Hochschule Darmstadt

Fachbereiche Mathematik
und Naturwissenschaften
& Informatik

Deep Learning Support of Transcatheter Aortic Valve Implantation using Neural Network-based Image Analysis

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M.Sc.)
im Studiengang Data Science

vorgelegt von

Nina Krüger

Matrikelnummer: 760213

Referent	:	Prof. Dr. Arnim Malcherek
Korreferentin	:	Prof. Dr.-Ing. Anja Hennemuth, Charité - Universitätsmedizin Berlin

Ausgabedatum	:	03. September 2020
Abgabedatum	:	12. Februar 2021

DECLARATION

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Frankfurt, 12. Februar 2021

Nina Krüger

ABSTRACT

PURPOSE: Treatment of severe aortic stenosis requires careful assessment of the aortic root to select an appropriate prosthesis for Transcatheter Aortic Valve Implantation (TAVI). For this purpose, pre-operative CT images of the heart are analyzed, and relevant parameters, such as the aortic annulus area or perimeter, are determined. Two software solutions are presently used at the German Heart Center Berlin for obtaining these measurements, the fully-automated HeartNavigator3 (HN) and the semi-automated 3mensio (3m). In this work, the feasibility of a neural network-based approach is assessed, which is independent of specific imaging protocols or vendors.

METHODS: To deduce the aortic annulus area and perimeter, image regions of interest are segmented using a cascade of Convolutional Neural Networks, following the U-Net architecture. A U-Net uses so-called transposed convolutions to predict each voxel's probability to be part of the sought-after segmentation. First, the region of interest surrounding the device landing zone is segmented, second, the aorta, including the aortic valve within that region, and third, the area around the annulus. From this final segmentation, the aortic annulus plane is deduced by principal component analysis. Area and perimeter are obtained from a segmentation of the annulus in this plane.

RESULTS: The neural networks were trained using a data set of 90 expert-annotated CT scans. Segmentation of the aorta within the device landing zone achieved an F1 score of 0.94 on a test set of seven patients; segmentation of the annulus in the two-dimensional plane reached an F1 score of 0.95. The deep learning model calculated an average annulus area of 543.2 mm^2 and an average perimeter of 83.9 mm on an evaluation data set of 100 patients. Those calculated means differ significantly from the two software solutions' measurements on the same data set (area: 481.5 mm^2 (HN), 463.5 mm^2 (3m); perimeter: 79.3 mm (HN), 77.2 mm (3m)). While the discrepancy between the two software solutions is consistent with reported inter-observer differences, the deep learning results deviate more than twice as much from the software solutions' measurements.

CONCLUSION: Even with a relatively small training set of 90 CT scans, the neural network approach enables the reliable assessment of the aortic root. However, further work is required to optimize the annulus plane detection for correct annulus measurement. An extended training data set is required to further improve this method's applicability and robustness. It should also include several examples of uncommon cases, such as pre-implanted artificial valves.

Key words: Deep Learning, Convolutional Neural Network, U-Net, ResNet, Image Analysis, Segmentation, Medical Data, CT

ZUSAMMENFASSUNG

MOTIVATION: Zur Behandlung einer schweren Aortenklappenstenose ist eine sorgfältige Analyse der Aortenwurzel von größter Bedeutung, um eine geeignete Prothese für eine Transkatheter-Aortenklappen-Implantation (TAVI) zu wählen. Hierfür werden vor der Operation CT-Aufnahmen des Herzens betrachtet und relevante Messwerte, wie Fläche und Umfang des Aortenannulus, berechnet. Am Deutschen Herzzentrum Berlin werden zur Bestimmung dieser Messwerte bisher die Softwarelösungen HeartNavigator (HN) und 3mensio (3m) genutzt, die teils manuelles Eingreifen erfordern. Diese Arbeit erprobt einen voll-automatisierten Ansatz basierend auf neuronalen Netzen, welcher unabhängig von bestimmten Bildgebungsprotokollen oder Softwareanbietern nutzbar ist.

METHODEN: Um Fläche und Umfang des Aortenanullus zu bestimmen, werden zunächst relevante Bildregionen segmentiert. Hierfür wird ein Deep-Learning-Ansatz mit einer Abfolge von Convolutional Neural Networks entsprechend der U-Net-Architektur verwendet. Das U-Net nutzt *transposed convolutions* (umgekehrte Faltungen), um den Voxeln eines Bildes eine Wahrscheinlichkeit zuzuordnen, ob diese zu einer gesuchten Segmentierung gehören. Zuerst wird die Region um die Aortenklappe, in der die Prothese eingesetzt werden soll, segmentiert, als zweites die Aorta inklusive Aortenklappe und zuletzt die Region um den Annulus. Mithilfe einer Hauptkomponentenanalyse wird diejenige Ebene abgeleitet, in der der Annulus liegt. Anhand der Segmentierung des Annulus innerhalb dieser Ebene werden letztlich Fläche und Umfang des Annulus bestimmt.

ERGEBNISSE: Die neuronalen Netze wurden mit 90 CT-Aufnahmen trainiert, die von Experten annotiert wurden. Die Aortensegmentierung erreichte einen F1-Wert von 0.94 auf einem Testdatensatz von sieben Patienten; die Segmentierung des Annulus erreichte einen F1-Wert von 0.95. Der Deep-Learning-Ansatz bestimmte durchschnittlich eine Fläche des Aortenannulus von 543.2 mm^2 und einen Umfang von 83.9 mm auf einem weiteren unabhängigen Datensatz von 100 Patienten. Diese Durchschnitte unterscheiden sich deutlich von denen, die die Softwarelösungen auf demselben Datensatz ermittelten (Fläche: 481.5 mm^2 (HN), 463.5 mm^2 (3m); Umfang: 79.3 mm (HN), 77.2 mm (3m)). Die Diskrepanz zwischen den Softwarelösungen entspricht etwa der beobachten Diskrepanz zwischen von Medizinern getätigten Messungen. Die Ergebnisse des Deep-Learning-Ansatzes weichen um mehr als das Doppelte davon ab.

FAZIT: Bereits mit einem recht kleinen Datensatz von 90 CT-Aufnahmen ermöglicht der hierarchische Deep-Learning-Ansatz die zuverlässige Segmentierung der Aortenwurzel. Allerdings muss die Detektion des Aortenannulus weiter optimiert werden, um diesen exakt vermessen zu können.

Ein erweiterter Datensatz ist erforderlich, um diesen Ansatz zuverlässiger und robuster zu gestalten. Dieser sollte ebenfalls einige untypische Fälle, wie bereits implantierte künstliche Aortenklappen enthalten.

Schlagwörter: Deep Learning, Convolutional Neural Network, U-Net, Res-Net, Bildanalyse, Segmentierung, Medizinische Daten, CT

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

CNN Convolutional Neural Network

CSO Contour Segmentation Object

CT Computed Tomography

DLZ Device Landing Zone

DICOM Digital Imaging and Communications in Medicine

FFNN Feedforward Neural Network

IoGT Intersection–over–Ground-Truth

IoU Intersection–over–Union

LVOT Left Ventricular Outflow Tract

MAE Mean Absolute Error

MSCT Multi-Slice Computed Tomography

NIfTI Neuroimaging Informatics Technology Initiative

PCA Principal Component Analysis

ReLU Rectified Linear Unit

TAVI Transcatheter Aortic Valve Implantation

WEM Winged Edge Mesh

Part I

THESIS

1 INTRODUCTION

The following chapter illustrates the motivation and aim of this thesis - deep learning support of Transcatheter Aortic Valve Implantation. The structure of the thesis is briefly described.

1.1 MOTIVATION

The human heart beats around 100 000 times a day. With each beat, 70 ml of blood is pumped through the aorta to supply the oxygen-enriched blood to the rest of the body [32]. Unfortunately, several conditions can affect the heart's ability to function, one of which is known as aortic stenosis. This disease especially occurs with advancing age. Over the years, the aortic valve weakens and possibly accumulates calcium deposits from the blood, thus stiffening the valve cusps. A congenitally abnormal aortic valve further increases the risk of developing the condition. As a result, the blood flow from the left ventricle to the aorta gets restricted, forcing the left ventricle to work harder to pump a sufficient amount of blood into the aorta and onward to the rest of the body. Eventually, this extra work can weaken the heart, ultimately leading to heart failure. Accompanying symptoms may include breathlessness, chest pain or fainting. [90]

Nowadays, aortic stenosis is treatable by surgically replacing the defective valve with a prosthesis [30]. However, some patients might not be fit for surgery due to lung problems or further comorbidities.

For such cases, a less invasive procedure was developed that generally does not require the opening of the chest, the so-called Transcatheter Aortic Valve Implantation (TAVI). In a TAVI, a catheter holding the valve replacement is advanced mostly through the femoral artery towards the faulty heart valve. The prosthesis is expanded within the original valve to take over its function [59].

Nevertheless, this procedure does have its own risks: as there is no direct access to the heart in contrast to open surgery, it is much harder to fit the prosthetic valve properly. An undersizing of the valve can lead to paravalvular leakage, blood flowing between the structure of the implanted valve and cardiac tissue [30]. In case of oversizing, the implant either does not unfold properly or even causes a rupture of the surrounding tissue, especially if the valve is forcibly expanded with a balloon [30]. All cases jeopardize the implanted valve's proper functionality and might result in further complications and required operations. Thus, proper sizing is of utmost importance.

1.2 AIM AND SCOPE OF THE THESIS

The proposed thesis aims at supporting exactly this task of valve sizing by facilitating automatic aortic root analysis. At the German Heart Center Berlin, two software tools are currently being used to obtain the required measurements for valve selection, the semi-automated 3mensio and the fully-automated HeartNavigator3. Both require some manual intervention for optimal measurements and depend on specific input formats [56].

In this thesis, a deep learning approach for aortic root analysis is assessed, which is fully-automated and independent of imaging protocols or vendors. As a proof of concept, a model obtaining the area and perimeter of the aortic annulus is developed and benchmarked against the two software tools. An essential challenge in this benchmarking is the lack of true values for evaluation. The two software packages deliver different measurements, resulting in a discrepancy of deduced valve size in 18% of the cases [56]. Knobloch et al. [45] report similar inter-observer differences. It is nontrivial to decide which value is the most correct. Instead, the evaluation will focus on the comparability of the results between the neural network-based approach and the two software solutions, to prove the feasibility of the neural network-based approach.

Such an approach could be the basis for a conceptually simple and fully automatic model, supporting TAVI device selection. Possible benefits in comparison to the current software solutions are vendor independence, reproducibility of results and suitability for different Computed Tomography (CT) imaging protocols with minimal effort of adaptation.

In order to simplify the task of annulus measurement, it is broken down into several steps:

- Detection of a uniformly sized bounding box around the so-called Device Landing Zone, the area in which a prosthetic valve shall be placed
- Segmentation of the aorta, including the aortic valve within the bounding box
- Detection of the valve plane, i.e., the plane in which the aortic cusps are anchored and where the aortic annulus is measured
- Segmentation of the aortic annulus in this resulting two-dimensional plane and derivation of measurements

For each step, a Convolutional Neural Network (CNN) is trained and evaluated against different metrics.

1.3 STRUCTURE

Chapter 2 first establishes the required medical terms, and a broad overview of the current approach to aortic annulus measurement for TAVI device selection will be presented. Further, the machine learning principles considered in this thesis are introduced. Based on selected research papers, the state-of-the-art applications in the medical field are examined. Chapter 3 details the author's contributions on applying machine learning techniques to support optimal aortic valve sizing. Chapter 4 presents the results of each individual step as well as the comparison to the benchmark. The thesis concludes with a summary and an outlook on future research opportunities.

2 FUNDAMENTALS AND RELATED WORK

The first section of this chapter introduces the medical background of the disease, aortic stenosis, and its possible treatment, Transcatheter Aortic Valve Implantation (TAVI), concentrating on aspects necessary for understanding TAVI's relevance and complexity. The current tools for annulus measurement, 3mensio and HeartNavigator3, are presented.

The second section gives an overview of the used machine learning techniques and exemplifies how they are presently applied for medical use.

2.1 MEDICAL BACKGROUND

Selection of a suitable model for TAVI requires sound understanding of the anatomical conditions. This section establishes the essential medical background and current technologies used to support valve selection.

2.1.1 *Anatomy and Physiology of the Cardiovascular System*

The human cardiovascular system comprises the heart, blood vessels and the blood itself. The heart is responsible for pumping the blood through the blood vessels to transport oxygen, nutrients, hormones and waste products around the body. The blood vessels can be divided into three major types, arteries, veins and capillaries. Arteries move blood away from the heart, veins carry blood towards the heart, and capillaries facilitate the exchange of different elements, such as oxygen, between blood and tissues.

The aorta is the largest artery in the human body. Ascending from the heart's left ventricle, it supplies the oxygenated blood to the rest of the body. With every heartbeat, the aorta stretches to receive a large volume of blood, then contracts to its original diameter to push the blood into the off-branching arteries [79].

The heart itself is a double pump, where the left and right side contract separately. Each side consists of an atrium and a ventricle, making up the four chambers of the heart [74]. There are four valves within the heart that allow for a controlled outflow of the blood while restricting the blood from flowing back. Between the left ventricle of the heart and the aorta sits the aortic valve [4], as depicted in Figure 2.1.

2.1.2 *The Aortic Valvar Complex*

The aortic valve controls the blood flow between the left ventricle and the aorta. Typically, the aortic valve is composed of three cusps, also called

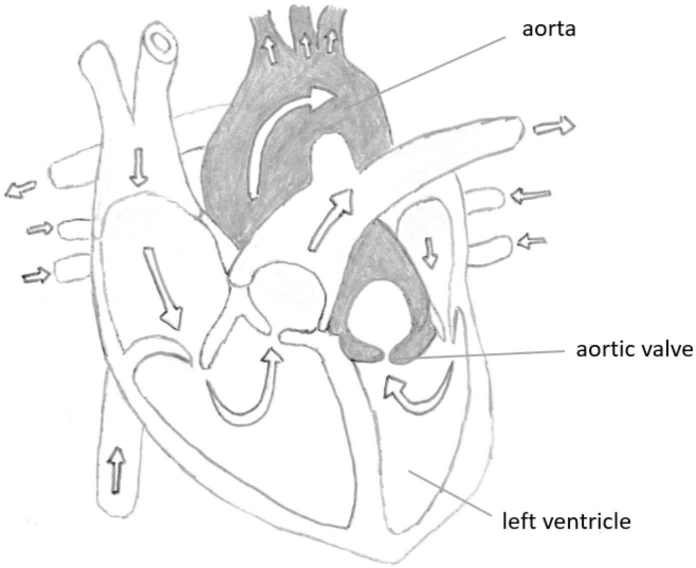


Figure 2.1: The position of the aortic valve within the heart (based on [4]): The aortic valve controls the blood flow between the left ventricle and the aorta.

leaflets. However, in 1.3% of the population examined in the United States, it is found to be congenitally bicuspid [65].

The aortic root is the direct continuation of the Left Ventricular Outflow Tract (LVOT). It comprises the whole aortic valve from its basal attachment of the leaflets within the left ventricle, the so-called annulus, to their attachment at the sinutubular junction, the region of the ascending aorta, where it becomes a tubular structure. The area where a prosthetic valve would be placed comprises the aortic root, including the annulus and LVOT. This area is also referred to as the Device Landing Zone (DLZ). Figure 2.2 provides an overview of the anatomy.

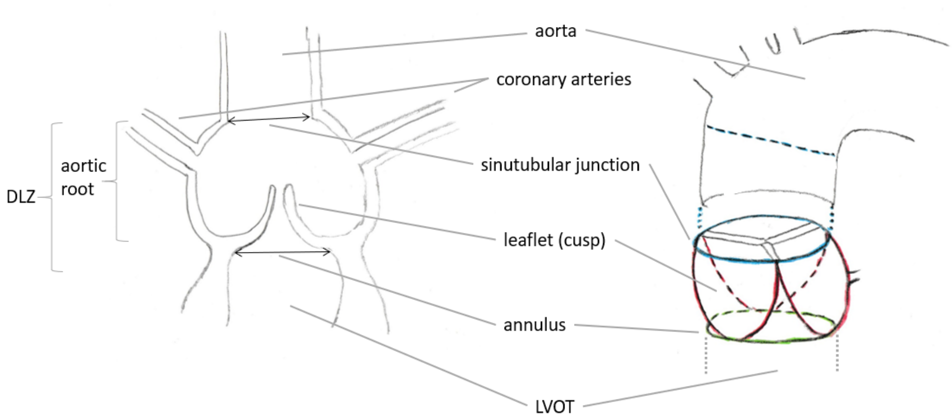


Figure 2.2: The aortic root components between LVOT and aorta (based on [78] and [12]): The aortic root comprises the whole aortic valve between annulus and sinutubular junction. The DLZ, the area where a prosthetic valve would be placed, additionally includes the annulus and the LVOT.

Several conditions can lead to a malfunctioning of the aortic valve, such as aortic stenosis.

2.1.3 *Aortic Stenosis*

Aortic stenosis is one of the most common and most serious valve diseases, affecting about 2% of the elderly population over 70 years of age, with a mortality rate of above 50% in patients with severe aortic stenosis undergoing conservative management [84].

Characteristic for this disease is a narrowing of the aortic valve opening, leading to a restriction of blood flow from the left ventricle to the aorta. A schematic comparison of a healthy and a stenotic valve can be found in Figure 2.3. As a result of aortic stenosis, the pressure in the left ventricle might increase, leading to a thickening of the heart's walls to maintain adequate pumping pressure. Without proper treatment, heart function can deteriorate. In light cases, medical treatment is possible. For patients with severe aortic stenosis, a replacement of the pathological aortic valve with a prosthesis might be advisable [25].

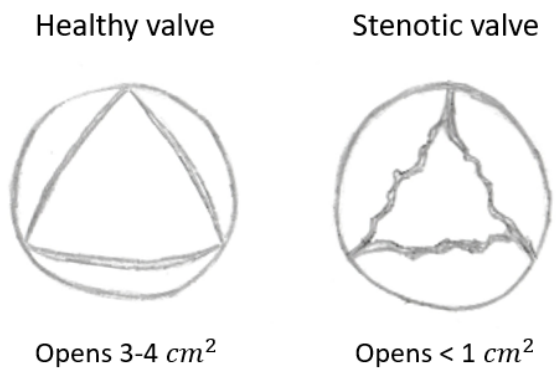


Figure 2.3: A schematic comparison of a healthy and a stenotic valve (based on [52]): A healthy valve opens to a cross-sectional area of about 3-4 cm^2 while a stenotic valve reaches an opening of less than 1 cm^2.

The most common procedure is open-heart transplantation, where the diseased aortic valve is removed and replaced with a new one [25].

For patients with severe comorbidities that are not fit for open-heart surgery, a less invasive procedure evolved, the so-called Transcatheter Aortic Valve Implantation (TAVI).

2.1.4 *Transcatheter Aortic Valve Implantation*

TAVI is a minimally invasive procedure for transplanting prosthetic aortic valves. The artificial valve is transported to its destined position via catheter, generally from the groin area through the femoral artery. Alternative access paths are possible. The prosthesis is then released inside the faulty aortic valve at the DLZ and either self-expands or is expanded with a balloon [19, 77]. In this procedure, in contrast to open-heart surgery, it is impossible to find the optimal valve by just testing likely fits. Thus, it is vital to determine the desired model size in advance. Under- or oversizing can lead to dramatic complications. Undersizing of the prosthesis can result in paravalvular leakage or stent migration [30]. Oversizing could even cause annular rupture, all cases ultimately endangering the patient's life [59]. As can be seen from the above complications, it is of utter importance to diligently select a suitable transplant. In order to properly size a TAVI device, the area and perimeter of the annulus are required. [61].

CT is seen as the gold standard for aortic root assessment as the basis for obtaining such measurements [8].

2.1.5 *CT Imaging and Software Tools*

Computed Tomography (CT) uses computer-processed combinations of multiple X-ray measurements taken from different angles to produce cross-sectional images of the body. As a result, CT images allow for visualization of various structures inside the body. For the present study, cardiac Multi-Slice Computed Tomography (MSCT) images were obtained using a standardized protocol, optimized for DLZ analysis [56]. Those multi-slice CTs result in three-dimensional images, represented by voxel values. A voxel is the three-dimensional equivalent to the two-dimensional pixel, which is an abbreviation for picture element.

According to Blanke et al. [8], an electrocardiogram-synchronized CT scan of the aortic root is required for TAVI planning, using a contrast agent to visualize blood vessels. The electrocardiogram-synchronization is necessary to ensure recording of each image at the same moment of the heartbeat. Otherwise, the aortic valve's movement would lead to a difference in measured dimensions of the valve, as shown by Horehledova et al. [33]. The CT scans are available in DICOM[1] format, respectively a MeVisLab internal storage format. For compatibility with the programming language Python, which is used to develop the deep learning approach, the NIfTI[2] format is used by choice. NIfTI is a standardized file format for saving three-dimensional images. Originally envisaged for neuroimaging, it is also suitable for other medical images, such as CTs.

Currently, heart specialists at the German Heart Center Berlin utilize two software solutions to analyze a patient's CT scan. The HeartNavigator3 [58] loads a cardiac MSCT series. The segmentation, the definition of the annular plane and the measurements of the annulus area and perimeter are performed fully automatically. 3Mensio [35] requires manual placement of three markers, indicating the tips of the three leaflets, determining the annular plane. The annulus can then be measured semi-automatically. Figure 2.4 shows a comparison of the two software solutions' interfaces. In each case, the resulting measurements need to be assessed by experts to infer the op-

1 https://www.dicomstandard.org/
2 https://nifti.nimh.nih.gov/

timal valve selection [56]. The vendors of both software solutions did not disclose any information on the methods used for segmentation, plane detection and measurement.

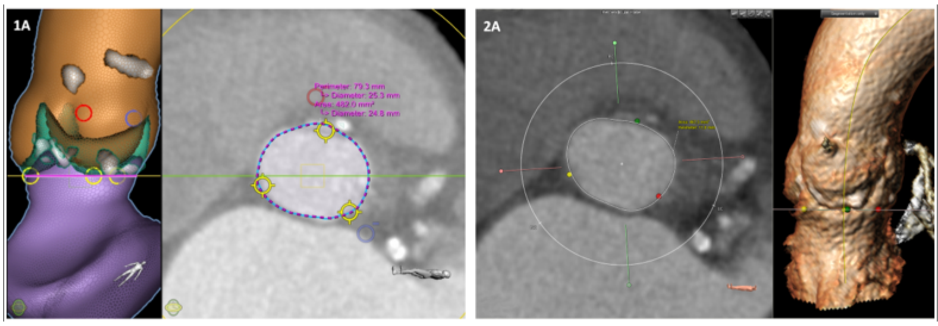


Figure 2.4: A comparison of the interfaces of the two software solutions for annulus measurement, the fully-automatic HeartNavigator (left) and the semi-automatic 3mensio (right) [56]: Both show a segmentation of the aorta and the LVOT in three dimensions, where the HeartNavigator differentiates between the two, and the detected annulus plane in two dimensions. The tips of the valve cusps are shown in the three- as well as the two-dimensional views.

Measurement of the annulus is non-trivial. Several studies evaluated the reliability and repeatability of measurements. Meyer et al. [56] observed a difference in resulting prosthesis size between HeartNavigator and 3mensio measurements in 18% of the considered patients, with a mean difference of 18 mm^2 for the annulus area. Knobloch et al. [45] report mean intra-observer differences of 1.5-5.7 mm^2 and mean inter-observer differences of 5.7-15 mm^2 for the annulus area.

2.2 MACHINE LEARNING

As elucidated above, model selection and sizing for TAVI is presently done by experts visually evaluating different measurements on the patient's CT scan while factoring in further patient data. Software tools support this selection process. Apart from being prone to errors, it also requires a lot of practice and adherence to research standards [6, 57].

Therefore, this thesis examines how machine learning can be used to assist medical practitioners in implant sizing. In machine learning, a computer algorithm learns to make predictions based on sample data. With a sufficient amount of examples, the algorithm can reach near-perfect accuracy. The required number of samples highly depends on the use case and can range from a few dozen to several thousand. Deep learning is a subfield of machine learning based on artificial neural networks. In the present study, deep learning is used to learn different segmentations in a CT scan, finally allowing for the deduction of the aortic annulus' area and perimeter.

The following section gives an overview of the used techniques and technologies.

2.2.1 *MeVisLab*

MeVisLab is a proprietary cross-platform software solution, providing a framework for image processing, especially focused on medical imaging [54]. With its proper support for CT images, MeVisLab was used to prepare and annotate training images.

The following modules were frequently used:

- **MLImageFormatLoad**
  loads images in .mlimage format
- **itkImageFileReader/-Writer**
  reads/writes Insight Toolkit[3] (ITK) formats, such as NIfTI
- **Load Base**
  loads XML marker lists, like valve plane markers or the centerline
- **WEMLoad**
  loads segmentations, saved as Winged Edge Mesh (WEM)
- **CSOLoad**
  loads Contour Segmentation Objects (CSOs)
- **Subimage**
  extracts defined sub-image
- **Resample**
  resamples an image to desired voxel dimensions to standardise the image's resolution; this will consequently alter the image's dimensions

A sample MeVisLab network is displayed in Figure 2.5.

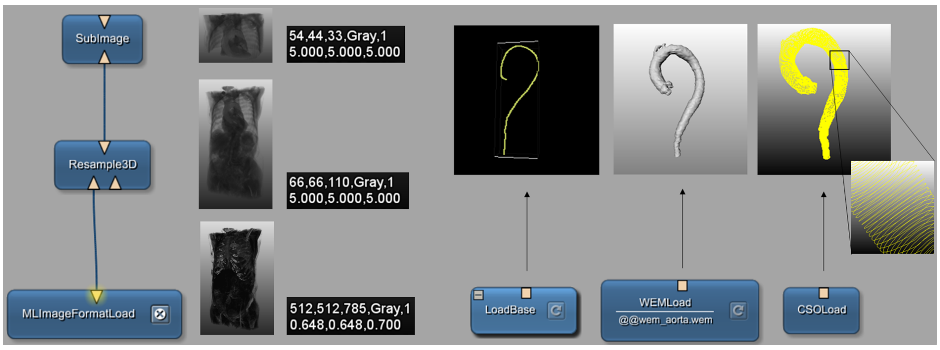


Figure 2.5: A sample MeVisLab network and output: The network shows the modules *MLImageFormatLoad*, *Resample3D*, *SubImage*, *LoadBase*, *WEMLoad* and *CSOLoad*. A sample output of each module is displayed.

More specific modules only used for special purposes will be indicated in the respective sections in Chapter 3 - Methods.

3 https://itk.org/

While working with MeVisLab, it needs to be distinguished between two different coordinate systems - voxel and world coordinates. Voxel coordinates refer to the image axes; they reach from zero to the size of the image. On the other hand, world coordinates refer to some real-world axes. In the context of CT scans, a point in the CT scanner is chosen as the origin of coordinates. The axes are arbitrarily defined, resulting in real numbered values, typically stated in millimeters, without any predefined range. Figure 2.6 illustrates this. MeVisLab allows for conversion between the two coordinate systems [53].

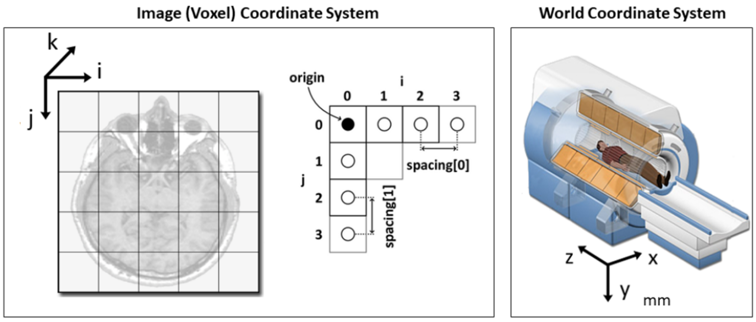


Figure 2.6: Voxel vs world coordinates (based on [1]): Voxel coordinates refer to the image axes while world coordinates refer to real-world axes, such as positions in a CT scanner.

2.2.2 *Training of Neural Networks*

Feedforward Neural Networks (FFNNs) provide the basis for deep learning. Inspired by the brain's functionality, a FFNN consists of artificial neurons conveying signals through several layers of the network to result in a desired output. The basic ideas presented in this subsection are extracted from *Pattern Recognition and Machine Learning* (Bishop [7]) and *Deep Learning* (Goodfellow, Bengio, and Courville [27]). In order to comprehend the concept of a neural network, its basic building block, the so-called perceptron, needs to be understood. A perceptron, depicted in Figure 2.7, transforms an input vector x to a scalar output y, by calculating a linear combination between the input vector elements $x_1, x_2, ..., x_n$ and a set of weights $w_1, w_2, ..., w_n$ and subsequently feeding this into a non-linear activation function. This activation function allows a network to learn non-linear patterns.

A neural network is then composed of several such perceptrons, which are arranged into layers, as shown in Figure 2.8. Three main components can be distinguished:

- The **input layer** corresponds to the input vector x.
- The **hidden layers** between the input and the output are mainly responsible for approximating a mapping between the input vector x and the desired output y.
- The **output layer** transforms the internal network state to the desired output dimension.

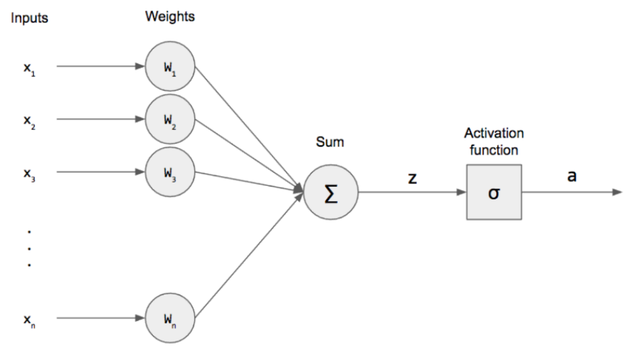


Figure 2.7: The structure of a perceptron [18]: A perceptron calculates a linear combination between input vector elements $x_1, x_2, ..., x_n$ and a set of weights $w_1, w_2, ..., w_n$. The resulting sum z is fed into a non-linear activation function to yield the output a.

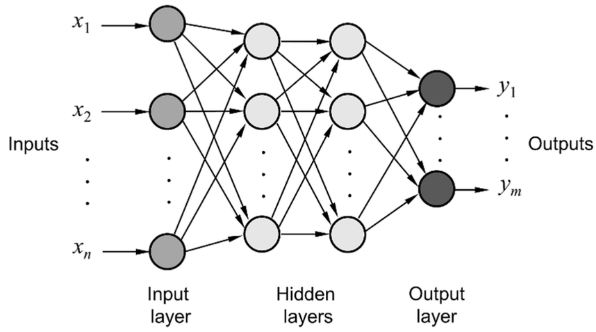


Figure 2.8: An example of a Feedforward Neural Network (FFNN) [62]: Several perceptrons are arranged into layers to build a network. An input vector $x_1, x_2, ..., x_n$ is fed into the network via an input layer. It passes through several hidden layers, resulting in an output vector $y_1, y_2, ..., y_m$ in the output layer.

The goal is then to approximate some function f^*, with $y = f^*(x)$ by defining a mapping $y = f(x, \theta)$. Specifically, the optimal values for θ need to be deduced. Here θ represents the aforementioned set of weights $w_1, w_2, ..., w_n$.

The most common approach for this deduction is the backward propagation of errors [9]. Iteratively, θ is adjusted to minimize the error between the actual target value y and the output $\hat{y}$. For this, first the input vector x is fed forward through the network, yielding the output $\hat{y} = f(x, \theta)$. The error between y and $\hat{y}$ is calculated with a loss function $L(y, \hat{y})$. In order to minimize this loss, it is backpropagated through the network, and θ is adjusted accordingly. This is commonly done by applying a gradient descent approach. Each weight or component of θ is usually randomly initialized. The gradient of the loss function with respect to each individual weight is calculated, and the weights are adjusted according to the direction of steepest descent. After adjustment, the steps are repeated until a defined stopping criterion is met.

A significant drawback of the standard FFNN is the high number of parameters that need to be optimized, especially in the case of high dimensional inputs, like images. Further, spatial relations cannot be retained, which is of concern in image analysis. Hence, an alternative approach emerged, the Convolutional Neural Network (CNN), which is especially suitable for high-dimensional image input.

2.2.3 *Convolutional Neural Networks*

The following subsection is largely based on *Deep Learning* (Goodfellow, Bengio, and Courville [27]) and *Deep learning: Technical introduction* (Epelbaum [22]). A CNN applies the same operation to several subsets of the input. This results in a sharing of weights, significantly lowering the number of trained parameters while also preserving spatial relations. Each operation generally consists of three steps, combined into a convolutional block:

1. **Convolution**
   The first layer in a convolutional block typically is a convolutional layer. A so-called kernel or filter is shifted over the input, repetitively performing a matrix multiplication between the kernel and the subset of the input to which the kernel is currently applied. This operation detects relevant features within the image, as illustrated in Figure 2.9.

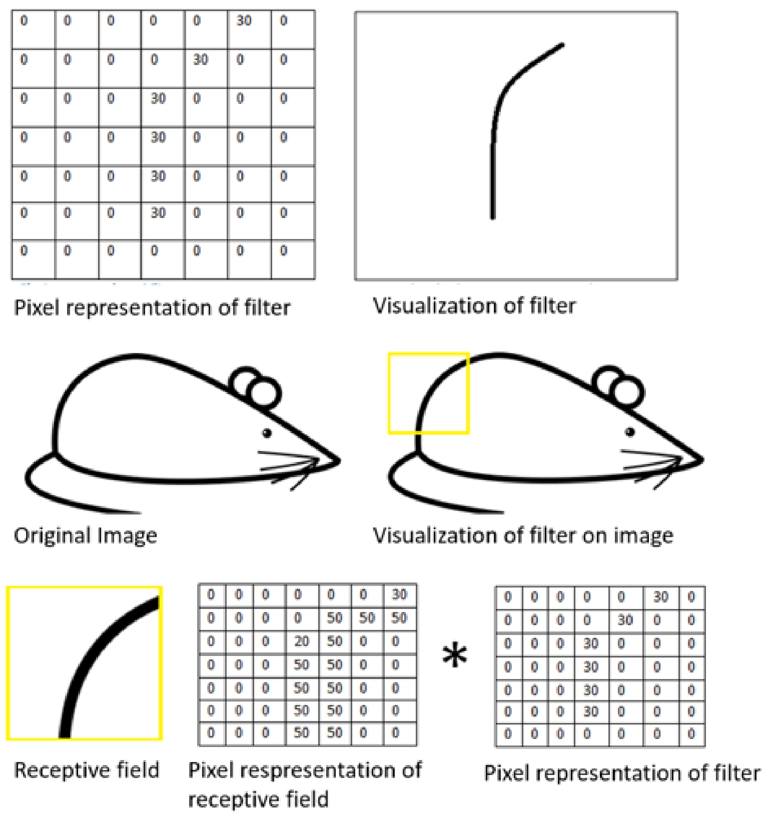


Figure 2.9: The convolutional operation [17]: A filter is shifted over the input image, repeatedly multiplying the pixel representations of the filter and the sub-image. The larger the resulting number, the more the sub-image is assumed to correspond to the filter. In this way, relevant features in the image are detected.

2. **Activation function**
   Concurrent with the standard architecture of a neural network, the convolutional layer's output is activated with a non-linear function. A typical choice for hidden layer activations is the Rectified Linear Unit (ReLU) [82]:
   $$f(x) = max(0, x).$$
   ReLU is a piecewise linear function, directly outputting the input if it is positive, and zero otherwise.

   For the output layer, the sigmoid function S (logistic function) is an appropriate activation if probabilities are expected in the output. It shows a return value in the range [0, 1].
   $$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$
   Both activation functions are visualized in Figure 2.10.

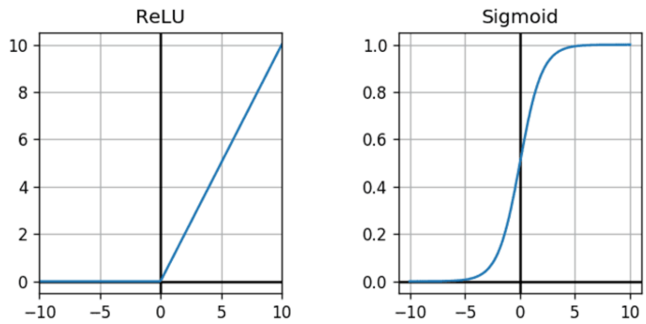


Figure 2.10: The activation functions ReLU and sigmoid [82]: ReLU is a piecewise linear function, directly outputting the input if it is positive, and zero otherwise, typically used for activation of hidden layer outputs. For the output layer, the sigmoid function is an appropriate activation if probabilites are expected in the output. It shows a return value in the range [0, 1].

3. **Pooling**
   A pooling layer concludes the basic convolutional block. Its purpose is to extract the most dominant features while also reducing the spatial dimension. Max or average pooling is employed, with max pooling typically yielding better results. It allows for translational invariance, since the output of the max pooling layer will generally be the same if features are slightly translated. Figure 2.11 exemplifies the pooling operation.

In a CNN, several different kernels are applied per convolutional layer. Each convolutional block's output can be fed into several more blocks, each block yielding higher level features, illustrated in Figure 2.12.

The standard CNN is completed with fully-connected layers generating the final output.

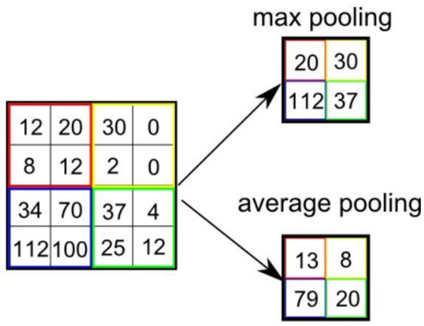


Figure 2.11: The pooling operation [68]: Max pooling reduces a sub-image of specific size (here 2 x 2) to its maximal value. Average pooling reduces the sub-image to the average of its values.

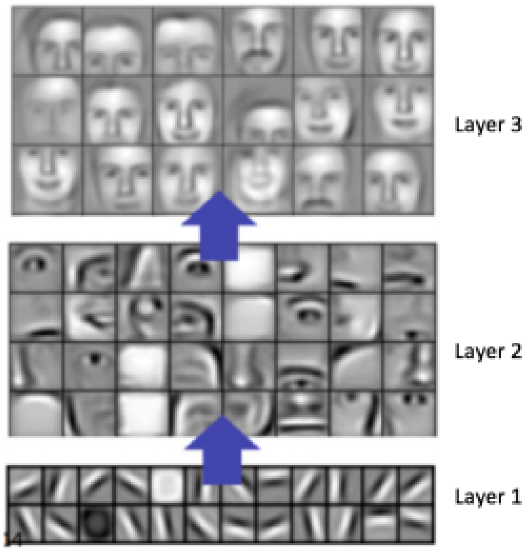


Figure 2.12: From low level to high level features [49]: The first convolutional layer detects low-level features, such as edges, the second layer already detects higher level features, such as eyes or noses, and the third layer detects even more abstract features, such as whole faces.

2.2.4 *Losses and Metrics*

A neural network is optimized by minimizing its error, represented by a loss function. Usually, minimization is achieved by backpropagating the error gradient through the network and adjusting the network's weights accordingly. Thus, the loss function needs to be piecewise-differentiable. The gradients optimally should be non-zero whenever the prediction and the target value do not agree and should also increase with an increasing discrepancy. At any non-differentiable point, a loss function is typically approximated [27]. A metric, on the other hand, is used to measure the algorithm's performance in an interpretable manner [34]. It is often defined in such a way that it is one if prediction and target agree perfectly and zero if there is no agreement at all.

In the following subsection, the functions used for model training and evaluation will be shortly introduced and possible difficulties highlighted. As will be further discussed in Chapter 3 - Methods, the different subtasks will be addressed with regression and segmentation approaches.

REGRESSION

Suitable loss functions for regression are the L1-norm, the mean absolute error (MAE) or the L2-norm, the root mean squared error (RMSE) [69]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \,, \tag{2.1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}, \tag{2.2}$$

with n being the number of samples, y the ground truth values and $\hat{y}$ the predictions.

Since those norms on their own do not provide information about the model quality, they need to be compared against a benchmark for evaluation purposes. Typical benchmarks are obtained by averaging, random walks or other very simple models [60].

For training of a neural network, the RMSE is advantageous, as its gradient increases with an increasing error while the gradient of the MAE is constant. Consequently, with RMSE, the adjustment of the weights will be proportional to the error. However, this also poses a disadvantage: The RMSE is much more sensitive to outliers. Thus, in the case of outliers due to corrupted data, the MAE might be favorable [60].

Other loss functions for regression are available, which will not be further considered but might be an interesting starting point for model improvements [29].

SEGMENTATION

The most commonly used loss function for the task of image segmentation is a pixel-wise binary cross-entropy loss. This loss examines each pixel individually, comparing the class predictions to the target mask [26]:

$$L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i), \tag{2.3}$$

where y is the target mask (1 if the pixel belongs to the segmentation and 0 otherwise) and $\hat{y}$ the predicted probability for each of the n pixels to belong to the segmentation.

This loss can be directly transferred to the three-dimensional case by considering each voxel. The binary cross-entropy loss is not easily interpretable. Thus, two metrics are regarded for model evaluation - Intersection–over–Union (IoU), also called Jaccard Index, and the Dice Coefficient or F1 score, as based on *Metrics to Evaluate your Semantic Segmentation Model* [83].

The IoU is given by the following formula:

$$IoU = \frac{Precision \cdot Recall}{Precision + Recall - Precision \cdot Recall} = \frac{TP}{TP + FP + FN} = \frac{Intersection}{Union}, \tag{2.4}$$

with $TP \equiv$ True Positive, $FP \equiv$ False Positive, $FN \equiv$ False Negative. The metrics used for calculation of the IoU can be derived from the confusion matrix:

		True Segmentation	
		positive	negative
Predicted Segmentation	positive	TP	FP
	negative	FN	TN

Table 2.1: Confusion matrix.

Precision and Recall are then defined as follows:

$$\begin{aligned} Precision &= \frac{TP}{TP + FP,} \\ Recall &= \frac{TP}{TP + FN.} \end{aligned}$$

The IoU describes the area of intersection between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth. In the three-dimensional case, the number of voxels belonging to both the predicted segmentation and the ground truth divided by the number of voxels belonging to either of the two is considered. This metric is illustrated in Figure 2.13a.

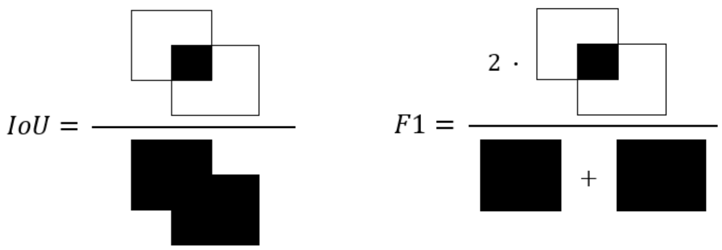
The F1 score is the harmonic mean of precision and recall.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = \frac{2 \cdot Intersection}{TotalArea/Volume} \tag{2.5}$$

It is a specialization of the more generic F_β score, which applies an additional weight β, such that recall is considered β times as important as precision.

$$F_\beta = \left(1 + \beta^2\right) \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}$$

With regard to image segmentation, the F1 score is given by twice the area of intersection between the predicted segmentation and the ground truth divided by the total number of pixels in both segmentations. In the three-dimensional case, it relates to twice the number of voxels belonging to both the predicted segmentation and the ground truth divided by the total number of voxels in both, illustrated in Figure 2.13b.



(a) Intersection-over-Union

(b) F1 score

Figure 2.13: Metrics for semantic segmentation: The Intersection–over–Union (IoU) and the F1 score are used to evaluate image segmentations. The IoU divides the area of intersection of the predicted segmentation and the ground truth by the area of union. The F1 score is given by twice the area of intersection divided by the total number of pixels in both segmentations.

Both are positively correlated. As shown in *IoU vs F1* [46]: "F score tends to measure something closer to average performance, while the IoU score measures something closer to the worst-case performance." The IoU tends to penalize single instances of bad classification more than the F1 score. Thus, the IoU should be the preferred metric if the model output is required as input for further steps to prevent completely wrong predictions. At the same time, the F1 score is favorable if outliers are expected in the data due to erroneous annotations.

Both metrics are applied to binarized predictions; hence, they cannot be used as loss functions for continuous network outputs. Different loss functions are available for segmentation, which are also easily interpretable as metrics. In future work, these might prove to be advantageous in the present case but will not be further analyzed in the context of this thesis [38, 41]. For now, the used loss function is confined to binary cross-entropy.

2.2.5 *Transfer Learning*

In transfer learning, knowledge obtained from one task is reused for another related task. With regards to CNNs, two main approaches of transfer learning can be distinguished, according to *CS231n: Convolutional Neural Networks for Visual Recognition - Transfer Learning* [88]:

CNNS AS FIXED FEATURE EXTRACTOR The last fully-connected layer of a pre-trained CNN is either retrained for the task at hand or replaced with suitable layers to generate the desired output. In contrast, the weights of the other layers are frozen and used as a fixed feature extractor.

FINE-TUNING THE CNN In this case, all layers of a pre-trained CNN are fine-tuned to the new task starting from the weights obtained from the pre-training. Potentially, some of the layers could still be kept fixed. As the earlier layers detect low-level features such as edges, fixing their weights does typically not impede model accuracy but can speed up the training process.

Overall, transfer learning offers two main advantages: first, it facilitates feature extraction of low-level features, requiring less training data for fine-tuning, second, due to the pre-training, models typically generalize better to previously unseen data. Hence, it is beneficial if the number of training samples for a specific task is limited, while there is an extensive database for a similar task.

2.2.6 *Ensemble Methods*

Ensemble methods construct multiple models and combine them to achieve one output. This combined output often performs better than a single model, as shown in several machine learning competitions and explained in theory [16]. Ensemble methods can either consist of the same base model, trained on different data sets, or different base models can be combined. Different strategies for combining the models can be utilized; averaging is the most straightforward one [20].

2.2.7 *Self-training*

In self-training, a model is first trained on a labeled data set, then generates pseudo-labels for an unlabeled data set, and is retrained on the combined data set with labels and pseudo-labels. The workflow is depicted in Figure 2.14. The steps can be repeated if desired [75].

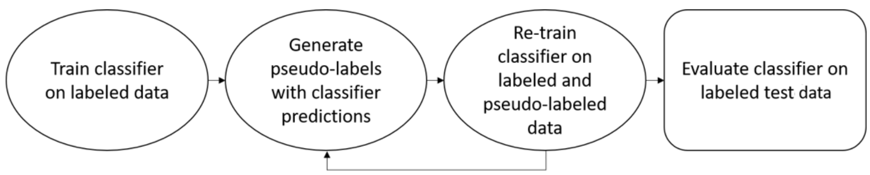


Figure 2.14: Self-training workflow [75]: A classifier is trained on labeled data. Afterwards, pseudo-labels are generated from the classifier predictions and the classifier is retrained on the combined data set of labels and pseudo-labels. It is then evaluated on labeled test data. The pseudo-label generation and re-training can be repeated if desired.

Zoph et al. [87] showed that self-training can tremendously improve results over pre-training and data augmentation, especially if the number of labeled training samples is limited.

2.3 RELATED WORK - MACHINE LEARNING ADVANCES FOR TAVI

Machine learning and artificial intelligence also found their way into medical research. A multitude of papers are published, dedicated research journals have emerged, and books have been printed [14, 21, 72].

Jaegere and Ribeiro [39] reflect on the use of artificial intelligence and advanced computer modeling in transcatheter interventions for structural heart disease, highlighting implications for clinical practice. They conclude:

> "The advent of Artificial Intelligence provides us with the possibility of processing large and extensive sets of data in such a way that complex patterns and relationships between variables that would never be accessible to the human eye and mind will eventually come to light. [...] Moreover, Artificial Intelligence, coupled with advanced simulation modeling, grants us the possibility of testing an invasive treatment in a patient-specific anatomic setting, thereby predicting which treatment is the most optimal for a specific patient."

Image-Based Computational Models for TAVI Planning: From CT Images to Implant Deployment (Grbic et al. [28]) follows precisely this approach of coupling machine learning techniques with modeling. TAVI planning is supported by segmenting the aortic apparatus from CT images, deducing a model of the aortic anatomy. This model allows for patient-specific selection of the optimal implant and, finally, simulation of the implant employment. Several machine learning algorithms are utilized for the individual steps, such as Marginal Space Learning (MSL). MSL is an efficient algorithm for object detection in three-dimensional images by incrementally learning classifiers in marginal spaces of lower dimensions. The problem is broken down into three subproblems - first, the object's position is estimated, second, its orientation, and lastly, its scale. This incremental approach results in high efficiency [86]. However, the aortic valve segmentation in Grbic et al. [28]'s approach is currently limited by the usage of generic parameters describing the thickness in a specific region of the segmented anatomy, impeding subsequent patient-specific measurements.

Al et al. [2] attempt to support clinicians in TAVI sizing by automatically detecting required landmarks using a colonial walk algorithm, which exploits a trained regression tree with directions to the target point in order to localize this point. However, this approach does not provide a segmentation of the full anatomical structure of the aortic valvar complex. Some characteristics might, thus, remain undetected.

Many recent approaches use model-based segmentation algorithms. For example, Lalys et al. [47] presented a comprehensive pipeline for TAVI analysis, including centerline detection, aorta segmentation, and aortic root segmentation using so-called atlas (average model) registration and deformable 3D snakes. Nevertheless, their approach requires the manual placement of a seed point in the aortic root region. Furthermore, the model-based approach might impede patient-specific measurements.

Also, deep neural networks have been applied to the task of automated aortic annulus measurement for TAVI. However, the problem was reduced to the two-dimensional aortic annulus plane [3]. Thus, this approach requires manual effort for the identification of the annulus plane.

This thesis now further pursues the application of deep neural networks for aortic root segmentation and annulus measurement from three-dimensional CT scans. Given the persuading performance of CNN methods in recent years, such an approach could enable fully automatic segmentation and measurement of any desired anatomical structure given appropriate annotations. It allows for patient-specific analysis, independent of generic parameters or models for the underlying anatomy.

3 METHODS

The following chapter portrays the methods used to develop a model for automatic inference of the aortic annulus perimeter and area from a three-dimensional CT scan before a TAVI procedure. This goal was split into four steps:

1. The region of interest around the DLZ is detected.
2. The aorta, including the aortic valve, is segmented. This segmentation can then be used as additional input for later predictions. Additionally, the aorta segmentation is valuable for TAVI access planning.
3. The valve plane is identified.
4. The aortic annulus in this plane is segmented and finally measured.

For each step, a so-called U-Net is trained to predict the desired segmentations. The U-Net won several awards in biomedical image segmentation challenges [66] and was thus the first choice for the present segmentation tasks. The following section will explain its architecture in detail.

Afterwards, the data basis for training the networks is presented, and features that influence model development are highlighted.

Finally, each of the four steps is explained in detail, highlighting data preparation, model specifics and applied postprocessing. Results for each step will be presented in Chapter 4 - Results.

3.1 THE U-NET ARCHITECTURE

CNNs have demonstrated tremendous achievements in image classification tasks, sometimes even exceeding the average human being's performance [43]. However, the standard architecture of CNNs does not perform well on the localization of desired image areas, for instance, segmentation of medical images. The U-Net architecture solves this demand by reversing the convolutions and thus allowing a pixel-wise class prediction.

This architecture is hence used to reduce the patients' CT scans to the relevant area around the DLZ and segment the desired structures. It is based on [67] and illustrated exemplarily for the two-dimensional case in Figure 3.1. The code basis is derived from [76]. For three-dimensional input images, the network layers are adjusted accordingly. The U-Net is implemented in Keras[1] with the TensorFlow[2] backend; for package specifications see Section B.2.

1 https://keras.io/
2 https://www.tensorflow.org/

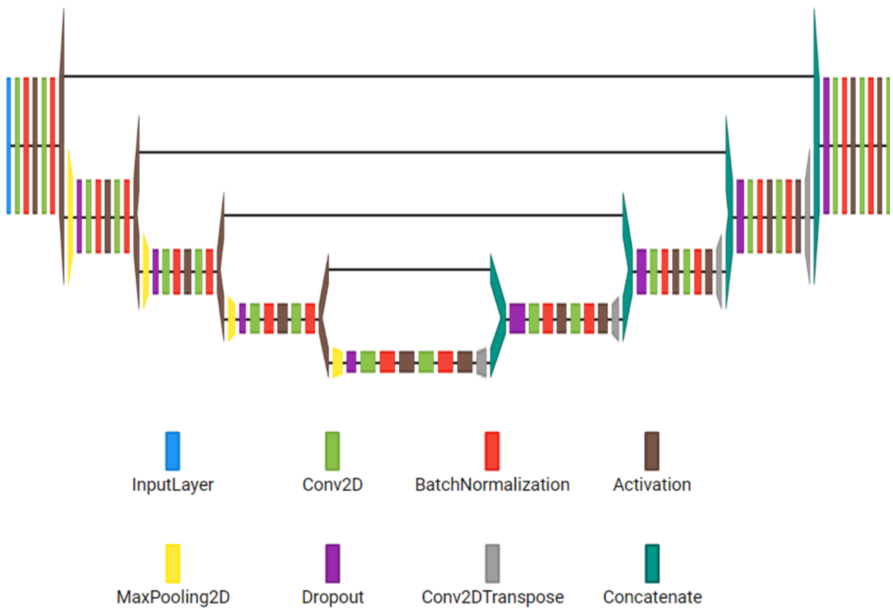


Figure 3.1: The U-Net architecture (visualization created with [5]): The U-Net consists of a contracting and an expansive path. The contracting path reduces the image dimension while increasing the deduced features. It contains several blocks, combining convolutional, batch normalization, activation, max pooling and dropout layers. The expansive path restores the original image dimension. Its blocks comprise convolutional, transposed convolutional, batch normalization, activation and dropout layers. Additionally, the output of blocks in the contracting path is concatenated with the input of the corresponding blocks in the expansive path.

The network consists of a contracting path and an expansive path. The contracting path corresponds to a standard convolutional network architecture, see Section 2.2.3. In the following, the layers are described for the two-dimensional case. In three dimensions, the third dimension has the same size as the other two. Several 3x3 convolutional layers are followed by a ReLU activation and a 2x2 max pooling for downsampling.

The expansive path alternates 3x3 convolutions with 2x2 transposed, or also called up-convolutions. Figure 3.2 illustrates the convolution and transposed convolution operations.

The contracting path contains five convolution blocks, where every block consists of two similar convolutional layers with same padding, maintaining the input dimensions. This is followed by batch normalization and a ReLU activation. In batch normalization, the layer input is normalized per batch. The training data set is subdivided into batches of equal size. During training, it is iterated over the batches, and the model's weights are adjusted after each batch. The first four blocks are respectively amended by max pooling and dropout layers with a dropout rate of 0.05. The dropout layer randomly sets input units to 0 with a frequency of the dropout rate at each step during training time, which helps prevent overfitting. The number of filters per convolutional layer, thus the number of extracted image features, is doubled

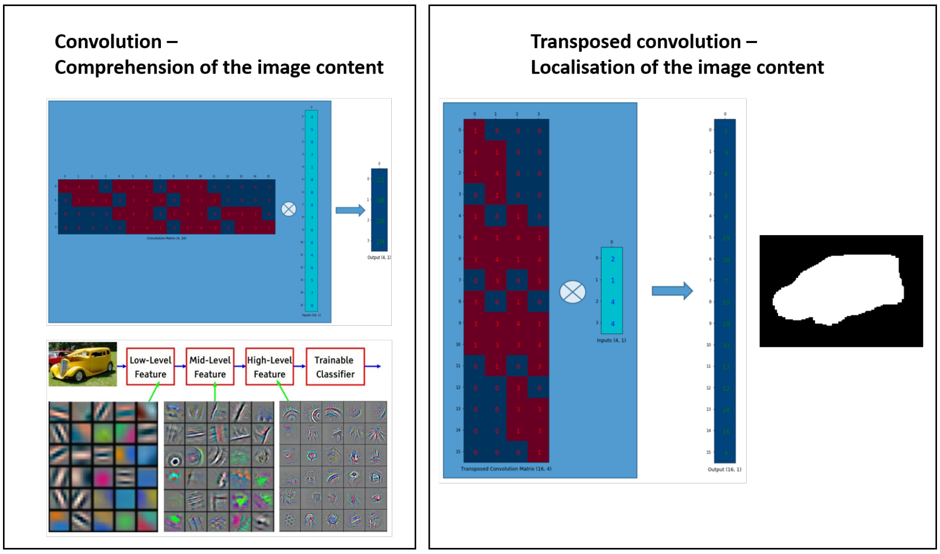


Figure 3.2: Convolution and transposed convolution (based on [24] and [71]): The convolution is responsible for comprehension of the image content by deduction of relevant image features via application of filters. The transposed convolution localizes the image content by reversing the convolution and assigning each pixel a probability to belong to a given class.

from block to block at the same time halving the spatial dimensions of the input with each max pooling layer.

The expansive path reverses this procedure via four transposed convolutional layers. With each transposed convolution, the number of extracted features is halved. The resulting output is concatenated with the corresponding output of the contracting path. Once more, a dropout layer and another convolutional layer block follow, maintaining the number of extracted features. As the last layer, a 1x1 convolution with sigmoid activation is used, resulting in a pixel- or voxelwise classification.

Ultimately, this architecture allows for the segmentation of the sought-for image areas. The U-Net outputs a probability for each voxel to belong to the segmentation. In each step, this is binarized with a threshold of 0.5 to result in the final segmentation. The used batch size and the number of filters in the first convolutional block are listed in Table 3.1. Both are not explicitly optimized but are limited by the available working memory; for information on the available hardware, see Section B.1. As loss function binary cross-entropy is employed. For the valve plane identification, this is weighted by the inverse class distribution, as described in [85]. The Adam optimizer [44] is used for stochastic gradient descent.

Early stopping is applied to avoid overfitting, with a learning rate reduction after five epochs[3] and stopping after ten epochs without validation loss improvement.

3 In one epoch, the full training data set passed once through the neural network.

	Batch Size	Number of Filters
Device Landing Zone Detection	2	18
Aorta Segmentation	2	18
Valve Plane Identification	2	18
Aortic Annulus Segmentation	10	24

Table 3.1: Parameter setting for the U-Net training.

3.2 DATA BASIS

A data set of 97 patients who underwent a TAVI procedure was randomly selected from the German Heart Center Berlin TAVI registry. For each patient, an annotated three-dimensional CT thorax scan is available. The annotations, shown in Figure 3.3, were generated by domain experts using a custom MeVisLab-based software prototype to obtain:

- a mask of the aorta lumen,
- the centerline through the aorta and LVOT,
- cross-sectional contours of the aorta and LVOT perpendicular to the centerline, and
- three valve plane markers, indicating the hinge points or tips of the aortic valve cusps.

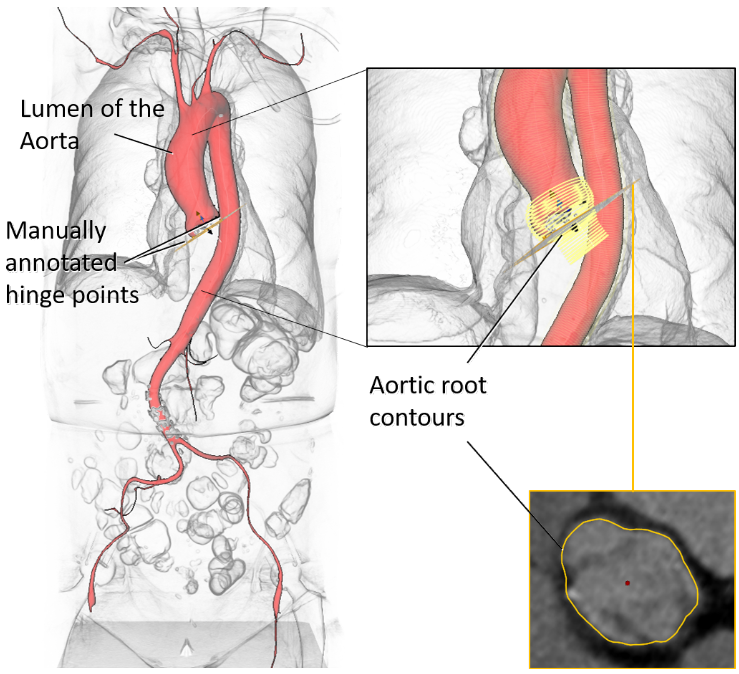


Figure 3.3: Overview of available annotations: The available expert annotations consist of a mask of the aorta lumen, the centerline of aorta and LVOT, cross-sectional contours of aorta and LVOT as well as manually placed hinge point markers.

The mask of the aorta lumen is used to generate a segmentation mask of the aorta and aortic valve, excluding the LVOT. In contrast, the cross-sectional contours are used to obtain conjunct segmentation masks of the aorta, including the LVOT, as depicted in Figure 3.4. Which of the two is selected for each individual step is noted in the respective sections within this chapter.

CSO - including the LVOT WEM - restricted to aorta and valve

Figure 3.4: Segmentations with and without LVOT: The CSO additionally comprises the LVOT, while the WEM is limited to the aorta and the aortic valve.

Seven patients were randomly selected as an individual test set, while the remaining 90 patients were split into 75 for training and 15 for validation in a six-fold cross-validation. To exploit the benefits of ensemble methods (see Section 2.2.6), the six model outputs are averaged to yield the final result. The impact of averaging the outputs is assessed in Section 4.2.1. The cross-validation is used for each of the four steps - DLZ detection, aorta segmentation, annulus plane identification and annulus segmentation.

To maintain the patients' anonymity, identifying information, such as name, date of birth or patient identifier, is replaced with a pseudonym. To retain the patient's age at the time of the TAVI procedure, it was previously calculated from the date of birth and the date of admission.

The CT scans used for this thesis have a slice extent of 512 x 512 voxels, with the number of slices ranging from 321 to 856. The number of slices is mainly affected by the voxel size (minimum 0.59 x 0.59 x 0.7 mm^3, maximum 0.65 x 0.65 x 2 mm^3).

Each image is resampled to a uniform voxel size, followed by a reduction to a sub-image of a specific extent. The details are given in the respective data preparation sections. The intensities of each input image are normalized to a range of [0, 1]. Details are given in Listing 3.1.

```
# subtract the image's minimum from each voxel value
# divide by the current intensity range
img = (img - np.min(img))/np.ptp(img)
```

Listing 3.1: Normalization

The image-wise scaling is chosen to ensure gradients of similar magnitude over different batches, such that each image has approximately the same influence on neural network weight adjustment during backpropagation. Alternative scalings are discussed in Section 5.3 - Future Work.

Several peculiarities can impede the deep learning model's training and application, some of which are visualized below.

PREVIOUSLY IMPLANTED PROSTHETIC VALVES, see Figure 3.5, appear as a very bright white ring or white dots on the CT, depending on the type of inserted valve. Due to the scaling of the image to the range between zero and one (see Listing 3.1), the rest of the image appears dark, aggravating information extraction.

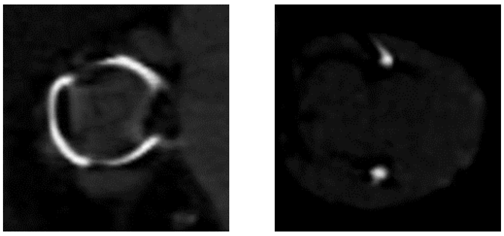
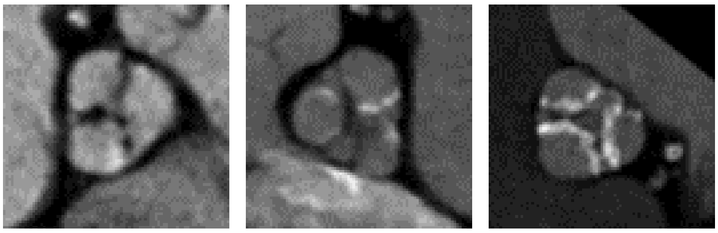


Figure 3.5: Pre-implanted prosthetic valves appear as a bright white ring or white dots.

CALCIFICATION, which is displayed in Figure 3.6, appears white on the CT scans. As each image is scaled to the range between zero and one, the valve itself appears darker if calcification is present. This fact and the calcification itself can impede the detection of the valve. Ideas for taking calcification into account are presented in Section 5.3 - Future Work.



(a) (Nearly) None (b) Mild (c) Severe

Figure 3.6: Different levels of calcification of the aortic valve.

BLURRING, as shown in Figure 3.7, can result from movement of the patient during CT-scanning, or it might be due to resampling a CT image with low resolution to a higher resolution. Consequently, the valve contours are less clear.

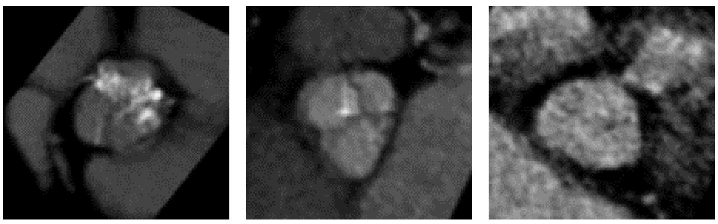


Figure 3.7: Blurry images.

All aforementioned peculiarities are present in the training data set.

3.3 DATA AUGMENTATION

Machine learning algorithms typically require an immense number of training samples to deliver satisfactory results. Brownlee [10] recommends *"Ideally, tens or hundreds of thousands for 'average' modeling problems."* In cases where a large amount of training data cannot be obtained, data augmentation techniques allow for an artificial increase of training samples by multiplying the input images through suitable transformations like rotations, translations or reflections [23]. This improves the trained model's precision and generalizability, as evaluated in Section 4.2.1.

For this thesis, the augmentation was confined to translation to preserve the CT scans orientation, as this orientation is standardized. A change in brightness and contrast was considered. However, it was dismissed immediately, as the model training in early experiments did not benefit from such augmentations. In case of three-dimensional segmentations in each epoch each training sample is randomly adjusted within the following constraints (visualized in Figure 3.8 in two dimensions):

```
shift_range = 5

xs = random.randrange(shift_range)
ys = random.randrange(shift_range)
zs = random.randrange(shift_range)

img_shifted = img[xs:x_dim+xs, ys:y_dim+ys, zs:z_dim+zs]
```

Listing 3.2: Translation

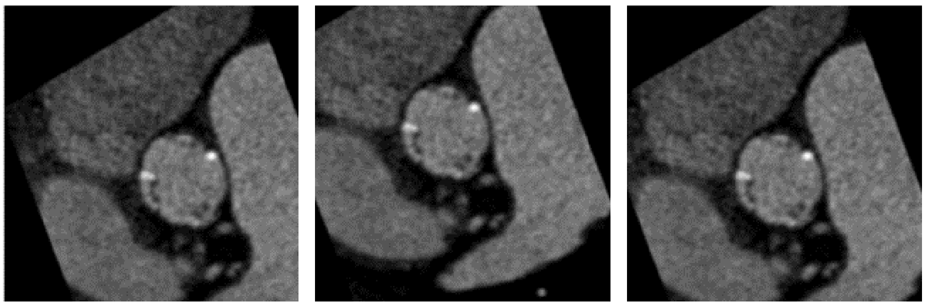


Figure 3.8: One image differently translated: The image is randomly shifted in the x- and y-direction.

For the two-dimensional segmentation in step 4, no data augmentation was used due to a sufficient number of training samples, as explained in Section 3.7.

3.4 STEP 1 - DEVICE LANDING ZONE DETECTION

The above presented U-Net architecture is now used to consecutively segment interesting areas from the patient's CT scan, resulting in a pipeline of hierarchically applied CNNs. First, the CT scan is reduced to a region of interest around the so-called Device Landing Zone (DLZ).

The DLZ was defined as the area including the aortic valve (i.e., the aortic annulus and valvular cusps) and the LVOT (until the junction point of the anterior mitral leaflet). This is the region where the artificial valve will be positioned during TAVI to replace the faulty valve. Thus, this area's anatomy is critical for properly fitting the prosthesis.

The following section presents the data basis and model setup for the DLZ detection.

Data Preparation

Firstly, the training data was prepared with MeVisLab. Initially, all input CT scans were resampled to a uniform voxel size of 2 x 2 x 2 mm^3 and clipped to a fixed size of 128 x 128 x 192 voxels. The U-Net architecture requires unified input dimensions which should be recursively divisible by two [89]; the unified resolution further facilitates common feature extraction across all input images. The above combination of voxel and image size ensures that the image contains all relevant anatomical structures while still fitting into the available working memory (see Section B.1).

Based on the centerline and the valve plane markers, an axis-aligned cubical bounding box of 68 x 68 x 68 mm^3 is placed in the image around the aortic valve. The midpoint of the three valve plane markers is selected as the midpoint of the bounding box.

This bounding box was then used as mask in a CNN to learn the detection of the landing zone, that is to say, the detection of the area around the aortic valve. The data preparation is visualized in Figure 3.9.

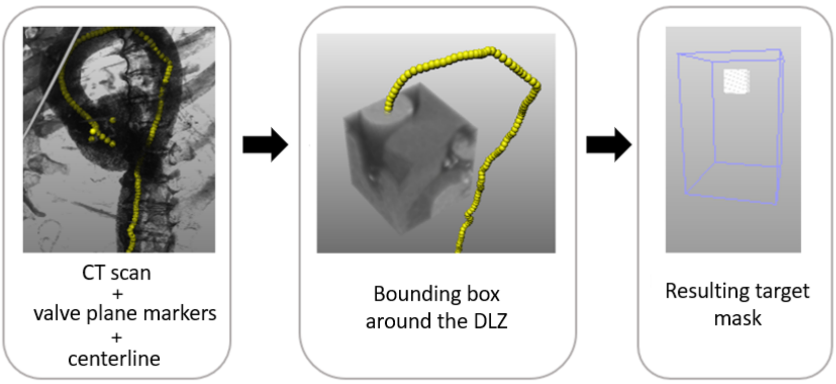


Figure 3.9: Data preparation: Basis for the training masks are the original CT scan, the valve plane markers and the centerline. A uniformly sized bounding box is centered around the valve plane markers. This bounding box inside the CT scan is used as target mask for the neural network training.

Model Specifics and Postprocessing

For each voxel in the input CT scan the CNN returns a probability for this voxel to belong to the uniformly sized bounding box around the DLZ. The probabilities are binarized with a threshold of 0.5, and a bounding box is drawn around all predicted voxels. This bounding box is resized to the desired dimensions of 68 x 68 x 68 mm^3, with the midpoint being the center of the predicted bounding box. The postprocessing step is illustrated in two dimensions in Figure 3.10.

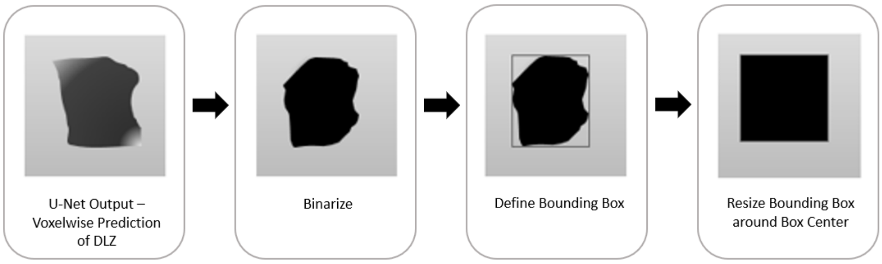


Figure 3.10: Postprocessing: The U-Net outputs voxelwise probabilities of the DLZ. This is binarized with a threshold of 0.5 and a bounding box is drawn around it. This bounding box is resized to the desired dimensions centered around its midpoint.

First experiments showed that the predicted bounding box does not always perfectly correspond to the desired bounding box. Thus, to ensure all relevant information is contained within that bounding box, a padding of 6 *mm* on both sides in x-, y- and z-direction is added to the predicted bounding box, leading to a box of size 80 x 80 x 80 mm^3. The padding of 6 *mm* was selected to result at a perfect overlay of the ground truth bounding box on the training data set. Chapter 4 - Results substantiates the decision for padding. Section 5.1 challenges this approach and mentions alternatives.

3.5 STEP 2 - AORTA SEGMENTATION

In the second step, a similar U-Net architecture as in the first step is used to learn the aorta segmentation within the DLZ.

Data Preparation

The segmentation masks are semi-automatically obtained from MeVisLab. For the aorta segmentation, the WEM is used, which only contains the aorta, including the aortic valve, but excluding the LVOT.

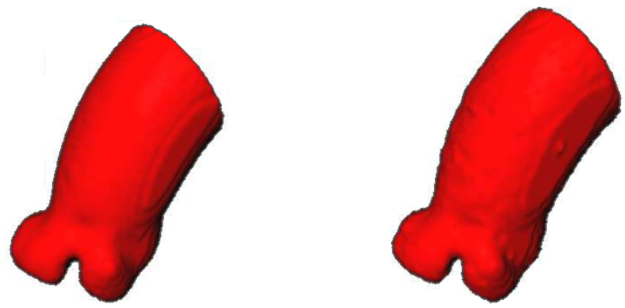
For all following steps, the original images are resampled to a uniform voxel size of 0.6 x 0.6 x 0.6 mm^3 before being cropped to the region of the DLZ. The resampling is again done to ensure maximal resolution under the hardware limitations. Aspects to consider for optimization of the voxel size are discussed in Section 5.3 - Future Work.

The segmentation masks for U-Net training are reduced to the region of the DLZ in two ways:

- First, they are reduced to the original DLZ bounding box, defined by the valve plane markers and the centerline.
- Second, they are reduced to the predicted DLZ as an output from step 1.

Thus, each patient in the training data set results in two masks for the neural network training in step 2. The original DLZs, as well as the predicted ones, were used for training to improve the model's generalizability. By training on the predicted DLZ, applicability to the outcome of step 1 shall be achieved.

Example segmentation masks are shown in Figure 3.11. It can be seen that the resulting sub-images have slightly different coverage.



Ground Truth DLZ Predicted DLZ

Figure 3.11: Segmentation masks resulting from the ground truth DLZ and the predicted DLZ from step 1: The predicted DLZ has slightly different coverage. In this example, it contains more of the ascending aorta.

Model Specifics and Postprocessing

Throughout the training process, in every epoch, each patient is used once as a training sample. It is randomly selected whether the original or the predicted DLZ is considered for each patient in each epoch anew. In Chapter 4 - Results, it is analyzed whether the model's performance benefits from taking the predicted DLZ into consideration. For this, one model which is only trained using the original DLZ is compared to a model that uses both, original and predicted DLZ for training.

In contrast to the model for step 1, the U-Net weights for this step are not initialized randomly. The weights obtained from step 1 are used for initialization to exploit already extracted information and speed up convergence. The results are compared to a model with random initialization in Chapter 4 - Results.

Except for the binarization of the network's output, no postprocessing is required in this step.

3.6 step 3 - annulus plane identification

In the third step, the region around the aortic annulus plane is segmented, with the goal to deduce the orientation and midpoint of the aortic annulus and finally enable the annulus measurements.

The aortic annulus is defined as the virtual ring passing through the three hinge points or tips of the valve cusps. Those would optimally be given by the valve plane markers. Since the valve plane markers are set manually in MeVisLab, the positioning will likely not be ideal, as shown in Figure 3.12. This leads to the assumption that regression might be a good strategy for learning the valve plane marker positions, as minor deviations would only lead to a minor increase in error. In contrast, the previously used segmentation approach cannot readily differentiate between slight deviations from the correct valve plane marker positions and completely wrong predictions.

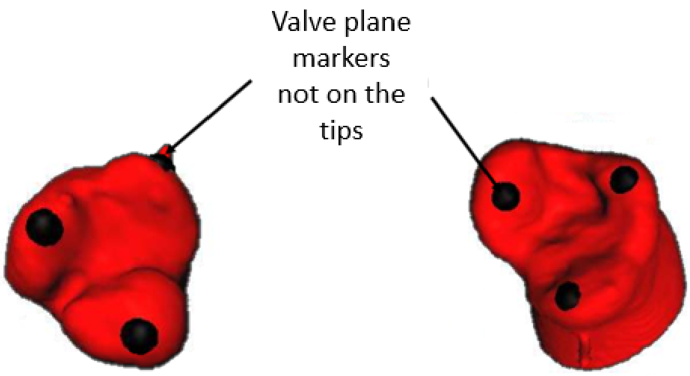


Figure 3.12: Examples of incorrect marker positions on segmented aortic valves: Some of the valve plane markers (black) are not positioned at the tips of the aortic cusps.

regression

Two initial approaches were pursued:

1. Using the coordinates of the three valve plane markers as targets, leading to nine output variables (x, y, z coordinates for each marker)
2. Using the valve plane markers' midpoint and the normal vector to the plane defined by the markers as targets, leading to six output variables (x, y, z coordinates of the midpoint and normal vector).

For both approaches, a ResNet was employed. ResNet, short for Residual Network, is a neural network architecture that achieved tremendous results in several visual recognition tasks [31]. However, the predictions were insufficient. Also, a simple CNN with two hidden layers was tested on both regression approaches, yielding the same outcome. Regression was thus primarily dismissed, and a segmentation approach was further explored. For details on the regression approach see Appendix A.

segmentation

In a first attempt, a U-Net was used to detect the three valve plane markers by segmentation. A padding of different sizes was added to each of the three markers in order to increase the marker size. This simplifies the segmentation task and accounts for the fact that the positioning of the valve plane markers is not ideal, and at the same time, a minor change only leads to a minor change in the resulting plane. The padded markers were used as target mask in the same U-Net architecture as for the previous two steps. The results again were insufficient, see Appendix A.

The task was further simplified by using the aortic root region within the annulus as mask, leading to the chosen strategy for step 3, which will be explained in the following subsections.

Data Preparation

The mask for segmentation of the aortic annulus region is prepared based on the manually defined valve plane markers. For each CT scan, the plane containing the three valve plane markers is identified. The three-dimensional aorta segmentation CSO that lies within 15 slices in the direction of the plane normal is then used to train a U-Net. One slice has a thickness of one voxel. The CSO also includes the LVOT and is hence used here to yield a segmentation mask including aortic valve and LVOT, with the annulus plane in the center. The resulting mask is visualized in Figure 3.13. This three-dimensional approach intends to make the segmentation robust against the input uncertainty of the valve plane markers. Additionally, usage of only one slice and 30 slices was evaluated but not further pursued, as justified in Chapter 4 - Results.

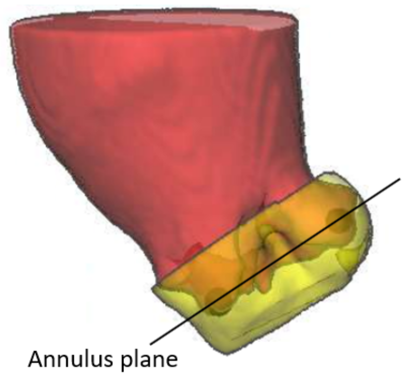


Figure 3.13: Segmentation mask (yellow), aligned along the annulus plane, defined by the valve plane markers (black) with the aorta segmentation (red) for reference.

Model Specifics and Postprocessing

The weights for this step's U-Net are initialized by the weights obtained from the previous aorta segmentation step. The number of voxels belonging to the mask is much smaller than the number of background voxels. On average, 19 500 voxels belong to the mask, in case 15 slices are considered, while 2 million voxels belong to the background. In such cases with high imbalance, it is advisable to use a weighted binary cross-entropy since an unweighted one will likely predict the more frequent class. A typical weighting is given by the inverse average class frequency [85]. This was applied here.

Deduction of the Annulus Plane

Principal Component Analysis (PCA) is applied to deduce the annulus plane from the predicted segmentation. PCA is a dimensionality reduction method, which reduces a data set to its uncorrelated components that maximize variance [37]. The PCA is visualized in Figure 3.14.

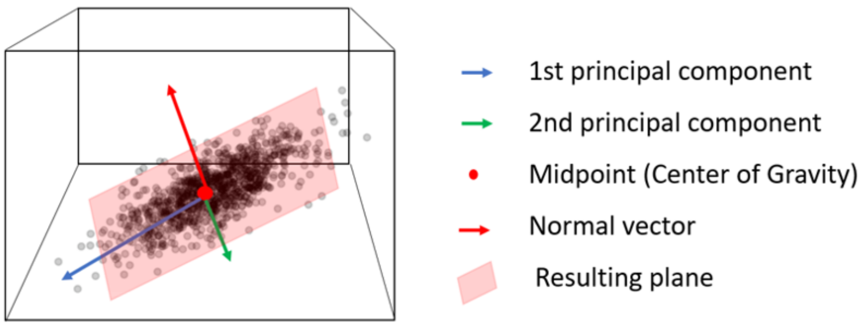


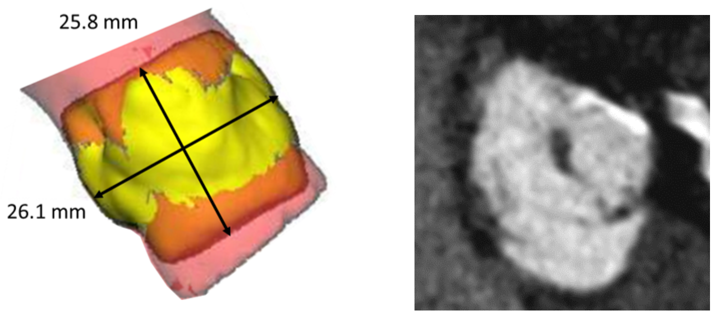
Figure 3.14: Principal component analysis: The first two principal components maximize the variance of the underlying data while being perpendicular to each other. A plane results from the normal vector orthogonal to the two principal components. The plane's midpoint is defined by the center of gravity of the underlying data.

The diameter of the segmentation mask along the annulus plane exceeds its height. Thus, the two components with the highest variance describe the orientation of the annulus plane. Therefore, the vector orthogonal to the two largest eigenvectors of the PCA is taken as an estimate of the plane's normal vector. The center of gravity of the predicted segmentation estimates the midpoint of the annulus plane. This midpoint and normal vector finally allow for approximation of the annulus plane. The MeVisLab module *XMarkerPCA* automatically calculates both values after conversion of the segmentation mask to marker positions with the module *MaskToMarkers*.

Improvement of the annulus orientation

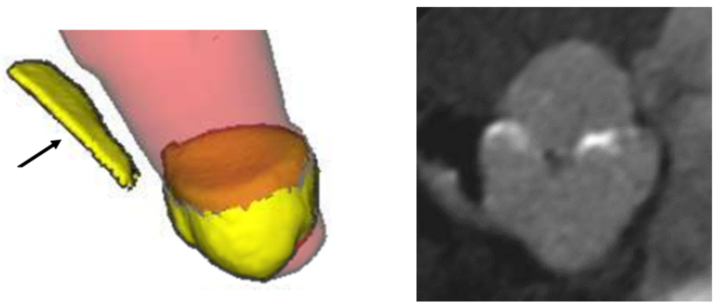
Initial experiments on the training data revealed two major issues in identifying the correct plane orientation:

- A segmented aortic valve region that is too thick, such that the PCA does not yield the desired normal vector, as visualized in Figure 3.15,
- Artifacts in the segmentation distorting the PCA, see Figure 3.16.



(a) Segmentation (b) Resulting plane

Figure 3.15: A segmentation that has a comparable height and diameter: The minimal and maximal diameter should exceed the height to ensure correct calculation of the plane's normal vector in PCA. The plane resulting from this inordinately thick segmentation has a wrong orientation, showing the valve leaflets from the side.



(a) Segmentation (b) Resulting plane

Figure 3.16: Artifact in the segmentation, marked by an arrow. Again the resulting plane orientation is incorrect, showing the valve leaflets from the side.

To ensure the correct orientation is detected, the predicted segmentation is masked with the contour of the three-dimensional aorta segmentation from step 2, illustrated in Figure 3.17. Further, any undesired segments are removed by only considering the largest connected component. This ensures that the input into the PCA is more optimally aligned along the annulus plane, albeit also shifting the center of gravity further into the valve. The effect of this masking is quantitatively analyzed in Chapter 4 - Results.

Viewing the resulting planes anew shows that the orientation now appears satisfactory. A sample is shown in Figure 3.19a.

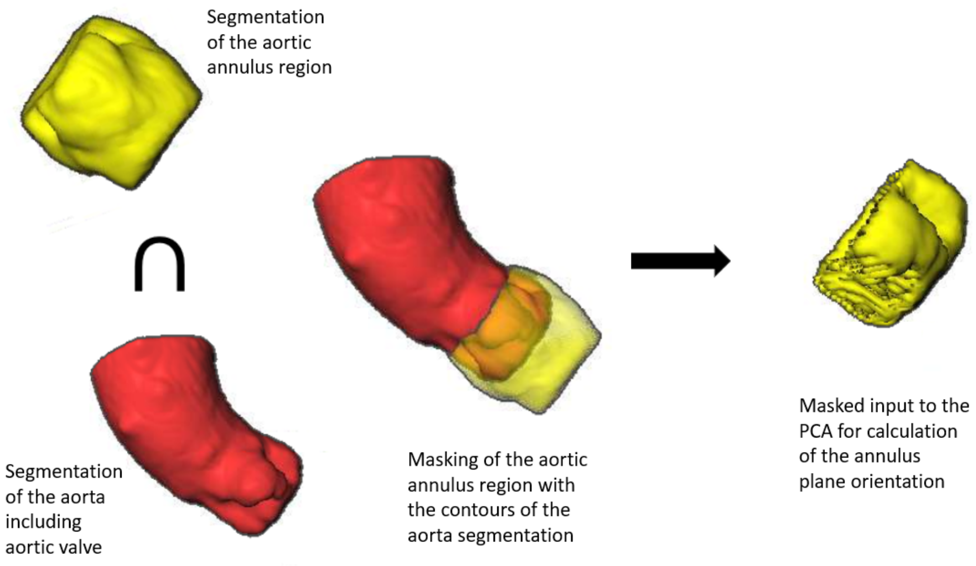
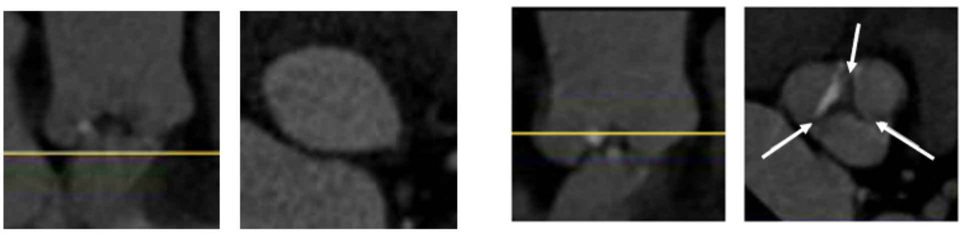


Figure 3.17: The segmentation of the annulus region is masked with the contours of the predicted aorta segmentation from step 2. The resulting segmentation is used as input to the PCA for calculation of the valve plane orientation.

Adjustment of the annulus midpoint

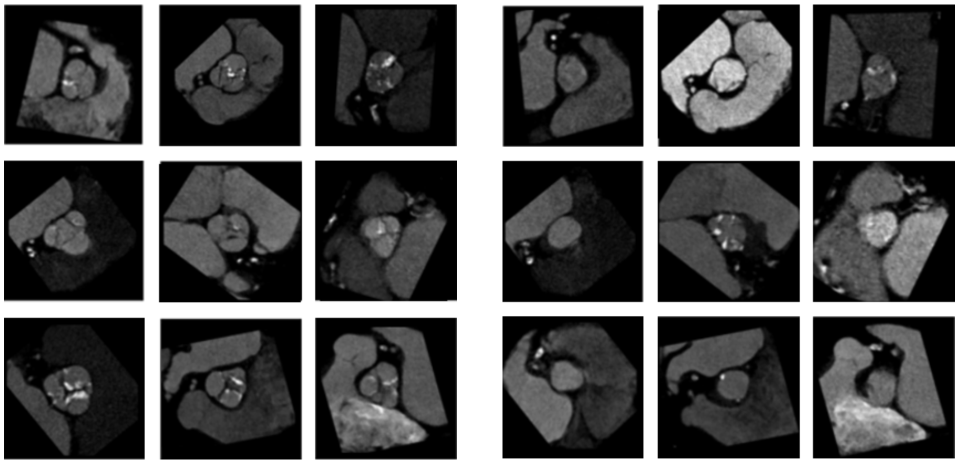
However, the detected midpoint often still lies within the valve and not just below on the annulus plane, which is used for area and perimeter measurement. As shown in Figure 2.2, the annulus plane is just below the aortic valve at the leaflet attachments. Figure 3.18 contrasts an annulus plane with a plane inside the aortic valve. It can be seen that the plane in Figure 3.18b is positioned inside the valve by the three indentations resulting from the valve leaflets, marked with arrows.



(a) Annulus plane (b) Plane inside valve

Figure 3.18: A correctly detected annulus plane in comparison to a plane inside the aortic valve; white arrows mark the leaflet indentations. A cross-section shows the position of the plane.

Thus, with the current outcome, the plane needs to be shifted further down to lie on the annulus. This is attempted by again taking into account the three-dimensional aorta segmentation and selecting the slice just below the deepest segmented point. Visual examination shows that this generally leads to a plane below the aortic valve, see Figure 3.19b.



(a) Before shift (b) After shift

Figure 3.19: Sample of planes resulting from valve plane markers: Before the plane is shifted according to the aorta segmentation, the planes are located inside the valve. After the shift, no indentations from the valve cusps can be identified, suggesting a positioning of the planes below the aortic valve.

3.7 STEP 4 - AORTIC ANNULUS SEGMENTATION AND MEASUREMENT

After the annular plane has been detected in the previous step, the annulus is segmented and finally measured. A U-Net applying two-dimensional operations is used for the segmentation. Instead of segmenting the annulus in two dimensions, it would also be possible to directly consider the segmented region from the previous step in the annulus plane. The two approaches are compared in Chapter 4 - Results.

Data Preparation

The two-dimensional segmentation masks are obtained by taking 30 consecutive slices from the three-dimensional aorta segmentation mask, including the LVOT of each patient, parallel to the valve plane, as visualized in 3.20. The most central slice is the one where the valve plane markers are positioned. This leads to $90 * 30 = 2700$ train samples. Data augmentation is omitted for this step. Each slice has a size of 256 x 256 voxels to ensure recursive divisibility by two [89], leading to a voxel size of 0.361 x 0.361 x 0.361 mm^3.

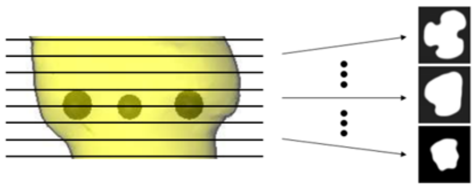


Figure 3.20: Data preparation for two-dimensional annulus segmentation: 30 consecutive slices from the aorta and LVOT segmentation (yellow) parallel to the valve plane markers (black).

Model Specifics and Postprocessing

A U-Net with similar layers as for the previous tasks is used, except for the fact that all operations are now applied in two dimensions.

To ensure the correct area is measured, only the most central connected component is regarded for measurement, as this should portray the annulus.

The area and perimeter of the annulus resulting from this two-dimensional segmentation are obtained with MeVisLab. A contour is drawn around the segmentation, using the MeVisLab module *CSOIsoGenerator*. The module *CSOInfo* then automatically calculates the area and perimeter.

4 RESULTS

This chapter first briefly summarizes the data basis for model development and benchmarking. Thereupon, the trained models for each step are evaluated, and the evaluation criteria are presented. Finally, the obtained measurements are compared with the benchmark.

4.1 PATIENT ANALYSIS - DATA BASIS FOR MODEL DEVELOPMENT VERSUS BENCHMARKING

The training data set for model development comprises 90 patients. An additional seven patients are used for the evaluation of each step. The final pipeline for annulus measurement is benchmarked against the two software solutions on 100 patients. Each patient was initially randomly selected from the German Heart Center Berlin TAVI registry.

The considered patients for training, evaluation and benchmarking show comparable demographics and procedural data, shown in Table 4.1. This suggests applicability of the developed model to the benchmarking data set. The procedural data is limited to the interesting aspects for annulus measurement - prosthesis size and pre-implanted aortic valves. For five patients in the training set, no information could be obtained of the procedural data, and for three patients, no information at all was available. In the evaluation set, procedural information for one patient was missing.

	mean (std) / count (%)					
	Training (90 Patients)		Evaluation (7 Patients)		Benchmark (100 Patients)	
Female	55	(63%)	5	(71%)	54	(54%)
Age [*years*]	80.6	(9.66)	77.7	(5.09)	79.2	(7.48)
Height [*cm*]	164	(8.32)	165	(5.05)	167	(8.78)
Weight [*kg*]	75.6	(17.6)	77.4	(25.0)	76.5	(19.5)
Body mass index $[kg/m^2]$	27.9	(5.99)	28.3	(8.38)	27.4	(7.10)
Body surface area $[m^2]$	1.85	(0.23)	1.87	(0.31)	1.81	(0.42)
Prosthesis size [*mm*]:						
20	1	(1.2%)	0	(0%)	0	(0%)
23	10	(11.8%)	0	(0%)	17	(17%)
25	3	(3.6%)	0	(0%)	9	(9%)
26	27	(31.8%)	3	(50%)	25	(25%)
27	5	(5.9%)	0	(0%)	12	(12%)

29	37	(43.5%)	2	(33.3%)	36	(36%)
34	2	(1.2%)	1	(16.7%)	1	(1%)
Previous valve implanted	5	(5.9%)	0	(16.7%)	2	(2%)

Table 4.1: Comparison of patients in the training, evaluation and benchmarking data set. Each set shows comparable demographics and procedural data.

4.2 TRAINED MODELS

For each step, an individual predictor is obtained by training six structurally identical models in a six-fold cross-validation and averaging the outputs. Depending on the underlying task, different evaluation criteria are applied. In the following subsections, the criteria are explained and the results evaluated accordingly.

4.2.1 *Step 1 - Device Landing Zone Detection*

In the device landing zone detection, a bounding box around the aortic valve centered around the valve plane markers is found by segmentation.

It should be noted that an optimal overlay of the detected bounding box to the mask does not necessarily result in optimal information content. Due to anatomical differences, the truly optimal bounding boxes might be of different sizes or different orientations. However, for simplicity, a uniform bounding box was selected, which demonstrates sufficient results.

Evaluation Criteria

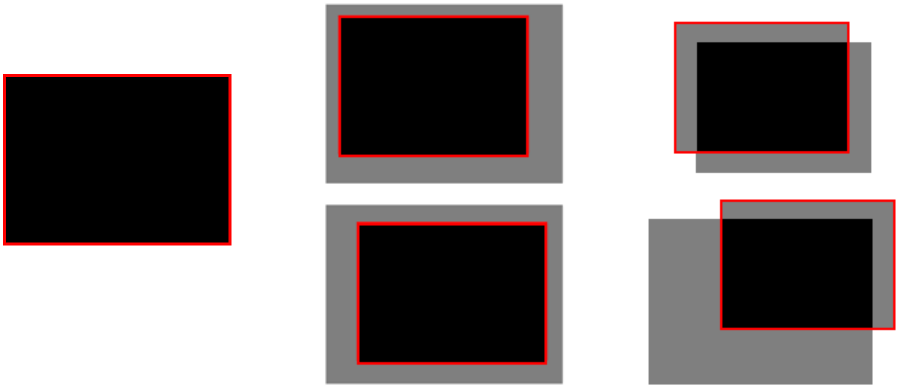
The device landing zone detection is achieved by segmenting the respective region.

Since the DLZ is required in the following segmentation steps, it is essential to avoid completely wrong predictions. As the IoU tends to penalize single instances of bad classification more than the F1 score quantitatively (see Section 2.2.4), the device landing zone detection is evaluated against a modification of the IoU (Equation 2.4).

It is desired to cover the full ground truth bounding box with the predicted segmentation to ensure that the whole aortic valve is contained in the resulting sub-image. The ground truth always refers to the respective masks for each step obtained from the annotated data set. A padding is later added to the predicted bounding box to ensure full coverage of the aortic valve. For clarification, this is visualized in Figure 4.2. Hence, for proper evaluation, the intersection is not divided by the union but by the ground truth bounding box.

In the following, this metric is called Intersection–over–Ground-Truth (IoGT).

$$IoGT = \frac{Intersection}{GroundTruth} \tag{4.1}$$



(a) Perfect without padding (b) Sufficient with padding (c) Insufficient without and with padding

Figure 4.1: Coverage of the ground truth bounding box (red outline).

It is not directly applied to the predicted segmentation but to the bounding box obtained after thresholding and reshaping, as explained in Figure 3.4.

Evaluation

Table 4.2 portrays the IoGT on the DLZ bounding boxes for the final model - the averaged segmentations from all six models in the cross-validation. Table 4.3 shows the IoGT for the individual models as well as the average of their IoGTs.

	Training (90 Patients)	Test (7 Patients)
IoGT	0.959	0.896

Table 4.2: IoGT on bounding box obtained from final model for step 1.

Model	Training (75 Patients)	Validation (15 Patients)
1	0.946	0.886
2	0.946	0.885
3	0.962	0.893
4	0.955	0.908
5	0.951	0.854
6	0.951	0.886
avg	0.952	0.885

Table 4.3: IoGT for each model in six-fold cross-validation and the averaged IoGTs.

As can be seen from the IoGT, a full coverage of the ground truth bounding box cannot be guaranteed. Thus, it was decided to pad the bounding box with three voxels in each direction to obtain an IoGT of 1 on the training set, leading to the following IoGT results:

	Training (90 Patients)	Test (7 Patients)
IoGT	1	0.995

Table 4.4: IoGT on padded bounding box obtained from final model for step 1.

Model	Training (75 Patients)	Validation (15 Patients)
1	1	0.962
2	1	0.994
3	0.999	1
4	1	0.995
5	0.999	0.979
6	1	0.989
avg	1	0.987

Table 4.5: IoGT for each model in six-fold cross-validation on padded bounding box and the averaged IoGTs.

The ground truth bounding box has a volume of 314 500 mm^3 (1.25% of the image), while the padded bounding box has a volume of 512 000 mm^3 (2% of the image). To ensure that the sub-images contain all required information for the following steps, this volume increase is accepted, and the padded bounding boxes are used hereafter.

Moreover, the impact of data augmentation is assessed by evaluating a model that was trained without any augmentation. Tables 4.6 and 4.7 show the IoGT without padding of the resulting bounding boxes.

	Training (90 Patients)	Test (7 Patients)
with augmentation	0.959	0.896
without augmentation	0.955	0.885

Table 4.6: IoGT obtained from final model for step 1 with and without data augmentation.

It is evident that the data augmentation improves the outcome and leads to more robust individual models without any high discrepancy in between model IoGTs. The results on the test set improve, suggesting enhanced generalizability. Furthermore, the positive impact of averaging the six model outputs for a final prediction can be observed by comparing the IoGT of the averaged model to the average of the individual models' IoGT.

Consequently, data augmentation and model averaging are utilized for each step without further assessment for the following steps.

Model	Training (75 Patients)	Validation (15 Patients)
1	0.903	0.912
2	0.956	0.866
3	0.971	0.887
4	0.972	0.886
5	0.847	0.771
6	0.954	0.904
avg	0.934	0.871

Table 4.7: IoGT for each model in six-fold cross-validation without data augmentation.

4.2.2 *Step 2 - Aorta Segmentation*

In the second step, the aorta, including the aortic valve within the obtained bounding box, is segmented.

Evaluation Criteria

Several circumstances might impede the segmentation, such as pre-implanted prostheses, anatomical anomalies or other medical or technical artifacts. For this task, an optimal solution for the average patient CT is desired while paying less regard to such exceptional cases. Thus, since the model should focus less on outliers, the F1 score (Equation 2.5) is the metric of choice, as explained in Section 2.2.4.

Evaluation

Table 4.8 portrays the F1 score in the case that the models were trained both on the ground truth DLZ calculated by the valve plane marker positions and the derived DLZ bounding box from the previous step. Each is enlarged with a padding to achieve bounding boxes of similar size.

	Training (90 Patients)		Test (7 Patients)	
	True DLZ	Predicted DLZ	True DLZ	Predicted DLZ
F1	0.952	0.952	0.947	0.944

Table 4.8: F1 score obtained from final model for step 2 with predicted and ground truth DLZ.

Table 4.9 shows the results evaluated on the ground truth and the predicted DLZ in case the models are only trained on the ground truth DLZ. As was to be expected, the results only differ marginally in case both are used, as the data augmentation already includes random shifts, which allows for better generalizability. Hence, in the following, the models' training and evaluation are confined to the ground truth DLZ for simplicity.

	Training (90 Patients)		Test (7 Patients)	
	True DLZ	Predicted DLZ	True DLZ	Predicted DLZ
F1	0.953	0.952	0.946	0.944

Table 4.9: F1 score obtained from final model for step 2 with ground truth DLZ.

Further, the models' weights are initialized with the final weights from the previous step to facilitate pattern recognition within the CT scans. This is also compared to the results with random weight initialization, which can be found below.

	Training (90 Patients)		Test (7 Patients)	
	True DLZ	Predicted DLZ	True DLZ	Predicted DLZ
F1	0.953	0.951	0.945	0.948

Table 4.10: F1 score obtained from final model for step 2 with ground truth DLZ with random weight initialization.

The difference in model accuracy resulting from weight initialization is negligible. However, the number of required epochs for model convergence is drastically reduced by usage of the weights from the previous step. With random initialization on average 38 epochs elapse before early-stopping while the weight initialization with the pre-trained weights reduces this number to 22.5.

4.2.3 *Step 3 - Annulus Plane Identification*

In the annotated data set, the annulus plane is determined by the three valve plane markers. However, those valve plane markers are not always optimally positioned and proved to be rather difficult to identify within the image. Consequently, a more extensive mask is employed from which the plane is deduced by principal component analysis, yielding a midpoint and a normal vector to the plane.

This final approach is evaluated in detail, while for all other attempts, an explanation and evaluation can be found in Appendix A.

Evaluation Criteria

All approaches to the annulus plane identification are evaluated by calculating the Mean Absolute Error (MAE) between the predicted midpoint and normal vector and the ground truth, obtained from the three valve plane markers, converting the segmentation task to a regression. Here the ground truth refers to the midpoint and normal vector obtained from the three valve plane markers. The midpoint is calculated by taking the mean over the valve plane marker positions within each image and dividing by the image dimensions. The normal vector is obtained by taking the cross product between two vectors connecting the valve plane markers and normalizing the resulting vector, see Listing 4.1.

```
# m is an array containing the indices of the valve plane markers within
    the image

midpoint = np.mean(m, axis=0)

orientation = np.cross(m[1]-m[0], m[2]-m[0])
normal_vector = abs(orientation/np.linalg.norm(orientation))
```

Listing 4.1: Ground truth midpoint and normal vector

The absolute error is favored over the squared error since outliers should not be excessively penalized, as these might be due to inaccuracies in the annotation. The errors for both variables are considered independently to examine plane orientation and position individually. To allow for proper assessment of the errors, they are contrasted against the errors resulting from a naive prediction - the arithmetic means of midpoint and normal vector of the training data set.

Evaluation

Table 4.11 shows the MAE compared to the benchmark for the midpoint and the normal vector, respectively. An extent of one, 15 and 30 slices is compared. An extent of 30 slices shows the worst result. For an extent of one and 15 slices, it is further attempted to improve the detected plane orientation with the following two measures. First, the predicted segmentation is masked with the three-dimensional aorta segmentation from step 2 (Seg); second, it is reduced to the largest connected component (Largest CC).

	Training (90 Patients)		Test (7 Patients)	
	Midpoint	Normal	Midpoint	Normal
30 Sclices	0.040	0.494	0.054	0.377
1 Slice	0.035	0.320	0.044	0.268
+ Seg	0.052	0.299	0.058	0.290
+ Largest CC	0.052	0.298	0.058	0.290
15 Slices	0.038	0.330	0.039	0.318
+ Seg	0.061	0.283	0.058	0.314
+ Largest CC	0.061	0.283	0.057	0.314
Training mean	0.039	0.376	0.033	0.297

Table 4.11: MAE for step 3 of the averaged model outputs.

It can be seen that the combination of model prediction and aorta segmentation improves the deduction of the normal vector while impairing the midpoint prediction. Reduction to the largest connected component slightly improves the results.

Evaluation on the training set shows the best results for an extent of 15 slices after the additional adjustments. On the test set, however, the results are worse than the model with an extent of one slice and even worse than the benchmark. Visual examination, however, shows that this mainly results from inaccurately placed valve plane markers on the small test set of only seven patients. After careful consideration, the model with an extent of 15 slices adjusted with the three-dimensional aorta segmentation, reduced to the largest connected component, is used in the further course.

For future work, it is advisable to reevaluate different extents with a more extensive annotated data set to optimize the annulus plane detection.

4.2.4 *Step 4 - Aortic Annulus Segmentation*

The annulus segmentation is approached as a simple segmentation task in the two-dimensional plane obtained in step 3. For each patient 30 consecutive slices are used for training and evaluation. From this segmentation, the annulus perimeter and area are measured. This final step is examined on the benchmark data set in Section 4.3.

Evaluation Criteria

For evaluation of the annulus segmentation the F1 score (Equation 2.5) is used. Similar to the three-dimensional aorta segmentation in step 2, outliers should not be overly penalized, but the segmentation should be optimized for the average patient CT scan.

Evaluation

Table 4.12 shows the F1 score of the two-dimensional U-Nets applied to the annulus segmentation task.

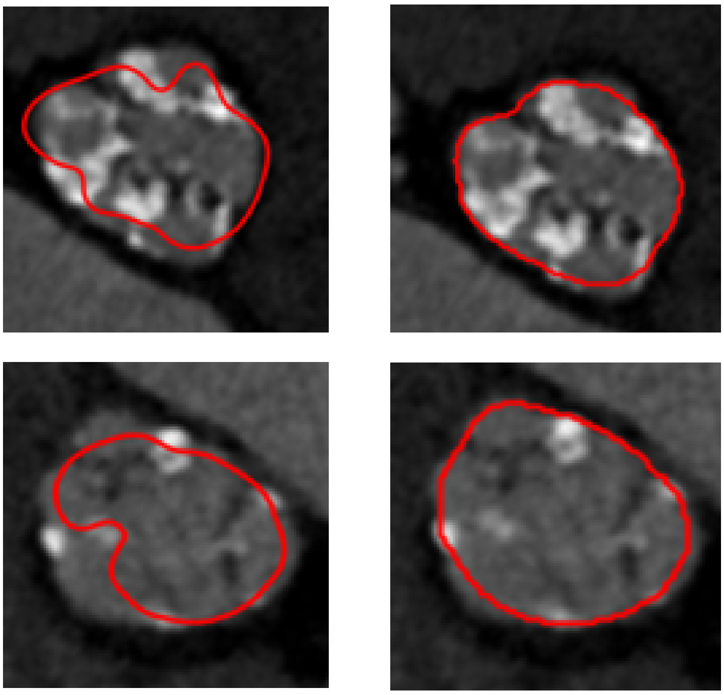
	Training (90 Patients à 30 images)	Test (7 Patients à 30 images)
F1	0.921	0.929

Table 4.12: F1 score obtained from final model for step 4.

Compared to the three-dimensional segmentation task (Section 4.2.2), the results are surprisingly poor. Visual examination reveals incorrectly annotated training data, see for example Figure 4.2a. However, it can be seen that the models still segment the desired areas, exemplified in Figure 4.2b.

Therefore, the model is improved with a variant form of self-training, see Section 2.2.7. The predicted segmentations are again used as training input for the models pre-trained on the annotated data.

Since there is no proper ground truth guaranteed, the models are subsequently only evaluated against the test set, for which the annotations have been manually corrected. It should be noted that also in Table 4.12 the test set had already been corrected.



(a) Incorrect annotations (b) Good predictions

Figure 4.2: Despite incorrect annotations, the U-Net predicts the desired annulus segmentation.

Table 4.13 shows the F1 score of the averaged predictions of the test set. The results are further improved by utilizing manually annotated valve planes as training input. For each patient, the plane indicated by the valve plane markers is used. Each model is retrained on the respective training subset, starting with the weights obtained from the self-training step. The resulting F1 score can be found in Table 4.13. Additionally, it is compared to the annulus segmentation from the previous step by considering the three-dimensional segmentation in the two-dimensional annulus plane.

	Test (7 Patients à 30 images)
Self-training	0.934
Re-annotated annulus	0.952
Segmentation from step 3	0.790
No further training	0.929

Table 4.13: F1 score obtained from final model after further training in comparison to the annulus plane segmentation obtained from step 3 and the initial model results without further training.

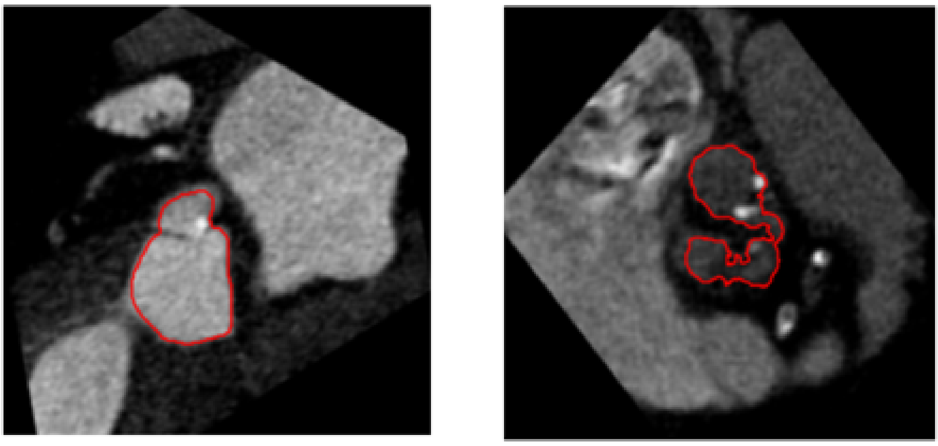
While the self-training improves the model's ability to segment the aortic valve, the properly annotated data still further enhances the model. In comparison, the segmentation from step 3 is inferior, and the additional step of two-dimensional annulus segmentation is adopted for the final pipeline.

The pipeline's suitability for medical application is verified in Section 4.3 - Comparison to the Benchmark by obtaining the annulus perimeter and area from the segmentation and benchmarking against the two software tools.

4.3 COMPARISON TO THE BENCHMARK

Meyer et al. [56] evaluated two software solutions for TAVI planning, the HeartNavigator and 3mensio, by obtaining measurements of annulus perimeter and area on 100 patients randomly selected from the German Heart Center Berlin TAVI registry. The above presented deep learning model is now benchmarked against those two software solutions. All four steps explained in Chapter 3 - Methods are sequentially applied to the same 100 patients' CT scans. Ultimately, the annulus perimeter and area are deduced and compared to the results obtained from HeartNavigator and 3mensio. The measurements of the two software tools are available as used for "Reliability and Influence on Decision Making of fully-automated vs. semi-automated Software Packages for Procedural Planning in TAVI." [56].

The two-dimensional segmentations obtained from step 4 are reduced to the connected component in the image center in order to disregard erroneous segments. In two cases, the center of the image does not belong to the segmentation. In those cases, the largest connected component is regarded as the annulus. The two cases are shown in Figure 4.3.



(a) (b)

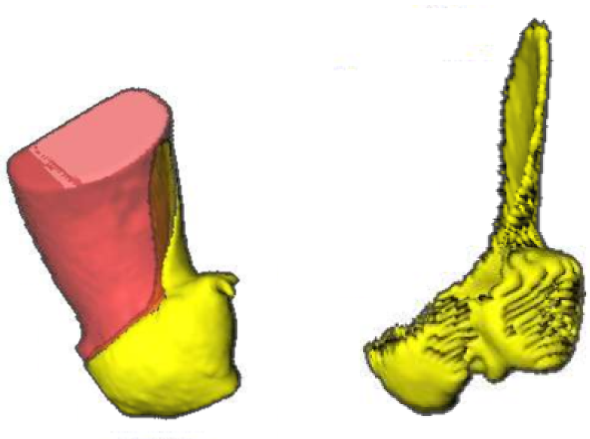
Figure 4.3: The largest connected component is considered as annulus: In these two cases the resulting segmentation does not contain the image center. Instead of the central connected component, the largest connected component is retained.

By proceeding in this way, measurements can be obtained for each of the 100 patients. The results are compared to the two software solutions in Table 4.14.

N = 100	Tool	mean ± std	min	max
Area $[mm^2]$	Deep Learning	535.4 ± 108.4	302.4	837.9
	HeartNavigator	481.5 ± 96.2	299.9	750.1
	3mensio	463.5 ± 88.0	219.9	702.5
Perimeter $[mm]$	Deep Learning	92.5 ± 13.5	70.5	149.4
	HeartNavigator	79.3 ± 7.8	63.1	99.7
	3mensio	77.2 ± 7.3	52.7	94.5

Table 4.14: Statistics of Deep Learning, HeartNavigator and 3mensio measurements.

In Figure 4.3a the orientation and position of the detected valve is incorrect, resulting from an inaccurate input into the PCA, see Figure 4.4. The segmentation of the annulus area erroneously extends into the aorta, superimposing the aorta's contours. Thus, masking with the contours still results in an incorrect input to the PCA, distorting the results. No apparent reason for this incorrect segmentation can be identified.



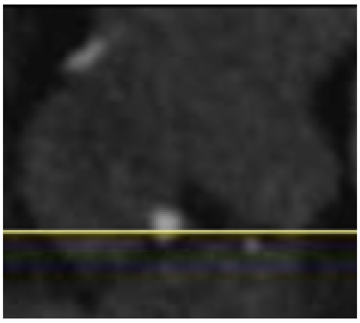
(a) Aorta segmentation (red) and incorrect segmentation of the annulus area (yellow)

(b) Resulting masked segmentation extending into the aorta

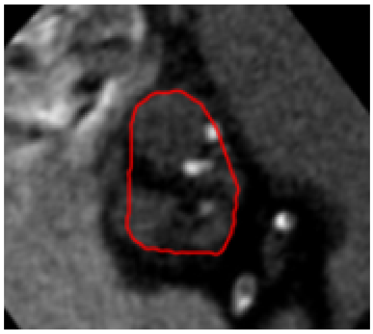
Figure 4.4: An incorrect segmentation of the annulus area (a) results in an incorrect input into the PCA (b) for the case of Figure 4.3a.

In Figure 4.3b, the detected plane is plausible, as can be seen from the cross-section in Figure 4.5a, although it might be optimized by a slight rotation. The detected plane is located on the hinge points of the valve leaflets. However, the annulus contours cannot be easily identified. Calcification and blurriness impede the segmentation. In the resulting segmentation after binarization, the center is not recognized as part of the desired segmentation. As a result, indentations distort the contour.

In the following, a smooth outer contour around the annulus shall be ensured for all cases. Thus, a two-dimensional convex hull around each segmentation is used for obtaining the measurements anew. The results are shown in Table 4.15 and visually exemplified in Figure 4.5b. The segmentation for the problematic case in Figure 4.3b improves. However, for optimal annulus segmentation, further fine-tuning is required. Approaches are suggested in Section 5.3 - Future Work.



(a) Cross-section



(b) Outer contour with convex hull

Figure 4.5: Cross-section and application of the convex hull for the case of Figure 4.3b.

N = 100	Tool	mean ± std	min	max
Area $[mm^2]$	Deep Learning	554.9 ± 115.4	317.4	881.7
	HeartNavigator	481.5 ± 96.2	299.9	750.1
	3mensio	463.5 ± 88.0	219.9	702.5
Perimeter $[mm]$	Deep Learning	88.3 ± 95.2	69.0	114.7
	HeartNavigator	79.3 ± 7.8	63.1	99.7
	3mensio	77.2 ± 7.3	52.7	94.5

Table 4.15: Statistics of Deep Learning, HeartNavigator and 3mensio measurements with convex hull.

To better understand the deep learning model's suitability, the number of cases in which the measurements obtained by the model are smaller or bigger than the respective software solutions' measurement is assessed.

		Deep Learning	
		smaller	bigger
Area $[mm^2]$	HeartNavigator	15	85
	3mensio	11	89
Perimeter $[mm]$	HeartNavigator	6	94
	3mensio	1	99

Table 4.16: Comparison of Deep Learning, HeartNavigator and 3mensio measurements with convex hull.

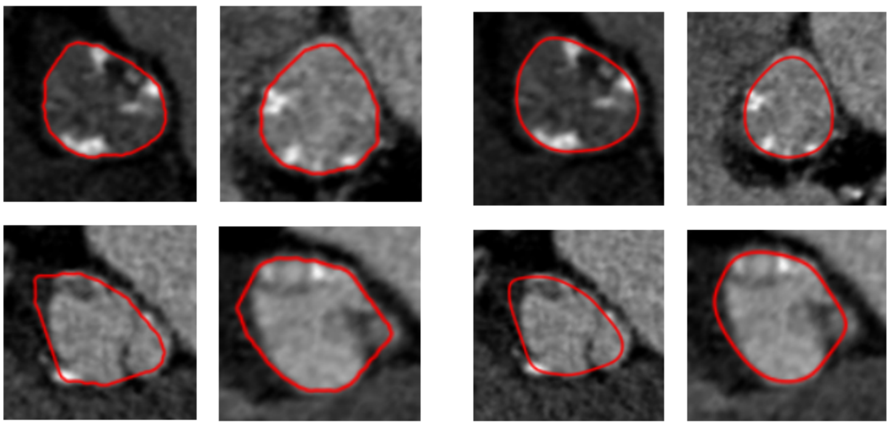
The measurements obtained by the deep learning model are in between the measurements of the two software tools in ten cases for the annulus area and in five cases for the annulus perimeter.

Figure 4.6a shows that the resulting annulus contours are still somewhat uneven, hence overestimating the annulus perimeter. MevisLab offers a built-in method for the so-called Laplacian smoothing [55]. In Laplacian smoothing, each vertex of the contour is shifted from its original position by a smoothing factor towards the average position of the neighboring vertices. The smoothing range determines the number of neighbors considered. The smoothing effect can be amplified by repeatedly applying the process in several passes [51]. This smoothing technique is exemplarily applied with the following parameters to counteract the uneven contours:

Parameter	Value
Number of Passes	20
Smoothing Factor	0.5
Smoothing Range	3

Table 4.17: Example parameters for Laplacian smoothing.

The effect is visualized in Figure 4.6b and quantitatively reported in Table 4.18 and Table 4.19.



(a) Uneven edges (b) Edges after smoothing

Figure 4.6: Effect of Laplacian smoothing: Uneven edges are smoothed, resulting in a reduced perimeter but also a reduced area.

Figure 4.6a shows the original convex contours, which have some irregularities. In Figure 4.6b the contours are smoothed, resulting in a reduced perimeter but also a reduced area.

Linsen [51] highlights a disadvantage of Laplacian smoothing - shrinkage in volume. The deep learning model frequently overestimates the measurements. Thus, Laplacian smoothing suggests an improvement in predicted measurements, which also results from the shrinkage and is not only an improvement in annulus segmentation. Hence, optimal parameters for Laplacian smoothing cannot trivially be deduced by optimization of the predicted measurements. Volume preserving alternative smoothing techniques, such as rescaling to the original volume, are likely more suitable but not in the scope of this thesis.

N = 100	Tool	mean ± std	min	max
Area $[mm^2]$	Deep Learning	543.2 ± 115.4	307.0	869.9
	HeartNavigator	481.5 ± 96.2	299.9	750.1
	3mensio	463.5 ± 88.0	219.9	702.5
Perimeter $[mm]$	Deep Learning	83.9 ± 9.3	65.0	11.0
	HeartNavigator	79.3 ± 7.8	63.1	99.7
	3mensio	77.2 ± 7.3	52.7	94.5

Table 4.18: Statistics of Deep Learning, HeartNavigator and 3mensio measurements after Laplacian Smoothing.

		Deep Learning	
		smaller	bigger
Area $[mm^2]$	HeartNavigator	18	82
	3mensio	17	83
Perimeter $[mm]$	HeartNavigator	25	75
	3mensio	16	84

Table 4.19: Comparison of Deep Learning, HeartNavigator and 3mensio measurements after Laplacian smoothing.

After smoothing the contours, the deep learning model's measurements are in between the two software tools' measurements in 13 cases for the annulus area and in 21 cases for the annulus perimeter.

The resulting distribution of the annulus area and perimeter is plotted in Figure 4.7, compared to the two software tools.

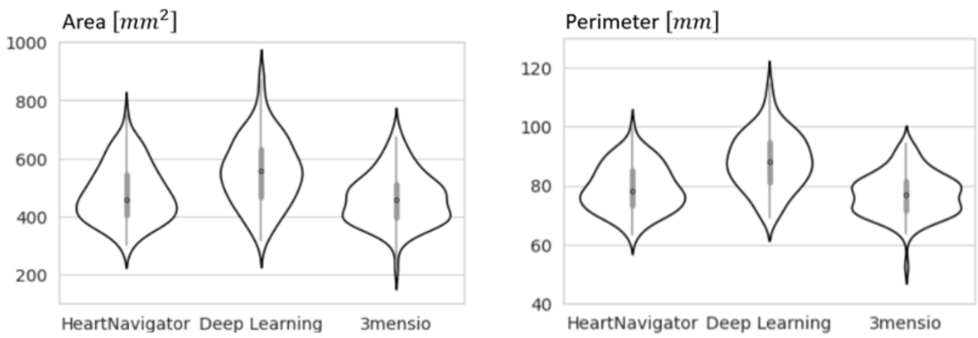


Figure 4.7: Distribution of measurements: The distribution of the deduced area and perimeter is compared for the two software solutions, HeartNavigator and 3mensio, and the deep learning model.

The deep learning model shows the broadest range of values. The minimum is close to the minimal value measured by HeartNavigator, both for area and perimeter, while 3mensio measured a lower minimum. Mean and maximum are highest for the deep learning model. It can be concluded that the deep learning measurements do not deviate by a fixed value.

In fact, over- but also underestimation occurs to different degrees, which is even more evident from a plot of agreement in Figure 4.8. The agreement between all three tools is plotted for a randomly selected subset of 30 patients after application of Laplacian smoothing.

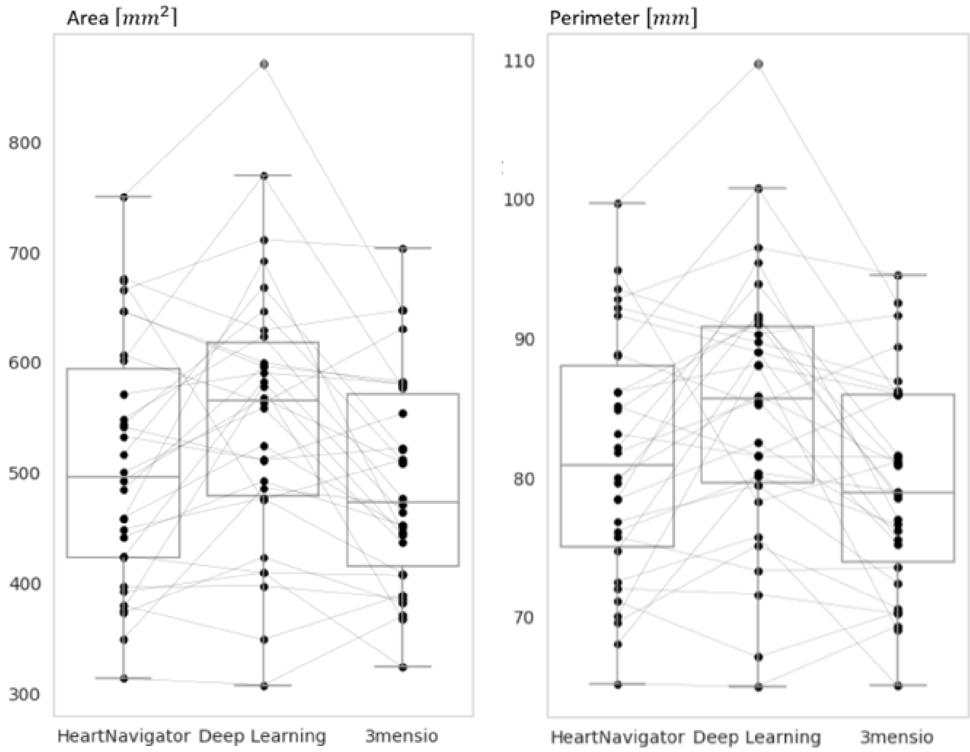
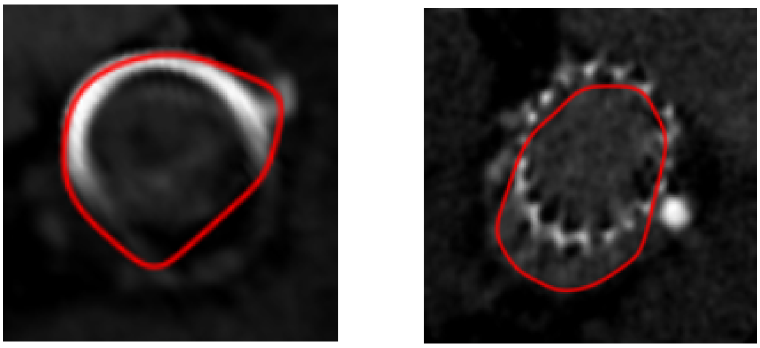


Figure 4.8: Agreement of measurements between the three tools: A boxplot with one dot for each patient and tool is plotted. Each line connects one individual patient, measured by the three tools. In several cases HeartNavigator and 3mensio diverge; on average the deep learning model reports higher values than the two software solutions.

Cases of agreement, under- and overestimation of the deep learning model compared to the software tools can be seen. In several instances, Heartnavigator and 3mensio diverge; in most cases, HeartNavigator reports the higher values. In some of these cases, the values obtained from the deep learning model are in between the two software tools' values. The deviations cannot be ascribed to a fixed bias.

Two cases of pre-implanted prosthetic valves are identified which compound the annulus segmentation. They are depicted in Figure 4.9 with the measurements obtained from the deep learning model after smoothing and the two software tools shown in Table 4.20. In retrospect of Figure 3.5, a pre-implanted valve as in Figure 4.9b had not been part of the training data and is thus not known to the model.

For the two cases of pre-implanted valves, the measurements of HeartNavigator and 3mensio diverge from each other, with the deep learning model's predictions leaning towards HeartNavigator.



(a) Pre-implanted valve as previously seen in the training data

(b) Pre-implanted valve not previously seen in the training data

Figure 4.9: Deduced annulus segmentations for the two cases with pre-implanted valves.

Tool	Case (a) Area $[mm^2]$	Case (a) Perimeter $[mm]$	Case (b) Area $[mm^2]$	Case (b) Perimeter $[mm]$
Deep Learning	386.7	71.3	755.8	100.9
HeartNavigator	385.8	71.5	639.0	91.1
3mensio	219.9	52.7	507.1	80.1

Table 4.20: Measurements in case of pre-implanted valves.

As a matter of fact, the main difficulty for annulus measurement still lies in the plane detection. Even after correction with the three-dimensional aorta segmentation, the detected plane often still lies within the valve, shown in Figure 4.10, and needs to be further shifted downwards to the annulus plane between the aortic valve and the LVOT. As no sufficiently annotated data set is readily available for training, this cannot be easily achieved. Possible approaches are proposed in Section 5.3 - Future Work.

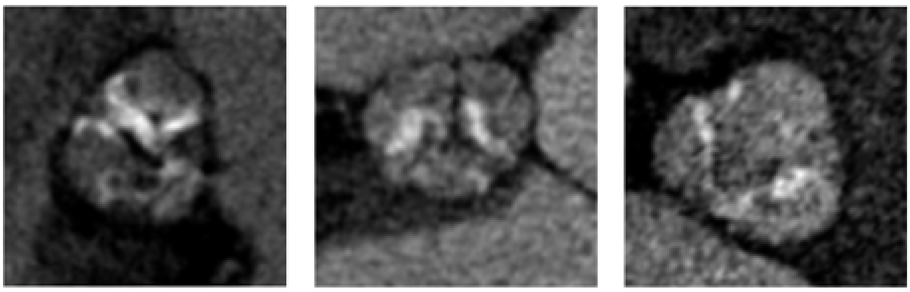


Figure 4.10: Detected planes within the valve.

5 CONCLUSION

In this concluding chapter, the positive and negative aspects of a neural network-based approach to annulus measurement are discussed. Impediments are highlighted, and the suitability for medical application is assessed. Finally, the conducted work is summarized, and possible future research tasks are proposed.

5.1 DISCUSSION

A cascaded approach of four sequentially applied CNNs, following the U-Net architecture, was pursued to achieve automatic measurement of aortic annulus area and perimeter. For each of the four substeps, the CNNs delivered satisfactory results. U-Nets are, thus, deemed an appropriate model for segmentation of CT scans.

The cascaded approach yields the advantage that previous results can be used in the following steps. In this way, the CT images could be reduced to the DLZ for the following steps, allowing for a higher image resolution without further memory requirement.

The size of the DLZ bounding box was defined manually by inspection of the CT images in the training data set. Two factors influenced the decision: on the one hand, the resulting sub-images need to be big enough to contain the relevant information for the following steps; on the other hand, the sub-images need to be small enough to fit into the available working memory while having the highest possible resolution (for hardware specifications see Section B.1). The bounding box size was initially chosen to exactly contain the aortic valve based on a subset of the training set. To ensure all relevant information is actually contained in the predicted bounding boxes, a padding was later added to the predictions. Optimization of this step was not considered of high importance for improvement of the overall deep learning approach, as the current predictions suffice for the following steps. Thus, this proceeding was accepted for the present work. Alternatively, the U-Nets could directly be trained on the bounding boxes of bigger size, or they could be trained to predict the bounding box midpoints, and the resulting sub-image could be chosen accordingly. The latter approach is not recommended, as step 3 showed that more expansive segmentations are easier to learn than individual points. Future work could evaluate further bounding box sizes and resolutions of the resulting sub-images.

Apart from the reduction to the DLZ, the cascaded approach produced the three-dimensional aorta segmentation, which could be used to improve the annulus plane detection in step 3. On the contrary, errors of previous steps can negatively impact the following steps. Such errors might be reducible with an end-to-end trained pipeline.

The derived measurements are in a reasonable range compared to the measurements obtained from the two software tools. Nevertheless, the mean of each measurement is higher. The difference of the means between the deep learning model and HeartNavigator is 61.7 mm^2 for area and 4.6 mm for perimeter. The difference between the deep learning model and 3mensio is 79.7 mm^2 for area and 6.7 mm for perimeter. The difference between HeartNavigator and 3mensio is 18.0 mm^2 for area and 2.1 mm for perimeter. While the discrepancy between the two software solutions is consistent with the inter-observer differences reported by Knobloch et al. [45], the deep learning results deviate more than twice as much from the software solutions' measurements.

The measurements are highly dependent on several factors, especially the position of the detected annulus plane. The accuracy of the model's predictions, in turn, depends on the annotations of the training data set. For this thesis, the training set comprised a relatively small number of 90 cases and was merely evaluated on seven cases. Further model improvement and evaluation with more comprehensive data sets is advisable. Further, the annotations exhibit several limitations. The valve plane markers are manually set on the tips of the aortic valve cusps. Perfect manual placement is challenging to achieve, resulting in less-than-ideal marker positions. Similarly, the segmentation of the aorta, especially the aortic valve, can be deficient. One limiting factor is that the aorta segmentation masks only comprise the aorta's lumen, not the tissue. The valve itself is, thus, underestimated. The present results could be further improved by either using a segmentation mask containing the aortic tissue, or the tissue thickness needs to be considered for deduction of the annulus position.

Apart from that, calcifications have a significant effect on TAVI device sizing. Knobloch et al. [45] exclude any calcifications from the annulus area. Calcifications were not explicitly treated in this thesis, which might explain the overestimation of the annulus dimensions in cases with calcification. No information could be obtained on how HeartNavigator and 3mensio treat calcifications.

The neural network-based approach, in theory, has no restrictions regarding imaging protocols or vendors. Provided the input CTs follow DICOM standards and contain the DLZ, automatic measurement of the annulus should be possible. Nevertheless, this could not be tested in detail with the present data. Data sets for training, evaluation and benchmarking were obtained from the German Heart Center Berlin, which uses a standardized imaging protocol and only two manufacturers for CT scanners. Before application to CT scans obtained from different manufacturers and imaging protocols, further evaluation on such is advisable. A re-training on the respective CTs might be required for optimal results.

In the present study, exceptional cases, such as pre-implanted artificial valves or bicuspid valves, were not explicitly considered. If several such cases are incorporated into the training data, likely the neural network-based model could also be able to treat these cases automatically.

The presently reported measurements shall give a first prospect of the neural network's potentialities. Delicate adjustment is required before application in the medical field.

5.2 summary

The purpose of this study was to prove the feasibility of a neural network-based approach to support aortic root analysis. The approach was benchmarked against two software solutions on the measurement of aortic annulus area and perimeter. A cascade of neural networks, each following the U-Net architecture, was employed to infer segmentations in four steps: detection of the region of interest around the aortic valve, segmentation of the aorta, including the aortic valve, detection of the aortic root area, from which the aortic annulus plane was approximated and segmentation of the aortic annulus in the two-dimensional annular plane. From this aortic annulus segmentation, area and perimeter were inferred and compared to the measurements obtained from the two software solutions.

The annular plane was approximated by applying a PCA to the predicted segmentation of the annulus area, masked with the outer edge of the aortic valve segmentation.

Evaluation of the neural networks showed promising results for each substep. For all 100 patients considered in the benchmark, annulus measurements could be obtained. The measurements are comparable to the ones obtained from the two software solutions. However, the area and perimeter are rather overestimated. The overestimation results from a deficient annulus plane detection, which often yields planes that are still inside the valve and not right on the aortic annulus.

To conclude, the cascaded neural network-based approach enabled the reliable detection and assessment of the aortic root and valve apparatus even with a relatively small training set. Further improvement of the annulus plane detection is required to yield optimal annulus segmentations for correct measurements. Still, the current results suggest neural network-based aortic root analysis as a promising approach to support fully automatic TAVI device sizing and selection, independent of a specific patient group or CT imaging protocol. An extended data set, including uncommon cases, such as pre-implanted artificial valves or bicuspid valves, could further improve the applicability and robustness of this method. Further measurements could be obtained in a similar fashion, such as coronary height, given appropriate annotations.

5.3 FUTURE WORK

As already mentioned, the current approach still needs to be improved before application in the medical field. Besides re-training the model with a more extensive data set, annotated by medical experts, several other possibilities for model improvement can be pursued.

Annotations: Apart from increasing the number of training samples and optimizing the annotations, several modifications of the annotations could facilitate the correct segmentation of the aortic annulus. Inclusion of the aortic tissue in the aorta segmentation would allow for the deduction of the annulus plane's position from the surface of the aortic valve segmentation, as the annulus lies just below the deepest point of the aortic valve. Examples of correct annulus planes could help decide whether a detected plane actually portrays the annulus. In this regard, the plane's gradient profile might be worth exploring, compared to typical annulus planes' profiles. If the detected plane does not resemble the annulus plane examples, the detected plane could be shifted or even slightly rotated until it resembles the annulus plane examples.

Moreover, a comprehensive data set of perfectly segmented annuli in the three-dimensional CT images might well be learnable by a CNN.

Neural Network training: The U-Net architecture proved to be well suited for segmenting different image regions from CT scans. With a more extensive training data set, it might be beneficial to explore different options for annulus plane identification anew, such as different extents for the segmentation masks, directly detecting the valve plane markers via segmentation, or the regression approaches in Appendix A. For regression of the annulus plane's normal vector, it might be beneficial to adjust the considered loss function. The MAE in the angle resulting from the normal vector, instead of the normal vector itself, is likely a better measure for the error. Another option would be to attempt the valve plane detection on the segmented aorta from step 2 instead of the original CT image to exploit the additional information. To further utilize ensemble methods, several approaches could be averaged into a final prediction. The segmentation accuracy might be further improved by exploring different loss functions or weights for the loss, as well as optimizers. A more expansive hyperparameter optimization is advisable.

Although a small batch size showed to help in the generalizability of a neural network [42], a higher batch size generally speeds up convergence, thus reducing training time and avoiding oscillation around local minima [70]. Consequently, fine-tuning the batch size can prove useful. Moreover, with an increase in the number of convolutional filters, it might be possible to further increase the segmentation accuracy. Batch size and number of filters both have an influence on the size of the resulting network, which is limited by the available working memory.

Memory optimization: With the current combination of batch size, number of filters and input image size, the available working memory was fully exploited. For details on the hardware, see Section B.1. Memory optimization is required to further increase the batch size or number of filters, hence the neural network size. Different options are available, such as mixed-precision computing [81] or gradient checkpointing [11]. Due to the satisfactory results and required time investment, these options were not further pursued in the present study.

Image preprocessing: Additional potential lies in preprocessing of the network's input images. Currently, each input image is scaled to the range [0, 1] to ensure gradients of similar magnitude over different batches. However, information from different light intensities between images is lost or distorted by calcifications or light artifacts. A CT-wise scaling might be advantageous. Instead of scaling to a fixed range, Lecun et al. [48] suggest normalization of the input to achieve a mean close to zero and a uniform variance. The mean is here typically taken with respect to the whole data set.

Apart from that, the current choice of the voxel sizes for steps 2 to 4 leads to resampling of the images to resolutions higher than the original images' resolution. This is achieved by interpolation, which might introduce inaccuracies. In future work, it might be beneficial to further analyze this aspect and potentially restrict the minimal resolution to the original input images' resolution.

Further information could be obtained by preprocessing steps, such as denoising, contrast enhancement or edge detection [36, 40].

Moreover, calcification could be considered in image preprocessing. According to Knobloch et al. [45], calcifications are not part of the annulus area. Thus, one option might be to remove calcifications from the image. In a simple manner, this could be achieved by harmonizing bright image regions with neighboring voxels. Nevertheless, Meyer et al. [56] stated that calcification patterns play a significant role in prosthesis selection, hence should not be disregarded. Optimally, calcifications could be individually segmented to assess severity and allocation.

Further TAVI support

In an ideal setting, machine learning models could support medical practitioners with automatically derived suggestions for TAVI device model and size. Research in this field is of high interest, which is also shown by the fact that part of the work in this thesis is accepted as a lecture presentation for the CARS 2021 congress [1].

Besides annulus measurements, additional parameters, such as coronary distances, affect TAVI device sizing [56]. Such parameters could be obtained in a similar fashion, given appropriate annotations.

1 https://www.cars-int.org/

Ultimately, a machine learning model could be trained on optimal TAVI procedures, targeted on prosthesis size and model. Different input parameters, such as the previously obtained measurements, further patient data or CT image data could be combined with different models for optimal predictive power. A mixed data neural network [50] using MedicalNet (see Section A.1), a ResNet pre-trained on medical images, for the CT image input could be a promising starting point. Prediction of likely complications after TAVI could be achieved in a similar manner.

Part II

APPENDIX

A

FURTHER ATTEMPTS AT VALVE PLANE IDENTIFICATION

Before arriving at the method for valve plane identification described in Section 3.6, several other approaches were assayed but dismissed due to insufficient results. Still, they pose an interesting opportunity for future work and will hence be elucidated in the following sections. For each approach, a single network was trained on 75 patients, with 15 patients used in validation to allow for early stopping and evaluated on the seven test patients. The same methods as in Section 3.3 are used for data augmentation.

A.1 MEDICALNET

MedicalNet [80] is a Pytorch implementation of "Med3D: Transfer Learning for 3D Medical Image Analysis." In Med3D Chen, Ma, and Zheng [13] attend to the issue that deep learning performance highly depends on the number of training samples and transfer learning can drastically improve the results by pre-training models on huge data sets, such as *ImageNet* [64]. However, there are fundamental differences between natural and medical images - two versus often three dimensions, colored versus grayscale - and also relevant features and task specifications differ [63]. Thus, pre-trained models on ImageNet cannot be directly transferred to medical applications. Consequently, Chen, Ma, and Zheng [13] aggregated several data sets from medical imaging competitions and designed the three-dimensional Med3D network trained on this aggregated data set, facilitating transfer learning specialized on medical images. The "Deep Residual Learning for Image Recognition" [31] architecture, ResNet for short, is used for Med3D, which will be explained in the following section.

A.2 THE RESNET ARCHITECTURE

Concepts and explanations in this section are primarily based on [15, 31]. Simonyan and Zisserman [73] have shown that the network depth is crucial for network performance, although simply stacking network layers does not necessarily lead to improvement. This can be understood by considering F, the class of functions a specific network architecture can reproduce. By adding another layer to the network, it will be able to reproduce a class of functions F'. However, if $F \notin F'$ it cannot be guaranteed that the new network with an additional layer is as good as the previous one. Thus, for the new model to be at least as effective as the previous model, it needs to be ensured that the newly added layer can map to an identity function $f(x) = x$. With the additional layer, the new model will be able to cover a broader space of functions and might, hence, perform even better at approximating the desired function. This was addressed by He et al. [31], who designed a so-called residual block to facilitate learning of the identity function.

Figure A.1 shows a regular convolutional block in comparison to a residual block.

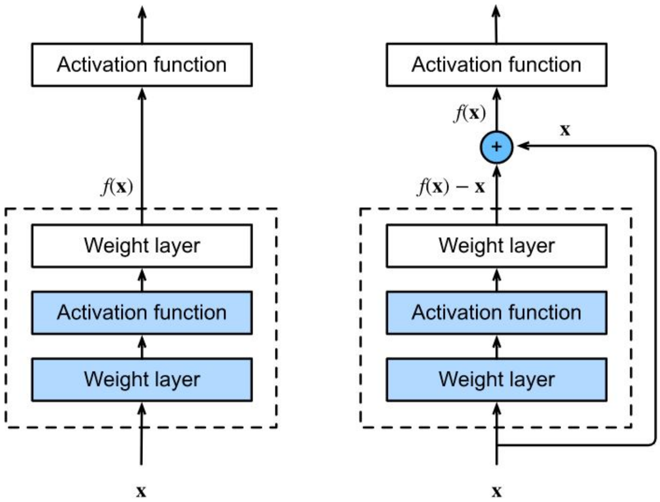


Figure A.1: A convolutional block (left) and a residual block (right) [15].

A regular block, within the dotted-line box, must learn $f(x)$. A residual block, on the other hand, only needs to learn the residual $f(x) - x$. If the desired mapping is the identity function, this can easily be achieved by the second weight layer approaching zero. The residual connection or shortcut connection will then propagate the identity further through the network.

To further increase network depth while balancing the required training time, Seif [69] designed a modification of the residual block, the so-called bottleneck, depicted in Figure A.2. Instead of skipping two convolutional layers with an identity shortcut, three layers are skipped in the bottleneck.

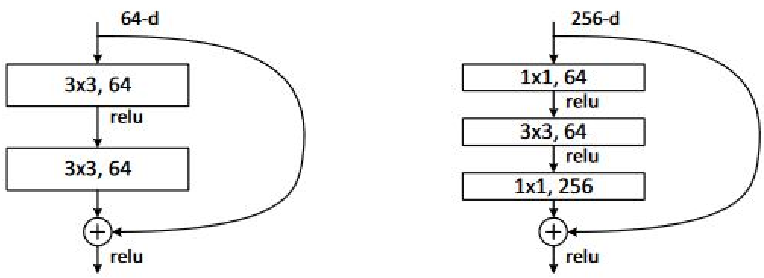


Figure A.2: A regular residual block (left) and a "bottleneck" block (right) [15].

Depending on the ResNet's depth, regular residual blocks are replaced by such bottlenecks to reduce required training time while still allowing for better performance, as shown by He et al. [31]. Typical ResNet depths, also implemented by Med3D, are 10, 18, 34, 50, 101, 152, 200, two of which shall be utilized in the following sections to identify the valve plane.

A.3 REGRESSION - VALVE PLANE MARKERS

In a first attempt, the valve plane shall be found by directly learning the valve plane marker positions. This was modeled as a regression task.

Methods

The valve plane marker positions were obtained by retrieving the indices of the valve plane marker masks within the image and dividing them by the image dimensions. Thus, nine values need to be estimated for each patient, the x-, y- and z-coordinate of each of the three valve plane markers.

Since the available training data set is limited, transfer learning was utilized with the Med3D pre-trained models. ResNets of depth 10 and 50 were compared. An even deeper ResNet could not be trained due to insufficient memory. The available hardware is listed in Section B.1. For reference, a simple CNN was trained with two convolutional layers, which was not pre-trained on medical data. The first layer applied 16 filters, the second layer four filters, each with a kernel size of three, a stride of one and zero padding. Batch normalization and max pooling with a kernel size of two were applied after each convolutional layer. Stochastic gradient descent was used to optimize both the simple CNN and ResNet. Early stopping was used, similar to the U-Net training. For all regression approaches the neural networks are implemented in PyTorch[1]; for package specifications see Section B.2. The MAE is used as a loss function and for evaluation.

Results

The midpoint and normal vector are calculated from the three predicted valve plane markers and compared to the ground truth with the MAE.

Table A.1 shows the results on the train, validation and test set for each model. It is again contrasted against the error resulting from the training mean.

Model	Training (75 Patients)		Validation (15 Patients)		Test (7 Patients)	
	Midpoint	Normal	Midpoint	Normal	Midpoint	Normal
ResNet-10	0.102	0.887	0.106	0.913	0.087	0.933
ResNet-50	0.122	0.996	0.154	0.667	0.117	1.025
CNN	0.113	0.453	0.107	1.110	0.109	1.195
Training mean	0.039	0.390	0.043	0.309	0.033	0.297

Table A.1: MAE for each model on the valve plane marker regression.

1 https://pytorch.org/

Each model's predictions are worse than the usage of the training mean. Consequently, all models were rejected.

A.4 REGRESSION - MIDPOINT AND NORMAL VECTOR

After the previous approach did not yield satisfactory results, it was attempted to identify the valve plane by learning its midpoint and normal vector.

Methods

The midpoint and the normal vector are calculated using Listing 4.1. Now six values per patient need to be learned by regression, the x-, y- and z-component of the midpoint and the normal vector. Again three ResNets and a simple CNN are compared using the MAE.

Results

Table A.1 shows the results on the train, validation and test set for each model.

Model	Training (75 Patients)		Validation (15 Patients)		Test (7 Patients)	
	Midpoint	Normal	Midpoint	Normal	Midpoint	Normal
ResNet-10	0.112	0.392	0.154	0.486	0.106	0.331
ResNet-50	0.188	0.478	0.139	0.405	0.115	0.281
CNN	0.141	0.342	0.151	0.317	0.141	0.371
Training mean	0.039	0.390	0.043	0.309	0.033	0.297

Table A.2: MAE for each model on the midpoint and normal vector regression.

Compared to the previous attempt to predict the three valve plane marker positions, the resulting midpoint prediction has approximately the same quality according to the MAE, while the normal vector prediction improves. Still, usage of the training mean prevails, and also this approach is not pursued any further.

A.5 SEGMENTATION - VALVE PLANE MARKERS

With the previous unsuccessful attempts, regression was dismissed, and it was attempted to detect the valve plane markers by segmenting the respective image regions.

Methods

A U-Net was employed to segment the valve plane markers. To simplify the segmentation, the region around the valve plane markers was enlarged with paddings p of different sizes: no padding, a padding of two and a padding of five in each direction. Again the loss - binary cross-entropy - was weighted with the inverse class frequency.

For evaluation, the predicted segmentation is binarized with a threshold of 0.5. The binarization will likely yield more than three possible points for the valve plane markers. Hence, a k-means algorithm was applied to the proposed valve plane marker positions to achieve three clusters, of which the midpoints are assumed to be the valve plane marker positions. Further, the minimal distance between the valve plane markers was calculated, and it was assessed whether the cluster midpoints satisfy that minimal distance. If not, the points that are too close together are averaged, and the next point with the highest predicted probability is considered. Again the distance between the points is evaluated. The new point is either taken as a valve plane marker or averaged with the other point if the minimal distance is not exceeded. This process is iterated until three points are found that all have at least the minimal distance to each other. These are assumed to be the valve plane markers.

Results

From these valve plane marker positions, the midpoint and the normal vector of the resulting plane are calculated and compared to the ground truth.

Table A.1 shows the results on the train, validation and test set for each model.

Padding	Training (75 Patients)		Validation (15 Patients)		Test (7 Patients)	
	Midpoint	Normal	Midpoint	Normal	Midpoint	Normal
$p = 0$	0.082	0.602	0.063	0.567	0.076	0.462
$p = 2$	0.123	0.831	0.132	0.653	0.165	0.894
$p = 5$	0.106	1.537	0.087	1.664	0.120	1.719
Training mean	0.039	0.390	0.043	0.309	0.033	0.297

Table A.3: MAE on the valve plane marker segmentation for different paddings of the markers.

It can be seen that with increasing padding, the prediction of the normal vector worsens, while the midpoint prediction is less affected by the padding. Nevertheless, the results cannot surpass the usage of the training mean, and the models are hence rejected, leading to the final approach, described in Section 3.6 - Step 3 - Annulus Plane Identification.

B TECHNICAL INFRASTRUCTURE

B.1 HARDWARE

A Linux server with the following hardware specifications was used for model development:

- 96 GB RAM,
- 2 intel Xeon Gold 5215 @ 1.7 GHz,
- 4 nVidia Titan V 12 GB HBM2 (GV100, SM7.0).

For training and application of the neural networks, the four nVidia GPUs were employed. Whenever possible, calculations were distributed equally across all four GPUs.

Depending on the regarded step and the weight initialization, average training time of one U-Net ranged between 20 minutes and three hours.

B.2 USED PYTHON PACKAGES

The following python packages were used in the context of this thesis:

Package	Version
h5py	2.10.0
ipykernel	5.1.4
ipython	7.13.0
ipython_genutils	0.2.0
ipywidgets	7.5.1
jupyter-core	4.6.3
jupyter-tensorboard	0.2.0
jupyter_client	6.1.0
keras	2.3.1
keras-applications	1.0.8
keras-preprocessing	1.1.0
matplotlib	3.2.1
nibabel	3.0.2
nilearn	0.6.2
numpy	1.18.2
opencv-python	4.4.0.44
pandas	1.0.3

pandas-profiling	2.6.0
pillow	7.1.2
pip	20.0.2
plotly	4.8.1
pydicom	2.0.0
python	3.7.7
python-dateutil	2.8.1
python-editor	1.0.4
scikit-image	0.17.2
scikit-learn	0.22.2.post1
scipy	1.4.1
seaborn	0.11.1
simpleitk	1.2.4
tensorboard	2.1.1
tensorflow-addons	0.9.1
tensorflow-estimator	2.1.0
tensorflow-gpu	2.1.0
torch	1.6.0
torchvision	0.6.0+cu101

Table B.1: Used python packages.

BIBLIOGRAPHY

[1] 3DSlicer. *Coordinate systems*. https://www.slicer.org/wiki/Coordinate_systems. Last visited: 24.11.2020.

[2] Walid Abdullah Al, Ho Yub Jung, Il Dong Yun, Yeonggul Jang, Hyung-Bok Park, and Hyuk-Jae Chang. "Automatic aortic valve landmark localization in coronary CT angiography using colonial walk." In: *PLOS ONE* 13.7 (July 2018), pp. 1–23. DOI: 10.1371/journal.pone.0200317. URL: https://doi.org/10.1371/journal.pone.0200317.

[3] Patricio Astudillo, Peter Mortier, Johan Bosmans, Ole De Backer, Peter de Jaegere, Matthieu De Beule, and Joni Dambre. "Enabling Automated Device Size Selection for Transcatheter Aortic Valve Implantation." In: *Journal of Interventional Cardiology* 2019 (2019). DOI: 10.1155/2019/3591314.

[4] Regina Bailey. *Anatomy of the Heart: Valves*. https://www.thoughtco.com/anatomy-of-the-heart-valves-373203. Last visited: 23.11.2020. 2019.

[5] Alex Bäuerle and Timo Ropinski. "Net2Vis: Transforming Deep Convolutional Networks into Publication-Ready Visualizations." In: *CoRR* abs/1902.04394 (2019). arXiv: 1902.04394. URL: http://arxiv.org/abs/1902.04394.

[6] Helmut Baumgartner et al. "2017 ESC/EACTS Guidelines for the management of valvular heart disease." In: *European Heart Journal* 38.36 (Aug. 2017), pp. 2739–2791. ISSN: 0195-668X. DOI: 10.1093/eurheartj/ehx391. eprint: https://academic.oup.com/eurheartj/article-pdf/38/36/2739/24596044/ehx391.pdf. URL: https://doi.org/10.1093/eurheartj/ehx391.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[8] Philipp Blanke et al. "Computed Tomography Imaging in the Context of Transcatheter Aortic Valve Implantation (TAVI)/Transcatheter Aortic Valve Replacement (TAVR)." In: *JACC: Cardiovascular Imaging* 12.1 (2019), pp. 1–24. DOI: 10.1016/j.jcmg.2018.12.003.

[9] Tom Breloff. *Learning without Backpropagation: Intuition and Ideas (Part 1)*. http://www.breloff.com/no-backprop/. Last visited: 09.12.2020. 2016.

[10] Jason Brownlee. *How Much Training Data is Required for Machine Learning?* https://machinelearningmastery.com/much-training-data-required-machine-learning/. Last visited: 27.01.2021. 2019.

[11] Yaroslav Bulatov. *Fitting larger networks into memory.* https://medium.com/tensorflow/fitting-larger-networks-into-memory-583e3c758ff9. Last visited: 21.01.2021. 2018.

[12] Efstratios I. Charitos and Hans-Hinrich Sievers. "Anatomy of the aortic root: implications for valve-sparing surgery." In: *Annals of Cardiothoracic Surgery* 2.1 (2013). ISSN: 2225-319X. URL: http://www.annalscts.com/article/view/1400.

[13] Sihong Chen, Kai Ma, and Yefeng Zheng. "Med3D: Transfer Learning for 3D Medical Image Analysis." In: *CoRR* abs/1904.00625 (2019). arXiv: 1904.00625. URL: http://arxiv.org/abs/1904.00625.

[14] Zwinderman-Aeilko H. Cleophas Ton J. *Machine Learning in Medicine - a Complete Overview*. Springer, 2015.

[15] Dive into Deep Learning. *Residual Networks (ResNet)*. https://d2l.ai/chapter_convolutional-modern/resnet.html. Last visited: 27.11.2020. 2020.

[16] Necati Demir. *Ensemble Methods: Elegant Techniques to Produce Improved Machine Learning Results*. https://www.toptal.com/machine-learning/ensemble-methods-machine-learning#:~:text=Ensemble%20methods%20are%20techniques%20that,winning%20solutions%20used%20ensemble%20methods.. Last visited: 24.11.2020. 2015.

[17] Adit Deshpande. *A Beginner's Guide To Understanding Convolutional Neural Networks*. https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/. Last visited: 24.11.2020. 2016.

[18] Mohit Deshpande. *Perceptrons: The First Neural Networks*. https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks/. Last visited: 24.11.2020. 2020.

[19] Herz und Diabeteszentrum NRW. *TAVI: Chirurgische Transkatheter-Aortenklappen-Implantation*. https://www.hdz-nrw.de/kliniken-institute/kliniken/klinik-fuer-thorax-und-kardiovaskularchirurgie/behandlung/herz-und-gefaesschirurgie/chirurgisches-tavi-verfahren.html. Last visited: 24.11.2020.

[20] Thomas G. Dietterich. "Ensemble Methods in Machine Learning." In: *Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg* 12 (2000), pp. 2825–2830. DOI: 10.1007/3-540-45014-9_1.

[21] Elsevier Science Direct. *Artificial Intelligence in Medicine*. https://www.sciencedirect.com/journal/artificial-intelligence-in-medicine. Last visited: 24.11.2020. 2020.

[22] Thomas Epelbaum. "Deep learning: Technical introduction." In: *ArXiv* abs/1709.01412 (2017).

[23] Arun Gandhi. *Data Augmentation | How to use Deep Learning when you have Limited Data - Part 2*. https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/. Last visited: 18.01.2021. 2020.

[24] Rohith Gandhi. *Build Your Own Convolution Neural Network in 5 mins*. https://towardsdatascience.com/build-your-own-convolution-neural-network-in-5-mins-4217c2cf964f. Last visited: 25.11.2020. 2018.

[25] Jane Racey Gleeson. *Weighing TAVR vs. SAVR for Aortic Stenosis Treatment*. https://healthblog.uofmhealth.org/heart-health/tavr-vs-savr. Last visited: 23.11.2020. 2018.

[26] Daniel Godoy. *Understanding binary cross-entropy / log loss: a visual explanation*. https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a. Last visited: 24.11.2020. 2018.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[28] Sasa Grbic, Tommaso Mansi, Razvan Ionasec, Ingmar Voigt, Helene Houle, Matthias John, Max Schoebinger, Nassir Navab, and Dorin Comaniciu. "Image-Based Computational Models for TAVI Planning: From CT Images to Implant Deployment." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Ed. by Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 395–402. ISBN: 978-3-642-40763-5.

[29] Prince Grover. *5 Regression Loss Functions All Machine Learners Should Know*. https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0#:~:text=2.,predictions%2C%20without%20considering%20their%20directions.. Last visited: 24.11.2020. 2018.

[30] Philippe Généreux et al. "Paravalvular Leak After Transcatheter Aortic Valve Replacement: The New Achilles' Heel? A Comprehensive Review of the Literature." In: *Journal of the American College of Cardiology* Volume 61, Issue 11 (2013), pp. 1125–1136.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[32] National Heart, Lung, and Blood Institute. *The Human Heart: A Living Pump*. DHEW publication ; no. (NIH) 76-1058. U.S. Department of Health, Education, Welfare, Public Heath Service, National Institutes of Health, National Heart, Lung, and Blood Institute, Office of Information, 1976. URL: https://books.google.de/books?id=z1axKjqTwCcC.

[33] Barbora Horehledova, Casper Mihl, Ellen Boswijk, Genevieve A. J. C. Crombag, Estelle C. Nijssen, Patty J. Nelemans, Leo F. Veenstra, Joachim E. Wildberger, and Marco Das. "Retrospectively ECG-gated helical vs. non-ECG-synchronized high-pitch CTA of the aortic root for TAVI planning." In: *PLOS ONE* 15.5 (May 2020), pp. 1–14. DOI: 10.1371/journal.pone.0232673. URL: https://doi.org/10.1371/journal.pone.0232673.

[34] Chen Huang, Shuangfei Zhai, Walter Talbott, Miguel Ángel Bautista, Shih-Yu Sun, Carlos Guestrin, and Joshua M. Susskind. "Addressing the Loss-Metric Mismatch with Adaptive Loss Alignment." In: *CoRR* abs/1905.05895 (2019). arXiv: 1905.05895. URL: http://arxiv.org/abs/1905.05895.

[35] Pie Medical Imaging. *3mensio*. https://www.3mensio.com/. Last visited: 12.01.2021.

[36] Anthony J.Bell and Terrence J.Sejnowski. "The "independent components" of natural scenes are edge filters." In: *Vision Research* 37.23 (1997), pp. 3327 –3338. ISSN: 0042-6989. DOI: https://doi.org/10.1016/S0042-6989(97)00121-1. URL: http://www.sciencedirect.com/science/article/pii/S0042698997001211.

[37] Zakaria Jaadi. *A STEP BY STEP EXPLANATION OF PRINCIPAL COMPONENT ANALYSIS*. https://builtin.com/data-science/step-step-explanation-principal-component-analysis. Last visited: 26.11.2020. 2019.

[38] Shruti Jadon. *A survey of loss functions for semantic segmentation*. 2020. arXiv: 2006.14822 [eess.IV].

[39] Peter PT de Jaegere and Joana Maria Ribeiro. "Original Research Paper: Artificial Intelligence and Advanced Computer Modelling in Transcatheter Interventions for Structural Heart Disease - Implications for Clinical Practice." In: *ESC Digital Health* (2020). https://www.escardio.org/Education/Digital-Health-and-Cardiology/Virtual-Journal/artificial-intelligence-and-advanced-computer-modelling-in-transcatheter-interve.

[40] Beaulah Jeyavathana, Balasubramanian Ramasamy, and Anbarasa Pandian. "A Survey: Analysis on Pre-processing and Segmentation Techniques for Medical Images." In: *International Journal of Research and Scientific Innovation (IJRSI)* (June 2016).

[41] Jeremy Jordan. *An overview of semantic image segmentation*. https://www.jeremyjordan.me/semantic-segmentation/. Last visited: 24.11.2020. 2018.

[42] Ibrahem Kandel and Mauro Castelli. "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset." In: *ICT Express* 6.4 (2020), pp. 312 –315. ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2020.04.010. URL: http://www.sciencedirect.com/science/article/pii/S2405959519303455.

[43] Andrej Karpathy. *What I learned from competing against a ConvNet on ImageNet*. `http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/`. Last visited: 25.11.2020. 2014.

[44] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: `1412.6980 [cs.LG]`.

[45] Gesine Knobloch et al. "Inter- and intra-observer repeatability of aortic annulus measurements on screening CT for transcatheter aortic valve replacement (TAVR): Implications for appropriate device sizing." In: *European journal of radiology* 105 (June 2018), pp. 209–215. DOI: `10.1016/j.ejrad.2018.06.003`.

[46] Tom Kwok. *IoU vs F1*. `https://tomkwok.com/posts/iou-vs-f1/'`. Last visited: 24.11.2020. 2020.

[47] Florent Lalys, Simon Esneault, Miguel Castro, Lucas Royer, Pascal Haigron, Vincent Auffret, and Jacques Tomasi. "Automatic aortic root segmentation and anatomical landmarks detection for TAVI procedure planning." en. In: *Minimally Invasive Therapy & Allied Technologies* 28.3 (May 2019), pp. 157–164. ISSN: 1364-5706, 1365-2931. DOI: `10.1080/13645706.2018.1488734`.

[48] Yann Lecun, Leon Bottou, Genevieve Orr, and Klaus-Robert Müller. "Efficient BackProp." In: (Aug. 2000).

[49] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." In: Jan. 2009, p. 77. DOI: `10.1145/1553374.1553453`.

[50] Laura Lewis. *Building a mixed-data neural network in Keras to predict accident locations*. `https://heartbeat.fritz.ai/building-a-mixed-data-neural-network-in-keras-to-predict-accident-locations-d51a63b738cf`. Last visited: 22.01.2021. 2019.

[51] Lars Linsen. *Lecture notes in Advanced Graphics - 1.7.1 Laplacian Smoothing*. 2015.

[52] Tanner Marshall. *Aortic valve disease*. `https://www.osmosis.org/learn/Aortic_valve_disease?from=/medicine-preclinical/cardiovascular/pathology/cardiac-diseases/heart-valve-disease`. Last visited: 25.01.2021. 2016.

[53] MeVisLab. *12.6. Coordinate Systems in the MeVisLab GUI*. `https://mevislabdownloads.mevis.de/docs/current/MeVisLab/Resources/Documentation/Publish/SDK/GettingStarted/ch12s06.html`. Last visited: 24.11.2020.

[54] MeVisLab. *About MeVisLab*. `https://www.mevislab.de/mevislab/`. Last visited: 24.11.2020.

[55] MeVisLab. *CSOIsoGenerator - Use Smooth*. https://mevislabdownloads.mevis.de/docs/current/MeVisLab/Standard/Documentation/Publish/ModuleReference/CSOIsoGenerator.html#mlabfield-usesmooth. Last visited: 19.12.2020.

[56] Alexander Meyer et al. "Reliability and Influence on Decision Making of fully-automated vs. semi-automated Software Packages for Procedural Planning in TAVI." In: *Scientific Reports* 10, 10746 (2020). DOI: 10.1038/s41598-020-67111-5.

[57] Catherine M. Otto et al. "2020 ACC/AHA Guideline for the Management of Patients With Valvular Heart Disease: Executive Summary: A Report of the American College of Cardiology/American Heart Association Joint Committee on Clinical Practice Guidelines." In: *Journal of the American College of Cardiology* 77.4 (2021), pp. 450 –500. ISSN: 0735-1097. DOI: https://doi.org/10.1016/j.jacc.2020.11.035. URL: http://www.sciencedirect.com/science/article/pii/S073510972037902X.

[58] PHILIPS. *HeartNavigator*. https://www.philips.de/healthcare/product/HCOPT20/heartnavigator-software-fr-behandlungsplanung-und-fhrung/uberblickh. Last visited: 12.01.2021.

[59] Miralem Pasic, Axel Unbehaun, Semih Buz, Thorsten Drews, and Roland Hetzer. "Annular Rupture During Transcatheter Aortic Valve Replacement: Classification, Pathophysiology, Diagnostics, Treatment Approaches, and Prevention." In: *Journal of the American College of Cardiology* Volume 8, Issue 1 (2015), pp. 1–9.

[60] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[61] Nicoló Piazza, Peter de Jaegere, Carl Schultz, Anton E. Becker, Patrick W. Serruys, and Robert H. Anderson. "Anatomy of the Aortic Valvar Complex and Its Implications for Transcatheter Implantation of the Aortic Valve." In: *Circulation: Cardiovascular Interventions* Vol. 1, No. 1 (2008), pp. 74–81.

[62] Abraham Pouliakis, Effrosyni Karakitsou, Niki Margari, Panagiotis Bountris, Maria Haritou, John Panayiotides, Dimitrios Koutsouris, and Petros Karakitsos. "Artificial Neural Networks as Decision Support Tools in Cytopathology: Past Present and Future." In: *Biomed Eng Comput Biol* 2016 (Feb. 2016), pp. 1–18. DOI: 10.4137/BECB.S31601.

[63] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. "Transfusion: Understanding Transfer Learning for Medical Imaging." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019, pp. 3347–3357. URL: https://proceedings.neurips.cc/paper/2019/file/eb1e78328c46506b46a4ac4a1e378b91-Paper.pdf.

[64] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. "CNN Features off-the-shelf: an Astounding Baseline for Recognition." In: *CoRR* abs/1403.6382 (2014). arXiv: 1403.6382. URL: http://arxiv.org/abs/1403.6382.

[65] Veronique Roger et al. "Heart Disease and Stroke Statistics-2011 Update A Report From the American Heart Association." In: *Circulation* 123 (Feb. 2011), e18–e209. DOI: 10.1161/CIR.0b013e3182009701.

[66] Olaf Ronneberger. *Our U-net wins two Challenges at ISBI 2015*. https://lmb.informatik.uni-freiburg.de/people/ronneber/isbi2015/. Last visited: 09.12.2020. 2015.

[67] 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.

[68] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Last visited: 24.11.2020. 2018.

[69] George Seif. *Understanding the 3 most common loss functions for Machine Learning Regression*. https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3. Last visited: 24.11.2020. 2019.

[70] Kevin Shen. *Effect of batch size on training dynamics*. https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e. Last visited: 21.01.2021. 2018.

[71] Naoki Shibuya. *Up-sampling with Transposed Convolution*. https://naokishibuya.medium.com/up-sampling-with-transposed-convolution-9ae4f2df52d0. Last visited: 25.11.2020. 2017.

[72] Jenni A. M. Sidey-Gibbons and Chris J. Sidey-Gibbons. "Machine learning in medicine: a practical introduction." In: *BMC Med Res Methodol* 19, 64 (2019). DOI: 10.1186/s12874-019-0681-4.

[73] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *ICLR* (2015). URL: https://arxiv.org/abs/1409.1556.

[74] Healthwise Staff. *Chambers of the heart*. https://www.healthlinkbc.ca/health-topics/tp10241. Last visited: 23.11.2020. 2019.

[75] Doug Steen. *A Gentle Introduction to Self-Training and Semi-Supervised Learning*. https://towardsdatascience.com/a-gentle-introduction-to-self-training-and-semi-supervised-learning-ceee73178b38. Last visited: 24.11.2020. 2020.

[76] Tobias Sterbak. *U-Net for segmenting seismic images with keras*. https://www.depends-on-the-definition.com/unet-keras-segmenting-images/. Last visited: 25.11.2020. 2018.

[77] Philipp Szalkiewicz et al. *The impact of device landing-zone morphology on the outcome after transcatheter aortic valve implantation*. https://www.researchgate.net/publication/338558935_The_impact_of_device_landing-zone_morphology_on_the_outcome_after_transcatheter_aortic_valve_implantation. Last visited: 24.11.2020. May 2019.

[78] Wataru Tatsuishi, Kiyoharu Nakano, Sayaka Kubota, Ryota Asano, and Go Kataoka. "Identification of Coronary Artery Orifice to Prevent Coronary Complications in Bioprosthetic and Transcatheter Aortic Valve Replacement." In: *Circulation journal : official journal of the Japanese Circulation Society* 79 (July 2015). DOI: 10.1253/circj.CJ-15-0415.

[79] Tim Taylor. *Cardiovascular System*. https://www.innerbody.com/image/cardov.html. Last visited: 23.11.2020. 2020.

[80] Tencent. *MedicalNet*. https://github.com/Tencent/MedicalNet. Last visited: 27.11.2020. 2020.

[81] TensorFlow. *Mixed Precision*. https://www.tensorflow.org/guide/mixed_precision?hl=en. Last visited: 21.01.2021. 2020.

[82] Dhruv Thakur. *ML Basics 4: Replace Negatives with Zeros!* https://dhruvs.space/posts/ml-basics-issue-4/, NOTE = Last visited: 03.02.2021. 2019.

[83] Ekin Tiu. *Metrics to Evaluate your Semantic Segmentation Model*. https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2. Last visited: 24.11.2020. 2019.

[84] Berne University Hospital Inselspital. *Morbidity and Mortality Due to Deferral of Aortic Valve Replacement in Patients With Severe Aortic Stenosis*. https://clinicaltrials.gov/ct2/show/NCT04333875. Last visited: 23.11.2020. 2020.

[85] Zichen Wang. *Practical tips for class imbalance in binary classification*. https://towardsdatascience.com/practical-tips-for-class-imbalance-in-binary-classification-6ee29bcdb8a7. Last visited: 25.11.2020. 2018.

[86] Yefeng Zheng and Dorin Comaniciu. "Marginal Space Learning." In: *Marginal Space Learning for Medical Image Analysis: Efficient Detection and Segmentation of Anatomical Structures*. New York, NY: Springer New York, 2014, pp. 25–65. ISBN: 978-1-4939-0600-0. DOI: 10.1007/978-1-4939-0600-0_2. URL: https://doi.org/10.1007/978-1-4939-0600-0_2.

[87] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, and Quoc V. Le. *Rethinking Pre-training and Self-training*. 2020. arXiv: 2006.06882 [cs.CV].

[88] cs231n. *CS231n: Convolutional Neural Networks for Visual Recognition - Transfer Learning*. https://cs231n.github.io/transfer-learning/. Last visited: 24.11.2020. 2020.

[89] cs231n. *CS231n: Convolutional Neural Networks for Visual Recognition*. https://cs231n.github.io/convolutional-networks/. Last visited: 11.02.2020. 2020.

[90] heart.org. *Aortic Stenosis Overview*. https://www.heart.org/en/health-topics/heart-valve-problems-and-disease/heart-valve-problems-and-causes/problem-aortic-valve-stenosis. Last visited: 03.12.2019. 2016.