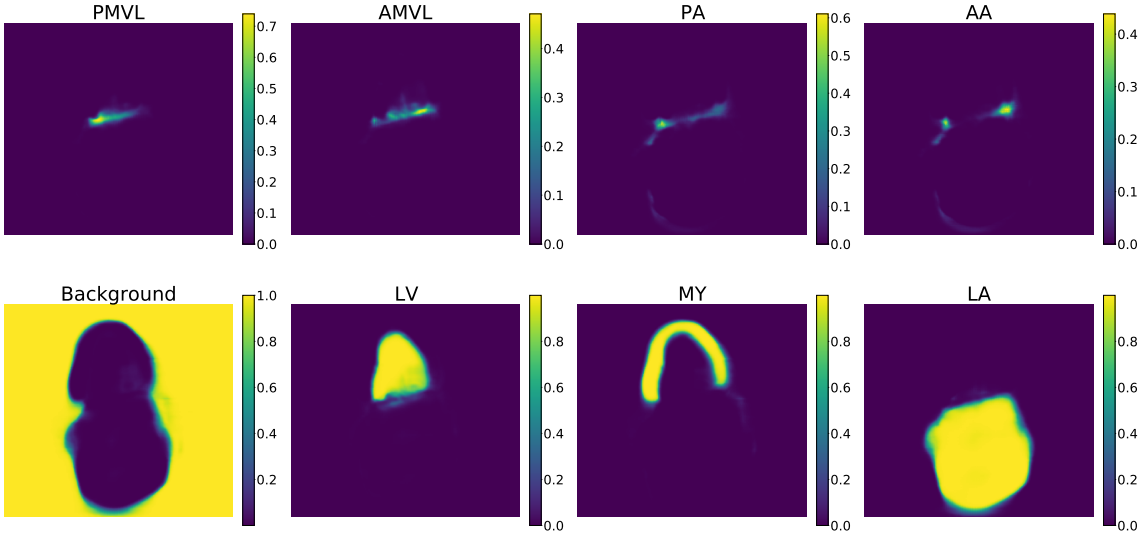



(b) Raw output channel segmentations

Figure 45: Test sample, index 51, of the ground truth segmentations (a), and the raw channel output from the U-Net Auto network.

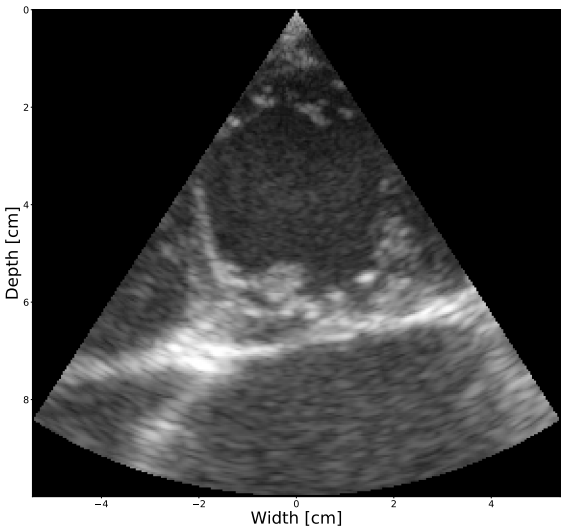



(a) One-hot encoded input channel segmentation

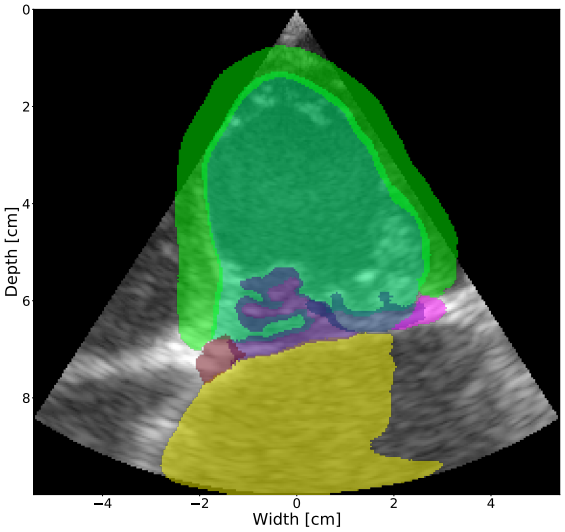



(b) Raw one-hot encoded output channel segmentations

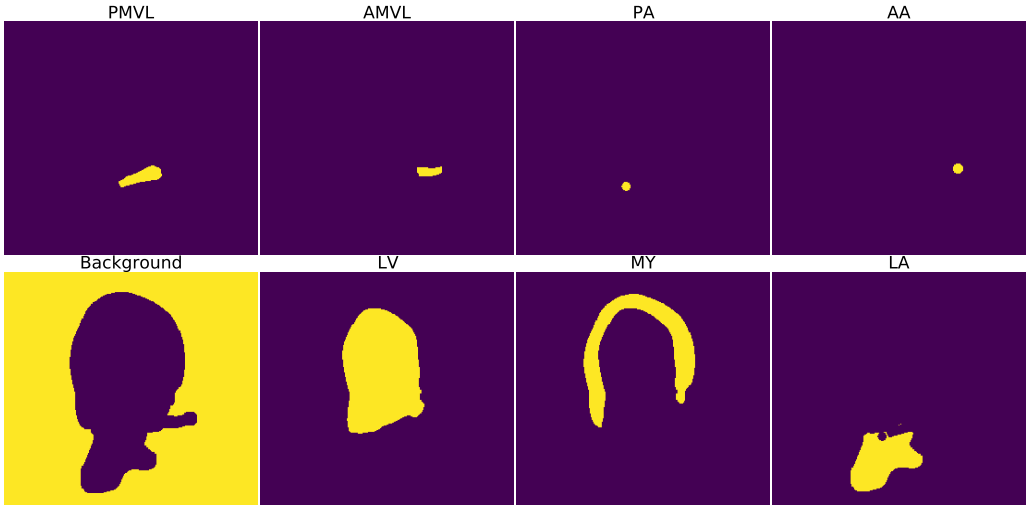
Figure 46: Test sample, index 51, of the ground truth segmentations (a), and the raw channel output from the U-Net Auto-R network.



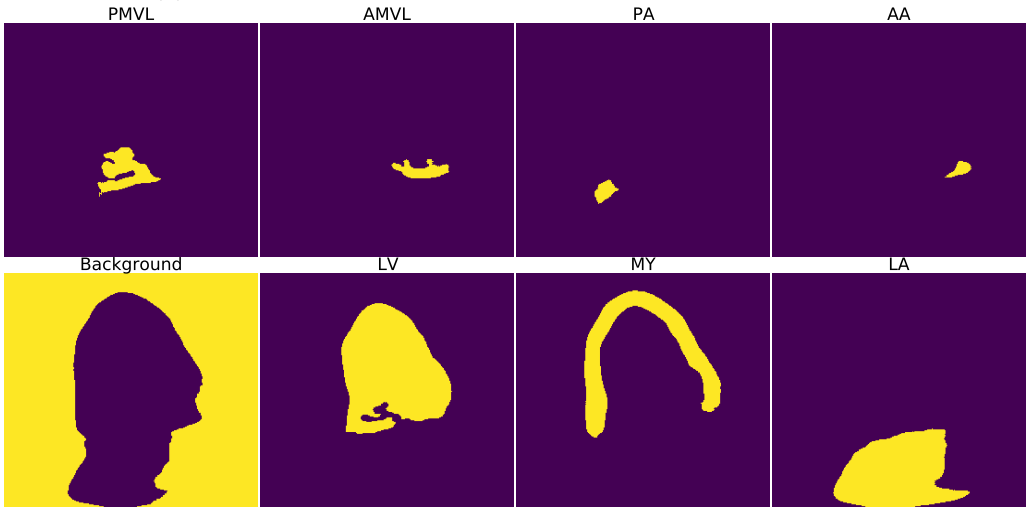

(a) Input B-mode image




(b) Predicted Segmentations

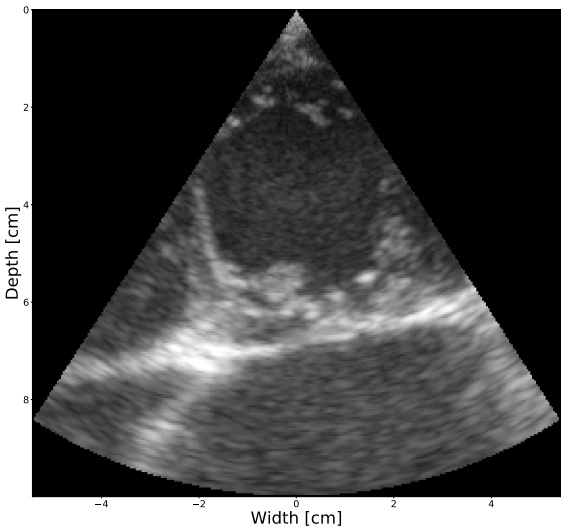



(c) One-hot encoded ground truth channel segmentations

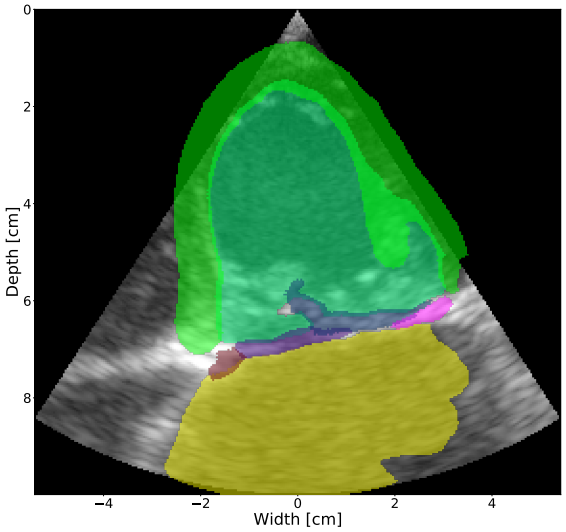



(d) One-hot encoded predicted channel segmentations

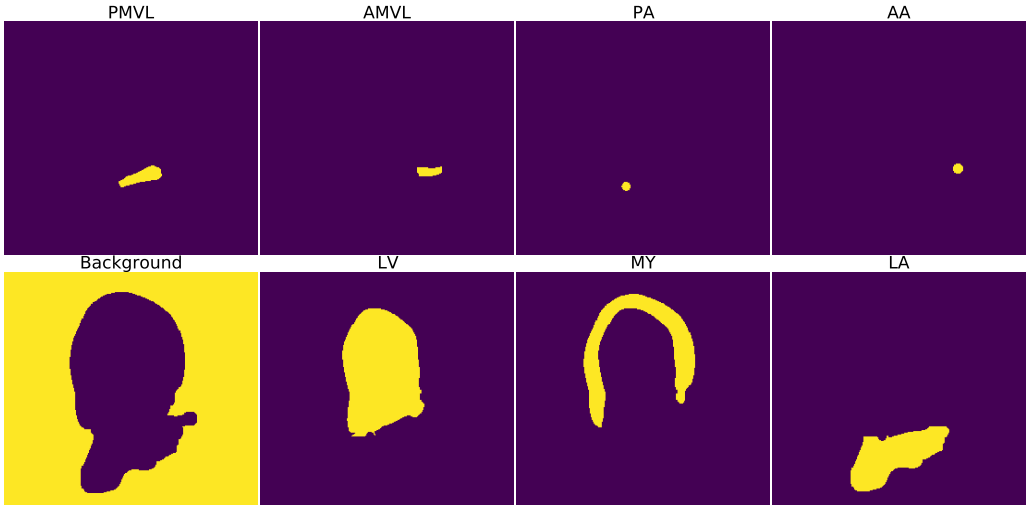
Figure 47: Test sample, index 116, produced by the U-Net Auto network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), AA (pink), LV (dark green), MY (light green), and LA (yellow). (c) and (d) show the one-hot encoded channels of the ground truth and predicted segmentations, respectively. The presented output segmentations have been post-processed.



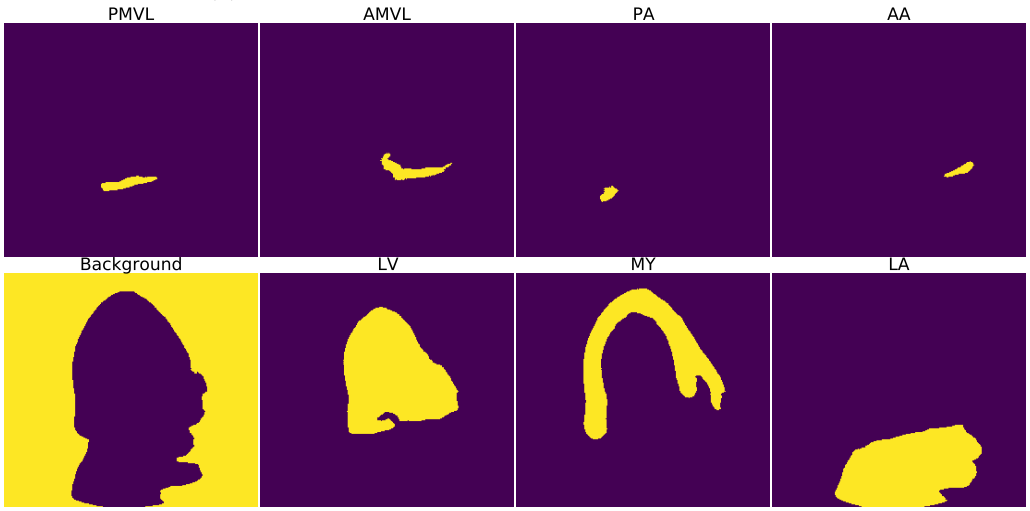

(a) Input B-mode image




(b) Predicted segmentations

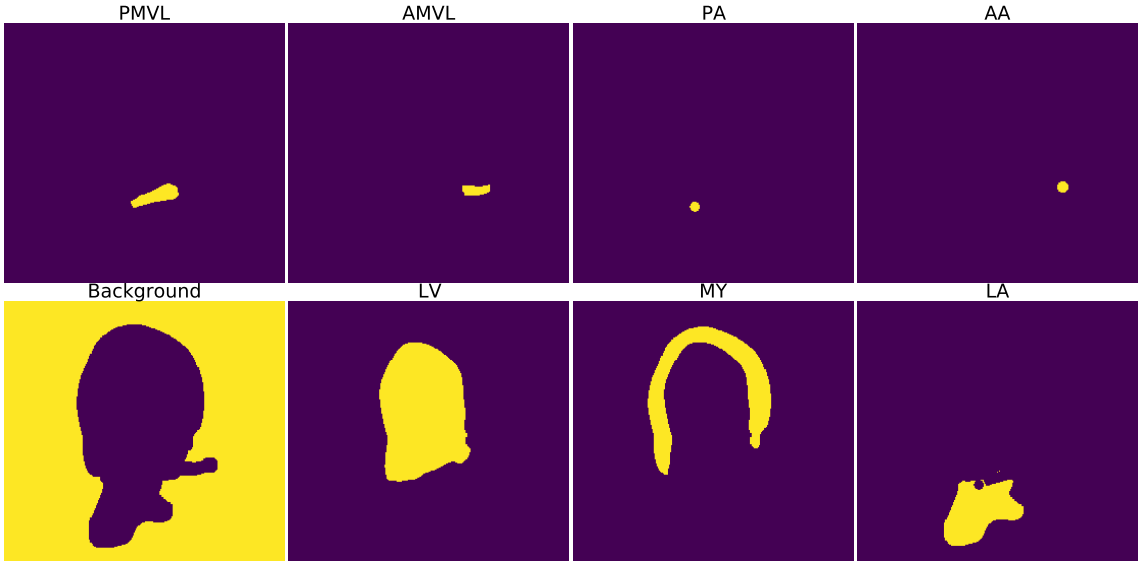



(c) One-hot encoded input channel segmentation

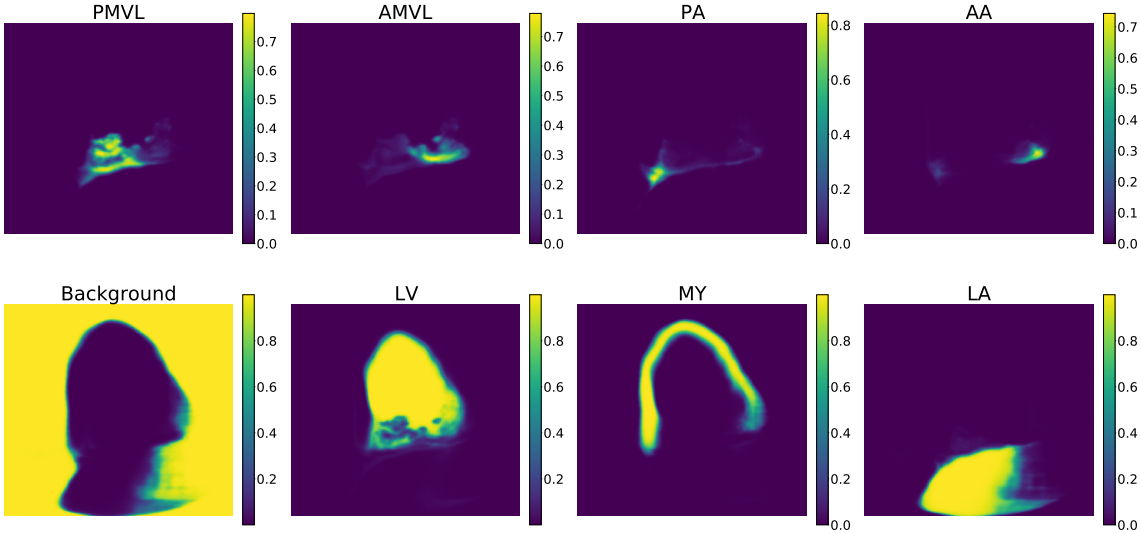



(d) One-hot encoded output channel segmentations

Figure 48: Test sample, index 116, produced by the U-Net Auto-R network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), AA (pink), LV (dark green), MY (light green), and LA (yellow). (c) and (d) show the one-hot encoded channels of the ground truth and predicted segmentations, respectively. The presented output segmentations have been post-processed.

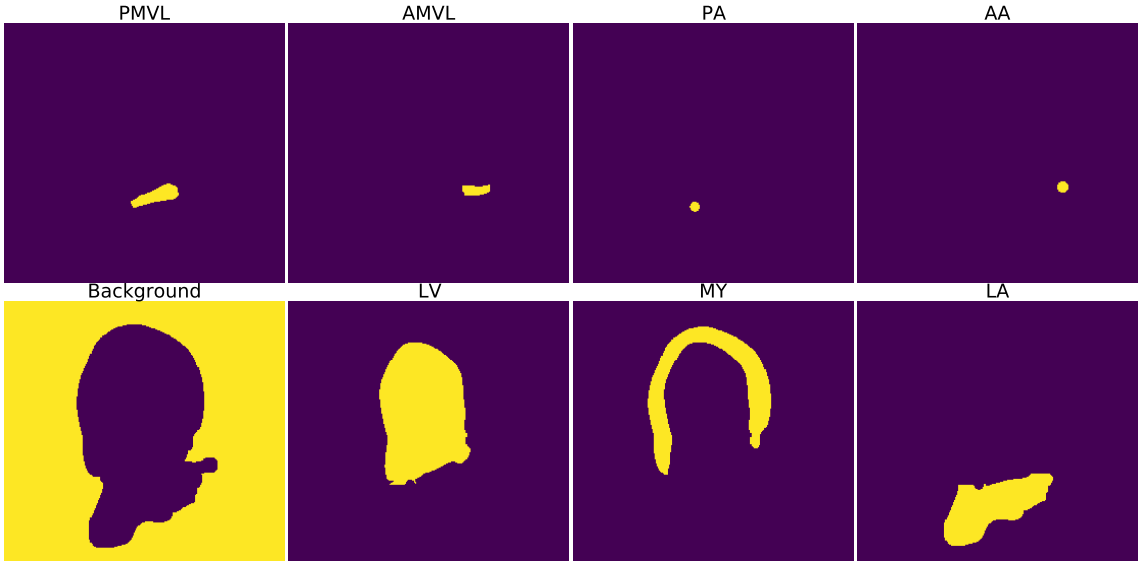



(a) One-hot encoded input channel segmentation

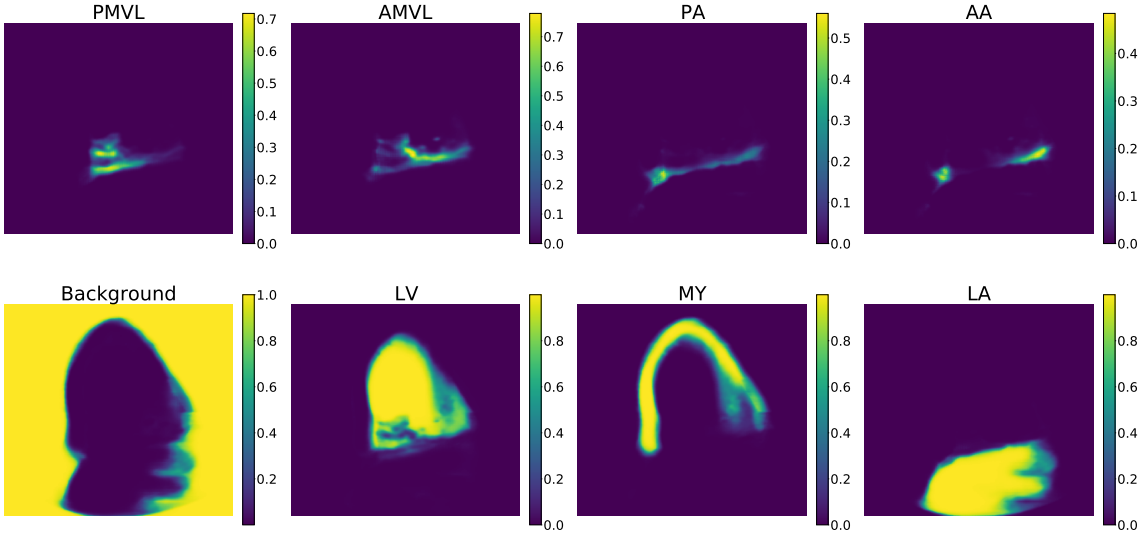



(b) Raw output channel segmentations

Figure 49: Test sample, index 116, of the ground truth segmentations (a), and the raw channel output from the U-Net Auto network.

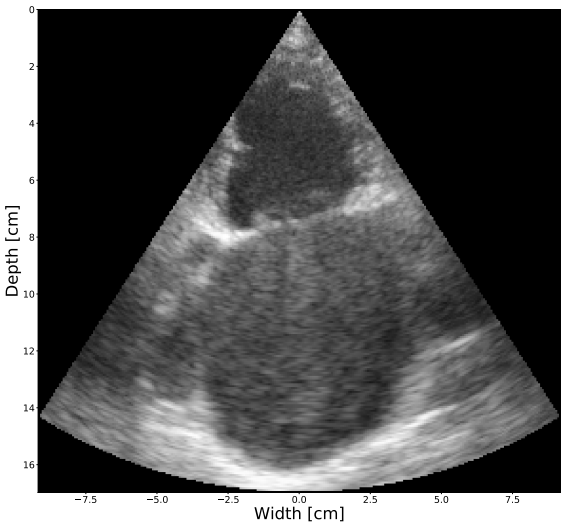



(a) One-hot encoded input channel segmentation

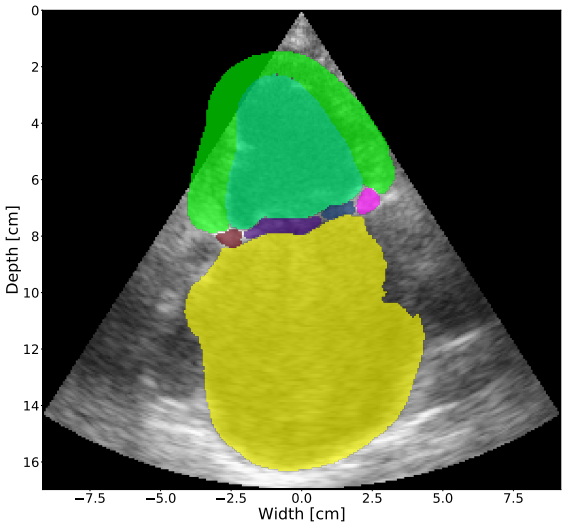



(b) Raw one-hot encoded output channel segmentations

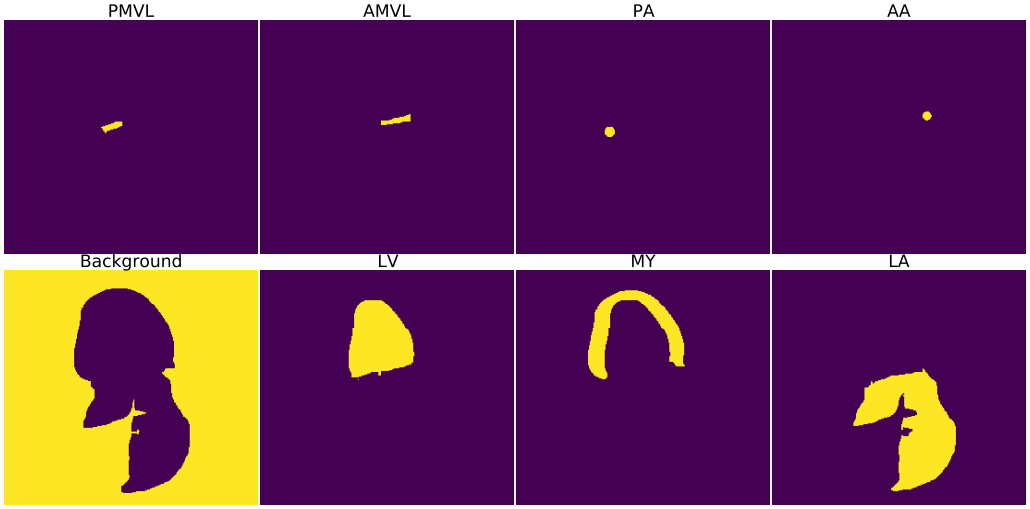
Figure 50: Test sample, index 116, of the ground truth segmentations (a), and the raw channel output from the U-Net Auto-R network.



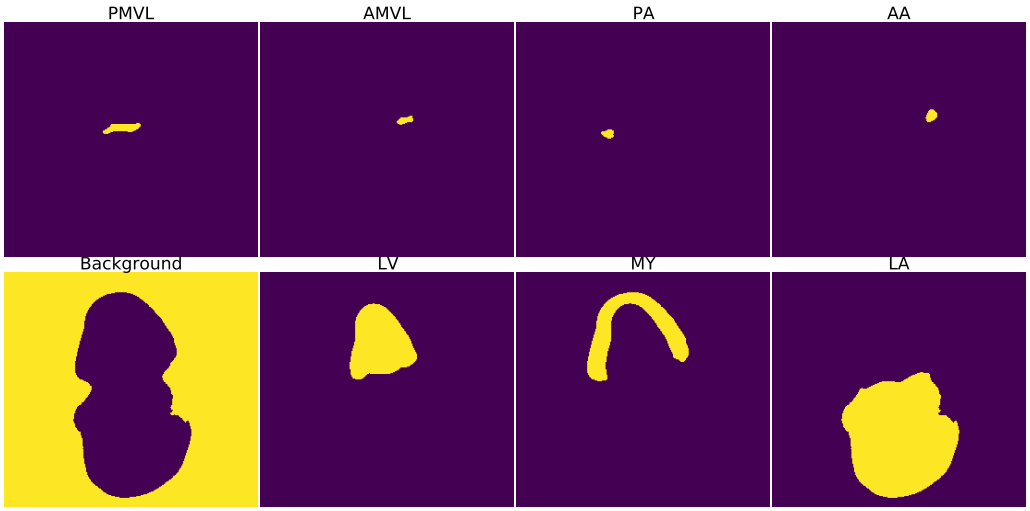

(a) Input B-mode image




(b) Predicted Segmentations

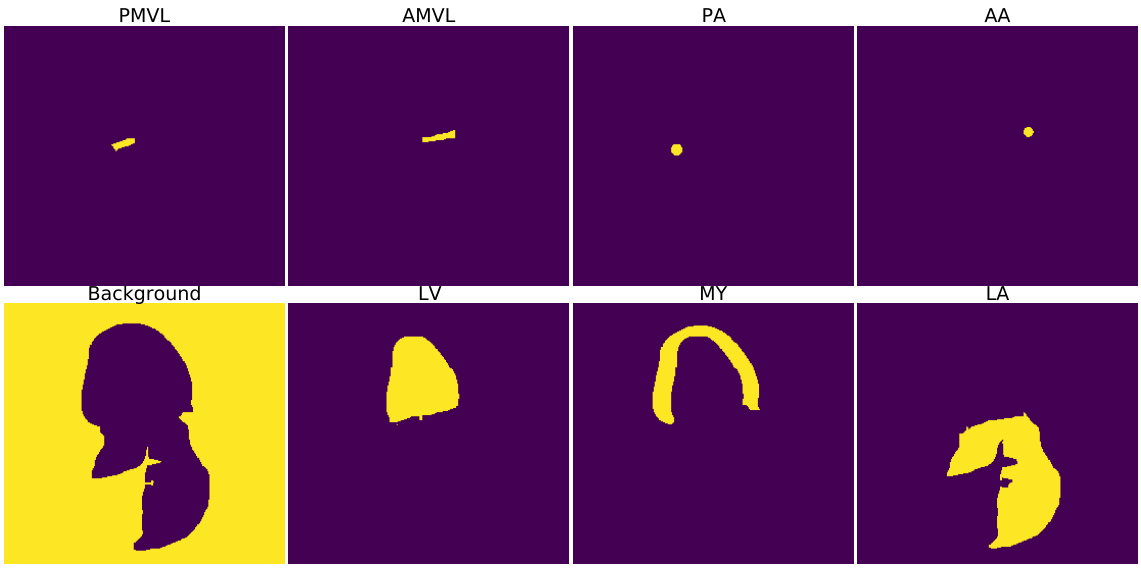



(c) One-hot encoded ground truth channel segmentations

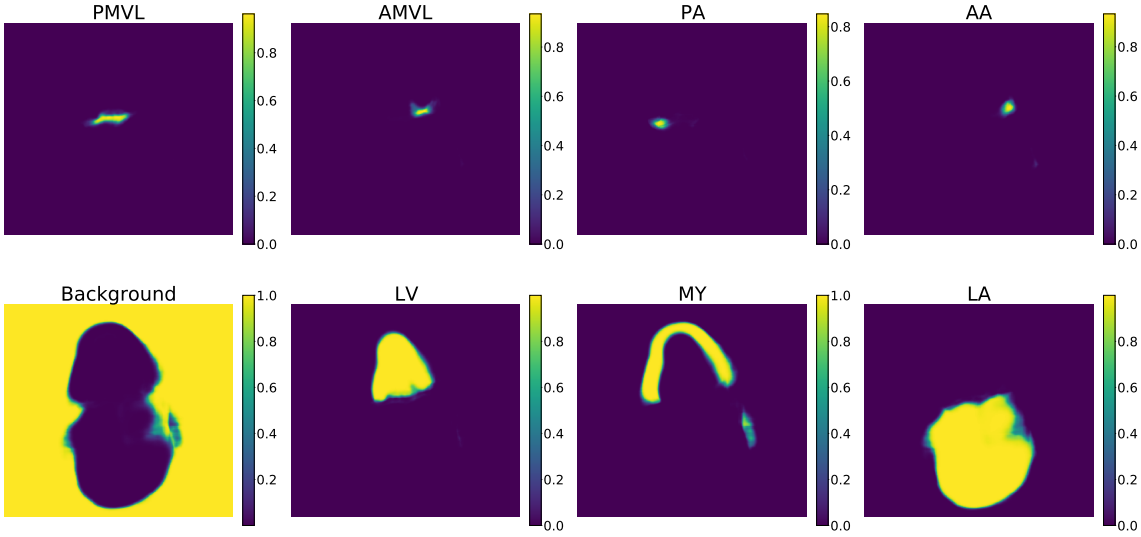



(d) One-hot encoded predicted channel segmentations

Figure 51: Test sample, index 51, produced by the U-Net Auto-R network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), AA (pink), LV (dark green), MY (light green), and LA (yellow). (c) and (d) show the one-hot encoded channels of the ground truth and predicted segmentations, respectively. The presented output segmentations have been post-processed.

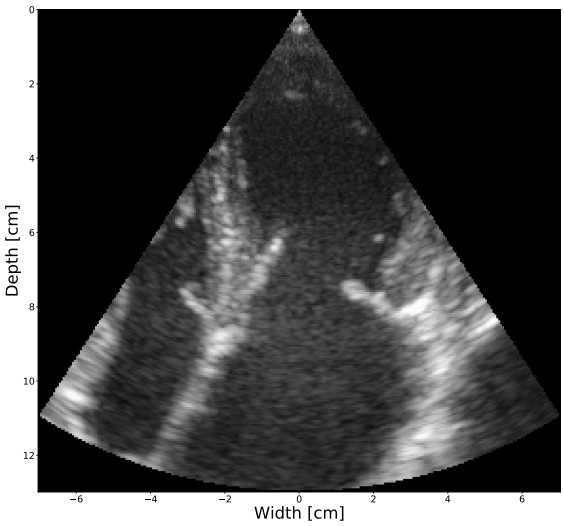



(a) One-hot encoded input channel segmentation

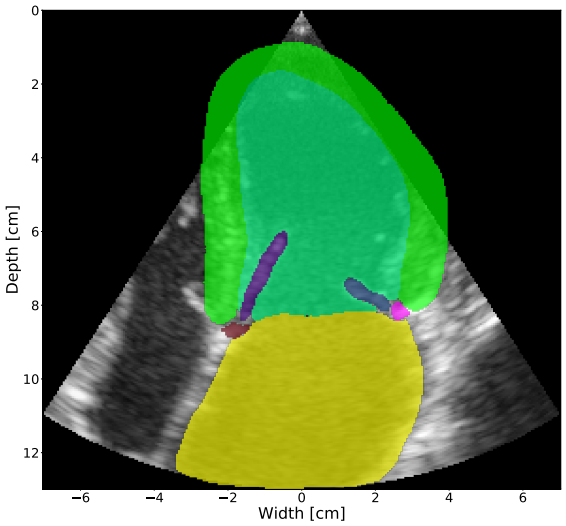



(b) Raw output channel segmentations

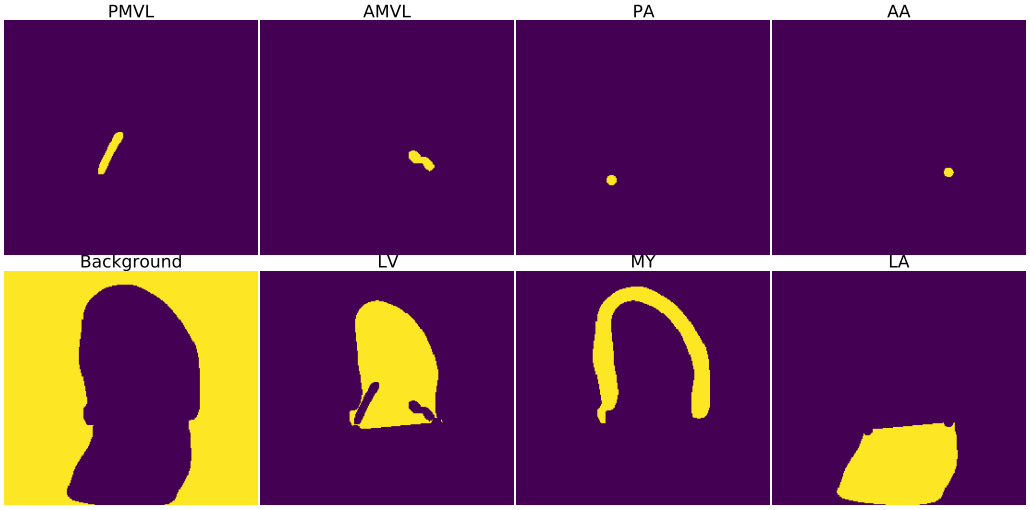
Figure 52: Test sample, index 51, of the ground truth segmentations (a), and the raw channel output from the augmented U-Net Auto-R network.



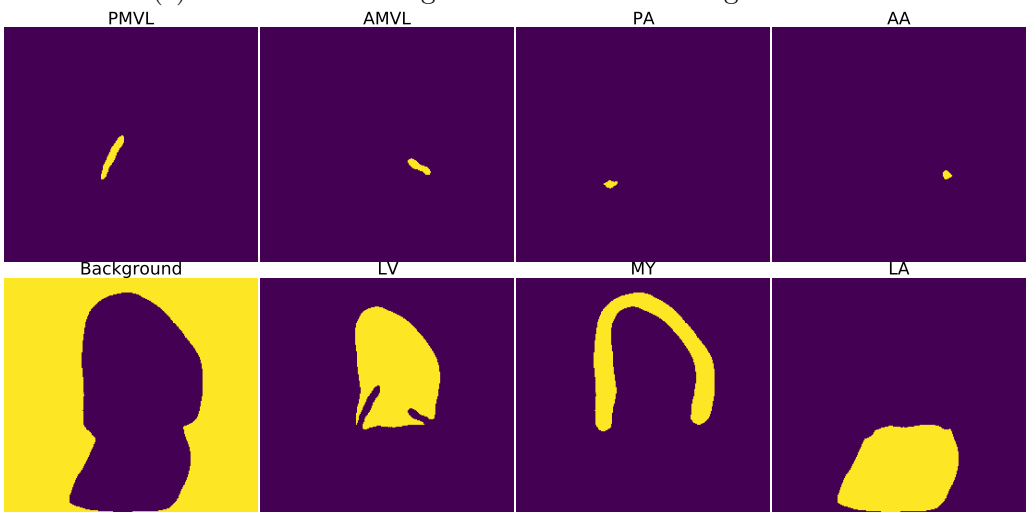

(a) Input B-mode image




(b) Predicted Segmentations

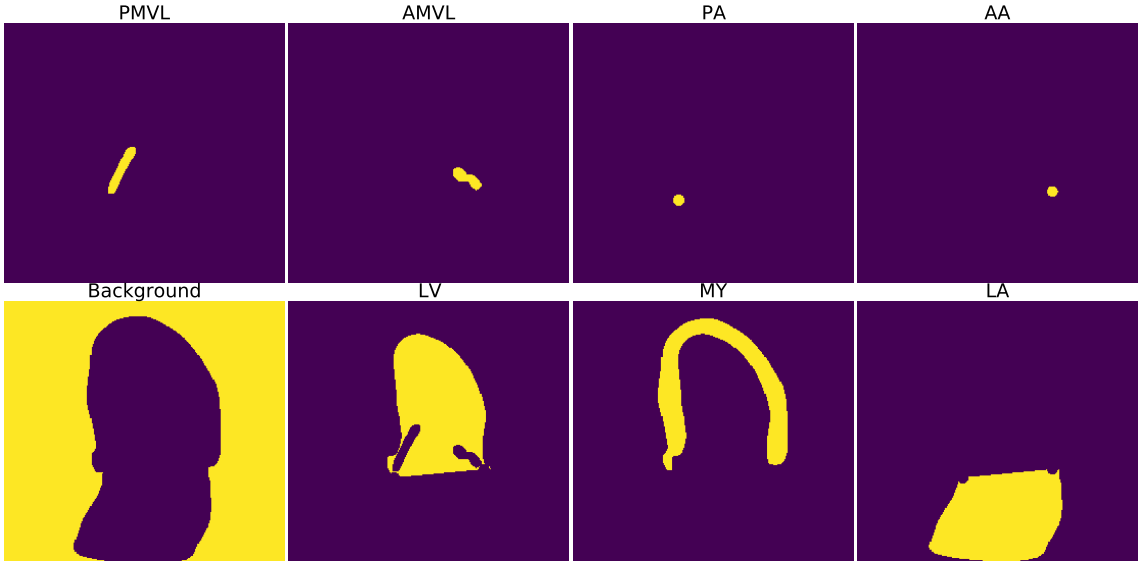



(c) One-hot encoded ground truth channel segmentations

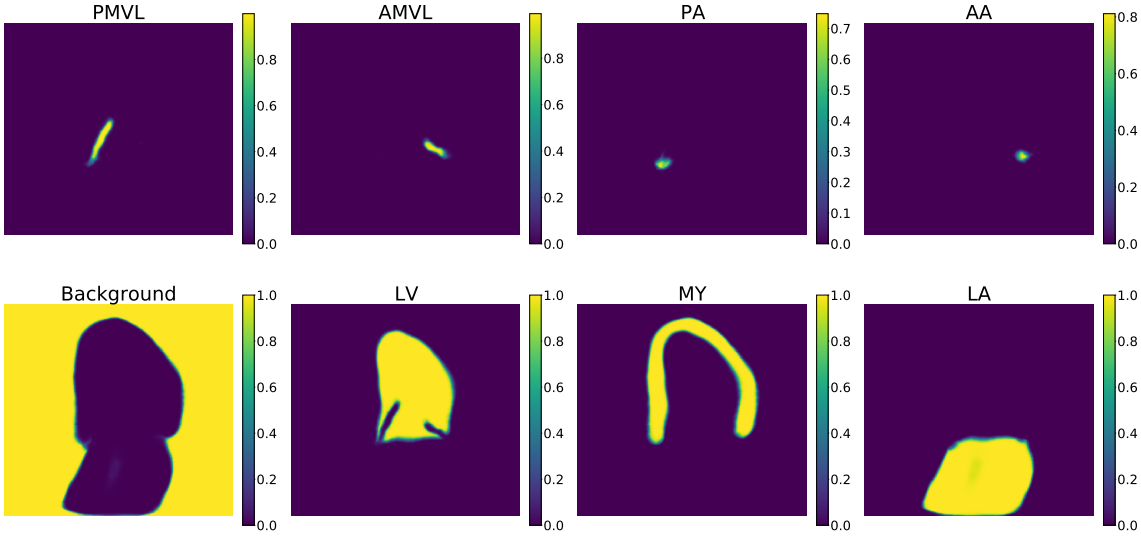



(d) One-hot encoded predicted channel segmentations

Figure 53: Test sample, index 35, produced by the U-Net Auto-R network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), AA (pink), LV (dark green), MY (light green), and LA (yellow). (c) and (d) show the one-hot encoded channels of the ground truth and predicted segmentations, respectively. The presented output segmentations have been post-processed.

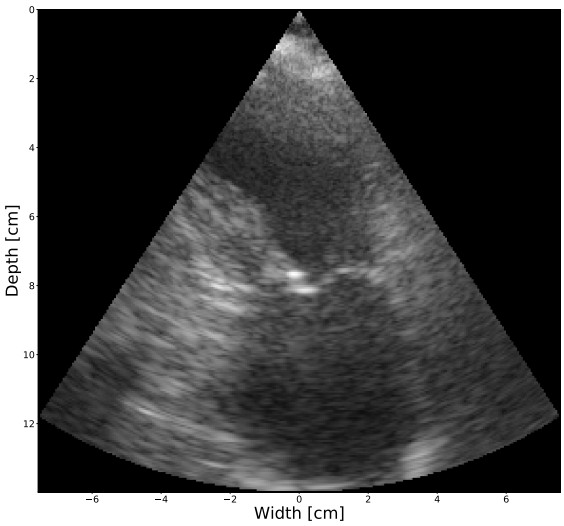



(a) One-hot encoded input channel segmentation

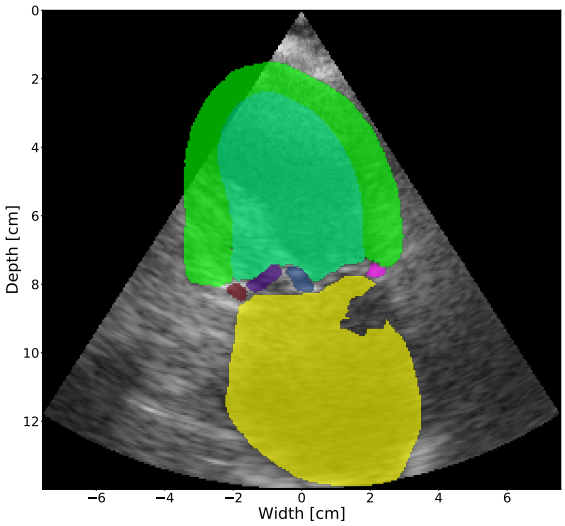



(b) Raw output channel segmentations

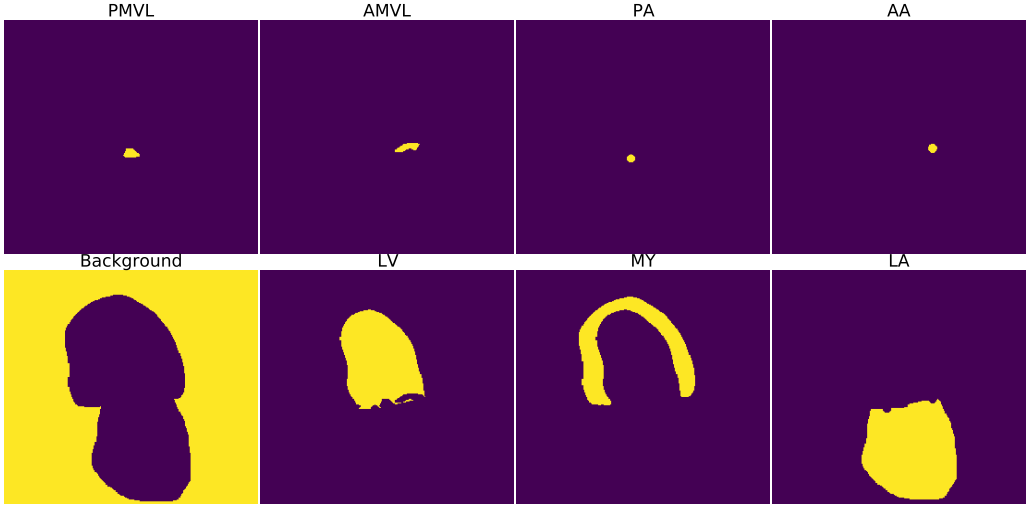
Figure 54: Test sample, index 35, of the ground truth segmentations (a), and the raw channel output from the U-Net Auto-R network.



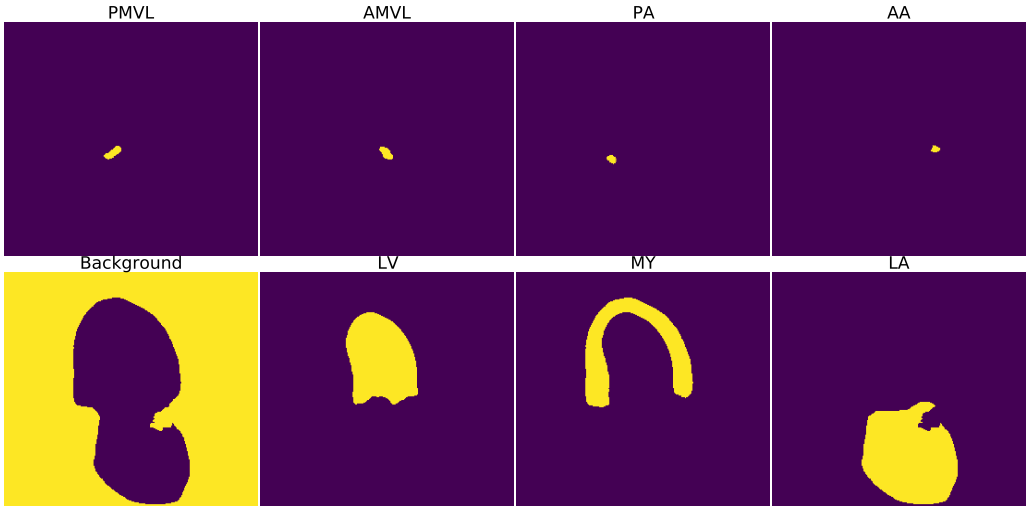

(a) Input B-mode image




(b) Predicted Segmentations

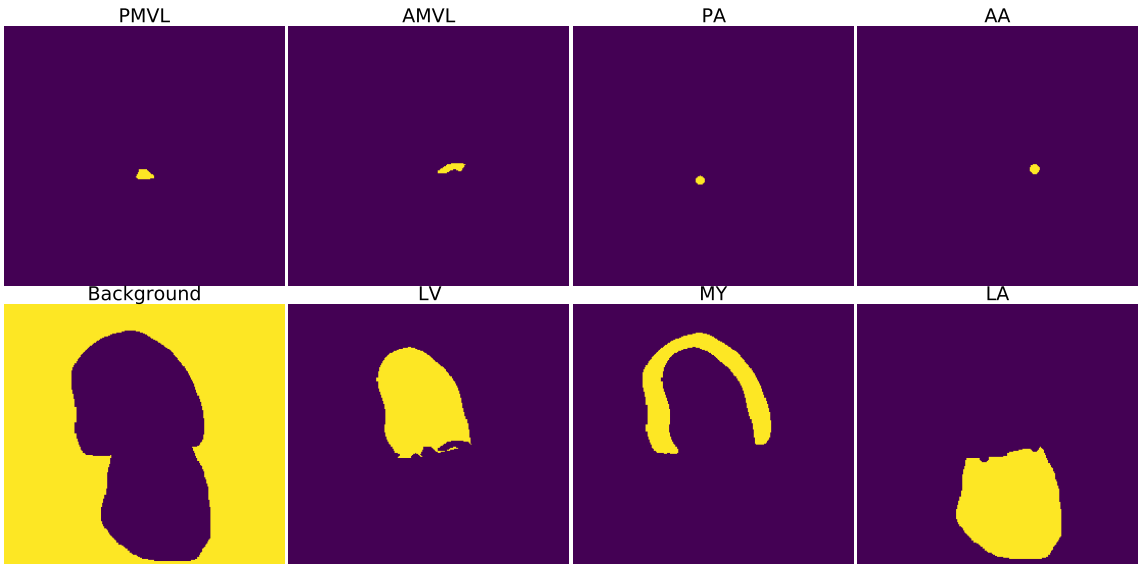



(c) One-hot encoded ground truth channel segmentations

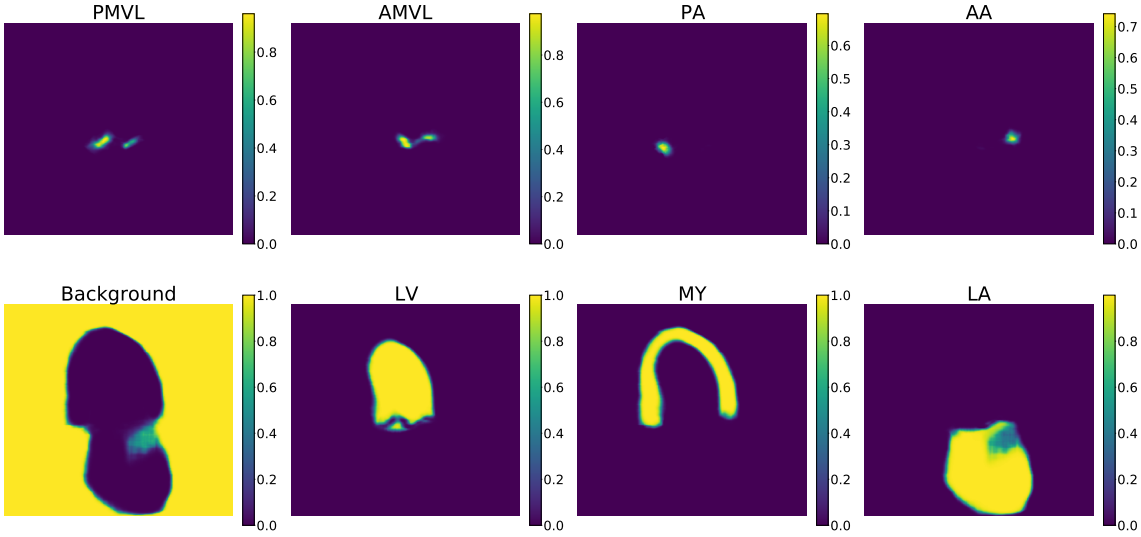



(d) One-hot encoded predicted channel segmentations

Figure 55: Test sample, index 77, produced by the U-Net Auto-R network. (a) show the input B-mode image. (b) show the output segmentations from the network of the PMVL (purple), AMVL (blue), PA (red), AA (pink), LV (dark green), MY (light green), and LA (yellow). (c) and (d) show the one-hot encoded channels of the ground truth and predicted segmentations, respectively. The presented output segmentations have been post-processed.

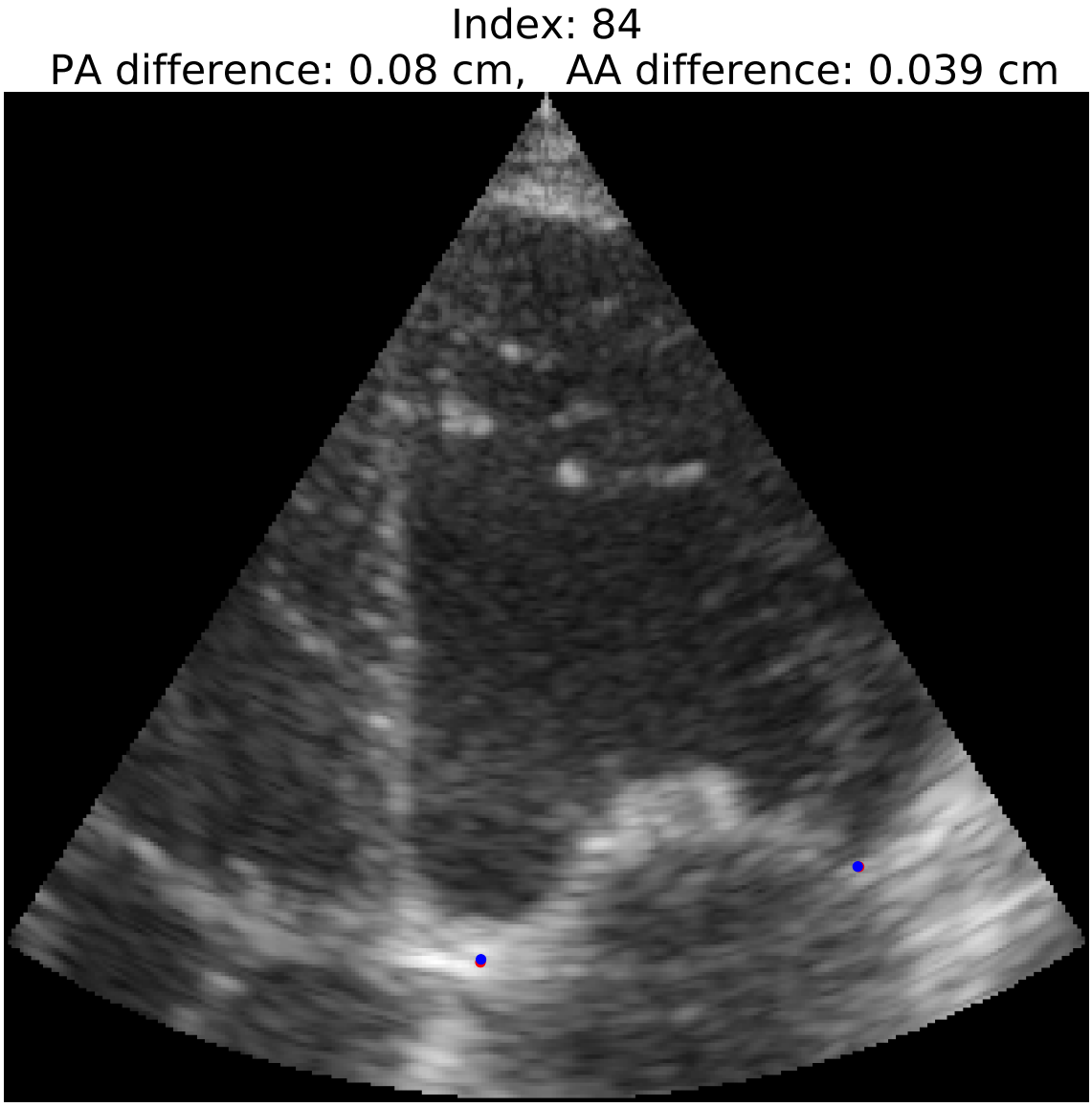



(a) One-hot encoded input channel segmentation




(b) Raw output channel segmentations

Figure 56: Test sample, index 77, of the ground truth segmentations (a), and the raw channel output from the U-Net Auto-R network.

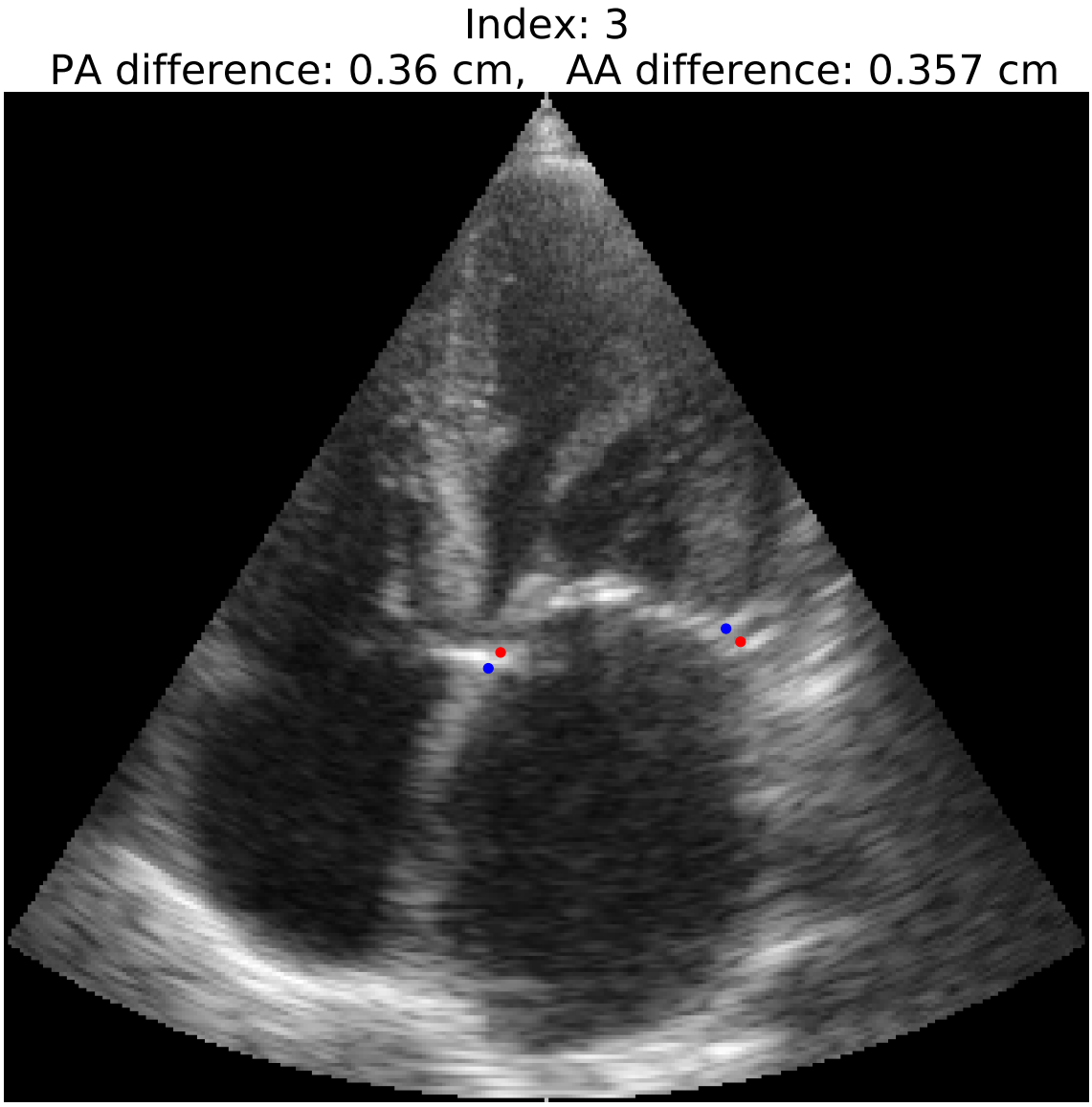



(a) Input B-mode image

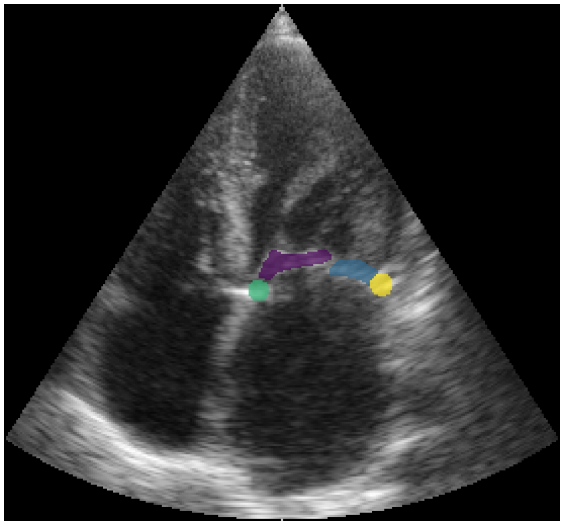
(b) Ground truth

(c) Prediction

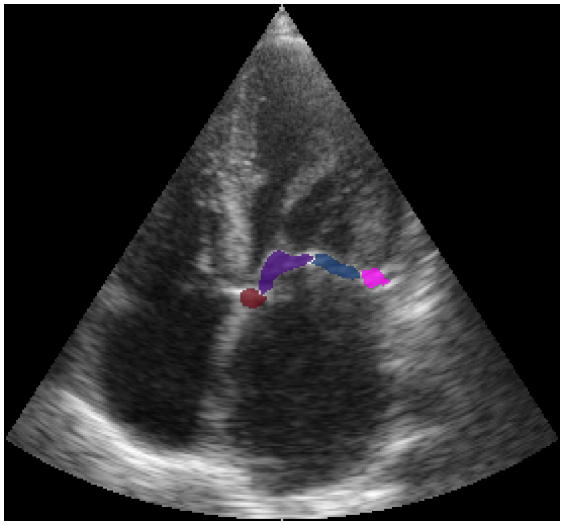
Figure 57: Sample with index 84 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net Auto-R model.




(a) Input B-mode image

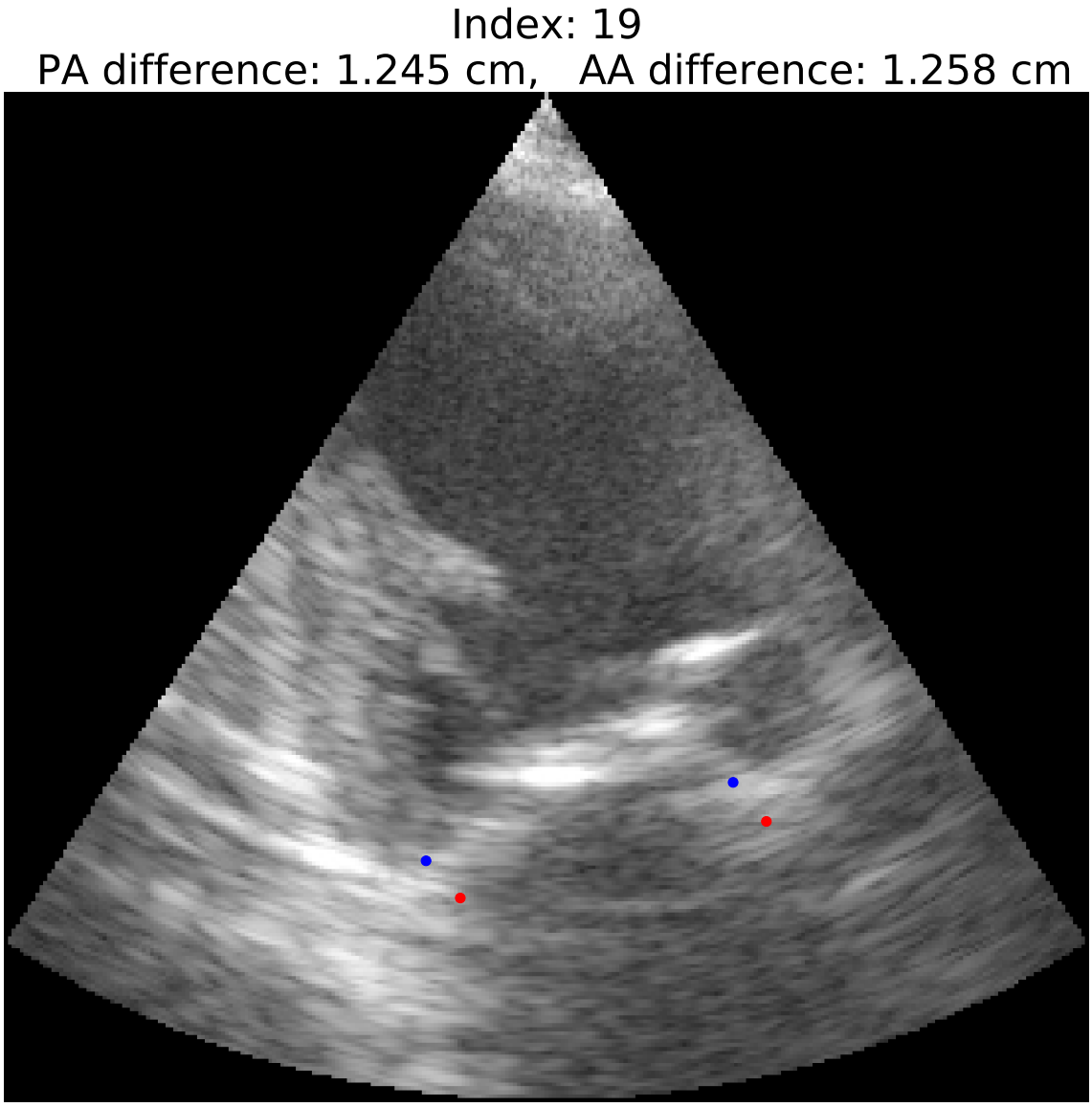


(b) Ground truth

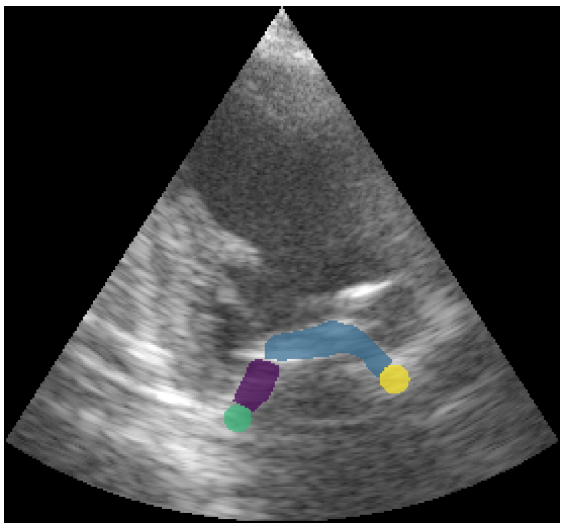


(c) Prediction

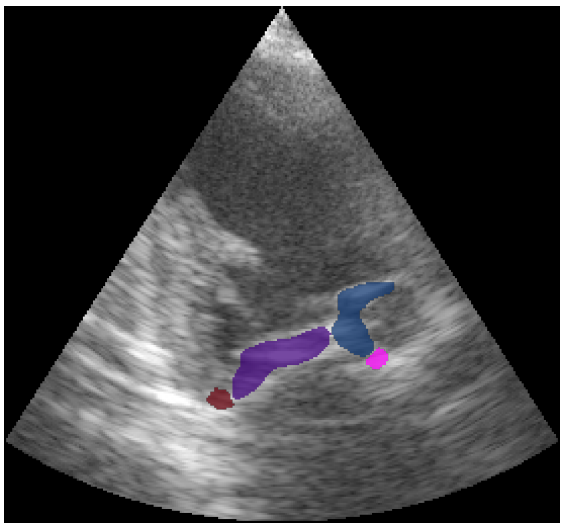
Figure 58: Sample with index 3 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net Auto-R model.




(a) Input B-mode image

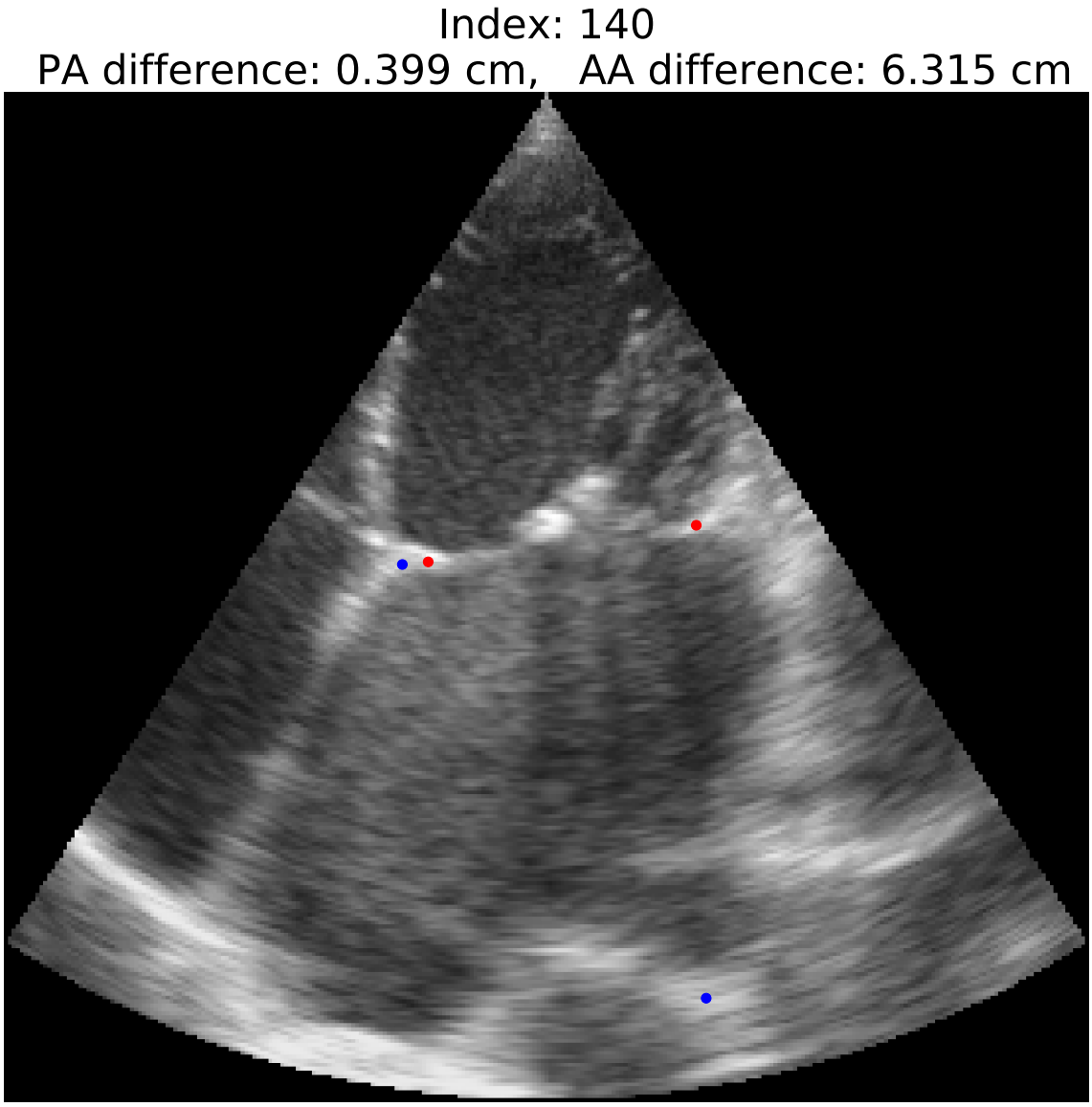


(b) Ground truth

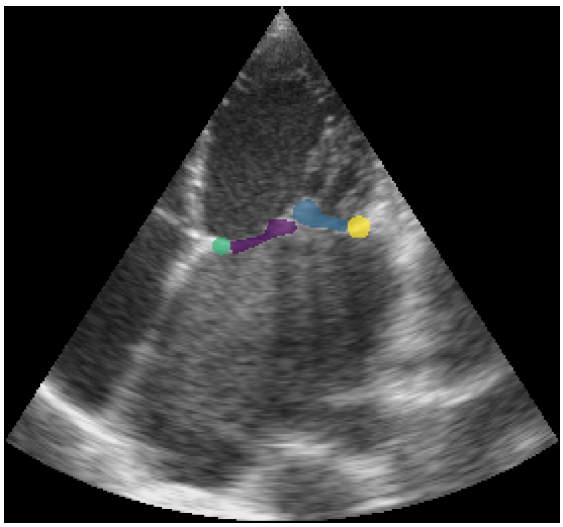


(c) Prediction

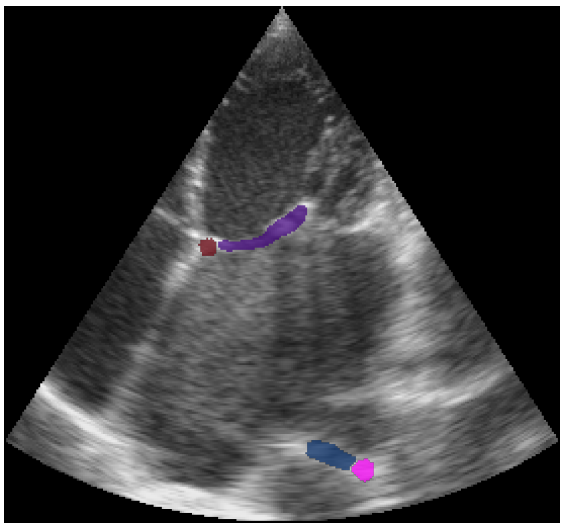
Figure 59: Sample with index 19 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net Auto-R model.




(a) Input B-mode image

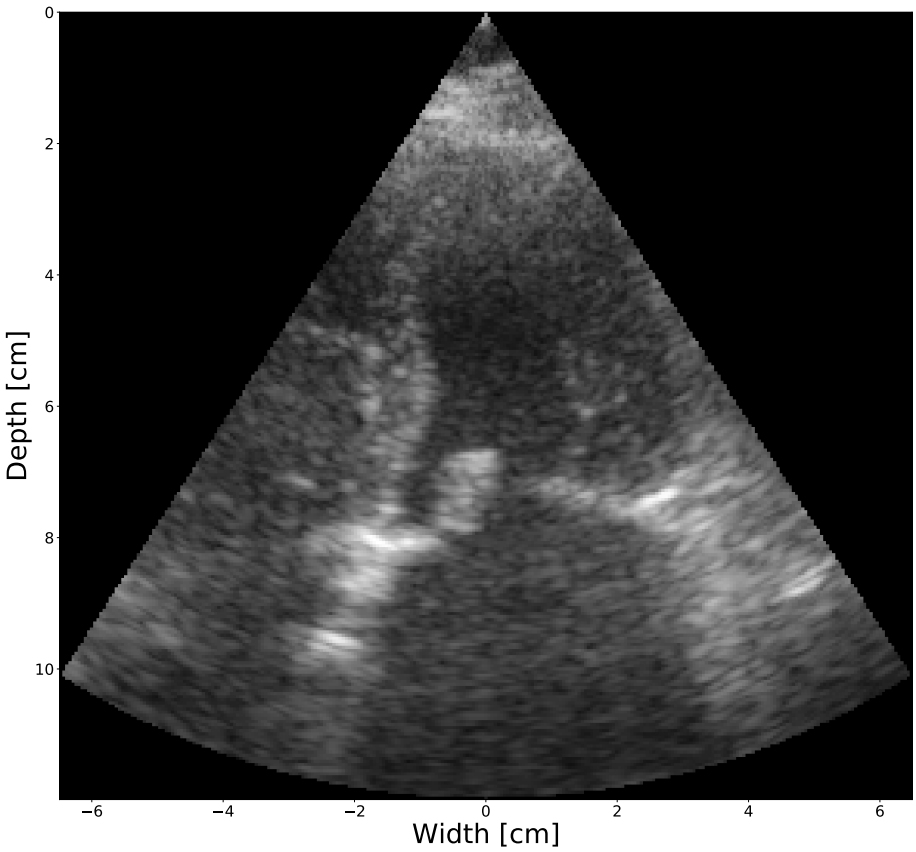


(b) Ground truth

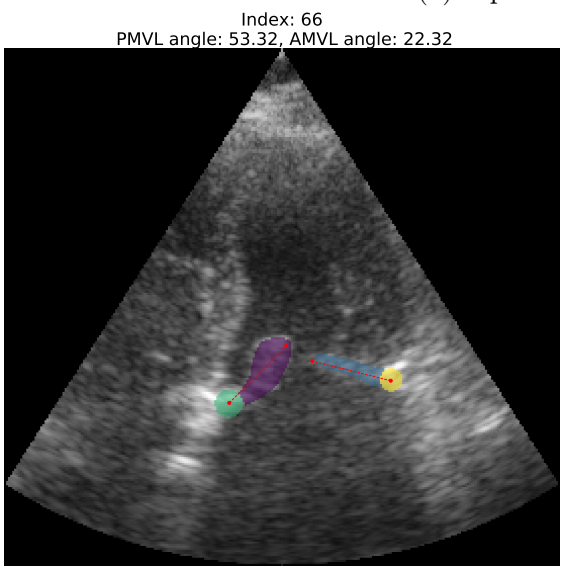


(c) Prediction

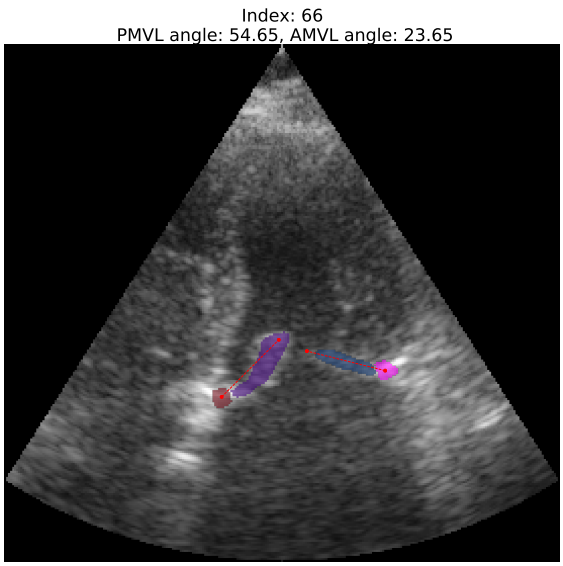
Figure 60: Sample with index 140 from the test set with the ground truth center points (red) and predicted center points (blue) layered on top of the input B-mode image (a), the ground truth segmentations (b) and the predicted segmentations (c) by the U-Net Auto-R model.




(a) Input B-mode image

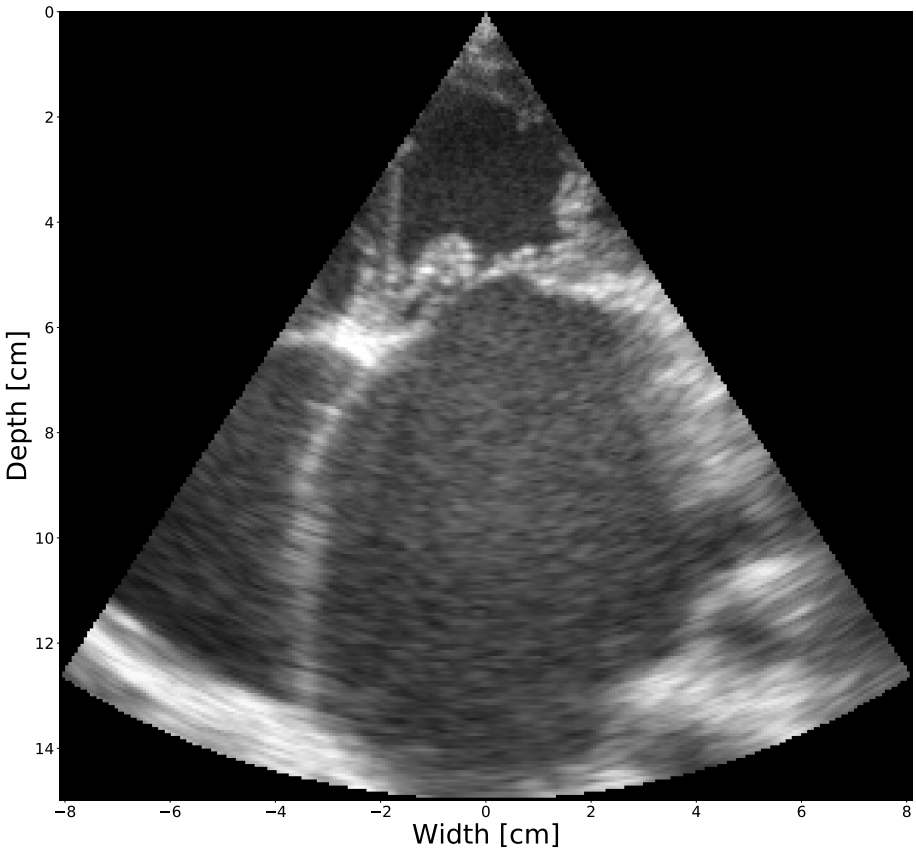



(b) Ground truth

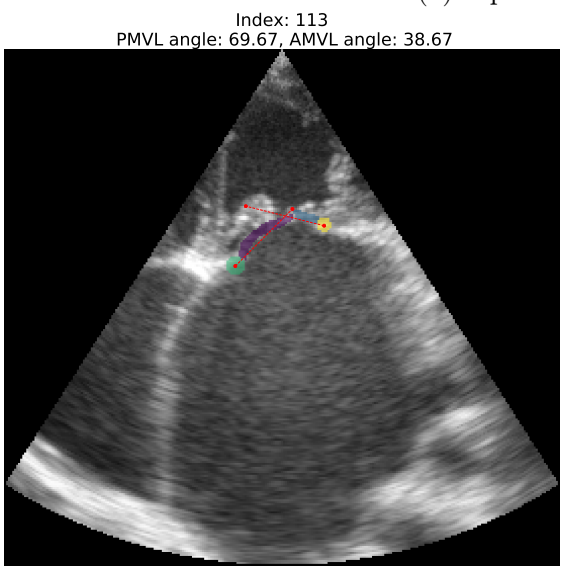



(c) Prediction

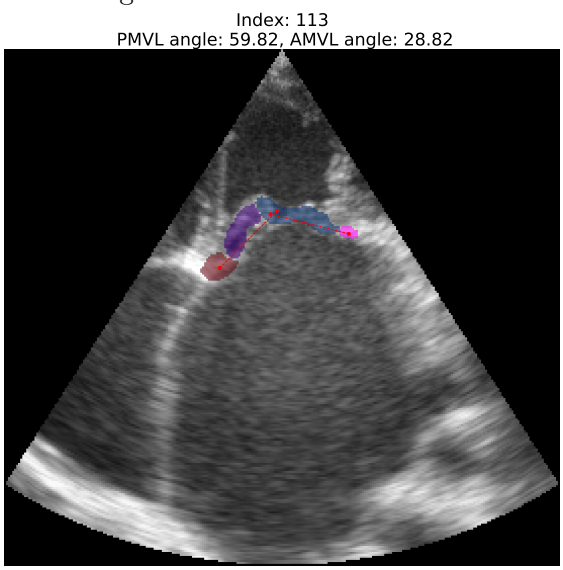
Figure 61: Angle estimation performed on the segmentations produced by the U-Net Auto-R network, sample with index 66 from the test set (a), where the estimated angles on the ground truth segmentations (b) and predicted segmentations (c) are similar.




(a) Input B-mode image

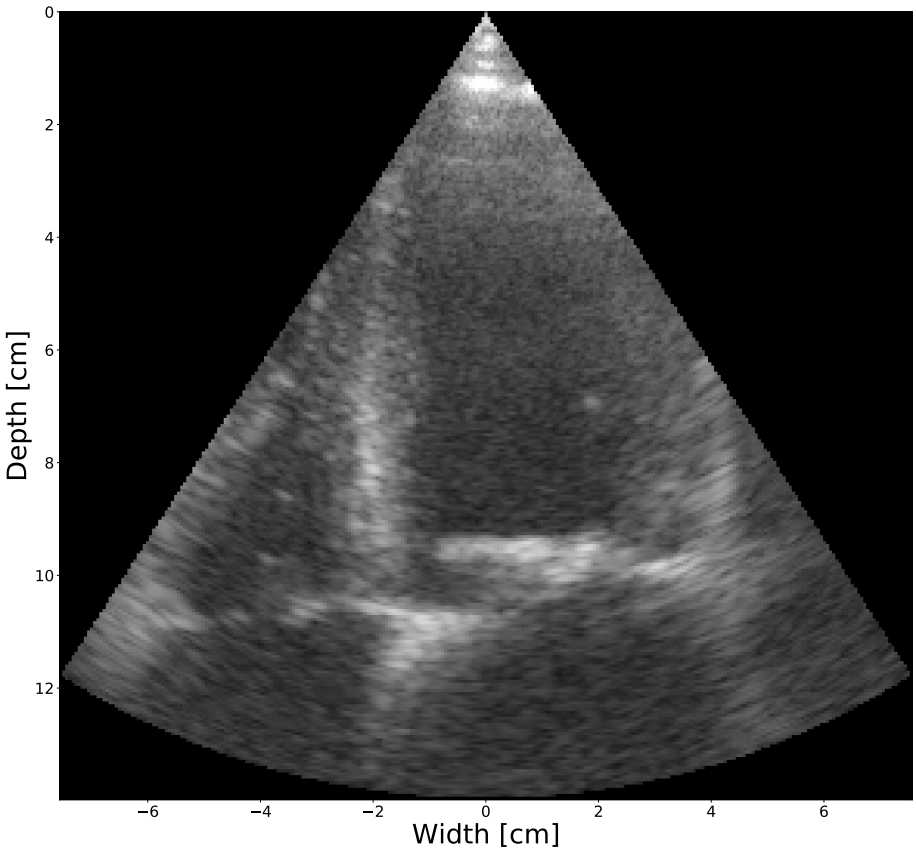



(b) Ground truth

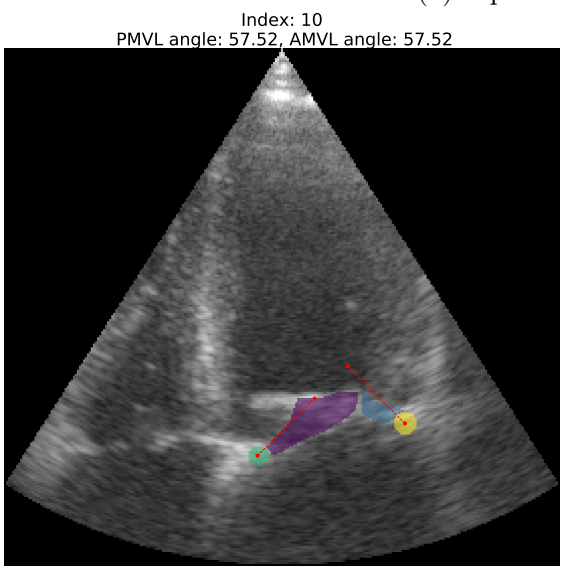



(c) Prediction

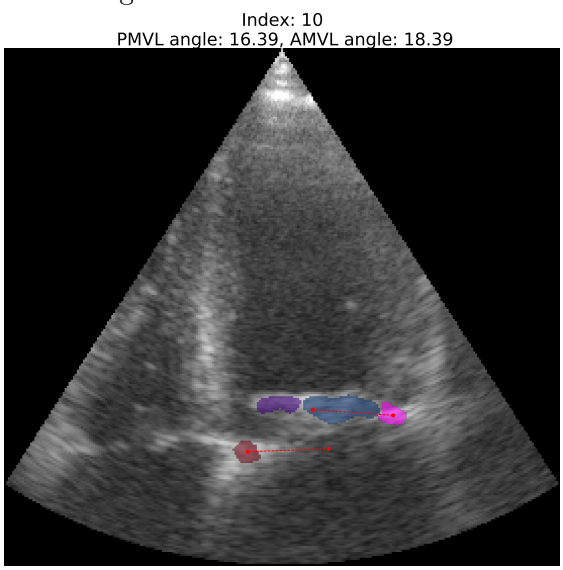
Figure 62: Angle estimation performed on the segmentations produced by the U-Net Auto-R network, sample with index 113 from the test set (a) with a median difference between the estimated angles of the ground truth segmentations (b) and predicted segmentations (c).




(a) Input B-mode image




(b) Ground truth




(c) Prediction

Figure 63: Angle estimation performed on the segmentations produced by the Auto-R network, sample with index 10 from the test set (a), where the estimated angles on the ground truth segmentations (b) and predicted segmentations (c) are far from each other.

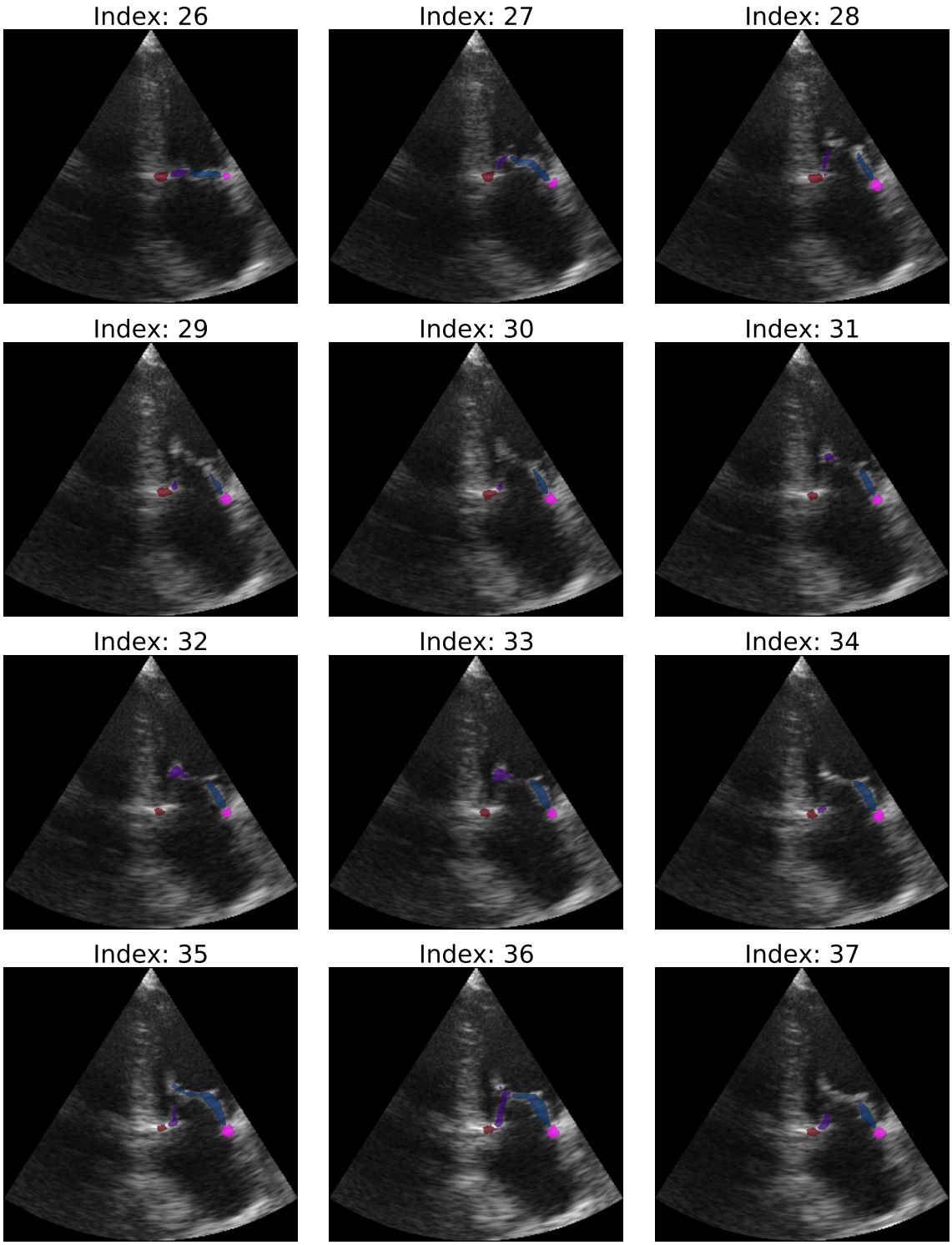
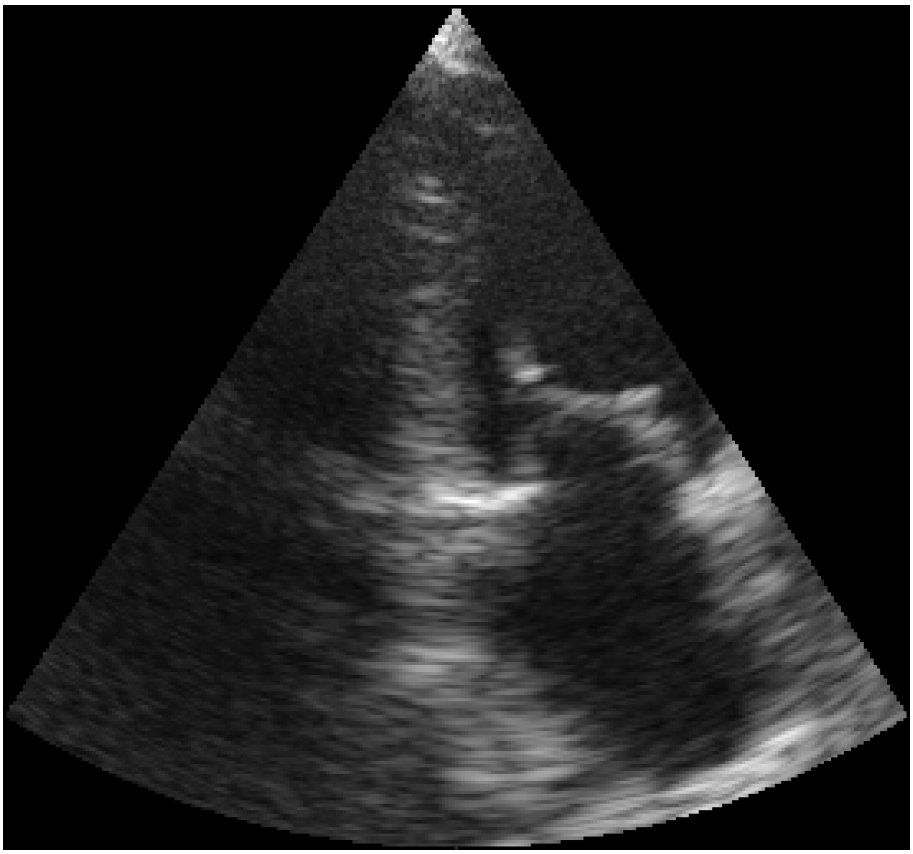
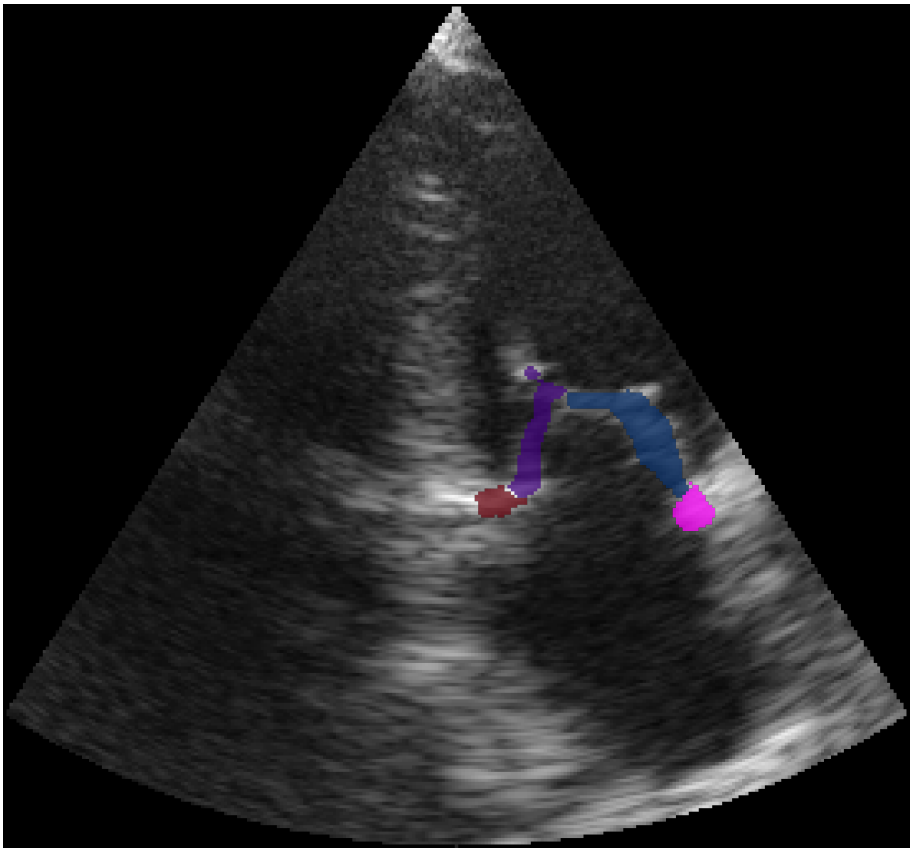



Figure 64: A selection of frames from DICOM-file number 5 in the test set. Each image is a combination of the input B-mode image with the predicted segmentations of the valve apparatus by U-Net Auto-R layered on top. In this selection, the valve starts closed (index 26), reaching maximum opening at approximately index 32, and then starts to close.



(a) Input B-mode image, index 36



(b) B-mode image with segmentations layered on top

Figure 65: Outtake from figure 64, frame 36, showing an incorrect segmentation by the network U-Net Auto-R. An artifact has be chosen as the correct anterior leaflet and annulus segmentations.

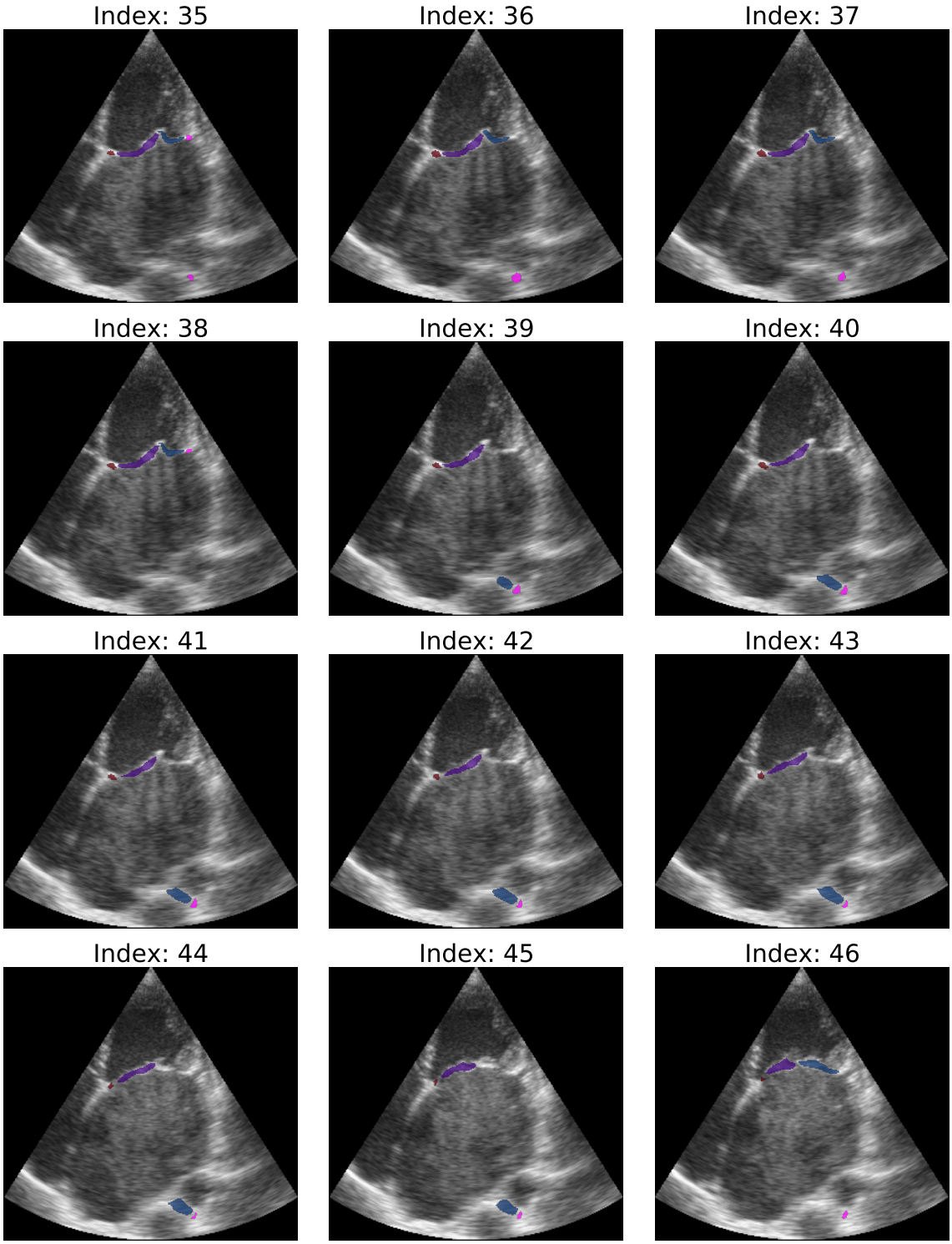
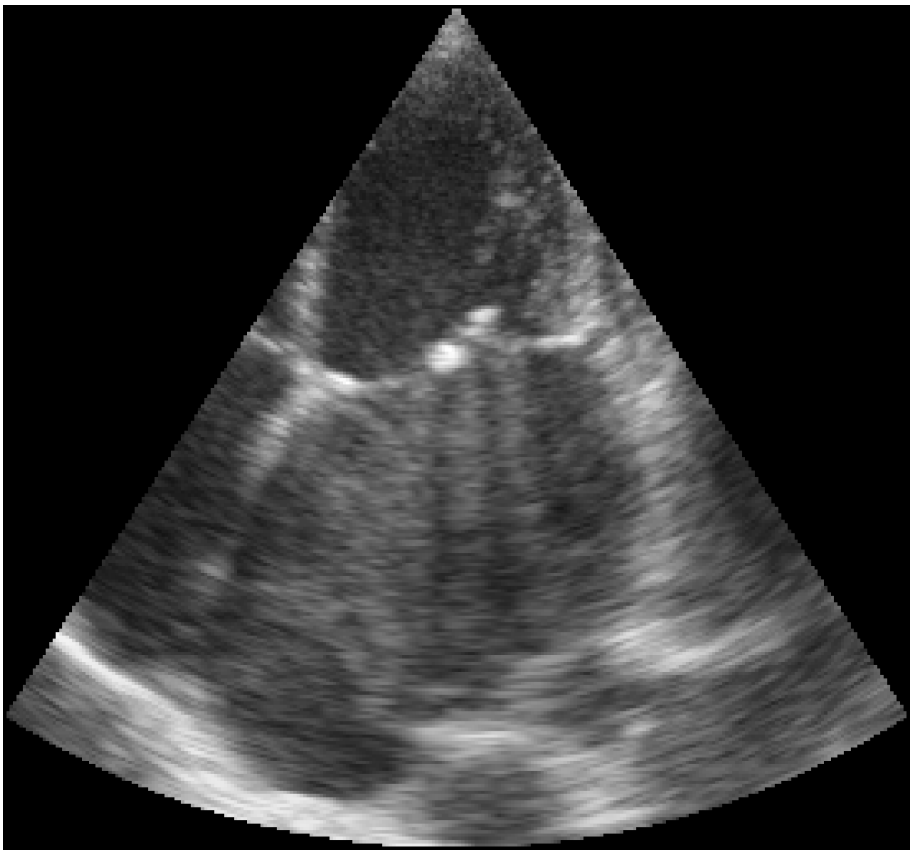
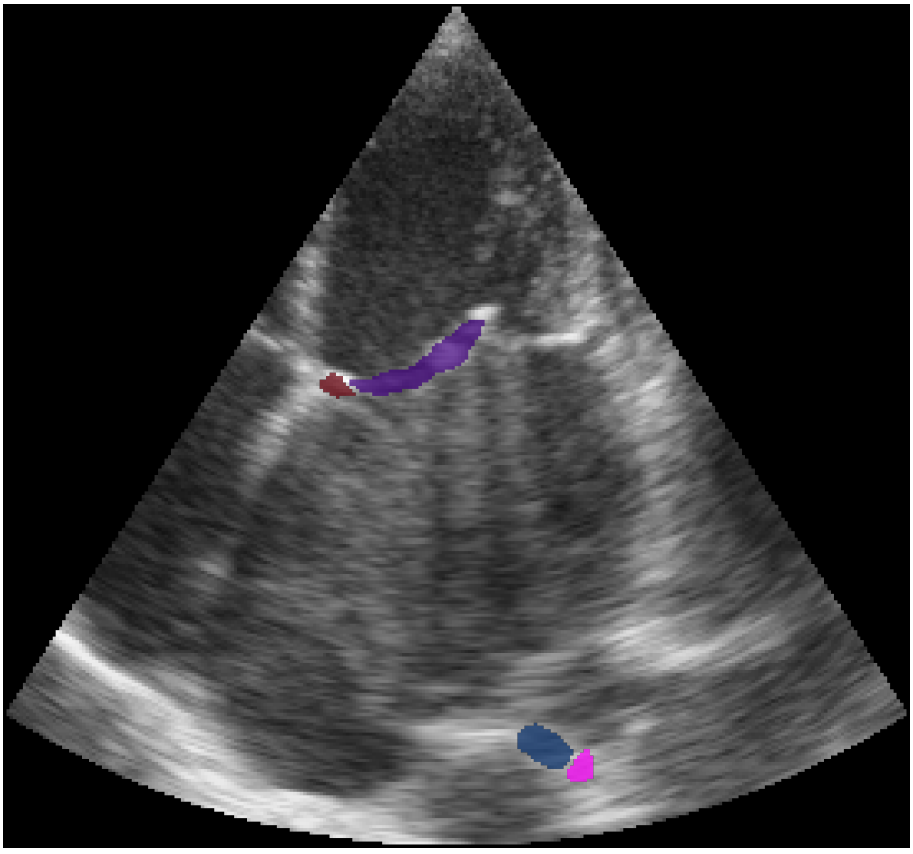



Figure 66: A selection of frames from DICOM-file number 35 in the test set. Each image is a combination of the input B-mode image with the predicted segmentations of the valve apparatus by U-Net Auto-R layered on top. In this selection, the valve is closed the whole time.



(a) Input B-mode image, index 35



(b) B-mode image with segmentations layered on top

Figure 67: Outtake from figure 66, frame 39, showing an incorrect segmentation by the network U-Net Auto-R. An artifact has be chosen as the correct anterior leaflet and annulus segmentations.

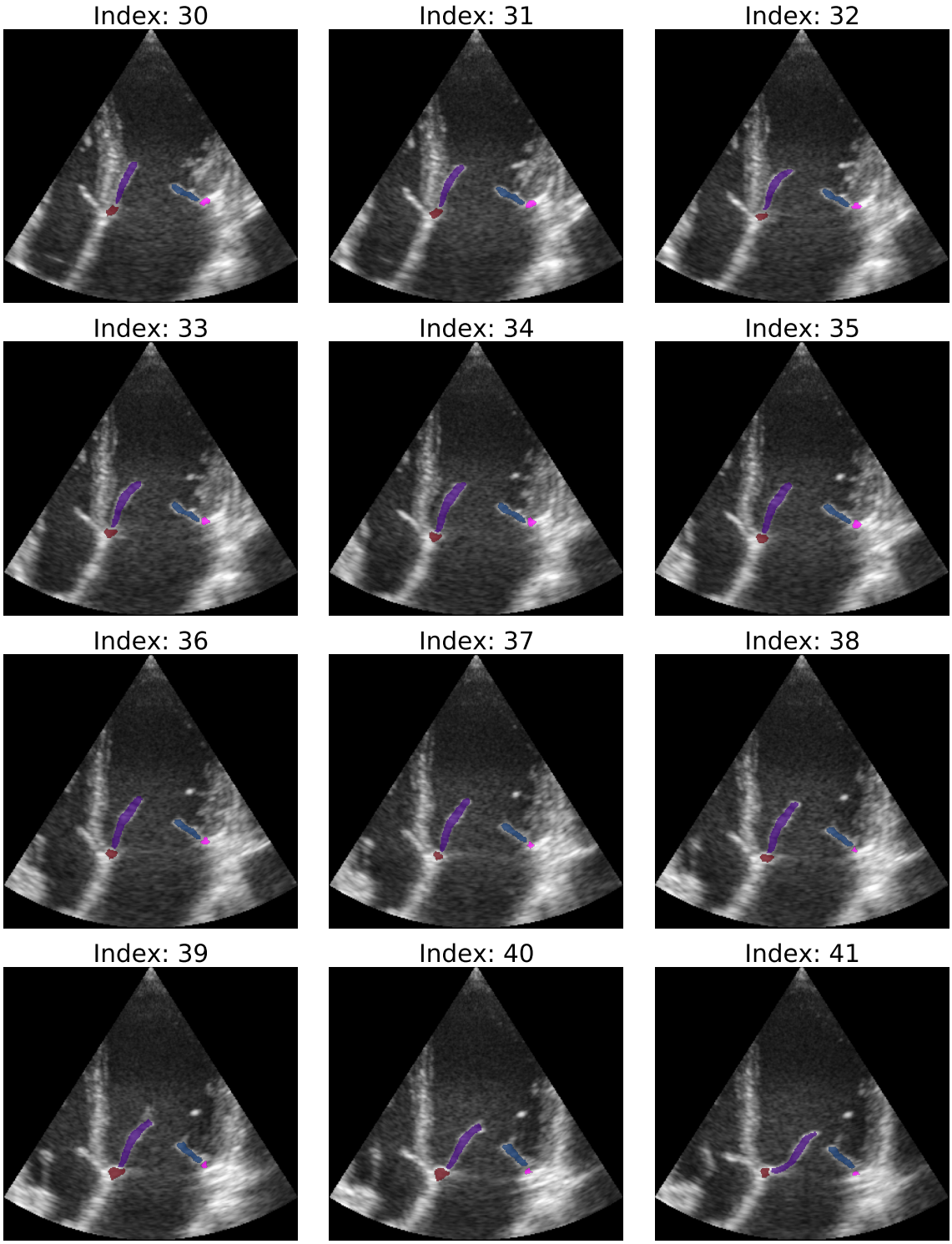



Figure 68: A selection of frames from DICOM-file number 7 in the test set. Each image is a combination of the input B-mode image with the predicted segmentations of the valve apparatus by U-Net Auto-R layered on top. In this selection, the valve starts opened (index 30) and starts to close as time moves.

## 4.4 Context Comparison

This section summarizes the best model results using only the valve, U-Net OV-C, and the best model using the auto-generated segmentations, U-Net Auto-R. Table 12 shows the DICE scores relating to the valve apparatus for both models. Table 13 summarizes the performance in terms of the annulus metric, and table 14 summarizes the performance in terms of the angle estimation metric.

Table 12: List of the DICE scores relating to the valve apparatus for the U-Net OV-C and U-Net Auto-R networks. The highest score between the two models for each measurement of each class are highlighted.

| Class | U-Net OV-C | | | | U-Net Auto-R | | | |
|---|---|---|---|---|---|---|---|---|
| | Min | Max | Mean | Median | Min | Max | Mean | Median |
| PMVL | 0.000 | **0.906** | **0.607** | 0.691 | 0.000 | 0.898 | 0.589 | **0.693** |
| AMVL | 0.000 | **0.901** | **0.641** | 0.696 | 0.000 | 0.898 | 0.631 | **0.700** |
| PA | 0.000 | **0.945** | **0.395** | **0.423** | 0.000 | 0.909 | 0.375 | 0.398 |
| AA | 0.000 | **0.938** | **0.477** | **0.548** | 0.000 | 0.879 | 0.399 | 0.438 |

Table 13: List of the annulus distance metric for the U-Net OV-C and U-Net Auto-R networks. The best score between the two models for each measurement of each class are highlighted. The numbers are given in centimeters.

| Class | U-Net OV-C | | | | U-Net Auto-R | | | |
|---|---|---|---|---|---|---|---|---|
| | Min | Max | Mean | Median | Min | Max | Mean | Median |
| PA | 0.020 | **1.750** | **0.436** | **0.339** | **0.007** | 3.181 | 0.492 | 0.378 |
| AA | **0.019** | 6.841 | **0.416** | **0.273** | 0.037 | **6.804** | 0.481 | 0.297 |

Table 14: List of the estimated angles metric for the U-Net OV-C and U-Net Auto-R networks. The best score between the two models for each measurement of each class are highlighted. The numbers are given in degrees.

| Class | U-Net OV-C | | | | U-Net Auto-R | | | |
|---|---|---|---|---|---|---|---|---|
| | Min | Max | Mean | Median | Min | Max | Mean | Median |
| PMVL | 0.050 | **88.24** | **15.12** | **6.73** | **0.010** | 94.00 | 16.74 | 7.43 |
| AMVL | 0.11 | **105.0054** | 20.42 | 10.58 | **0.012** | 137.16 | **20.25** | **10.30** |

## 4.5 Python Package

During development, a custom Python package, called ValveSeg, has been created for the project. The package is a collection of functions and scripts used for data extraction, model creation, and pipeline engineering. This has made the process of pre-and post-processing and model training easy to modify. The package also makes it easier to continue the work on project in the future.

# 5 Discussion

## 5.1 Data Set Discussion

Limitations in the design of the annotation tool used to create the valve annotations have resulted in a limited extraction of data. The current annotation tool bundles both leaflets into one class and both annulus points as one class. This is not a problem when there is only one annotation for each object since their relative position can be used to determine their position (anterior vs. posterior). However, if there are more or fewer annotations on an image, it is much harder to decide which is which. If there are three annotations with the valve class, it is not trivial to determine which annotations belong to which leaflet without going through every file manually. In future development of the annotation tool, it would be beneficial to implement one class for each object one wishes to annotate to ease the extraction process. Another possible improvement of the annotation tool would be to save more parameters that classify the images. More information about the data makes it easier to analyze the performance of a model and tailor the data set for different purposes. For example, adding a view classification parameter during the annotation process would be beneficial because the data set could be split into subsets containing only one view and then evaluate each subset's performance. Another helpful parameter would be an indicator for an open or closed valve. This plays into a central weakness of the data set at hand.

From all the different configurations of the U-Net architecture, it is apparent that every model struggles to segment out the valve when the leaflets are pointing up into the left ventricle. In figure 29, we can see an example where the valve is open and a subpar segmentation performed by the network. There are several possible explanations for this. The most probable reason is an imbalance in the data set. The data set is almost exclusively composed of patients with some form of mitral valvular disease. Some of these diseases restrict the valve's opening, resulting in a valve that is not entirely opened during diastole. Figures 23e - 23f show an example where a healthy valve would be fully open, but is only partially opened due to deformation of the valve. In the whole data set, there are approximately 25% fully opened valves. An imbalance in a data set often leads to a bias, for closed valves in this case. The bias occurs because the neurons in the network will be tuned more towards the imbalance (closed valve) in the data set.

To combat this problem, we need more data. Either by annotating more data or augmenting the open valve annotations we already have. Both methods consume considerable time. Augmentation would be the easiest solution, but more data could also solve another problem with the data set. As mentioned above, the data set contains almost no healthy patients. Most of the screened patients were late in the process of valve deformation. As a result, the models trained with this data set will have a bias towards deformed leaflets.

One of the goals of this project is to help detection of valve deformation at an early stage in the deformation process. To accomplish this, the produced results indicate that the model needs to be trained on a wide variety of valve samples, both deformed valves, and healthy valves. Adding more unique annotated data would increase the versatility of the models.

## 5.2 Only Valve Apparatus Discussion

Two main variations of the U-Net network using only the valve apparatus have been created; one using boolean values for the ground truth segmentation masks (U-Net OV-B) and one using integer values for the ground truth segmentation masks (U-Net OV-C). Both variations had problems with convergence. Some configurations of the models output a constant value for the loss during training and output zero predictions on the test data. In practice, this means that the network does not learn anything. There are several probable reasons for this. One possible reason could be related to the fact that the ground truth segmentations of the valve apparatus contain a large majority of zeros and only a small handful of ones. Small objects are often a problem for CNNs. This comes from the nature of the convolutional operations used. The convolutional kernel used by the max pooling layers will shrink the image during the compression layer of the network. Small objects will therefore have less impact on the network for each compression stage. The skip-connections in the U-Net architecture aim to solve this problem, but they are not always successful.

The convergence problem is also present in the configurations where the background class has been added. The background layer contains an abundance of ones, but the result is the same, a dying network. This could be caused by the use of the ReLU function as the activation function for the convolutional layers. The ReLU function has a known problem called the dying ReLU problem, which describes a state where ReLU neurons become inactive and output 0 regardless of the input. Lu et al. [32] shows that a network with infinite depth using the ReLU activation function will eventually die. If the valve annotations used to create the background class are enlarged above a certain threshold before inverting the mask, the networks converge. This method is used for the U-Net OV-B network, as a transformation of the masks is the only method that can be used to change the boolean masks. It is difficult to determine the direct cause of this improvement, but the enlargement changes the balance in the data. Before enlargement, every pixel is assigned a class. After enlargement, there are a handful of pixels without any class associated to them, thus changing the activation behavior of the neurons around the valve. These extra zeros in the background layer have similar functionality to the dropout layers used in the model structure since they act as a reset of the activation of the neurons. The U-Net OV-C network solves this problem by changing the integer values from 0 and 1, to 1 and 255. This ensures that all the neurons always get a small activation, thus hindering the death of the ReLU activated neurons.

The DICE scores for the two models, shown in table 2, indicate that the U-Net OV-C model has the best overall performance, but the difference between the models is not very substantial. Figures 25 and 26 show predictions by the U-Net OV-B and U-Net OV-C networks on the same test sample, respectively. The predictions are reasonably similar. This example mirrors the general performance of the two models. The raw output from the U-Net OV-C network gives higher confidence for some classes. The raw output in figure 25 shows that the AMVL and PA channels contain lower confidence values compared to the raw output in figure 26. This difference in performance is most likely related to the degree of enlargement used for the U-Net OV-B network since it is the main difference between the two models.

The enlargement of the valve segmentations before creating the background class used in the U-Net OV-B model uses $\sigma = 2$ in the Gaussian kernel used for enlarge-

ment. This value is empirically chosen to be as low as possible while still maintaining convergence for the network. This value may not be ideal and is a possible source for error. The convergence of the network depends on the chosen $\sigma$ value, making training more challenging for the U-Net OV-B network compared to the U-Net OV-C network. In addition, there are no indications that the U-Net OV-B model gives any benefit over the U-Net OV-C network. During development, the U-Net OV-B was therefore abandoned in favor of the U-Net OV-C network.

## 5.3 Auto-Generated Segmentations Discussion

The U-Net structure has also been trained with auto-generated segmentation of the left ventricle, myocardium, and left atrium, in addition to the manual valve segmentations. Two pre-trained neural networks have created these segmentations. This is done to explore the possible benefits of giving the U-Net more context, with the hope being that the network will perform better. Two variations have been trained, one using the original auto-generated segmentations (U-Net Auto) and one using reassigned versions of the auto-generated segmentations (U-Net Auto-R).

Figure 43 and 47 show predictions performed by the U-Net Auto network on two data samples from the test set with index 51 and 116, respectively. Figure 44 and 48 show predictions performed by the U-Net Auto-R network on the same data samples. These examples illustrate the general performance of the models. The predictions of the left ventricle, myocardium and left atrium by the U-Net Auto network are not reliable approximately half of the test set and reasonable for the other half. The U-Net Auto-R network produces reasonable predictions for most cases in the test set.

In addition, the raw network output is more stable for the U-Net Auto-R network. Two examples of the raw output from the networks are shown in figure 45 and 49 for the U-Net Auto, and in figure 46 and 50 for the U-Net Auto-R. The raw output of the U-Net Auto has more artifacts in the raw output, which most likely results from the poor state of some of the original auto-generated segmentations. In the raw output of the U-Net Auto-R, we can also see some artifacts, but not of the extent as the U-Net Auto. During development, the U-Net Auto was therefore abandoned in favor of the U-Net Auto-R network.

Despite all the pre-processing steps, we see that some of the auto-generated ground truth segmentations are still not ideal. In some cases, there is almost no myocardium or left ventricle annotation, as shown in figure 14. Another problem with the automatic annotations is that some of the annotations of the left atrium, like the example shown in figure 13 are not correct. This will affect the performance of the network and should be addressed in future development.

One solution could be the removal of these samples from the data set. However, given the small size of the data set, this would not be ideal. Table 8 shows the DICE scores for the valve apparatus performed by the U-Net Auto-R network trained on the original data set and a cleaned version. Augmentation has been applied to both data sets to increase the number of data samples to approximately 4000 images for both networks. We can see from table 8 that the network using the original data set outperforms the cleaned version. The cleaned data set is created by removing 88 samples with subpar auto-generated segmentations from the original data set.

However, table 9 shows the DICE scores for the auto-generated segmentations, and for these classes, the network using the cleaned data set outperforms the original data set. These results show that the removal of 88 valve samples has a more considerable impact on the DICE score for the valves than the removal of 88 subpar auto-generated segmentations.

Another more promising solution for the data set could be to compare all the auto-generated segmentations produced from the same recordings. In figure 24 we can see that the segmentations for two of the images are good, but one of them is missing the annotation of the left ventricle. All three input B-mode images are almost identical, which means that the good segmentations could be transferred to the image with the wrong annotation. These segmentations will not be a perfect fit, but the overall fit will be better. Calculating the median segmentation of all the annotated frames could be a good place to start.

Finally, the most obvious solution for the problem is the addition of more unique data samples. These results comparing the cleaned data set to the original show that removing only 88 have a considerable effect on the performance. Given the extensive effect these removed samples have, it is reasonable to assume that more data will increase the model's performance.

## 5.4 Context Inclusion Discussion

When a CNN is tasked to classify, the amount of information the network needs to learn is proportional to the number of classes it is tasked to detect. This follows the logic that more classes correlate to a more nuanced relationship between the features in input data. As a result, the network needs to fine-tune the feature maps to a vaster extent compared to a network with the same base structure tasked to detect fewer classes. More fine-tuning requires more training and more data. This can be seen in the raw outputs of the U-Net OV-C and U-Net Auto-R.

### 5.4.1 Augmentation Impact

When both networks are trained on the data set without augmentation the raw output from the U-Net Auto-R network contains more artifacts and outputs lower confidence in its predictions compared to the U-Net OV-C network trained on the same data. However, this is not the case after augmentation. As an example, we look at data sample with index 51 from the test set for all the four instances.

Figure 26 and 30 show example predictions from the U-Net OV-C network using no augmentations and the U-Net OV-C using all augmentation methods, respectively. The raw output in the figures are almost identical and are similar to the ground truth segmentations, showing that the augmentation did not have a large impact on the performance.

Table 3 shows a summary of the DICE scores for the U-Net OV-C network. We can see from the table that most of the augmentation methods have a small improvement of a couple of percent on the DICE score for the leaflets, and some methods actually decrease the performance for the leaflets. However, the performance of the annulus points increases for every augmentation method. The increase in annulus accuracy is generally larger than the decrease in leaflet accuracy. This indicates that the augmentation methods and values used for this variation of the network are not

optimal. When augmentation is applied, one aims to create new realistic samples, but whether the new samples are realistic enough is hard to determine. Further investigation into the augmentation pipeline is therefore needed.

The raw output from the U-Net Auto-R using no augmentation and the U-Net Auto-R using all augmentation methods are shown in figures 46 and 52, respectively. There is a notable difference between these two examples. The raw output channels in figure 46 have imprecise predictions, while the raw output channels in figure 52 are more concise when compared to the ground truth channels; thus showing the large impact the augmentation methods have on the U-Net Auto-R network.

Table 6 shows a summary of the DICE scores for the valve apparatus for the different augmentation runs of the U-Net Auto-R network. In this case, we can see that the augmentations have a much larger impact on the scores, and that the network using every augmentation at the same time shows a substantial improvement compared to the other configurations. These results back up the assertion that the U-Net Auto-R network needs more data to perform well when compared to the U-Net OV-C network because of the difference in the number of classes. However, the use of augmentation can not be directly compared to the addition of more unique data, and these results suggest that more unique data is needed.

These results make it difficult to determine whether the addition of context can improve the performance of the valve segmentations. The best performance with and without context are summarized in section 4.4. Table 12 shows the DICE scores for both models. The mean scores are higher for the U-Net OV-C network for all four classes. However, the median scores for the leaflets are higher for the U-Net Auto-R network, but only by a small margin. The scores indicate that the U-Net OV-C model has the best performance, but the difference between the networks is not huge. The results indicate that a network using context has the possibility to perform on par with the same network without context, given that enough data is available.

### 5.4.2 Feature Extraction

From the results of the feature extraction metrics, shown in section 4.2.3 and 4.3.4, we can observe some of the weaknesses of the models. From tables 4 and 10 we can see that both models have a maximum difference of over 6 cm between the ground truth center points and the predicted center points. This is a huge error and is the result of a weakness in the post-processing method. The post-processing of the raw output from the networks assumes that the largest region in each channel is the correct segmentation. This is not always the case. An artifact can sometimes be larger than the valve or annulus segmentation, thus resulting in incorrect predictions as shown in figures 34 and 60 produced by the U-Net OV-C and U-Net Auto-R, respectively. In these extreme cases, the resulting difference in distance for the center points becomes enormous, affecting the average performance of the annulus metric. Because of this, the median score for the distance is the best metric to use when comparing the general performance of the two models. The same problem occurs for the angle estimation differences. The maximum difference between some of the estimated angles are huge and also results from the post-processing method's weakness.

When it comes to the annulus metric, we can see that both models perform similarly

in median distance difference, summarized in table 13. From the table, we can see that the median difference for the posterior annulus is 0.339 cm for the U-Net OV-C and 0.378 cm for the U-Net Auto-R. The difference between the two models is 0.039 cm in favor of the U-Net OV-C. The U-Net OV-C produces a media difference of 0.273 cm for the anterior annulus, and the U-Net Auto-R produces a median difference of 0.297. The U-Net OV-C outperforms the U-Net Auto-R by 0.024 cm.

The annulus points marked as ground truths by the clinicians are the same points used for quantitative assessment of longitudinal ventricular function [33]. When performing the assessment, the clinician sets the annulus point manually. To be able to do this, one needs years of experience to get accurate results. If this assessment could be automated, it would reduce the time and resources of experienced clinicians by enabling less experienced clinicians to perform the assessment. In further developing the project, one could investigate if the networks' predictions are accurate enough to automate this assessment.

One problem we can observe for both the models is related to the chordae tendineae, which are responsible for the movement of the leaflets. Both the U-Net OV-C and U-Net Auto-R networks sometimes segment out parts of the chordae tendineae as a part of the leaflet. Examples of this behavior are shown in figure 27 for the U-Net OV-C, and in figure 59 for the U-Net Auto-R network. There are several possible explanations for these errors. One of them is that the networks have not been trained with enough data to teach them that the chordae tendineae is not a part of the valve. It is also possible that some of the ground truth annotations have pieces of the chordae tendineae in the annotations. Multiple clinicians have created the valve annotations, and there is a possibility that they have differing opinions on where the leaflets stop and chordae tendineae start.

Another possible source of error could be that there are not enough data samples in the data set where the chordae tendineae are visible. The smaller a data set is, the more exposed it is to these kinds of possible imbalances. The inclusion of the chordae tendineae as a part of the valve will affect the angle estimation drastically for the examples where it appears. This is evident by looking at the example in figure 37, where the estimated angle of the posterior leaflet is too large due to segmentation of the chordae tendineae as a part of the leaflet.

The two models have similar scores regarding the difference between the predicted estimated angle and the ground truth. Table 14 shows the performance for the two models. We can see that the estimated angles from the U-Net OV-C predictions are 6.73 degrees off compared to the ground truth for the posterior leaflet and 10.58 degrees for the anterior leaflet. The U-Net Auto-R predictions result in a media distance difference of 7.43 degrees for the posterior leaflet and 10.30 degrees for the anterior leaflet. In this case, the U-Net Auto-R outperforms the U-Net OV-C by a small margin for the posterior leaflet. Thus showing that the U-Net Auto-R does not always underperform for the given data set.

The estimation of the valve angles is a feature that could be useful to help clinicians when assessing a patient. The angles between the leaflets and the annulus plane could be used to estimate blood flow between the left atrium and the left ventricle. Small angles when the valve is open indicate a restricted opening of the valve and limit the amount of blood that can flood between the chambers per heart cycle.

This could, with enough tests and automation, give clinicians an indication about the state of the blood flow and potentially help the detection of irregularities.

The angle estimation presented in this thesis functions as a proof-of-concept, as the development started towards the end of the project. There are most likely several edge cases that need to be considered to make the implementation more reliable. A possible improvement of the angle estimation could be to extract the centerline of each leaflet prediction before estimating the angle. This could be done by skeletonization, a process where the segmentation is gradually shrunken until one centerline remains.

The estimation of the angles is subject to uncertainty, as it is only an estimate. This is apparent when we look at examples of the estimated angles. Figure 35 produced by the U-Net OV-C network. The difference between the estimation on the ground truth and the predictions are about 1 degree for both the estimated leaflet angles. These numbers appear excellent. However, when we look at the images, the difference seems larger. This comes in part from the uncertainty introduced by the annulus center points. The estimated angles are relative to the annulus plane, and the annulus plane produced by the ground truth is not identical to that of the predicted annulus plane.

The implemented solution only extracts the estimated angle and does not use it for any other type of measurement. The accuracy of the angle is only compared to the estimated angle from the ground truth segmentations. Whether or not these estimations are good enough to use for any clinical purposes is difficult to ascertain without further investigation. Further development of the method is therefore needed to assess its possible application for clinical purposes properly.

### 5.4.3 Sequence Testing

The sequence testing results, shown in section 4.2.4 and 4.3.5, show that both the networks struggle to segment out the valve when it is open. This is most apparent when the opening between the two valves is small or when the leaflets point straight up into the left ventricle. This problem can be observed in the test sequence using DICOM-file 5 from the test set. Figure 38 shows a selection of frames from the sequence with the segmentations produced by the U-Net OV-C network, and figure 39 shows one highlighted frame from the selection. Figure 64 shows the same selection from DICOM-file 5 performed by the U-Net Auto-R network. The results from both the models are similar, almost no segmentation for leaflets when the leaflets are pointing up into the left ventricular. These results echo the findings discussed in section 5.1, namely that there are not enough examples in the training set where the valve is open. The models therefore develop a bias for closed valves.

This bias is also manifested as merging of the two leaflets when they are supposed to be open. Figure 40 shows a selection of frames and the resulting predictions performed by the U-Net OV-C network on DICOM-file 12 from the test set. In this example, we can see that for some of the frames, the predictions of the leaflets merge together to form a closed valve. The valve in the sequence reaches maximum opening at approximately index 32, and then starts to close. At index 34 the network segments out a closed valve, but at index 37 it segments out an open valve. Frames 34 and 36 are shown in full size in figure 41. The segmentations of frame 34 are weak and show a weakness of the model.

The same weakness can be found in the U-Net Auto-R model. In the selection of frames from DICOM-file 5 we can see that the network also connects the leaflets when the valve is supposed to be open. This is highlighted in frame 36, shown in figure 65, where the segmentation of the anterior leaflet is extended so the two leaflets form a closed valve. Given that this problem occurs for both models, it enforces the idea that the imbalance in the data set affects the performance and that more data is needed.

The problem with the post-processing method is clearly visible in some of the test sequences, for example in the sequence test from DICOM-file number 35 shown in figure 66 produced by the U-Net Auto-R network. In the sequence, we can see that for some of the frames the anterior leaflet and annulus point are marked at the bottom of the image, while the posterior leaflet and annulus point looks to be correct. Frame number 39 from the same DICOM-file is shown in figure 67 showing an example where this is the case. This is a result of the post-processing method used and is something that needs to be addressed in the future by modifying the post-processing pipeline. One possible improvement could be to compare the largest regions in each channel after post-processing. We know that the valve leaflets will always be in close proximity to each other, and that the annulus points will not be far from their associated leaflet. We can therefore compare the positions of the largest regions to each other, and then choose the regions that are the closest to each other.

The problems outlined above are not present in all the test recordings. Both models perform well when the recording quality is good and show tendencies to perform better when the valve is more in focus. Figure 42 shows a selection of frames from DICOM-file 8 in the test set and the predictions performed by the U-Net OV-C network. In this recording, the image quality is good and the valve occupies a relatively large section of the image. The predictions produced by the network are excellent. Frame 31 in the selection shows a good segmentation of the posterior leaflet, which points upward into the left ventricle and lies close to the myocardium.

This is the case for the U-Net Auto-R network as well. Figure 68 shows a similar selection of frames from DICOM-file 7. DICOM-files 7 and 8 are different recordings from the same patient, and we can see that both networks produce excellent segmentations for recordings of this kind of quality. One of the aspects of both these recordings is that the presence of the chordae tendineae are not substantial. As we have seen from previous examples, for example figure 37 and 59, the network sometimes segments out parts of the chordae tendineae as a part of the valve. This is an indication that there are not enough samples in the data set where the chordae tendineae are present, which once again ascertains the need for more unique data samples.

### 5.4.4 Multipurpose Network Advantages

In general, we can see that the U-Net OV-C network outperforms the U-Net Auto-R network with a small margin on each presented metric. Even though the performance did not increase, there are still several benefits to having one network that can do all the different segmentations at the same time. One aspect is time. A network doing everything at the same time will use less time than two networks of the same size doing two different segmentation tasks separately. Another benefit of one network is the ease of use. Implementation of one network into other applications

is more practical compared to several networks. Also, dealing with several networks requires fusion of the output data from the different networks, making the process more complicated.

Having all the different classes available is also advantageous for post-processing purposes. The mitral valve will always be above the left atrium and below the myocardium and left ventricle. A possible post-processing pipeline could extract the center point of each structure, compare the predicted regions for the valve apparatus, and choose the region that best fits these center points. The current post-processing pipeline fits well when dealing with large structures and can be used for the large structures before the center point extraction.

## 5.5 Future work

Going forward, there are many different possible steps one can take to improve the overall performance of the models. As mentioned earlier in the discussion, the addition of more data is the first logical next step. More data have the potential to improve both the U-Net OV-C and U-Net Auto-R networks. The general problem of the data set at hand is the imbalance of closed and open valves. Then new data should also add data from healthy patients because it will strengthen the robustness of the network. This can easily be done using the custom Python package mentioned in section 4.5.

The need for more data became apparent early in the project, but the acquisition of more data requires time. As a result, time was put into developing the custom Python package to save time in the future when more data is available. The package is designed to extract new data and train the network with this data quickly. The package also includes the two best models with and without context (U-Net OV-C and U-Net Auto-R) and a script that can be used to run the models and get predictions on new data. As mentioned earlier in the discussion, further development of the angle estimation is needed. The source code for the angle estimation is included in the package, thus making it easy to continue developing the metric.

Every model proposed in this thesis uses the same base structure of the U-Net because of its track record concerning medical image segmentation tasks. However, the base U-Net is a time-invariant system. The predictions are made independently. Since the mitral valve moves as time passes, several other network structures could be interesting to have a close look at in the future. An example of such a structure is the Long Short-Term Memory (LSTM) architecture. LSTM is an artificial recurrent neural network (RNN) that takes advantage of feedback connections. This means that the network can process data sequences, for example videos, and "remember" predictions from the previous data. Tasks with a solid temporal link have shown promising results using LSTM networks [34]. Given the temporal aspect of the motion of the valves in the heart, it could be interesting to investigate the possibilities of segmentation using a variation of the LSTM architecture. This would, however, require more annotations. There are only approximately four annotated frames for each recording in the current data set. The current data set is not suited well for the training of a LSTM network; more annotations would therefore be necessary.

# 6 Conclusion

In this master thesis, a variation of the deep neural network architecture U-Net has been implemented and trained to segment out the mitral valve in the heart. The networks are trained using ground truth segmentations of the valve apparatus. These segmentations have been created by trained clinicians using a custom annotation tool tailored for the task. The clinicians have annotated the valve leaflets and the points where the leaflets are anchored to the fibrous annulus. The data set used for training are composed of mostly sick patients with differing degrees of deformation of the valve. Investigation into the impact of context in the form of addition of other cardiac structures is performed. Segmentations of the left ventricle, myocardium, and left atrium are automatically generated by two pre-trained neural networks. Two main variations of the model are trained and compared, one trained with segmentations of the valve (U-Net OV-C) and one using the auto-generated segmentation in addition to the valve apparatus (U-Net Auto-R). Training using the initial data set results in a substantial in performance in favour of the U-Net OV-C network. However, analysis of the data set indicates that there is not enough data for the U-Net Auto-R network to be competitive. Artificial addition of data samples trough augmentation results in similar performance for the models. When augmentation is used, the U-Net OV-C model produces segmentations with a DICE score accuracy of 0.691, 0.696, 0.423, and 0.548 for the posterior leaflet, anterior leaflet, posterior annulus, and anterior annulus, respectively. The accuracy of the U-Net Auto-R model given in the same order is 0.693, 0.700, 0.398, and 0.438.

Two features of the valve apparatus are extracted from the predictions produced by the networks. The center points of the fibrous annulus points and an estimation of the angle between the each leaflet and the plane between the two annulus points. The two models have similar accuracy scores for the predicted annulus center point and estimated angle. The segmentations produced by the U-Net OV-C network results in a median error 3.39 mm for the posterior annulus and 2.73 mm for the anterior annulus. The U-Net Auto-R produces a median error of 3.78 mm and 2.97 mm for the posterior and anterior annulus, respectively. The same procedure is performed for the angle estimation, where the U-Net OV-C has a median error of 6.73 and 10.58 degrees for the posterior and anterior leaflet angles. The U-Net Auto-R has a median error of 7.43 and 10.30 degrees. These results show the potential of deformation analysis using deep learning. However, the angle estimation method proposed in this thesis is a proof-of-concept. The method shows promising results, but further investigation into its application potential is needed.

Given the small data set at hand, it is difficult to provide a solid conclusion to whether or not the addition of context of other structures of the heart help the U-Net structure to segment out the mitral valve. The results in this thesis indicates that the inclusion of context does not improve performance for this particular data set. However, the augmentation results indicates that the inclusion of the context does not affect the general performance in a substantial manner if enough data is provided during training of the network.

# References


[1] Haidong Wang et al. "Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the Global Burden of Disease Study 2015". In: *The Lancet* 388.10053 (2016), pp. 1459–1544. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(16)31012-1. URL: https://www.sciencedirect.com/science/article/pii/S0140673616310121.

[2] Henry C. McGill, C. Alex McMahan, and Samuel S. Gidding. "Preventing Heart Disease in the 21st Century". In: *Circulation* 117.9 (2008), pp. 1216–1227. DOI: 10.1161/CIRCULATIONAHA.107.717033. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCULATIONAHA.107.717033. URL: https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.107.717033.

[3] Martin J O'Donnell et al. "Global and regional effects of potentially modifiable risk factors associated with acute stroke in 32 countries (INTERSTROKE): a case-control study". In: *The Lancet* 388.10046 (2016), pp. 761–775. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(16)30506-2. URL: https://www.sciencedirect.com/science/article/pii/S0140673616305062.

[4] Spyros Makridakis. "The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms". In: *Futures* 90 (2017), pp. 46–60. ISSN: 0016-3287. DOI: https://doi.org/10.1016/j.futures.2017.03.006. URL: https://www.sciencedirect.com/science/article/pii/S0016328717300046.

[5] Ilya Kuzovkin et al. "Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex". In: *Communications Biology* 1.1 (Aug. 2018), p. 107. ISSN: 2399-3642. DOI: 10.1038/s42003-018-0110-y. URL: https://doi.org/10.1038/s42003-018-0110-y.

[6] Chen Chen et al. "Deep Learning for Cardiac Image Segmentation: A Review". In: *Frontiers in Cardiovascular Medicine* 7 (2020), p. 25. ISSN: 2297-055X. DOI: 10.3389/fcvm.2020.00025. URL: https://www.frontiersin.org/article/10.3389/fcvm.2020.00025.

[7] E. Costa et al. "Mitral Valve Leaflets Segmentation in Echocardiography using Convolutional Neural Networks*". In: *2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG)*. 2019, pp. 1–4. DOI: 10.1109/ENBENG.2019.8692573.

[8] Sigvard Johansen Seljelv and Lasse Løvstakken. "Deep Learning for Deformation Analysis in Echocardiography (Project thesis)". In: (2020).

[9] Lindsay M. Biga et al. *Anatomy and Physiology: Cardiac Cycle*. URL: https://open.oregonstate.education/aandp/chapter/19-1-heart-anatomy/ (visited on 12/14/2020).

[10] Nina Ajmone Marsan and Aniek L. van Wijngaarden. "Valve Strain: A Further Step Toward a Full Understanding of Mitral Valve Function and Dysfunction". In: *JACC: Cardiovascular Imaging* (2021). ISSN: 1936-878X. DOI: https://doi.org/10.1016/j.jcmg.2021.02.006. URL: https://www.sciencedirect.com/science/article/pii/S1936878X21001777.

[11] Lindsay M. Biga et al. *Anatomy and Physiology: Cardiac Cycle*. URL: https://open.oregonstate.education/aandp/chapter/19-3-cardiac-cycle/ (visited on 12/12/2020).

[12] American Heart Association. *Problem: Mitral Valve Regurgitation*. 2020. URL: https://www.heart.org/en/health-topics/heart-valve-problems-

and-disease/heart-valve-problems-and-causes/problem-mitral-valve-regurgitation (visited on 05/28/2021).

[13] M. Liu et al. "Rheumatic Heart Disease: Causes, Symptoms, and Treatments". In: *Cell Biochem Biophys* 72.3 (July 2015), pp. 861–863.

[14] American Heart Association. *Problem: Mitral Valve Prolaps*. 2021. URL: https://www.heart.org/en/health-topics/heart-valve-problems-and-disease/heart-valve-problems-and-causes/problem-mitral-valve-prolapse (visited on 05/28/2021).

[15] American Heart Association. *Problem: Mitral Valve Stenosis*. 2021. URL: https://www.hopkinsmedicine.org/health/conditions-and-diseases/rheumatic-heart-disease (visited on 05/28/2021).

[16] Alaa A. Mohamed, Ahmed A. Arifi, and Ahmed Omran. "The basics of echocardiography". In: *Journal of the Saudi Heart Association* 22.2 (2010), pp. 71–76. ISSN: 1016-7315. DOI: https://doi.org/10.1016/j.jsha.2010.02.011. URL: http://www.sciencedirect.com/science/article/pii/S1016731510000345.

[17] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[18] Ian GoodFellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. Cambridge, Massachusetts: The MIT Press, 2016. ISBN: 9780262035613.

[19] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *ArXiv* abs/1609.04747 (2016).

[20] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[21] H. Gao et al. "Pixel Transposed Convolutional Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.5 (2020), pp. 1218–1227. DOI: 10.1109/TPAMI.2019.2893965.

[22] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[24] Deepak Kumar and A G Ramakrishnan. "Power-law transformation for enhanced recognition of born-digital word images". In: *2012 International Conference on Signal Processing and Communications (SPCOM)*. 2012, pp. 1–5. DOI: 10.1109/SPCOM.2012.6290009.

[25] George Stockman and Linda G. Shapiro. *Computer Vision*. 1st. USA: Prentice Hall PTR, 2001. ISBN: 0130307963.

[26] Fred E. Szabo. "M". In: *The Linear Algebra Survival Guide*. Ed. by Fred E. Szabo. Boston: Academic Press, 2015, pp. 219–233. ISBN: 978-0-12-409520-5. DOI: https://doi.org/10.1016/B978-0-12-409520-5.50020-5. URL: https://www.sciencedirect.com/science/article/pii/B9780124095205500205.

[27] Richard O. Duda and Peter E. Hart. "Use of the Hough Transformation to Detect Lines and Curves in Pictures". In: *Commun. ACM* 15.1 (Jan. 1972), pp. 11–15. ISSN: 0001-0782. DOI: 10.1145/361237.361242. URL: https://doi.org/10.1145/361237.361242.

[28] *Module: tf.keras*. Google. URL: https://www.tensorflow.org/api_docs/python/tf/keras.

[29] *Plotly, Open-Source Python Package*. Plotly. URL: https://plotly.com/.

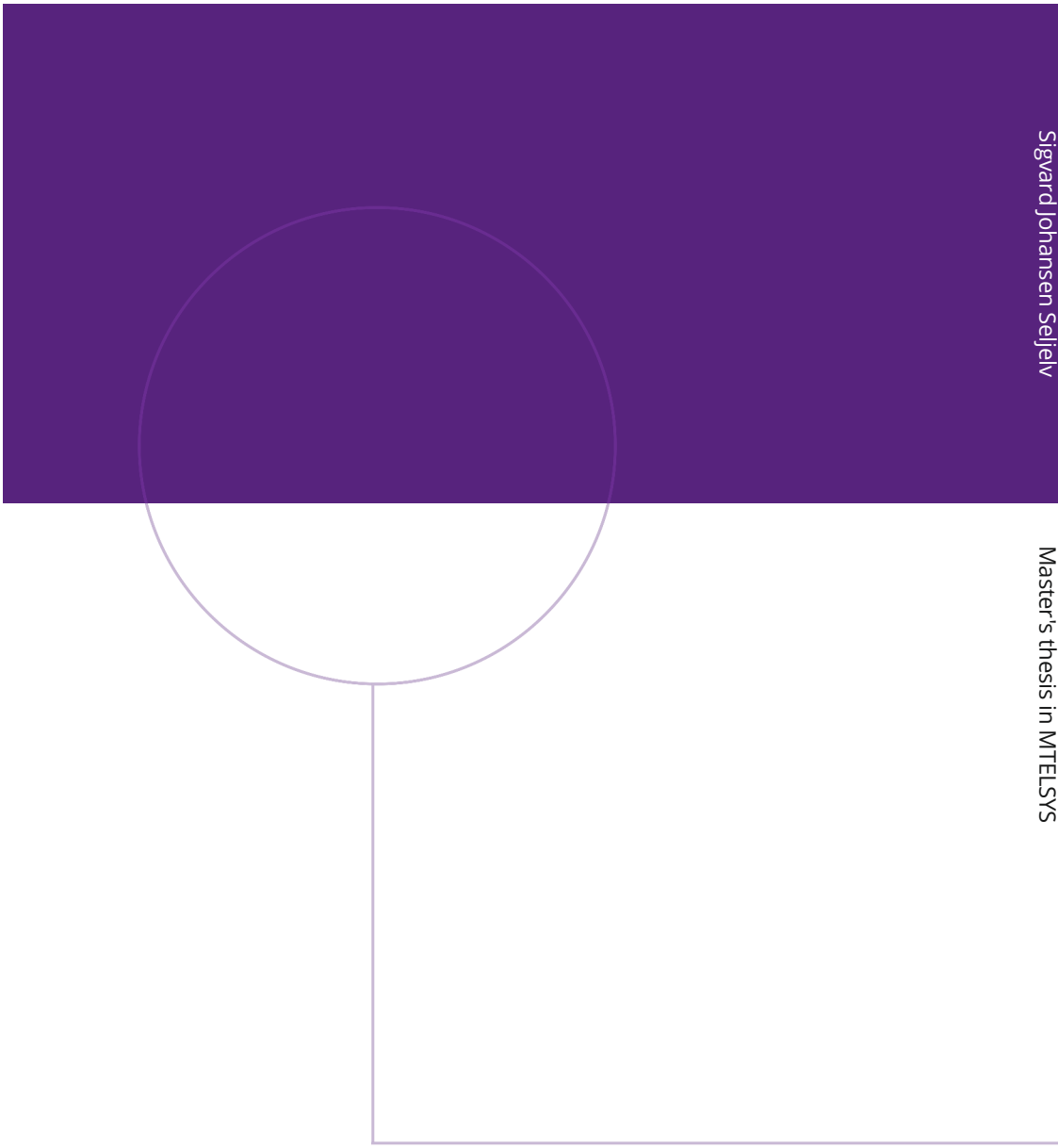
[30] E. Smistad et al. "Segmentation of apical long axis, four- and two-chamber views using deep neural networks". In: *2019 IEEE International Ultrasonics Symposium (IUS)*. 2019, pp. 8–11. DOI: 10.1109/ULTSYM.2019.8926017.

[31] *Keras: Adam optimizer*. URL: https://keras.io/api/optimizers/adam/ (visited on 06/08/2021).

[32] Lu Lu et al. "Dying ReLU and Initialization: Theory and Numerical Examples". In: *Communications in Computational Physics* 28.5 (2020), pp. 1671–1706. ISSN: 1991-7120. DOI: https://doi.org/10.4208/cicp.OA-2020-0165. URL: http://global-sci.org/intro/article_detail/cicp/18393.html.

[33] Derliz Mereles. *Quantitative assessment of longitudinal venticular function*. Cardiology Department at the University Hospital of Heidelberg. URL: https://echobasics.de/long-en.html (visited on 06/16/2020).

[34] Rusul L. Abduljabbar, Hussein Dia, and Pei-Wei Tsai. "Unidirectional and Bidirectional LSTM Models for Short-Term Traffic Prediction". In: *Journal of Advanced Transportation* 2021 (Mar. 2021), p. 5589075. ISSN: 0197-6729. DOI: 10.1155/2021/5589075. URL: https://doi.org/10.1155/2021/5589075.



Sigvard Johansen Seljelv

Master's thesis in MTELSYS

NTNU
Norwegian University of
Science and Technology