Trym Nordal

Deep Learning Based Tracking of Anatomical Structures from Transesophageal Recordings of the Left Ventricle

Project report
Supervisor: Lasse Løvstakken
Co-supervisor: Gabriel Kiss
Trondheim, December 2018

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

Acknowledgement

I would like to express my gratitude to The Operating Room of the Future (FOR) at St. Olavs Hospital Trondheim for providing me with the opportunity to work on such an interesting project and for supporting me with the necessary resources.

I would also like to thank my supervisors Gabriel Kiss and Lasse Løvstakken for great guidance and assistance during this project, and I would like to thank Erik Andreas Berg Rye for collecting the data used in this project.

Abstract

Perioperative monitoring of a patient's heart during cardiac surgery is vital to ensure that the heart restores and maintains desired functionality, and transesophageal echocardiography (TEE) is widely accepted as a routine monitoring tool for these measures. However, the assessment of cardiac function from transesophageal echocardiography is currently performed qualitatively by echocardiographers. Performing the assessment quantitatively and automatically could potentially lead to more accurate and consistent functional parameter values, and speed up the monitoring process. Methods using speckle tracking have tried to do this, but satisfactory results have not been presented. In recent years, deep learning has been shown to perform with extremely high accuracy in complex tasks such as image segmentation and classification, also within the field of medical imaging. By using a convolutional neural network (CNN) with an encoder-decoder architecture, the feasibility of using deep learning to recognize and track two anatomical landmarks on images of the left ventricle has been examined in the work presented in this report. The movement of the two landmarks in a recorded sequence of the heart cycle can be used to calculate *mitral annular plane systolic excursion* (MAPSE), an important functional parameter. The current research task has been formulated as a segmentation problem where the CNN estimates the location of the two landmarks in raw TEE images. The accuracy of the output produced by the CNN is not satisfactory to our expectations. The results also show that the CNN is overfitting on the training data. This is likely due to an insufficient amount of training data. However, the results are promising when evaluating the output of the CNN visually. The accuracy of the network can be improved by increasing the amount of training data. By also incorporating temporal information from the image sequence, the network might provide more consistent tracking results. The work presented in this report serves mainly as a foundation for the author's master thesis due this following spring, which will be a continuation of this project.

TABLE OF CONTENTS

LIST OF FIGURES

Abbreviations

CNN	=	Convolutional neural network
MAPSE	=	Mitral annular plane systolic excursion
ReLU	=	Rectified linear unit
RNN	=	Recurrent neural network
TEE	=	Transesophageal echocardiography
TTE	=	Transthoracic echocardiography

CHAPTER 1
INTRODUCTION

1.1 Background

Echocardiographers at St.Olavs Hospital have identified a need for automatically performing quantitative perioperative monitoring of cardiac function. Based on this, the research group The Operating Room of the Future proposed this problem as a research topic for a master thesis.

1.1.1 Cardiac Monitoring

Before, during and after cardiac surgery the patient's heart is monitored carefully to ensure that the heart maintains and restores desired functionality. Today, the monitoring process involves measuring blood pressure, heart rate, respiratory rate, blood oxygen saturation, doing hemodynamic monitoring, clinical observation and echocardiographic evaluation [3]. This is a time consuming and complex process which is currently performed by trained personnel.

Echocardiography using medical ultrasound is a widely accepted tool for perioperative monitoring during cardiac surgery. Using the acoustic properties of muscular tissue, echocardiography is done by emitting high frequency ultrasound waves from a transducer (probe) to obtain structural images of the heart. Functional parameters derived from echocardiography are considered essential to the assessment of cardiac function [1]. During surgery, these parameters are currently determined by visual estimation done by experienced personnel [19]. This method of assessment is a time consuming, resource heavy, qualitative measure, which is susceptible to intra-observer variability. Additionally, if the pre- and postoperative cardiac function differs, time is a crucial aspect for ensuring the patient's safety. Performing these assessments quantitatively and automatically could potentially lead to more consistent and correct parameter values, while at the same time speeding up the process. These improvements will increase patient safety.

1.1.2 Automatic Assessment

Methods using speckle tracking have been tested for automatically deriving heart function parameters. However, transesophageal echocardiograpic (TEE) images and ultrasound images are inherently noisy, making speckle tracking based methods susceptible to drifting [20, 6]. This gives inaccurate results, and promotes an urgent need for improved automatic tracing methods.

In recent years, machine learning, and specifically deep learning methods, have given valuable results in tasks such as image segmentation, object classification and speech recognition [12]. Deep learning has also been successfully applied to medical imaging [14].

1.2 Aim of Study

The aim of this study is to examine the feasibility of automatically recognizing and tracking anatomical structures of the left ventricle from TEE using deep learning. If the physical structures of the left ventricle of the heart can be tracked automatically, this opens up for the possibility of automatically deriving functional parameters such as strain, annular plane displacement and tissue velocities. To examine this, a convolutional neural network will be trained to recognize two points on the left ventricle, one point on each side of the left atrioventricular valve. These two points are the basis of automatic derivation of the annular plane displacement, or *mitral annular plane systolic excursion* (MAPSE), which means that if these points can be tracked, MAPSE values can be derived automatically.

Additionally, the work presented in this report serves as foundation work for my master thesis. Intermediate objectives for this project includes getting familiar with the deep learning framework PYTORCH, process raw ultrasound images and getting familiar with similar work.

1.3 Structure of Report

This report starts with chapter 1, covering the motivation for the work done this semester and the aim of the study. In chapter 2, the theoretical concepts needed to understand the work is explained. This chapter covers heart physiology, basic concepts of ultrasound technology and deep learning, more specifically convolutional neural networks. Chapter 3 covers the processing and labelling of the data, and the implementation and justification of the deep learning model used in this study. In chapter 4, the results of the implemented system is presented, and chapter 5 discusses these results and presents future work. The report ends with a conclusion in chapter 6.

CHAPTER 2
THEORY

2.1 The Human Heart

The human heart is a muscular organ responsible for pumping blood through the blood vessels in the circulatory system. It is located in the thoracic cavity and connects to the systemic and pulmonary circulation. The systemic circulation transports oxygenated blood from the heart to the rest of the body and returns blood low in oxygen back to the heart. The pulmonary circulation transports the deoxygenated blood from the heart to the lungs, where it is oxygenated and returned to the heart.

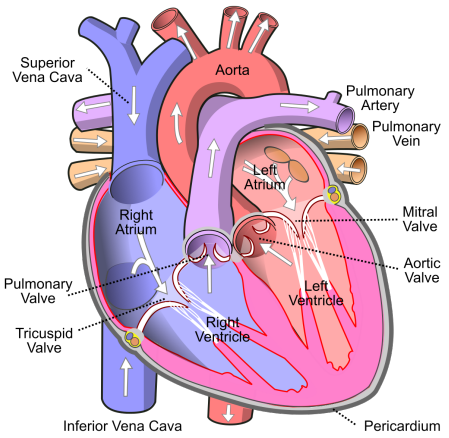


Figure 2.1: The human heart.

The heart consists of four chambers, two upper atria and two lower ventricles. The right and left atria, and the right and left ventricle are separated by the septum. Four

valves control the blood flow in the heart.

In the first phase of the heart cycle, the left and right ventricles receive blood from the left and right atria respectively, through the atrioventricular valves. The left atrium is filled up with blood from the pulmonary vein and the right atrium is filled up from the superior vena cava. This phase is called the *diastole*. In the second phase, the atrioventricular valves are closed, and the heart muscle (myocardium) contracts. The blood is pushed out from the left and right ventricle through the aortic and the pulmonary valves respectively. This phase is called the *systole*.

2.1.1 Cardiac Function

The assessment of cardiac condition and functionality is referred to as cardiac function. Global cardiac function is a measure of overall heart function. The size of the heart and pumping ability between end-diastole and end-systole are two measures of the global function. The displacement of the atrioventricular plane between end-diastole and end-systole is an example of a global measure. This parameter is known as *mitral annular plane systolic excursion* (MAPSE) and provides information about the systolic- and the diastolic function of the heart. Regional cardiac function is typically a measure of contraction patterns across the myocardium. Myocardinal strain is an example of regional cardiac function that measures the local deformation in the myocardium.

2.2 Medical Ultrasound Imaging

Ultrasound technology was first used for diagnostic purposes in the early 1950's, and is now one of the most widely used medical imaging modalities [2]. Compared to other imaging modalities, ultrasound is relatively inexpensive, portable and radiation free.

Modern medical ultrasound is performed by transmitting focused, high frequency sound waves from the transducer and reading the reflected waves, called the echo. The transmitted sound waves propagate through the human body and get partly absorbed, scattered and reflected when traveling through mediums of different acoustic property, such as muscle tissue, fat, bone and blood. The emitted waves are longitudinal and have a frequency in the range of 1 to 20 MHz. Increasing the wave frequency results in higher resolution, however it shortens the depth of penetration. Decreasing the frequency provides images with more depth, but with lower resolution. Using ultrasound to visualize cardiac structures is referred to as echocardiography.

2.2.1 Transesophageal Echocardiography

The most widely used tool for monitoring and assessing heart function is echocardiography. The two methods used for echocardiography are transthoracic echocardiography (TTE) and transesophageal echocardiography (TEE). TTE is done by placing the ultrasound probe on the exterior of the patient's chest. TEE is done by inserting the ultrasound probe into the patient's esophagus to obtain images from within the patient's body, closer to the heart. This method provides higher spatial resolution compared with TTE, due to the close proximity between the transducer and the heart. Another benefit of TEE is the

possibility of having the probe located within the patient's body while performing heart surgery. This is not possible using TTE, as the probe is then placed on the outside of the chest. The high resolution images obtained and the internal placement of the probe are two of the reasons why TEE is widely accepted as a routine tool during cardiac surgery to assess the placement of surgical tools and implants. TEE can also be a useful monitoring tool to detect changes in the anatomy and function of the heart, because the probe is inside the patient's esophagus during the entire procedure [15].

2.3 Deep Learning

Deep learning is a broad family of machine learning methods utilizing deep neural networks to learn data representation. Deep neural networks can do both unsupervised and supervised learning. In recent years, deep learning methods have shown to be effective solving complex tasks where classical machine learning methods fall short. The ability to learn representations in raw data with multiple abstraction levels is what separates deep learning methods from classical machine learning methods [12].

A standard neural network consists of a number of processing units, called neurons, ordered in layers. If every neuron in every layer is connected to every neuron in the previous and the next layer, the network is *fully connected*. Layers between the input and the output layer are called *hidden* layers. The input data is passed through the network and transformed by the neurons. At the output layer, the output from the network is compared to a reference, and the neuron parameters are adjusted to minimize the error between the output and the reference. The reference, also referred to as *ground truth*, can for instance be the correct classification of the input data, if the task performed by the network is classification. Having reference labels for the input data is what separates supervised learning from unsupervised learning. The universal approximator theorem states that a neural network with more than one layer and non-linear neuron activation can emulate any computable function [8]. The term deep learning refers to the act of teaching deep neural networks the complex features of a set of input data. The number of hidden layers in a neural network is the depth of the network.

There are many different types of neural networks. Standard fully connected neural networks perform well in classification tasks where the input consists of features of the different classes. Recurrent neural networks (RNN) perform well in tasks where the data contains temporal information, such as in natural language processing and speech recognition. Convolutional neural networks (CNN) perform well in tasks where the data contains spatial patterns, such as in image classification and image segmentation. The rest of this chapter will be focused on supervised learning with CNNs, as the current task is recognizing anatomical landmarks on the left ventricle by a CNN.

2.3.1 Convolutional Neural Networks

Convolutional neural networks are neural networks with a specialized architecture for processing data with a grid-like structure, such as images or time-series data [5]. These models are composed of a number of convolutional layers, which transform the input data into an output, while simultaneously learning increasingly abstract spatial features of the data.

The spatial feature learning in a CNN is inspired by how the mammalian brain does visual visual recognition, specifically in the primary visual cortex, or the V1 [5]. This is the first stage of visual processing where complex processing is performed. A convolutional layer is constructed to emulate three properties of the V1:

- The V1 is structured as a two-dimensional spatial map, where light arriving a certain location in the retina only affects a corresponding area of the V1. Similarly, the input to a convolutional layer is transformed to a feature map where the spatial relationships between the neurons are kept intact, in a two-dimensional structure.
- The V1 has *simple cells* with activity that can be described as applying a linear function on a small area on the input image. In a convolutional layer, this is facilitated by performing convolution on the input image.
- The V1 has *complex cells* which is similar to the simple cells. However, they are invariant to small changes in position. This inspires an operation called *pooling* in a convolutional layer

CNNs have been researched since the late eighties, and the first successful real-world application was used to recognize hand-written digits [13]. The breakthrough for deep learning, however, came in 2012 where the deep CNN AlexNet outperformed every other image classification system by a large margin [**?**]. More recently, this architecture has proven very accurate in tasks such as image classification and segmentation. The rest of this chapter will assume that the input data for the CNN is two-dimensional image data.

Convolutional Layers

A convolutional layer typically consists of three operations: **convolution**, **non-linear activation** and **pooling**.

The first operation, **convolution**, is a mathematical operation of two real-valued functions. In the context of a CNN, the mathematical expression is given by:

$$x^l_{i,j} = \sum_m \sum_n w^l_{m,n} o^{l-1}_{i+m,j+n} \tag{2.1}$$

$w^l_{m,n}$ is the weight at position $[m, n]$ at layer l where w^l is a two dimensional $k_1 \times k_2$ vector. The weight vector w^l is often referred to as a *kernel* or a *filter*. $o^{l-1}_{i+m,j+n}$ is the output from layer $l-1$ at position $[i+m, j+n]$. If l is the first layer in the network, then the vector o^{l-1} is the input image and $o^{l-1}_{i,j}$ corresponds to the pixel value at position $[i, j]$. The values from the previous output layer involved in a convolution, the leftmost gray area in figure 2.2, is referred to as a *local receptive field*. By summing over the product of the kernel and the local receptive field, we obtain $x^l_{i,j}$. By applying the kernel to the whole image in a sliding window fashion, we perform a convolution. The vector x^l is often referred to as *feature map* and contains extracted spatial features. Figure 2.2 shows the convolution operation.

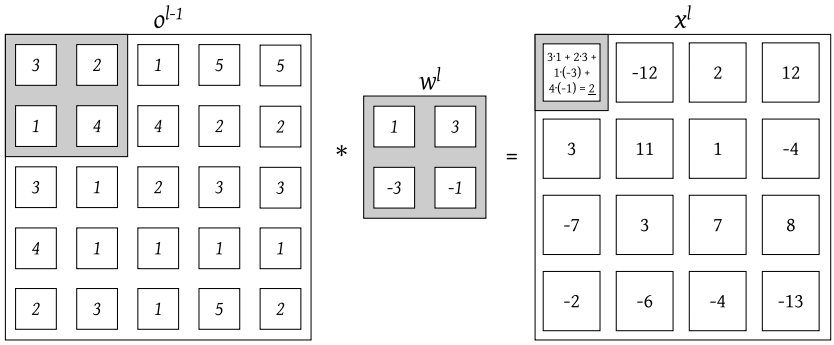


Figure 2.2: Convolution.

Non-linear activation is performed on the feature map to introduce non-linearity in the CNN. The non-linear activation is needed for the network to be able to learn non-linear patterns in the data [8]. The general expression for the activation function is given by:

$$a_{i,j}^l = f(x_{i,j}^l) \tag{2.2}$$

Many different activation functions exists, but the most used is the *Rectified linear unit* function, which is given by:

$$f(x) = max(0, x) \tag{2.3}$$

where x is a pixel value in the feature map. The ReLU function is said to be "the single most important factor in improving the performance of a recognition system" and it has shown to make learning more feasible when compared to networks with activation functions where both sides are saturated [9, 4]. ReLU is also computationally efficient, speeding up the training process. The ReLU operation is shown in figure 2.3.

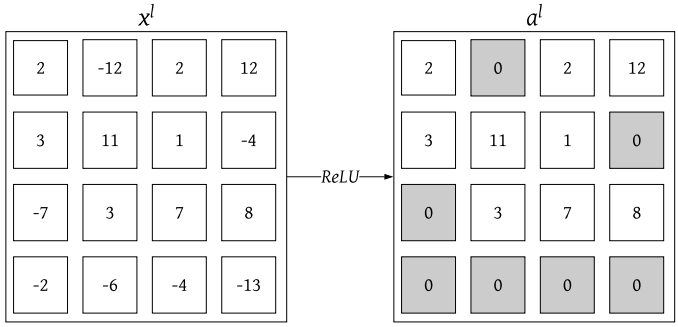


Figure 2.3: ReLU activation.

Pooling is done to simplify the feature map. The most used pooling function is called *max pooling*. Max pooling selects the highest value in a sliding $m \times m$ window of the output from the non-linear activation map, and creates a condensed feature map with only the highest values of the $i \times j$ activation map. The output from this operation, o^l, is the output from layer l. Max pooing with a 2×2 grid is shown in figure 2.4.

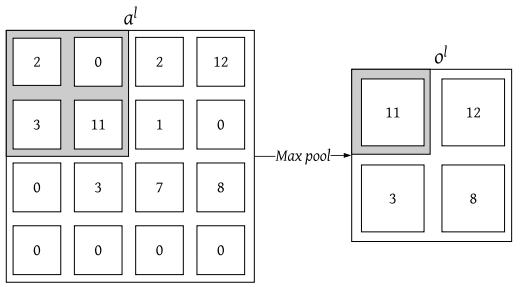


Figure 2.4: Max pooling with 2×2 grid.

As seen in figures 2.2 - 2.4, a typical convolutional layer will perform downsampling of the input image. The learning phase will in turn make the layer able to extract the most important information. When constructing a convolutional layer, one must specify the output size of the layer. The output size corresponds to how many feature maps the layer produces and states how many learnable kernels the convolutional layer has.

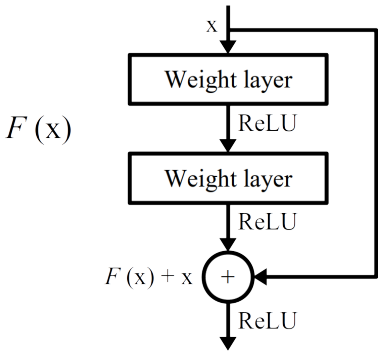


Figure 2.5: ResNet's skip connections.

In the past five years, configurations have been made to the convolutional layers to gain better results with CNNs. A well known and appreciated configuration is the introduction of shortcut connections in a network, called ResNet. The shortcut connections are shown in figure 2.5. ResNets shortcut connections makes it possible to stack more convolutional layers consecutively. This in turn increases the complexity, while still increasing classification accuracy. ResNet-152 (152 convolutional layers) beats the human performance in image classification of the ImageNet dataset, which is a dataset comprised of more than 14 million labeled images of 1000 different classes [7, 18].

Training a Convolutional Neural Network

In order to obtain useful results with a CNN, the network needs to be trained to extract the necessary features for the given task. To train a CNN by supervised learning, input data and reference labels are needed.

The first phase of the training process, and the only phase when doing inference, is called *forward pass*. In this phase, image data is passed through the convolutional layers as described in the section above. The second phase is where the learning happens. When the data is passed through the whole network, the output is compared to the reference labels by the **loss function**.

The loss function, or the criterion, computes the loss, or the error, between the output and the reference. The loss is the value the network is aiming to minimize. The most commonly used loss function in deep learning is called *cross entropy loss*, which is given as:

$$L_{crossentropy} = -\frac{1}{n}\sum_{x}\sum_{j}\left(y_j \ln a_j^L + (1-y_j)\ln(1-a_j^L)\right) \tag{2.4}$$

n is the number of training samples, x is a training sample and j is an output neuron that which corresponds to a class. y_j is the reference value at output j and a_j^L is the activation of neuron j at the output layer L. Based on this loss, the weights of the network should be adjusted. The weights that contributes the most to the error should be the most adjusted. The gradient of the loss function with respect to the weights gives this value, $\frac{\partial L}{\partial w}$. The expression for the gradient of the loss function w.r.t. every weight in the network can be found by applying the chain rule:

$$\frac{\partial L}{\partial w_{i,j}^l} = \sum_{i=0}^{H-k_1}\sum_{j=0}^{W-k_2}\frac{\partial L}{\partial x_{i,j}^l}\frac{\partial x_{i,j}^l}{\partial w_{m',n'}^l} \tag{2.5}$$

L is the loss function. $w_{m',n'}^l$ is the weight at position $[m', n']$ in layer l with size $k_1 \times k_2$. $x_{i,j}^l$ is the pixel value in feature map x at position $[i, j]$ in layer l. The last term, the gradient of $x_{i,j}^l$ w.r.t $w_{m',n'}^l$, can be thought of as how the feature layer x^l is affected when weights in w^l are changed. This is equal to the output from the previous layer, $o_{i+m',j+n'}^{l-1}$. The first term after the summation, the gradient of the loss function w.r.t. the feature map x^l, can be thought of as the way in which the loss function is affected when pixel values in x^l are changed. These gradients are referred to as deltas. The gradient of the loss function w.r.t. the weights can be written as:

$$\frac{\partial L}{\partial w_{i,j}^l} = \sum_{i=0}^{H-k_1}\sum_{j=0}^{W-k_2}\delta_{i,j}^l o_{i+m',j+n'}^{l-1} \tag{2.6}$$

In a CNN the gradients are computed using an algorithm called **backpropagation**. By applying backpropagation, information produced by the loss function propagate backwards through the network in order to compute the gradients.

When the gradient of the loss function w.r.t. every weight in the network has been found by backpropagation, the weights need to be adjusted according to how much they

contribute to the loss. This is done by **optimization** methods, and the most common method is called *stochastic gradient descent* (SGD). SGD applies an update rule for the weights given by:

$$w_{i,j}^{l} \leftarrow w_{i,j}^{l} - \eta \frac{\partial E}{\partial w_{i,j}^{l}} \tag{2.7}$$

where η is the learning rate. The reason for the term *stochastic* is because SGD introduces a source of noise, the random sampling of n training examples. Another optimizer commonly used in deep learning is called the *Adam* optimizer. The name Adam is derived from *adaptive moment estimation*. The difference between SGD and Adam is that while SGD has one learning rate for the whole training process, Adam gives each parameter in the network an adaptive learning rate. The Adam optimizer has been shown to outperform other optimization methods in terms of rate of convergence [11].

When training a CNN and neural networks in general, the most common method is to divide the training data into *minibatches*. When using minibatches for training, n number of images are randomly chosen for a minibatch. Every image in the minibatch completes forward pass, and their individual losses are computed. The final minibatch loss is the average of these computed individual losses. Backpropagation computes the gradients for the minibatch loss and the weights are updated with 2.7. The loss computed on the minibatch serves as an estimate of the total loss on the training data. A large minibatch size will typically give a more correct estimate of the total training loss. However, it has been shown that smaller minibatch sizes often perform better because the network avoids sharp local minimizers [10].

An important aspect of finding a good model is to measure how well it generalizes to new data that it has not been trained on previously. Neural networks can be trained too much, which results in a model that is specialized on the training data and fails to generalize to new data. To measure how well the CNN generalizes, the dataset is divided into a *training set* and a *validation set*. For every time the training data has been passed through the network and the weights have been updated, the validation data is fed to the network for evaluation of accuracy. For this evaluation to be valuable, the weights cannot be updated based on the validation loss. One iteration through the training and validation data is referred to as one *epoch*. The accuracy on the validation data should increase as the loss on the training data decreases. At some point the validation accuracy will start to decrease despite the training accuracy continuing to increase. This is called overfitting, and it happens when the network is becoming too specialized on the training data. The optimal point for stopping the training is when the validation accuracy is the highest.

All the parameters set before starting to train are called *hyperparameters*. These include the learning rate of the optimizer, the number of images in a minibatch etc.

2.3.2 Image Segmentation

The process of estimating a class for every pixel in an image is called image segmentation. Different models utilizing deep learning has provided promising results in this task, and segmentation for medical imaging by deep learning has consequently been heavily researched [12, 14].

A network architecture widely used in image segmentation tasks is the *decoder-encoder* architecture. The encoder part of the network uses standard convolutional layers described in the section above. The decoder part of the network can be implemented by an operation called transpose convolution. This operation reverts the spatial resolution by applying a learnable filter.

Within the field of medical image analysis, image segmentation is the most frequent subject of research applying deep learning [14]. The most well known architecture for deep learning applied to medical imaging is the U-Net CNN. This architecture has equal amounts of up-sampling layers and down-sampling layers, and it introduced skip connections where opposing convolutional layers are concatenated with transpose convolutional layers[17]. This makes the network able to directly produce segmentation masks from the given input, and the network can be trained end-to-end, with standard backpropagation and optimization.

The output from a network performing segmentation is a number of segmentation masks which corresponds to the number of classes that are being segmented from the input image, one mask per class. These masks can be described as probability maps, where the pixel values are the probability of the class being present is this area. A common metric used to determine the accuracy of the estimated segmentation is called the Dice coefficient. This is a metric that is measuring the overlap between the reference segmentation and the estimated segmentation, and it is used to determine the accuracy on the validation data to keep track of how well the network is learning. The Dice coefficient is given by:

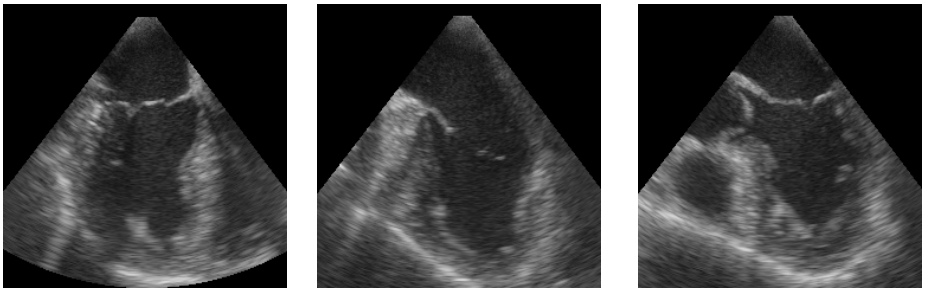
$$DSC = \frac{2TP}{2TP + FP + FN} \tag{2.8}$$

CHAPTER 3
MATERIAL AND METHOD

The task of recognizing and learning structures of the left ventricle using deep learning was formulated as a segmentation problem. Two anatomical landmarks in TEE images of the left ventricle were then chosen as the two points that the CNN would learn to find. These two points are located on each side of the mitral valve and can be used to derive MAPSE.

3.1 Material

The dataset utilized for this project contains approximately 4000 TEE images. The images are collected from five patients which all underwent coronary artery bypass surgery. There are recordings of the left ventricle from three different views, and in every patients' data except one there are recordings from both before and after surgery. The three views captured are 4 chamber, 2 chamber, and long axis views of the heart. The ultrasound scanners used were Vivid S70 and Vivid E9 from GE Vingmed in Horten. A 3D TEE probe 6VT-D was used to acquire the images. These images are recorded over 3 to 5 heart cycles. All of the recordings and the use of the recorded datasets is done with both patient consent and approval from the ethics committee at St. Olavs University Hospital. The recordings were not subjected to any form of selection process.



(a) 4 chamber view. **(b)** 2 chamber view. **(c)** Long axis view.

Figure 3.1: TEE images from three different views.

3.2 Method

3.2.1 Data Labelling

For the deep learning method to work, the TEE images had to be labeled. For this, a MATLAB script with a simple user interface was made. This script loads an image series from h5 format, displays the first image, and lets the user click somewhere in the image to select the desired point of reference. This point is saved as a binary segmentation mask where n pixels around the point are set to 1 while the rest of the pixels are set to 0. When the user has clicked and selected two points, the next image is shown. To get some consistency between two following images, the segmentation mask of the previous image is laid on top of the new image. After every image in the image series was labeled, the ground truth segmentation masks and the images were saved as image files. The reason n pixels around the selected point in the image was set to 1 was due to clutter, making the exact locations hard to pin down in every image. For this reason, the segmentation masks were referred to as reference masks and not ground truth. Another reason for setting a larger area to 1 was to provide the CNN with a better spatial understanding of the surrounding elements of the point. The final segmentation masks had a radius $n = 8$ pixels.

3.2.2 Network Architecture

The inspiration behind the implemented CNN architecture was a project called DeepLabCut [16]. DeepLabCut implemented a CNN for tracking points on animals and insects, and their implementation was greatly successful. Based on the DeepLabCut project, ResNet-50 was chosen as this projects encoder network. Unfortunately, DeepLabCut does not describe their method of upsampling the features from the encoder. For the current project, the decoder was therefore chosen to utilize the same architecture as U-Net. The resulting architecture looked like this:

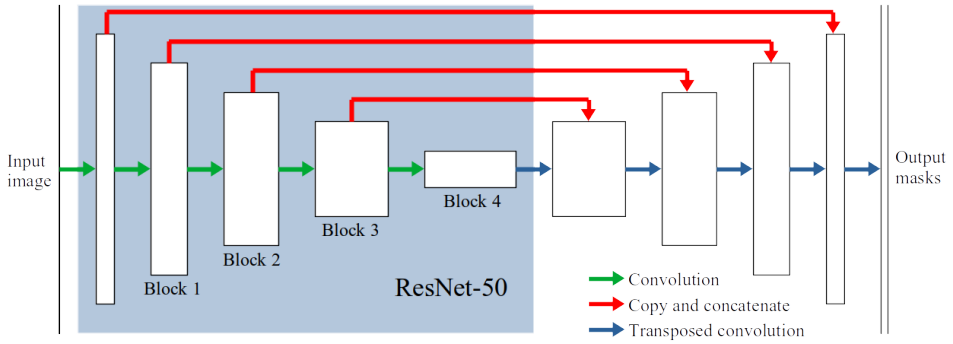


Figure 3.2: Network architecture.

3.2.3 Implementation

The framework used to implement, train and evaluate the model was PYTORCH. Pytorch is an open source Python library that offers GPU accelerated tensor computation and deep learning functionality. First, the network architecture was built. The ResNet-50 model was available for download with weights pretrained on image-net. Next, the ResNet model had to be somewhat modified. ResNet's input layer is built for RGB-images, and it therefore contains three channels. The input layer had to be changed to receive only one channel, due to the TEE images being grayscale. Secondly, because the model was not going to be used for image classification, but segmentation, the final and fully connected layer had to be removed. Each block of the decoder was constructed with one transposed convolution layer and one convolutional layer. After the transposed convolution, the output from the corresponding ResNet block was concatenated with the output from the transposed convolution. The concatenated outputs were then fed to the convolutional layer, which gave the output from the block. This output was fed to the transposed convolution layer of the next block. The output of the network produced three segmentation masks, one for each point on the two sides of the mitral valve, and one for the background.

The data loader module was made to automatically feed the network batches of images and their corresponding ground truth masks. Data augmentation was included. The images were originally 240×240, while ResNet input is 224×244. The images and their corresponding segmentation masks were first randomly cropped to 224×224. To emulate varying rotation in the TEE images they were rotated randomly ± 5 degrees before being fed to the network.

The final step was to configure the training regime. The number of epochs was set to 200, and for every epoch the validation set was used to determine the accuracy of the network. The accuracy metric chosen was the Dice similarity coefficient, described in section 2.3.2, at thresholds 0.5 and 0.75. For every tenth epoch the weights of the network were saved. The selected loss function was the cross entropy loss and the optimizer was chosen be the Adam optimizer.

After the implementation, the model was trained on a remote GPU cluster provided by FLOYDHUB.COM. The dataset was divided into a training set and a validation set and uploaded anonymized. To keep the validation and training data as different as possible, every

sequence displaying a view was taken from different patients. The validation sequences were chosen randomly. The GPU was an Nvidia Tesla K80 with 12 GB memory and 64 GB RAM.

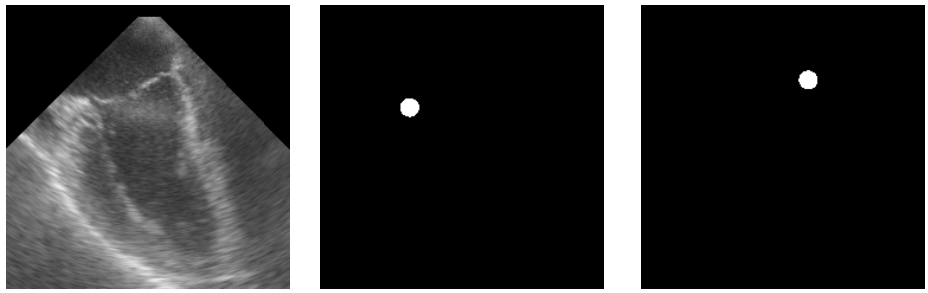
3.2.4 Post Processing of Network Output

The output from the CNN was the estimated probability masks of the location of the two points. To visually evaluate the estimated points, a threshold of 0.5 was applied to the probability maps to obtain the binary segmentation masks. To obtain the final coordinate of the estimated points, the centroid of the estimated area, (pixel value 1) was calculated.

CHAPTER 4
RESULTS

4.1 Data Preparation

The MATLAB script made for loading, displaying, labelling and converting the TEE images produced reference segmentation masks, which are required for the deep learning model. These segmentation masks were quality controlled by FOR researcher, Gabriel Kiss. In many of the images, the location of the AV points were easily detectable. Several images had clutter, both from reverberation and shadowing. Figure 4.1 shows an example image and the corresponding reference segmentation masks. Figure 4.2 shows the reference masks on top of the TEE image.



(a) The TEE image **(b)** Left segmentation mask **(c)** Right segmentation mask

Figure 4.1: TEE image with reference segmentation masks

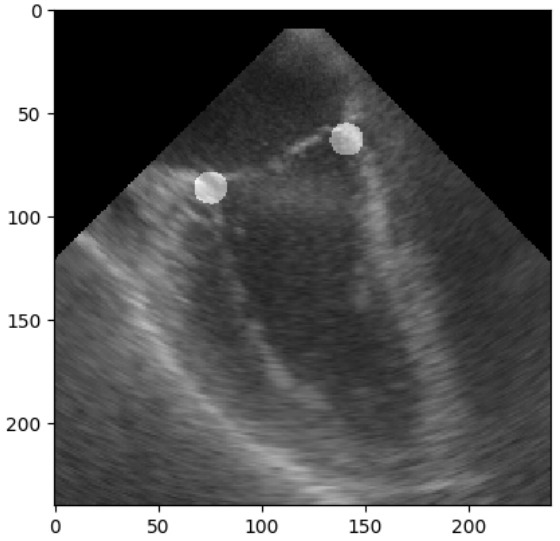


Figure 4.2: Image with segmentation masks on top.

4.2 Deep Learning

The results of training the implemented CNN can be seen in figures 4.3 - 4.5. Here, the accuracy is plotted against the number of epochs trained.

With a threshold of 0.5, the Dice coefficient of the training data increases rapidly during the first 5 epochs, before it slows down and becomes stationary for a few epochs with an accuracy value of approximately 0.75. Then, after approximately 70 epochs, the accuracy again starts to increase. The validation accuracy increases for less than 20 epochs, reaching a peak of 0.4 after 15 epochs. After 30 epochs it appears to decline. After 80 epochs, the accuracy stabilizes at just over 0.3.

The plot in figure 4.4, where threshold is 0.75, shows that the increase in training accuracy grows more slowly compared with the plot above. After near 80 epochs, the training set accuracy stabilizes at just below 0.8. The validation accuracy grows steadily for approximately 60 epochs, before settling around 0.3, with a peak after 69 epochs at 0.34. The difference between these validation accuracies decreases as the number of epochs increases.

Figure 4.5 shows the loss value plotted against number of epochs for the training and validation data. The loss on the training data is continually decreasing. The loss on the validation data is seen decreasing during the first 4 epochs, before steadily increasing.

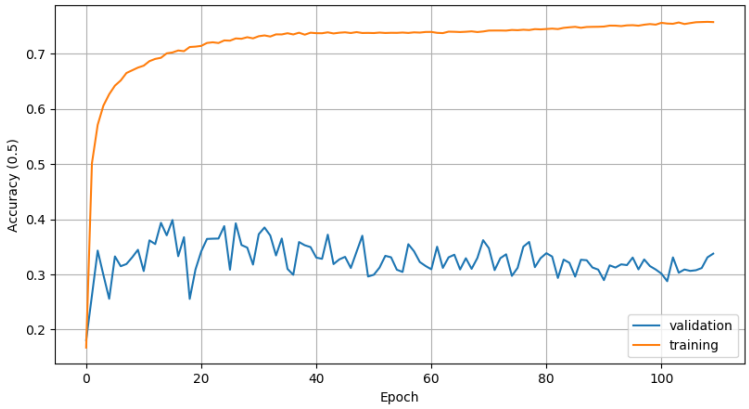


Figure 4.3: Training and validation accuracy with threshold 0.5.

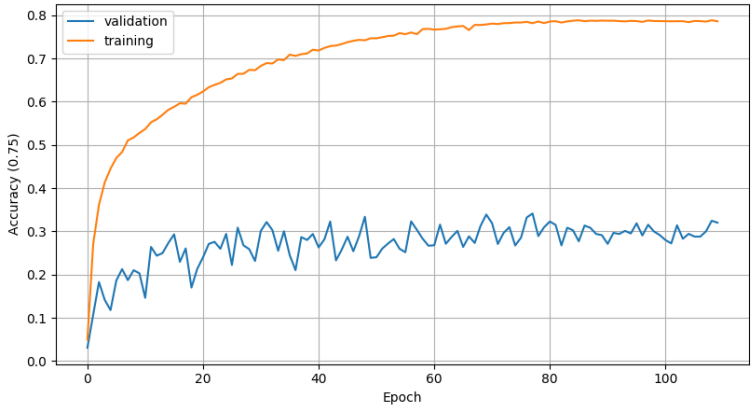


Figure 4.4: Training and validation accuracy with threshold 0.75.

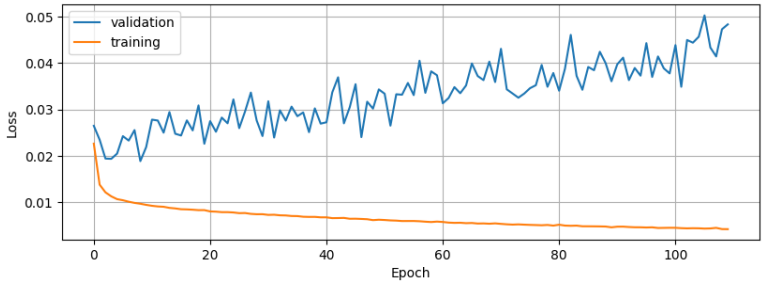


Figure 4.5: Training and validation loss.

4.3 Visual Inspection

Figure 4.6 shows TEE images with the estimated AV points in marked red. The network has not been trained on these images.

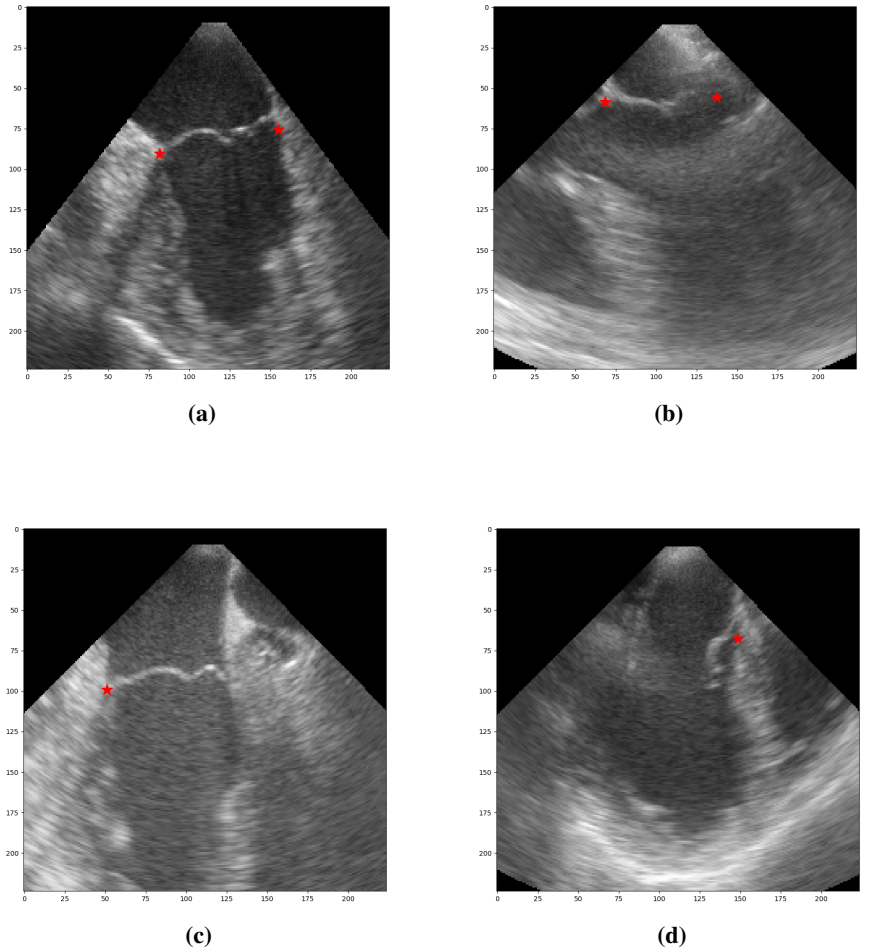


Figure 4.6: Estimated points with four different results.

Figure 4.6a shows a clear 2 chamber view image. Both points are estimated within the correct area. Figure 4.6b shows an image with reverberation clutter on the right side. Both points has been correctly and satisfactory estimated. Figure 4.6c shows the left point correctly estimated, and figure 4.6d shows an image with reverberation clutter one the left side and the right point correctly estimated.

4.4 Use Case

Figure 4.7 shows the two AV points of an image series in the validation set. The first plot shows the left point, with estimated location in blue and reference location in orange. In one of the frames the model could not find an estimate, however the distances between the remaining estimated and reference points have an average of 1.51 pixels. The second plot shows the right point estimated location and reference location. Here, four frames are missing. Further, the distances between the estimated points and reference points are larger, with an average of 3.43 pixels.

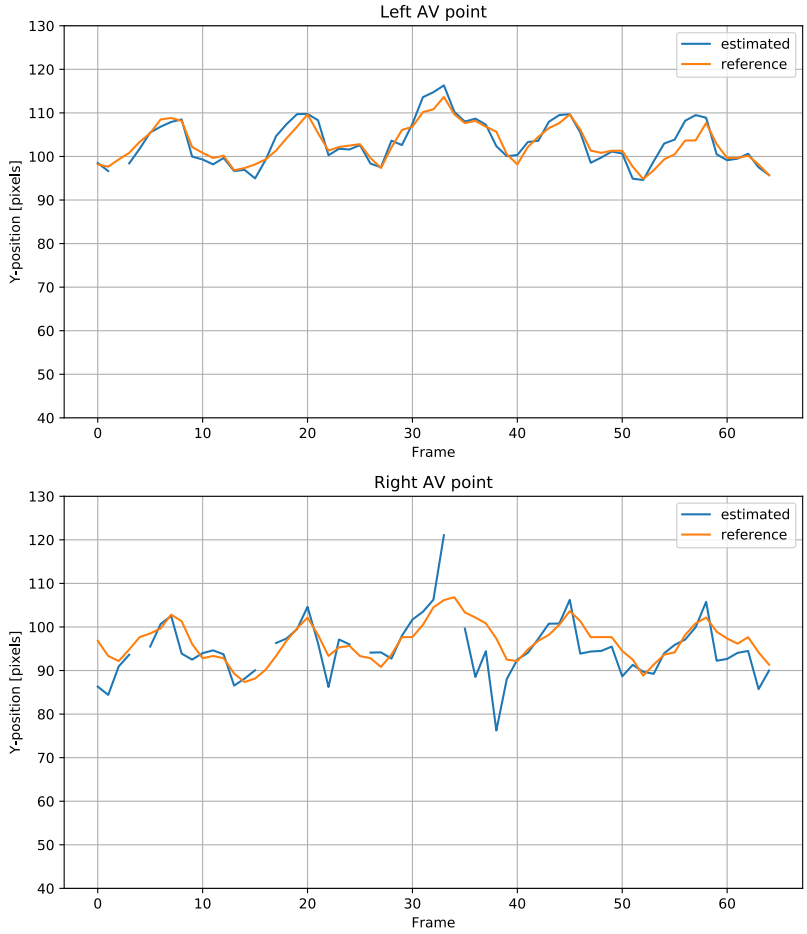


Figure 4.7: Y-coordinate of the estimated points and reference points for every frame in a sequence. The left point in the top plot and the right point in the bottom plot.

CHAPTER 5
DISCUSSION

5.1 Data Preparation

The script used for image labelling produced solid reference segmentation mask for the CNN. The number of pixels near the chosen points were set to 1 with a radius $r = 8$ pixels. This number was set prior to the labelling, and solely the masks produced with this radius were used for the training. Increasing this number might improve the network's ability to recognize the area surrounding the selected point, as the area would be more distinct compared with the rest of the image. However, the estimated coordinates might then show a higher variance, because the area the two points could be located in would be expanded. On the other side, decreasing this number might make it more challenging to locate the correct area because similar patterned areas could be present several places in the image.

5.2 Deep Learning Model

There is no obvious best approach to solving the problem of recognizing landmarks on TEE images of the left ventricle. Deep learning might also not be the best method of doing so. However, seen as more simple methods such as speckle tracking have not produced satisfactory results in this application, and that great advancements have been made in the field of medical imaging with deep learning during the recent years, it is expected that deep learning methods could be suitable and advantageous for a complex task such as this one.

Due to the time constraint of this project, only one model was implemented. The network was created based on a similar project, tracking specific points on animals and insects in grayscale images. The project model was based on the encoder-decoder architecture, with convolutional and transpose convolutional layers. With data being TEE images, ResNet-50 is a good choice for feature extraction. Firstly, ResNet-50 is one of the

CNNs with the highest accuracy in the task of image classification, which means that it is outstanding at extracting spatial features from images. Secondly, it can be downloaded with pretrained weights, and it can be trained for any visual recognition purpose. Using ResNet as the encoder makes the network able to understand the spatial structures in the TEE images. The decoder was based on the architecture of U-Net. When using transpose convolutional layers in the decoder, the CNN learns how to use the spatial features extracted by the encoder to recognize the areas where the points could be and generate a full sized output map. By copying and concatenating the outputs from the convolutional blocks in the encoder, features from different abstraction layers are utilized to produce the output.

Deep learning methods require a large amount of data to produce satisfactory results. Based on the results presented in section 4.2, it can be seen that the model is overfitting. This can be seen from the accuracy plots with threshold 0.5, figure 4.3. The validation accuracy increases for a few epochs, and then starts to decrease, while the training accuracy continues to increase. The large gap between the validation accuracies and the training accuracies is another an indication that the CNN is overfitting. The third and most definite indication that the model is overfitting is that the loss on the valuation data does not decrease while the loss on the training data continues to decrease, as seen in figure 4.5. The model is specializing on the images from the training dataset, and does not generalize well to new data.

There are several factors that might have caused this model to overfit. The most obvious explanation is the small amount of unique training data. The dataset used in this experiment contains almost 4000 TEE images. However, every image sequence contains images which are all very similar. In addition, there are only three different views recorded for every patient, making it only three truly unique sequences per patient. As the population consists of five patients, this amounts to only 15 truly unique image series in total. For this reason, the most effective way to avoid overfitting would be to increase the amount of training data.

The images fed to the network were randomly shuffled, and the estimation of the image points were conducted in a one shot manner, with only spatial information. Images from TEE, however, are from recorded sequences displaying the heart as it contracts and relaxes, often over multiple cycles. An alternative approach for estimating the location of the two points could be to utilize the temporal information in the image sequences. As the heart contracts, the two points normally move downwards in the y-direction, and upwards when the heart relaxes. The motion of these two points is periodic, and a model estimating the points based on both spatial information and temporal information could possibly lead to better performance.

5.3 Visual Inspection and Use Case

From a visual inspection of the output from the network, figure 4.6, the estimated points are accurately located in the correct areas when they are found. Figure 4.6a shows a clear image with both points marked in the correct area. Correspondingly accurate results should be expected from a system performing tracking in a real world setting. Figure 4.6b shows both estimated points in an image containing some reverberation noise. The

network should be robust to clutter, as clutter is often present in TEE images. Figure 4.6c however, shows a clear image where the right point is missing. A model performing this task should have no problem locating the right point in this image. Figure 4.6d, shows the right point in the correct location but no left point. Reverberation clutter can also be seen in the left area, and this is likely the reason why the left point is missing.

By inspection of the plots in figure 4.7 and the average distance from the reference for both points, it is clear that the network recognizes the point on the left side of the mitral valve more often, and with higher accuracy, compared with the right point. The error between the estimate and the reference is impressively small for the left point. For the right point, the error is larger, but for most of the points, the tracking is adequate. Upon further inspection, it should also be mentioned that the image series used for these results contains images from 2 chamber view without much clutter. The results from this image series might therefore be better than the average. However, since these points are to be used for MAPSE estimation, outliers in the y-direction will provide false and misleading values and must be avoided.

5.4 Future Work

Based on the results from the training of the network, it became clear that the amount of training data is not sufficient. The most important next step for further work should therefore be to acquire more training data.

Another important improvement would be to implement and test different models and architectures. Using models that utilize both the spatial and temporal information available in the data should lead to improved estimation accuracy. This can be done by adding RNN layers or perform convolution in the time domain by inputting a sequence of images.

Future work should also aim at finding accuracy metrics better fitted for this application. For example, this could be the distance between estimated points and the reference mask centroid, or the presence of estimated points in a defined radius from the reference mask centroid. This system should eventually be able to estimate global and regional cardiac function parameters. An important and necessary module to add is a module calculating function parameters based on the estimated point coordinates for the images in a sequence.

A separate test data set would be valuable and useful in future work. Having a test data set is important when evaluating more models for good comparison. The test set would benefit the evaluation and comparison of different implemented models. This dataset should ideally consist of new data from several patients, to emulate the performance of the trained system in a real world setting.

Finally, the work presented in this report serves mainly as foundation work for the author's future master project. After familiarizing with the deep learning framework PYTORCH by building and training a CNN for the task presented, and having made the data processing script, future work will focus predominantly on implementing and testing different network architectures and configurations, with more data.

CHAPTER 6
CONCLUSION

By using a CNN with an encoder-decoder architecture, the feasibility of using deep learning to recognize points has been examined. The research task has been formulated as a segmentation problem. Reference segmentation masks have been made for the whole dataset. The implemented and tested model in this project is based on DeepLabCut and U-Net.

The results from training and evaluating the validation data show that the model described in this report does not produce satisfactory results. The accuracy of the validation data does not continue to improve after just a few epochs, and after running a few more epochs it decreases. The accuracy of the training data first stabilizes after an initial increase, before starting to improve further, indicating that the model is overfitting. This is likely a result of a limited amount of training images. The network architecture itself might not be optimal, as it uses solely spatial information when estimating the location of the two AV points. However, by visual inspection the estimated points show promising results. The work presented in this report serve mainly as foundation for further work, which will be focused on acquiring more data, and implementation and testing of different architectures incorporating temporal information. This report presents work that has provided promising results and valuable insight into the task of performing quantitative monitoring with deep learning. The author of this report cannot wait to continue with the project after a short Christmas vacation.

BIBLIOGRAPHY

[1] James D.Thomas and Zoran B.Popović. Assessment of left ventricular function by cardiac ultrasound. *Journal of the American College of Cardiology*, 48:2012–2025, 2006.

[2] I. Edler and K. Lindström. The history of echocardiography. *Ultrasound in Medicine and Biology*, 30(12):1565–1644, 2004.

[3] Vincent et al. Perioperative cardiovascular monitoring of high-risk patients: a consensus of 12. *Critical Care*, 19:224, 2015.

[4] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

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

[6] Martin Henrik Hassel. Perioperative monitoring of cardiac function based on transesophageal echocardiographic data. Master's thesis, NTNU, 6 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[9] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, 2009.

[10] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016.

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.

[14] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M.van der Laak, Bramvan Ginneken, and Clara I.Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

[15] MartinaNowak-Machen. The role of transesophageal echocardiography in aortic surgery. *Best Practice Research Clinical Anaesthesiology*, 30:317–329, 2016.

[16] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21:1281–1289, 2018.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[19] Annette Vegas and Massimiliano Meineri. Three-dimensional transesophageal echocardiography is a major advance for intraoperative clinical management of patients undergoing cardiac surgery: A core review. *Anesthesia Analgesia*, 110:1548–1573, 2010.

[20] Jens-Uwe Voigt. Definitions for a common standard for 2d speckle tracking echocardiography: consensus document of the eacvi/ase/industry task force to standardize deformation imaging. *European Heart Journal - Cardiovascular Imaging*, 16:1–11, 2015.