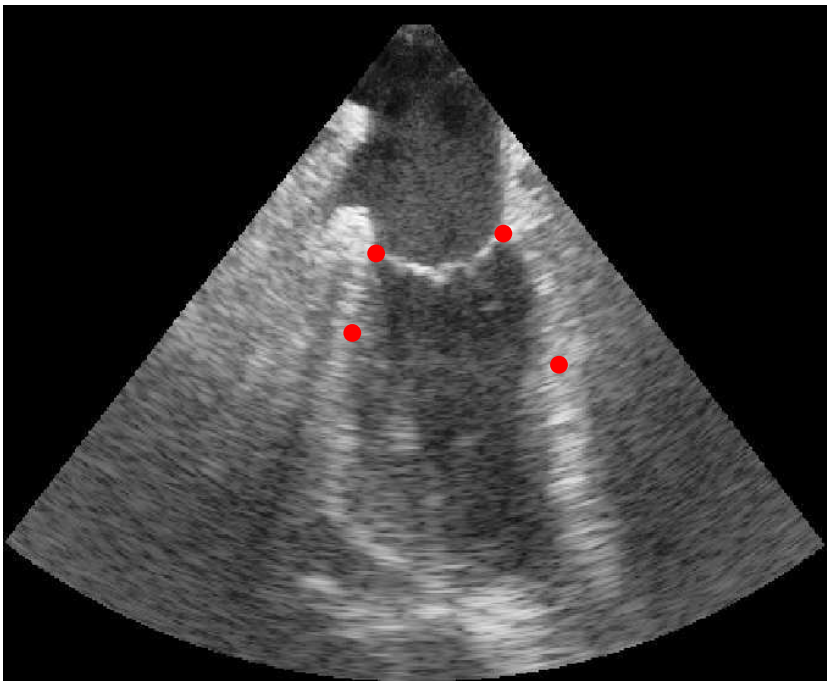
Master's thesis

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

Torjus Haukom

Basal Strain Estimation in Transesophageal Echocardiography using Unsupervised Deep Learning

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Gabriel Hanssen Kiss and Ilangko Balasingham
June 2019



NTNU
Norwegian University of Science and Technology

Torjus Haukom

Basal Strain Estimation in Transesophageal Echocardiography using Unsupervised Deep Learning

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Gabriel Hanssen Kiss and Ilangko Balasingham
June 2019

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

NTNU
Norwegian University of
Science and Technology

Summary

Patients undergoing cardiac surgery run the risk of serious complications during and after the intervention, and have their hearts monitored through the perioperative period. Echocardiographic assessment of the contractility of the heart can be an important component in this monitoring, but is often highly qualitative as visual inspection remains the dominant technique. In recent years, efforts have been made to develop standardized quantitative measures of cardiac function, strain being one of them. Strain imaging technology is available from the major vendors of ultrasound equipment and software, but currently requires manual annotation of the images. In addition, the available technology suffers from high inter- and intra-observer variability, making automation of the strain estimation task desirable. Continuing advances in transesophageal echocardiography (TEE) are believed to facilitate this automation.

This thesis aims to contribute towards the full automation of perioperative echocardiographic monitoring through investigating the feasibility of fast, automatic longitudinal strain estimation in the basal segments from unselected 4-chamber, 2-chamber, and long-axis TEE images using unsupervised deep learning methods. A strain estimation pipeline is proposed, composed of two major components: myocardial landmark detection and frame-to-frame displacement estimation. Using the estimated displacements, the detected landmarks can be tracked through the cardiac cycle and used to estimate strain. The landmark detection algorithm assumes known mitral annulus location and employs a series of filtering operations to highlight a suitable landmark in the myocardial segment below it. The displacements are estimated using a fully convolutional neural network (CNN) and cubic B-spline interpolation, inspired by recent work in image registration. The CNN is trained in an unsupervised manner, removing the need for manual annotation of the ground truth, and estimates a low-resolution displacement field. This low-resolution field is then interpolated to produce a dense displacement field describing the motion of each individual pixel between two consecutive frames.

Three CNN models were trained and evaluated on samples from 94 patients (57 for training, 14 for validation, 23 for testing). The most succesful model shows promising results in the 4- and 2-chamber views, especially when the images are of high quality. Notably, it achieves a mean absolute difference (MD) of $(2.96 \pm 3.13)\%$ on strain estimates in the inferoseptal segment in the 4-chamber view when compared to a commercially available method. In the other segments, the MD ranged from 4.04% to 6.17%, performing worst on the long-axis samples. The largest differences were observed in samples where the image quality was poor, leading to the conclusion that strain estimation using this method is feasible if efforts are made to improve robustness or if image quality can be guaranteed.

Sammendrag

Pasienter med behov for hjertekirurgi løper en risiko for alvorlige komplikasjoner under og etter inngrepet, og blir derfor overvåket gjennom den perioperative perioden. Ekkokardiografisk evaluering kan være en viktig del av denne overvåkningen, men ettersom visuell inspeksjon er den dominerende metoden er slik evaluering i høy grad kvalitativ. I senere år har det vært stor interesse for å standarisere kvantitative indikatorer for hjertefunksjon, og en mye brukt slik indikator er myokardiell strain. Strainavbildningsteknologi er i dag tilgjengelig fra de fleste større leverandører av ultralydutstyr og -programvare, men disse krever at personell med opplæring annoterer bildene manuelt. Det er også vist at det er stor variasjon i slike målinger, både mellom observatører og mellom leverandører av utstyr, noe som gjør det ønskelig å standardisere automatiske strainmålinger. Fortsatt rask utvikling innen transøsofagal ekkokardiografi (TØE) er ventet å fasilitere denne automatiseringen.

Denne oppgaven har som mål å bidra til full automatisering av ekkokardiografisk overvåkning i den perioperative perioden. Her undersøkes muligheten for rask automatisk longitudinal strainestimering i basalsegmentene fra transøsofagale 4-kammer-, 2-kammer- og lang-aksebilder gjennom bruk av ikke-veiledet (eng: unsupervised) dyp læring. Det blir foreslått en metode i fire steg, der de to viktigste er deteksjon av gjenkjennbare punkter på myokardium og estimering av bilde-til-bilde forflytning. De estimerte forflytningene brukes til å følge de detekterte punktene gjennom hjertesyklusen, og avstanden mellom dem gir et estimat av strain. Punktdeteksjonen antar at mitralanullus' posisjon er kjent, og filtrerer bildene i flere omganger for å fremheve de mest distinkte punktene på basalsegmentet under. Forflytningene blir estimert av et konvolusjonelt nevralt nettverk og kubisk B-splineinterpolasjon, inspirert av nylig publisert arbeid innen bilderegistrering. Nevralnettet estimerer et lavoppløst forflytningsfelt som så interpoleres til et felt med full oppløsning, med én forflytningsvektor per piksel i bildene. Nevralnettet trenes uten veiledning slik at de sanne forflytningene ikke behøver å være kjent.

Tre konvolusjonelle nevralnett ble trent og evaluert på undersøkelser fra 94 pasienter (57 til trening, 14 til validering, 23 til testing). Det beste nettverket viser lovende resultater på 4- og 2-kammerbildene, spesielt der bildekvaliteten er høy. Dette nettverket oppnår en gjennomsnittlig absolutt differanse (GD) på $(2.96 \pm 3.13)\%$ i det inferoseptale segmentet i 4-kammerbilder når det sammenlignes med en kommersiell metode. I resten av segmentene ligger GD mellom 4.04% og 6.17%, med verst ytelse på lang-aksebildene. Størst avvik ble observert i undersøkelser med lav bildekvalitet. Det leder til konklusjonen om at strainestimering med denne metoden er mulig dersom robustheten forbedres eller om bildekvaliteten kan garanteres.

Preface

This thesis represents the end of my studies, for now. These five years have gone by faster than I could have ever imagined, and have left me with lots of knowledge, lots of new friends, and a girlfriend. All of which have been instrumental to completing this thesis.

I chose this project because I believed it would be challenging, and that it would require a wide range of the skills and knowledge that I have acquired throughout my studies. I can now say that I was right. This project leans on knowledge of mathematics, physics, and electronics and has required me to leverage skills in programming, statistics, and, perhaps most important, communication. Combining the work and results into a report has been one of the biggest challenges of my life, and it is with relief that I now hand it in.

Still, looking back, I have enjoyed the opportunity to dive deep into one project, and I have learned a lot that I never would have otherwise. Meeting and working together with experts in another field than my own has also been a great experience, and I emerge with greater appreciation and respect for the knowledge of others.

Acknowledgements

Materials needed for this thesis were funded by the Norwegian centre for Minimally invasive Image guided Therapy and medical technologies (NorMIT), a collaborative infrastructure between the St. Olavs hospital, the Norwegian University of Science and Technology (NTNU), and SINTEF AS, Trondheim, Norway.

Håvard Dalen (Ph.D., M.D.), Espen Holte (Ph.D., M.D.), and Bjørnar Grenne (Ph.D., M.D.) acquired the ultrasound images required to perform the experiments presented in this thesis. Without them, this project would have been impossible to complete. Erik Andreas Rye Berg (M.D.) provided reference values of strain in a subset of these samples. He also coauthored an abstract describing this work that was submitted to the 2019 IEEE International Ultrasonics Symposium and has contributed with valuable knowledge about strain estimation and echocardiography in general.

My supervisor, Dr. Gabriel Hanssen Kiss, probably deserves a full page of his own. Through the duration of the work with this thesis, he has always made time, and provided helpful comments and suggestions. Countless emails have been answered swiftly with a well thought out reply, and for this, he has my most sincere gratitude.

For her continuing support, and help with wording and proof-reading, I would like to thank my girlfriend, partner, and fellow student Camilla Sterud. For moral support and help with practical issues, I also owe a big thank you to my parents, Hanne Haukom and Svend Andersen.

Torjus Haukom
Tønsberg, June 17, 2019

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

Cardiac surgery can be a complex and comprehensive intervention and is not without risk. Procedures such as bypass surgery and valve replacements have been shown to negatively impact cardiac function, often causing decreased myocardial contractility, and in some cases, atrial fibrillation and myocardial infarction[1, 2, 3]. Consequently, patients undergoing such procedures have their hearts monitored through the perioperative phase. Currently, this monitoring is done by evaluating vital signs, such as blood pressure, heart rate, blood oxygen level, and respiratory rate, as well as through manual echocardiographic assessment of the cardiac function by an anesthesiologist[4, 5, 6].

The term echocardiography refers to ultrasonic imaging of the heart and is one of the primary applications of ultrasound. It is an inexpensive and simple imaging technique compared to other methods, such as computed tomography or magnetic resonance (MR) imaging. Relative to its price and ease of use, the images are of high quality, and acquisitions can be made in two or three dimensions in addition to time[7, 8]. Echocardiography has long been an indispensable tool for assessing cardiac health and function in the diagnostic setting, and with advancements in transesophageal echocardiography (TEE), ultrasonic imaging is increasingly used to monitor patients cardiac function throughout the perioperative period[4, 6].

Echocardiographic assessment of cardiac function in the perioperative period is often performed by simple visual inspection. In the last decade, however, efforts have been made to standardize quantitative indicators of cardiac function, as they rely less on the individual echocardiographer's experience and preferences[4, 9]. Myocardial strain is one such quantitative indicator, measuring global or regional myocardial deformation, and has been shown to have prognostic value in patients undergoing cardiac surgery[10]. Reference values for strain in healthy patients have also been proposed, making it an ideal metric for objective assessment of cardiac function[11].

Strain measurements depend on tracking the motion of the myocardium through the cardiac cycle. Currently, this is done either through tissue Doppler imaging (TDI) or speckle tracking methods. TDI is a one-dimensional velocity measurement along the ultrasound beam direction and is thus dependent on the beam angle being relatively parallel to the myocardium[12]. As the velocities measured by TDI samples are relative to the probe, interpolation is required to estimate a velocity gradient. Once this estimate is made, the strain rate may be calculated by integrating the gradient along a myocardial segment. To get an estimate of the regional strain the strain rate is integrated over time[13].

Speckle tracking approaches are based on comparing the similarity of local speckle pat-

terns between consecutive frames to track the distances between material points, or landmarks, on the myocardium. These methods are not angle dependent, and strain measured with speckle tracking shows a higher correlation than TDI when compared to higher quality MR images[14, 15]. These methods do, however, suffer from poor temporal resolution, especially in 3D, as speckle similarity must be optimized for each consecutive frame in the sample[16]. Furthermore, head-to-head comparisons have shown significant inter-vendor variations[17]. These methods also currently require trained personnel to manually annotate landmarks or ventricle contours in the images for the tracking to work, making strain estimation a time- and resource consuming task, unsuited for the operating theater.

A third approach to the strain estimation problem is elastic or deformable image registration. These methods estimate displacements between two consecutive frames in an effort to align them. Using these displacement vectors the distance between landmarks on the myocardium can be tracked to produce a strain estimate, as with speckle tracking. These methods are less used in practice, although performance has been shown to be similar to speckle tracking when compared to a gold standard reference measured by sonomicrometry[18].

Efforts have been made to provide fully automatic strain measurements. Knackstedt et al. demonstrated the reliability of the AutoLV algorithm (TomTec-Arena 1.2, TomTec Imaging Systems, Unterschleissheim, Germany), which detects the contour of the myocardium, to measure global longitudinal strain (GLS) in transthoracic images. Their experiments showed a high correlation with manual methods[19]. A more recent approach aimed at on-site analysis, proposed by Østvik et al., uses supervised deep learning to classify the view, crop the samples to the myocardium, and track its motion through the cardiac cycle and estimate GLS. Interestingly, the motion estimation was performed using a flownet type neural network trained on a synthetic dataset, while the view classification and cropping networks were trained on manually annotated data. Their results show promise but are still preliminary[20, 21].

Machine learning methods, and in particular those based on deep learning, have revolutionalized several fields of research in recent years, including speech recognition[22], natural language processing[23], and computer vision tasks such as object detection and classification in images[24, 25]. These advancements have not escaped the medical research community, and particularly computer vision models based on convolutional neural networks (CNNs) have been applied to a variety of medical imaging problems. Recent efforts have shown great promise, matching or beating trained physicians in tasks such as distinguishing melanoma from moles[26], detecting breast lesions in mammograms[27], and polyp detection in colonoscopy images[28, 29]. In the field of medical image registration, images have been successfully aligned by using CNNs to estimate the displacements between pairs of images[30, 31, 32]. In echocardiography, deep learning has been successfully applied to tasks such as view classification[33, 34], chamber segmentation[34, 35], and automatic measurement of global cardiac function indicators such as ejection fraction, mitral annular plane systolic excursion, and, as mentioned, global longitudinal strain in transthoracic images[20, 34, 36].

1.2 Aim and method

To ensure patient safety during cardiac interventions, echocardiographic evaluation of cardiac function is becoming a standard procedure during the perioperative period. As this evaluation is commonly done by visual inspection, it remains highly qualitative. Quantitative assessment, while possible, is time-consuming, and there is currently a push to automate such measurements. This thesis aims to contribute towards the full automation of perioperative echocardiographic monitoring through investigating the feasibility of fast, automatic longitudinal strain estimation in the basal segments from unselected 4-chamber, 2-chamber, and long-axis TEE images using unsupervised deep learning methods.

Inspired by its success in image registration, the Deep Learning Framework for Unsupervised Affine and Deformable Image Registration introduced by de Vos et al.[32] was adapted to estimate motion vector fields describing the movement of each pixel from one frame to the next in the TEE recordings, similar to estimating optical flow. Following these vectors through the recording for two points on a basal segment, the distance between them can be used to estimate longitudinal strain.

1.3 Outline of thesis

In this first chapter, the motivation behind automatic strain estimation was covered, along with a summary of previous efforts applying deep learning to medical imaging problems, including echocardiography. The theoretical background needed to follow the rest of the thesis is presented in Chapter 2, covering the basics of human cardiac physiology, ultrasound imaging, and deep learning. Chapter 3 covers the specific methods applied and the dataset used, including preprocessing steps and model architecture. Chapter 4 presents the results of the strain estimation. These results are discussed in Chapter 5, and a concluding summary is found in Chapter 6.

In addition to this thesis, an abstract describing this work was accepted to the 2019 IEEE International Ultrasonics Symposium in Glasgow, Scotland. A copy of the abstract is included in Appendix A. Appendix B covers B-spline interpolation with transposed convolutions. Appendix C contains the results of visually inspecting the basal landmark tracking. There is also a digital appendix accompanying this thesis that contains some examples of landmark tracking on the test set.

2 Theory

2.1 The human heart

The human heart is a muscular organ in the thoracic cavity, which is responsible for the distribution of blood in the body. As shown in Figure 2.1, its structure is made up of a left and right section, working together as a parallel pump. The sections are separated by the septum, and each section contains two chambers. The upper and lower chamber in each section are referred to as the *atrium* and *ventricle*, respectively, and are separated by valves to prevent the blood from flowing back[37].

The heart receives deoxygenated blood from the body via the venae cavae into the right atrium. The blood then flows through the tricuspid valve into the right ventricle. From there, the blood is pumped into the lungs, via the pulmonary arteries, where it is replenished with oxygen. From the lungs, the replenished blood flows through the pulmonary veins into the left atrium. It then flows through the mitral valve into the left ventricle. From there, the blood is pumped out into the aorta and on to all parts of the body[38].

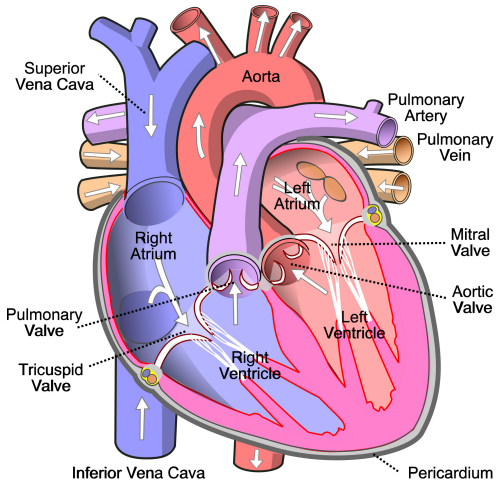


Figure 2.1: Illustration of the cardiac structure. White arrows show the direction of blood flow. Illustration by Wikimedia user Wapcaplet[1], reproduced under the CC BY-SA 3.0 license [CC BY-SA 3.0 (`http://creativecommons.org/licenses/by-sa/3.0/`)]

[1] `https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_(cropped).svg`

The cardiac cycle consists of two phases referred to as *systole* and *diastole* and correspond roughly to contraction and relaxation of the ventricles, respectively. The systolic phase begins when the pressure in the left ventricle surpasses that of the left atrium and the mitral valve closes. This point in the cycle is referred to as end-diastole (ED). With the atrioventricular valves closed, the ventricles rapidly contract to push the blood into the arteries while the atriums relax to allow new blood to arrive from the veins. When the ventricle pressure is lower than that in the arteries, the arterial valves close. The closing of the aortic valve marks the end of the systolic phase and the beginning of the diastolic phase, referred to as end-systole (ES). Now the atriums contract while the ventricles relax and expand. This causes a pressure gradient that draws the blood into the ventricles until the pressure gradient is reversed, and the atrioventricular valves close again[8]. In Figure 2.2, a Wiggers diagram is shown. It shows how the atrial, ventricular, and aortic pressures evolve through the cardiac cycle with mitral and aortic valve opening and closure annotated.

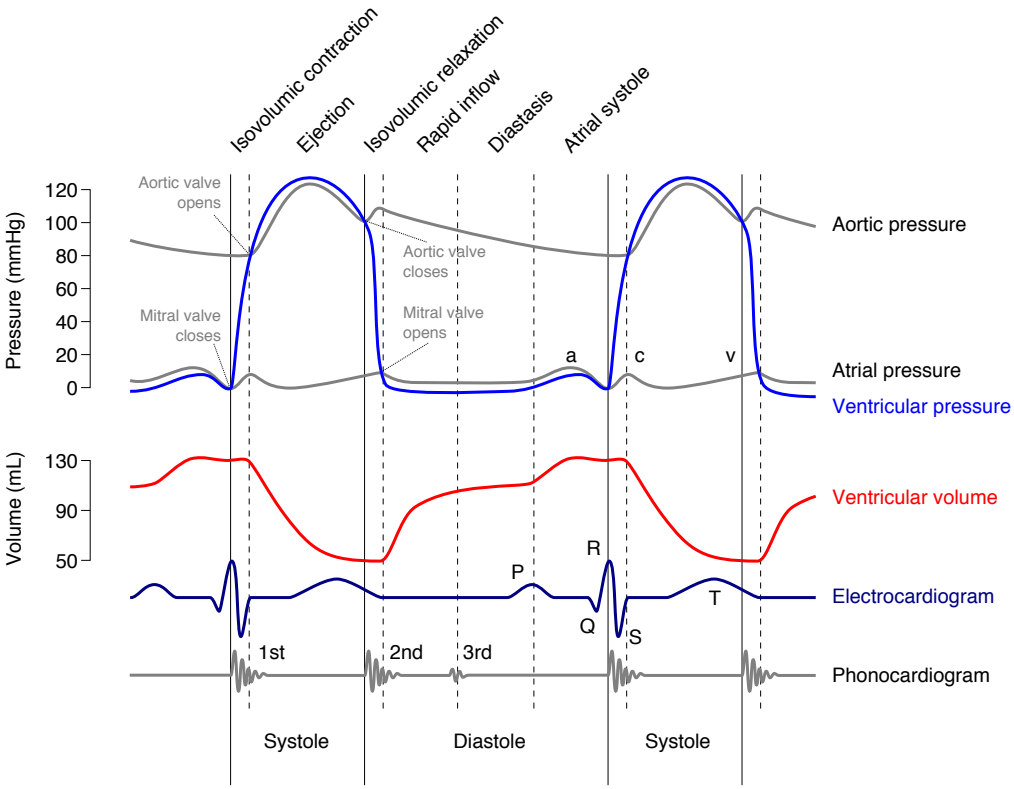


Figure 2.2: Wiggers diagram illustrating the relation between different pressures, volumes, and measurements through the cardiac cycle with important events annotated. Illustration by Wikimedia user DanielChangMD revised original work of DestinyQx; Redrawn as SVG by xavax[2]. Reproduced under the CC BY-SA 2.5 license [CC BY-SA 2.5 (`https://creativecommons.org/licenses/by-sa/2.5`)]

2.2 Ultrasound imaging

Ultrasound imaging is a widely used and relatively inexpensive diagnostic tool, perhaps most known for imaging fetuses in the uterus[8]. It is based on echos produced by the reflection of ultrasonic waves in tissue and can produce many different types of images in both two and three dimensions in addition to time. Covered here are 2D *B-mode* images, where *B* is for brightness, which are the most common[7].

To form a B-mode image several ultrasonic pulses are transmitted one by one from a probe at different angles, scanning a plane intersecting the object to be imaged, and the intensity of the echos that come back are recorded. Using the speed of sound, which is approximately $c = 1450\,\mathrm{m\,s^{-1}}$ in human tissue, the distance d to the object which produced the echo at time t can be calculated as $d = \frac{ct}{2}$. This technique is known as echo ranging and is illustrated in Figure 2.3. The intensities of the echos from each pulse are then plotted as a function of their distance to the probe, forming *B-mode lines*. As the acoustical properties are different for different types of tissue, different types of tissue may be distinguished. These B-mode lines are visualized in a polar plot forming the final image, as shown in Figure 2.4[7].

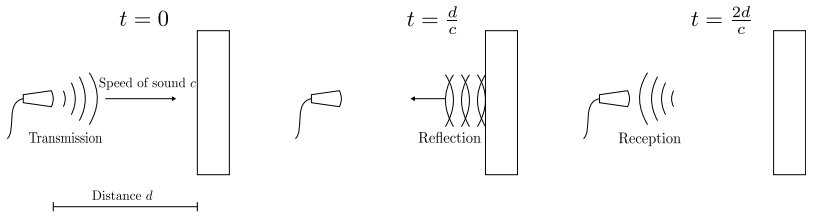


Figure 2.3: Echo ranging. The distance to an object is given by the time of arrival of the echo and the speed of sound.

[2] https://commons.wikimedia.org/wiki/File:Wiggers_Diagram.svg

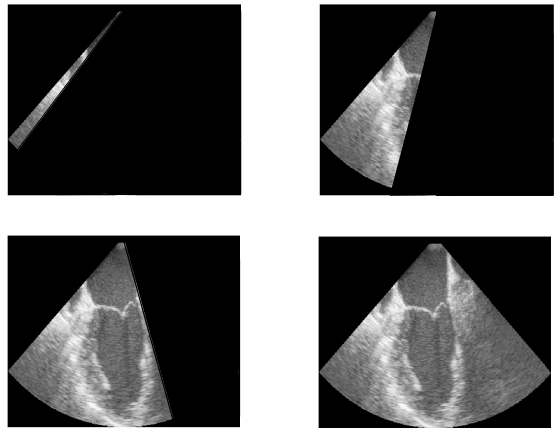


Figure 2.4: Scanning procedure to generate B-mode images. Recording the times of echos from pulses transmitted at different angles, we get samples in a polar coordinate system which form the image.

Two main factors thus determine the frame rate of an ultrasound recording. The first is the depth of the scan, as a deeper scan means that the echos take longer to reach the probe. The second is the width of the scan. Wider scans require more B-mode lines at more angles, making each scan slower. The spatial resolution of the images also depends on two main factors, the first being the frequency of the waves. At higher frequencies, echos with shorter time intervals between them may be distinguished. The second is the number of discrete angles used in the scanning procedure. Also, the distance between the B-mode lines increases with depth. Thus, the spatial resolution decreases further away from the probe.

2.2.1 Echocardiography

Echocardiography refers to ultrasonic imaging of the heart and is done in one of two ways. They differ in the placement of the probe and the invasiveness of the procedure, as illustrated in Figure 2.5. Transthoracic echocardiography (TTE) is performed by placing the probe on the exterior of the patient's chest, aiming the beam between the ribs. This method is quick to set up and is non-invasive, but suffers from noise from the lungs, and the ribs limit the probe placement. The probe must also be held still by the examiner throughout the exam.

The alternative is transesophageal echocardiography (TEE) where the probe is placed in the patient's esophagus. In humans, the heart rests upon the esophagus, giving TEE several advantages. The position of the heart relative to the probe is more or less constant, the probe is kept still by the esophageal wall, and there is less noise from the lungs. The shorter distance from the probe to the structures also allow for higher frequencies to be used, yielding a higher spatial and temporal resolution. These advantages come at a cost. TEE is a more demanding procedure and may be very uncomfortable for the patient, which

may require sedation. It is therefore mostly used for pre and post-operative assessment of cardiac health[4].

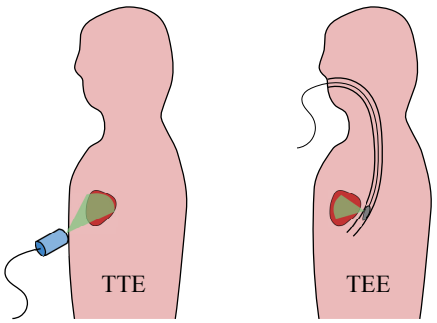


Figure 2.5: Probe placement for TTE and TEE. When performing TTE, the probe is placed on the outside of the patient's chest. When performing TEE, the probe is placed in the patient's esophagus.

2.3 Cardiac function

Assessing the cardiac function, and in particular left ventricle (LV) systolic function is one of the most common uses of TEE[4]. A distinction is made between global and regional function. Global function serves as an indicator of overall cardiac health and is usually assessed by measuring a difference in some size or distance between ED and ES. One common metric is the ejection fraction (EF) which is based on estimates of the end-diastolic and end-systolic LV volumes EDV and ESV and is given by $\mathrm{EF} = \frac{\mathrm{EDV - ESV}}{\mathrm{EDV}}$[9].

Regional cardiac function is assessed by observing the deformation or shortening of segments in the myocardium and has been shown to have prognostic value in cardiac surgery patients[10]. A standardized 17-segment model, as recommended by the American Heart Association, is often used to define the regions and is visualized in Figure 2.6[9]. The assessment is commonly done by visual inspection by a physician, which makes both inter- and intra-observer variability high. Thus, standardized quantitative measurements are desirable, and in recent years, the adoption of *strain imaging* has increased, and efforts to standardize these methods have been made[11, 39].

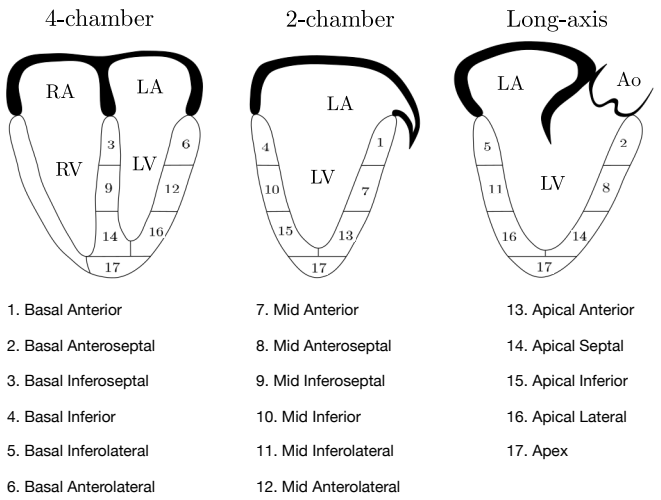


Figure 2.6: The 17-segment model of the left ventricle as recommended by the American Heart Association illustrated for three different views.

The term *strain* in cardiology refers to local shortening, thickening or lengthening of the myocardium, and is used as a measure of local LV function. This deformation is described by a tensor with six components: shortening along the x, y, and z-axes along with shear in the three planes between them. When estimating strain from images, a simplified strain metric is used, measured by tracking the distance L between two material points on the myocardium relative to the initial length L_0, typically at ED. This metric is referred to as Lagrangian strain and is defined in Equation (2.1)[39, 40].

$$\epsilon(t) = \frac{L(t) - L_0}{L_0} \tag{2.1}$$

Three principal types of strain can be measured by 2D echocardiography: longitudinal, radial, and circumferential[40]. In Figure 2.7, the measurements required for end-systolic radial and longitudinal strain in the basal segments are illustrated. Using ED as the initial time, the end-systolic strain is calculated by setting $t =$ ES in Equation (2.1) and quantifies the contractile ability of the myocardial region.

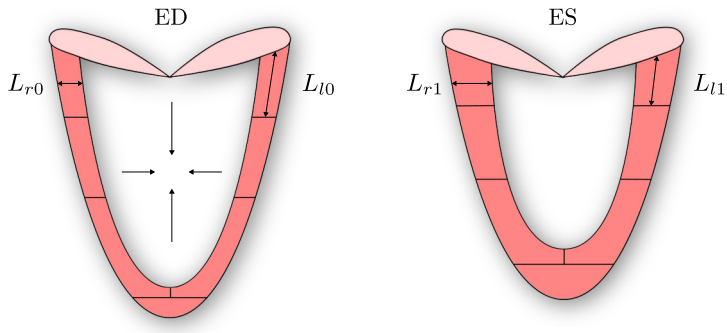


Figure 2.7: Measurements involved in end-systolic radial and longitudinal strain calculation on the basal segments of the left ventricle.

2.4 Deep learning

Machine learning (ML) is a field of research concerned with algorithms and statistical models that enable a computer system to perform a specific task without being explicitly programmed. It is a highly interdisciplinary field, relying on methods from, among others, optimization theory, statistics, and computer science. ML methods are applied to a variety of tasks, including recommender systems, recognition of speech and images, and control systems[41, 42]. ML methods are commonly divided into three categories: supervised learning, unsupervised learning, and reinforcement learning. The differences between them are illustrated in the flow charts in Figure 2.8.

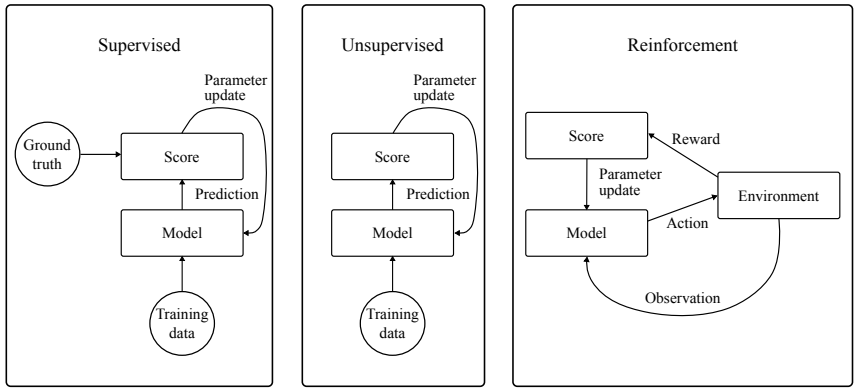


Figure 2.8: Flow charts illustrating the differences between supervised, unsupervised, and reinforcement learning.

Supervised learning can be thought of as learning from examples. Assume that training data x, which is a subset of all possible model inputs $\mathcal{X}$, is available. In the supervised setting, the desired outputs of the model y, called *ground truth*, must also be available.

We assume that there exists some unknown mapping $y = f(x)$, and the model should approximate this mapping as $\hat{y} = \hat{f}(x)$. During training, the model is fed examples from the training data and makes a prediction $\hat{y}$. The prediction is then compared to the ground truth y. This comparison results in some score, which is used to update the model parameters to get the prediction closer to the ground truth[43].

Unsupervised learning is a more flexible framework and is primarily used for exploring possible groupings in data in an approach known as clustering. However, it may refer to any learning algorithm that does not rely on a ground truth. The score may only be based on the training data, and in the case of clustering, a similarity metric between the examples is used[43].

Reinforcement learning can be thought of as learning by doing and is heavily used in artificial intelligence. These methods do not need data in the same way that supervised and unsupervised methods do. Instead, the model explores an environment on which it performs some action, based on observing it. The result of this action is some reward or penalty, indicating how good the action was. Then, a new observation is made, and the cycle starts over. During training, the model parameters are optimized to choose the actions that maximize the reward[43].

Deep learning (DL) is a subfield of ML. The models used in DL are called artificial neural networks (ANNs), so named because they are inspired by the way the human brain performs computations. The term *deep* comes from the fact that ANNs are complex estimators built from layers of simpler estimators, and the number of layers is referred to as the *depth* of the network. Each of these layers learns a more abstract representation of the input features until finally combining them into a prediction. The popularity of DL methods has exploded in the previous decade, as the large datasets and computing power needed has become more widely available[41, 44], and DL methods have revolutionalized several fields of research, including computer vision[24, 25], speech recognition[22], and natural language processing[23].

2.4.1 Feed-forward neural networks

A feed-forward neural network or multilayer perceptron (MLP) is the simplest form of ANN[41]. Still, these models have been shown to approximate any continuous function[45, 46], and have been successfully applied to a variety of tasks, including playing backgammon, noise filtering of ECG signals, and driving cars[47].

The fundamental building blocks of feed-forward networks are called hidden units or *neurons*, and several neurons are combined to form a layer. Many such layers may be stacked to form a deeper network, but every deep feed-forward network consists of one input layer, one or more hidden layers, and one output layer. Each neuron in a layer connects to all of the neurons in the next layer. For this reason, such layers are often referred to as *fully connected* or *dense* layers. Inside a neuron, the contribution of each input is weighted by a weight w, and added to a potential bias term b. Both of which are optimized during training. To make the neuron non-linear, an activation function a is applied, as shown in Equation (2.2). Commonly, sigmoid, tanh, or rectified linear units (ReLUs) are used as activations[41].

$$h_i = a(\vec{w}_i^T \vec{x} + b_i) \tag{2.2}$$

This form of connected computations forms a directed acyclic computational graph, as illustrated in Figure 2.9. It is possible to include computations that make the graph cyclic. Such graphs are called recurrent neural networks and are beyond the scope of this thesis.

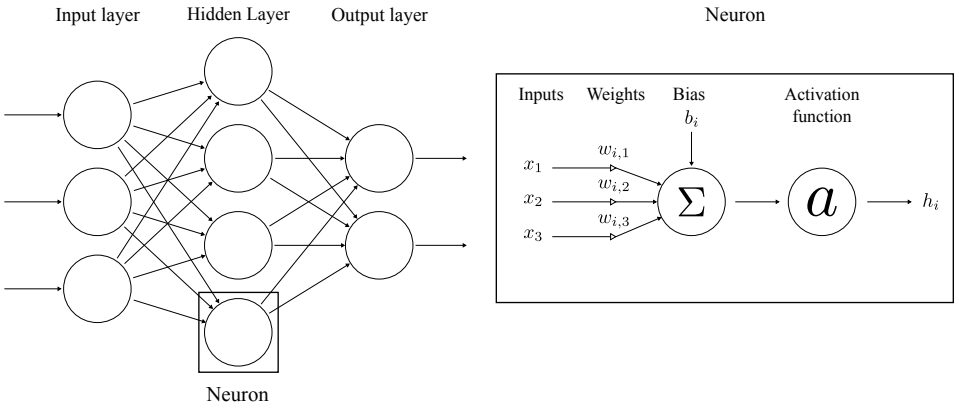


Figure 2.9: A feed-forward neural network, or MLP, with three input features, four hidden neurons and two outputs, along with an illustration of the calculation performed in a neuron.

2.4.2 Convolutional neural networks

Convolutional neural networks (CNNs), named after the convolution operation, are a different kind of ANN and are the first choice for computer vision applications[48]. In digital image processing, the 2D convolution operation is used for linear spatial filtering of images with fixed 2D filters. Such filters, commonly referred to as *kernels*, may be used for smoothing, noise reduction, or feature extraction, such as edge and corner detection, depending on the filter coefficients[49].

When convolving, each pixel in the filtered image is given by a linear combination of a neighborhood of pixels in the original image, weighted by the kernel. If the image I is viewed as an $M \times N$ matrix and the kernel K is an $L \times L$ matrix with $L < M$ and $L < N$, then a pixel in the filtered image H is given by Equation (2.3)[3][49].

$$h_{i,j} = \sum_{m=0}^{L-1} \sum_{n=0}^{L-1} k_{m,n} \cdot i_{i+m,j+n} \tag{2.3}$$

Performing this calculation for all the pixels in the filtered image can be viewed as sliding the kernel over the original image, as shown in Figure 2.10. The resulting image has

[3]Equation (2.3) is actually the correlation, which is how convolution is commonly implemented in practice. They are equivalent if the kernel is mirrored before filtering.

dimensions $M-(L-1)\times N-(L-1)$. If identical dimensions are desired, the original image may be zero-padded.

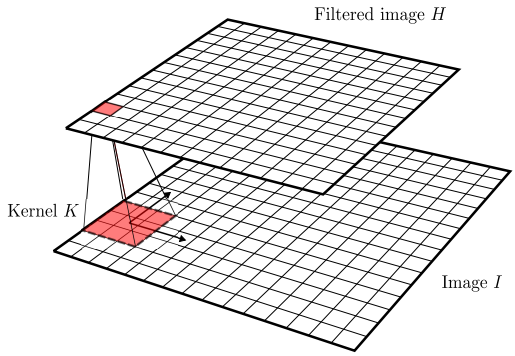


Figure 2.10: Filtering of an image I with a 3×3 kernel K through convolution.

A CNN layer consists of one or more kernels that filter incoming images. The entries in the kernel matrices are trained to produce more abstract features from the raw pixels. The resulting images for each kernel are stacked to produce a 3D feature map of dimension $M-(L-1)\times N-(L-1)\times D$, where D is the number of kernels in the layer. After the convolution operation, non-linearity may be introduced by activating the pixels in the filtered image. Commonly, ReLUs are used[50].

When stacking several convolutional layers, the *receptive field* of the resulting pixels increases. The result is that more abstract feature maps are produced in each successive layer, with the first layers usually ending up as edge and corner detectors. The later layers combine these into higher-level features such as eyes, windows, or cars, depending on the application[50].

Between every few convolutional layers, it is customary to add a downsampling operation known as *pooling*. A pooling layer divides an image into non-overlapping rectangles and produces one pixel from each rectangle in the resulting image. This reduces the memory footprint and computational load of the network, and also adds a regularizing effect. Commonly average or max pooling is used, which averages or picks out the maximum value of the neighborhood, respectively[41, 50]. An illustration of max pooling is shown in Figure 2.11.

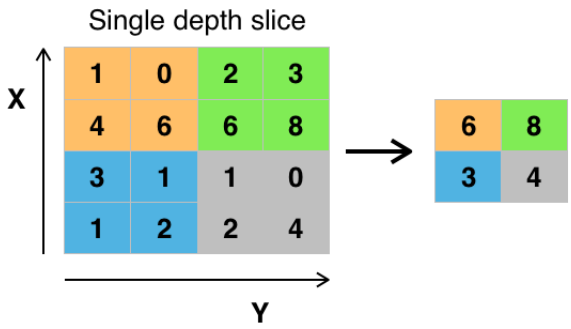


Figure 2.11: 2×2 max pooling with stride 2. The resulting image is made up of the maximum from each 2×2 rectangle. Illustration by Wikimedia user Aphex34[4]. Reproduced under the CC BY-SA 4.0 license [CC BY-SA 4.0 (`https://creativecommons.org/licenses/by-sa/4.0`)]

2.4.3 Training neural networks

Training a neural network is a particular kind of optimization problem that is solved with gradient-based methods. In general, we minimize a cost function $J(\theta)$ to obtain optimal network parameters θ. This cost function is the expected value over the training set $\mathbf{x}$ of some loss function $L(\hat{f}(\mathbf{x}, \theta))$: $J(\theta) = \mathbb{E}_{\mathbf{x}}\left[L(\hat{f}(\mathbf{x}, \theta))\right]$, where $\hat{f}(\mathbf{x}, \theta)$ is the network output. The expectation will, in practice, be estimated by an average. To efficiently train the network the loss function should reflect the goal of the training, and in the supervised case, this means some comparison with the ground truth[41].

The parameters are updated stepwise in an approach known as gradient descent, with a *learning rate* α, as shown in Equation (2.4).

$$\theta_i = \theta_{i-1} - \alpha \nabla J(\theta_{i-1}) \tag{2.4}$$

Using the chain rule, it becomes clear that for the loss of a network $\hat{f}$ with N layers, the partial derivatives with respect to a parameter can be decomposed as

$$\frac{\partial}{\partial \theta} L(\theta) = \frac{\partial L}{\partial \hat{f}(\mathbf{x}, \theta)} \frac{\partial \hat{f}(\mathbf{x}, \theta)}{\partial \theta}$$

$$\frac{\partial \hat{f}(\mathbf{x}, \theta)}{\partial \theta} = \frac{\partial \hat{f}^{(N)}(\hat{f}^{(N-1)}(\cdots \hat{f}^{(2)}(\hat{f}^{(1)}(\mathbf{x}))))}{\partial \hat{f}^{(N-1)}(\hat{f}^{(N-2)}(\cdots \hat{f}^{(2)}(\hat{f}^{(1)}(\mathbf{x}))))} \cdots \frac{\partial \hat{f}^{(2)}(\hat{f}^{(1)}(\mathbf{x}))}{\partial \hat{f}^{(1)}(\mathbf{x})} \frac{\partial \hat{f}^{(1)}(\mathbf{x})}{\partial \theta}$$

where $\hat{f}^{(i)}$ is the output of the ith layer from the input.

Thus, the gradients can be found by simply differentiating each layer separately and multiplying. This forms the basis for the backpropagation algorithm. The gradient of the very

[4] `https://commons.wikimedia.org/wiki/File:Max_pooling.png`

last layer is computed first, with respect to the second to last layer's output. Then, the second to last layer's gradient is computed with respect to the third to last and so on, propagating back to the input layer. When all gradients are calculated, the parameters can be updated according to Equation 2.4.

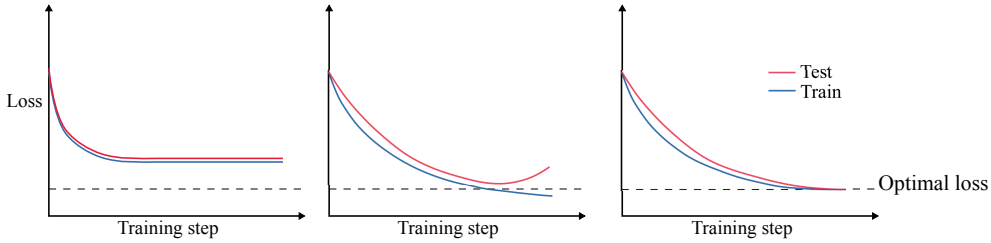
Computing the gradients is computationally intensive. Also, neural networks may have millions of parameters, thus requiring large amounts of data to optimize them. This makes the training a time-consuming task, often requiring powerful computers with graphics processing units (GPUs) for it to be feasible. The upside is that once the network is trained, predictions can be made in a fraction of the time used for backpropagation.

In practice, due to memory limitations and the large datasets required, the training set is almost always divided into non-overlapping random batches. The model is then optimized with one update per batch according to Equation (2.4). This is known as mini-batch gradient descent. Because of the computer architecture of GPUs, performance gains are made if the batch size is chosen to be a power of two, with the exact choice depending on the memory available. The batch size also impacts how well the model is able to learn, and the speed at which it converges; a smaller batch size means that more updates, with higher variance in the cost function estimate, are made for each run through the training set[41].

Several adjustments to the straight-forward gradient descent update in Equation (2.4) have been proposed to improve performance in particular use-cases. One example is the Adam optimizer, where the gradients are smoothed by an exponential moving average (EMA) to reduce variance in the updates. These smoothed gradients are then scaled by the square root of EMA smoothed squared gradients, effectively adapting the learning rate to the current region of the cost function[51]. Adam has become hugely popular since its release in 2014 and has been shown to be a good first choice for optimizing neural networks.

When evaluating neural network models, two metrics determine its ability to perform: the estimated loss on the training set, and the estimated loss on a separate, unseen test set. Together, the training and test set losses can be used to diagnose two fundamental issues with any ML model if monitored throughout the training process: underfitting and overfitting. For a model to perform well, both training and test set loss should be as low as possible, and the distance between the two should be small. If the model is unable to reach an optimal loss value on both the training and test set, it is said to be underfitting. Underfitting most often occurs when training a model that is too simple for the task. Overfitting happens when the model becomes too specialized to the training examples and fails to generalize to unseen data, causing increased test set loss. This can happen if the model is too complex, or training data is scarce[41].

Figure 2.12 shows examples of learning curves for models that are underfitting, overfitting and fitting optimally. Figure 2.12a shows learning curves for an underfitting model. The training and test set losses do not reach the optimum, as they converge at a higher value. Figure 2.12b shows a model where overfitting occurs. As the training set loss decreases beyond the optimal value, the test set loss is increasing, indicating a loss of generality. Lastly, Figure 2.12c shows an optimal fit, where both training and test set losses converge to the optimal value.



(a) Underfitting **(b)** Overfitting **(c)** Optimal fit

Figure 2.12: Underfitting, overfitting, and optimal fit. Neither the training nor the test set loss reaches the optimal value in an underfitted model. Overfitting occurs when further training yields worse performance on the test set. A good fit is found when both losses are near the optimal value.

3 Materials and Method

To estimate basal longitudinal strain from transesophageal echocardiographic (TEE) images, two things are needed: the location of at least two landmarks on each basal segment and a method of tracking these points. Machine learning methods have proven successful in segmenting of the left ventricle[35] and tracking the mitral annulus[34], the latter also being the subject of an ongoing thesis project at our department. For these reasons, the location of the mitral annulus was assumed to be known and was used as one of the basal landmarks. Thus, a means of detecting the second landmark in the basal segment was needed along with estimates of the trajectories of these points through the cardiac cycle.

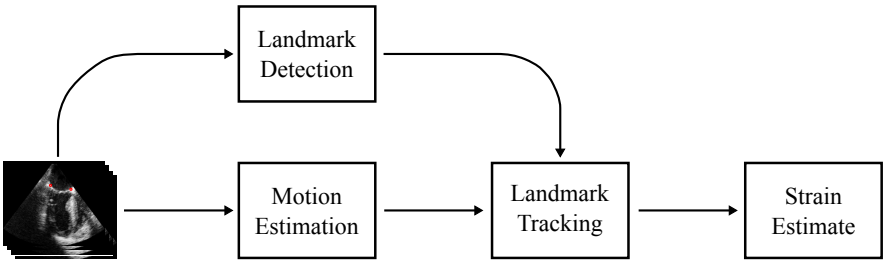


Figure 3.1: Proposed pipeline for estimating regional strain in the basal segments using TEE images with annotated mitral annulus.

In Figure 3.1, the proposed pipeline for automatic strain estimation is illustrated. A landmark detection algorithm was used to locate suitable points on the basal segments in the first frame of the sample using the mitral annulus as a reference. Tracking of these landmarks was done using an estimate of the pixel motion between each frame produced by a convolutional neural network (CNN). Using the distance between the landmarks in the end-diastolic (ED) and end-systolic (ES) frames a strain estimate could be calculated using Equation (2.1).

3.1 Data

For training and evaluation of the strain estimation pipeline, TEE B-mode images were obtained by cardiologists with echocardiographic expertise from 94 patients using GE Vivid E95 and E9 systems with a 6VT-D probe (GE Vingmed Ultrasound, Horten, Norway). 89 of these patients were examined in the clinic for diagnostic purposes, and five patients were examined before and after undergoing cardiac surgery (coronary artery bypass grafting in four cases, mitral valve clipping in one case). At least three complete cardiac cycles

were captured in three views: 4-chamber, 2-chamber, and long-axis. The pixel brightness was recorded in the range $[0, 255]$. The frame rate of the recordings was in the range of 30 to 60 frames per second, and the resolution ranged from 255×180 to 537×380 depending on the width and depth of the scan. No selection of the images was performed, and all samples were anonymized before analysis. To facilitate processing, the images were converted from the proprietary DICOM format to 2D images by applying a polar-Cartesian transform on the raw B-mode lines. During this conversion, the images were flipped from left to right by coincidence, which is the reason why some figures in this thesis are mirrored.

To assess the performance of the strain estimation pipeline, reference values for basal longitudinal strain were provided by a trained physician. The reference values were acquired by manually annotating the images and tracking the myocardium using the EchoPAC (GE Vingmed Ultrasound, Horten, Norway) speckle tracking software.

3.1.1 Preparation of datasets

The data was divided into three separate datasets. A training set consisting of samples from 57 patients chosen randomly was used for training the CNN. For hyperparameter tuning and to monitor the model performance during training a validation set consisting of the samples from 14 patients was used. In both of these datasets, the frames were zero-padded to match the resolution of the sample with the highest resolution within each set to enable training on batches. All frames of the samples in the training and validation sets were organized into pairs of consecutive frames. During training, these pairs were drawn randomly to construct batches.

The samples from the remaining 23 patients were used for testing. As the goal of the method was to track points through the cardiac cycle, the frames of these samples were kept in order. The test set samples were divided into smaller samples showing a single cardiac cycle from ED to ED, with ES annotated. These points in time were assumed to be known, as ED and ES can be found reliably from electrocardiogram signals, and previous efforts have shown that detection of these frames is possible from the raw images[52, 53]. As the location of the mitral annulus was also assumed to be known, these points were manually annotated in the initial frame (ED) of each of the divided samples.

All datasets were preprocessed by applying a proprietary contrast enhancement algorithm, courtesy of GE (GE Vingmed Ultrasound, Horten, Norway), and all pixels were scaled to $[0, 1]$.

3.2 Basal landmark detection

For a point to be a suitable landmark for strain estimation, it should have some properties that facilitate tracking. Firstly, it should be in a relatively bright neighborhood, as such areas are less likely to be obfuscated by random noise. Secondly, the landmark should be at some clearly defined feature, such as the endocardium edge, as these are easier to track.

Motivated by these desired properties, a series of filters were applied to the initial frame

of the samples. First, the frame was filtered by a 7×7 median filter to suppress noise. Then, the filtered frame was thresholded to remove dark areas and further reduce noise. To emphasize bright neighborhoods, rather than separate pixels, the resulting image was filtered with a 3×3 Gaussian filter with $\sigma = 3$. Lastly, to highlight edges, a Sobel filter was applied. The filter sizes and parameters were chosen empirically from a subset of the test set. Using the mitral annulus as a reference, a triangular sector was defined perpendicular to the mitral annular plane. From this sector, the brightest pixel in the final filtered frame with a distance of 3 to $5\,\mathrm{cm}$ from the mitral annulus was picked as the landmark. This procedure is illustrated in Figure 3.2.

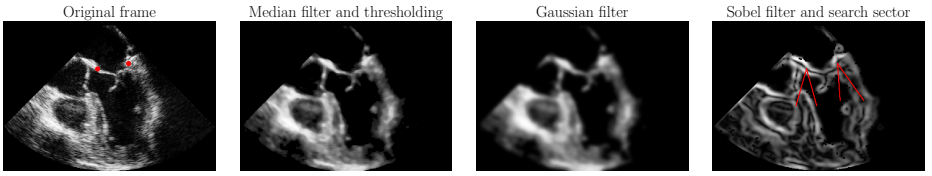


Figure 3.2: Detection of landmark points on the basal segments. The initial frame with annotated mitral annulus is filtered with a 7×7 median filter, thresholded, and blurred with a Gaussian filter before edges are highlighted with a Sobel filter. This processed frame is searched for the brightest pixel in the triangular sectors below the mitral annulus.

3.3 Motion estimation and landmark tracking

To track the landmarks on the basal segment, the approach used for deformable image registration by de Vos et al.[31, 32] was adapted to this application. At the heart of the method is a CNN that takes two consecutive frames I_i and I_{i+1} from a sample, and outputs a low-resolution displacement field $\vec{D}$ describing the motion between the two frames in x and y directions. This displacement field is then upsampled using cubic B-splines to make a dense displacement field $\vec{D}^d$ with one motion vector per pixel such that $I_i(x, y) \approx I_{i+1}(x + D^d_x(x, y), y + D^d_y(x, y))$. Thus, to perform point tracking, one can follow the displacement vector from frame to frame.

The CNN architecture, shown in Figure 3.3, consists of a concatenation layer, merging the two consecutive frames into one tensor, followed by alternating convolutional and average pooling layers. The number of pooling layers determines the resolution of the displacement field and thus, the number of B-spline control points. These low-resolution feature maps are passed through two more convolutional layers before finally two 1×1 convolutions are performed yielding the estimated displacements.

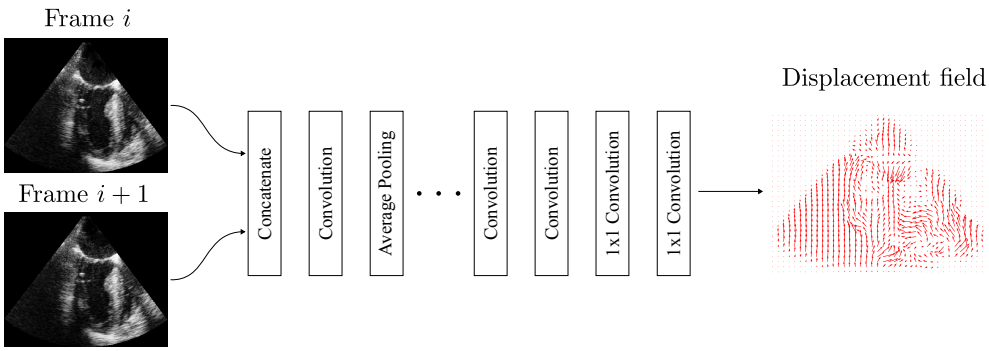


Figure 3.3: Architecture of the convolutional neural network used. As input, it expects two consecutive frames from an ultrasound sample. The output is a low-resolution displacement field to be interpolated. Several alternating convolutional and average pooling layers may be added before the final convolutions to achieve the desired spacing between the B-spline control points.

Figure 3.4 shows the desired result of the motion estimation task. A coordinate grid is warped by the estimated motion vectors between the ED and ES frames of a sample, resulting in a deformed grid illustrating the movement between these points in time. By taking the distance between the landmarks one these grids a strain estimate can be calculated using Equation 2.1.

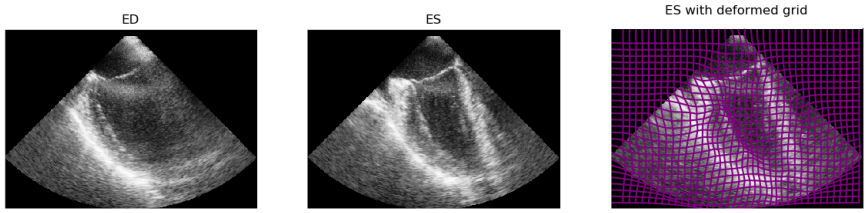


Figure 3.4: Frames from a sample at ED and ES, including a deformed grid illustrating the movement between these frames.

In Figure 3.5, a flow chart visualization of the training procedure for the motion estimator is shown. Following this procedure, consecutive frame pairs are fed to the CNN, which produce low-resolution displacement fields. These fields are interpolated using B-splines and used to warp the second frame. Then the warped frame is compared to the first frame using normalized cross-correlation. This approach has two major advantages. The neural network consists of only convolutional layers, meaning that the trained network can make estimates on frames of any resolution, and the training is done unsupervised, eliminating the need for costly ground truth annotation. In essence, this means that the method can be repurposed for any type of image, medical or other, with minimal adjustments.

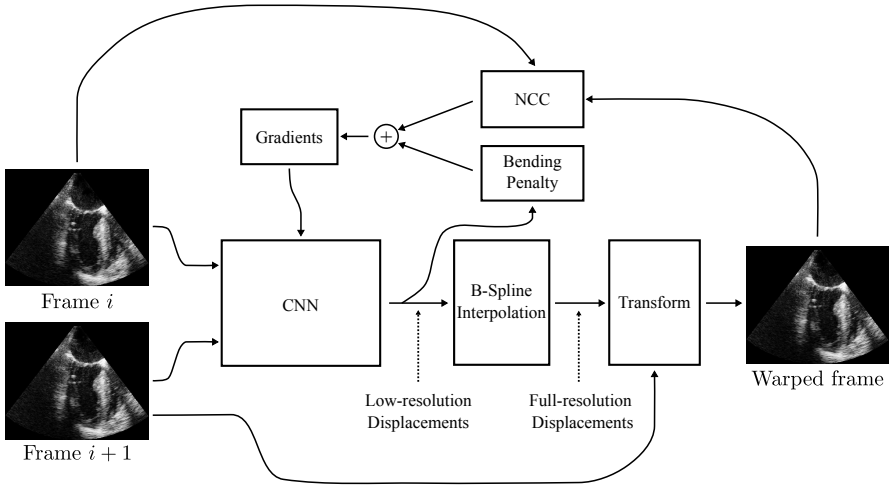


Figure 3.5: Training procedure for the motion estimation network. The CNN produces a low-resolution displacement field, which is then interpolated and used to warp frame $i + 1$ into frame i. Frame i and the warped frame are then compared using normalized cross-correlation. A differential based bending penalty is calculated from the low-resolution displacements and added to the cross-correlation to form the loss function. The loss is differentiated, and the gradients are used to update the CNN parameters.

To ensure spatial smoothness of the displacements, a differential based bending penalty can be added to the negated cross-correlation for regularization. This sum forms the loss function used to optimize the parameters of the network. The bending penalty P, given in Equation (3.1), minimizes the second order spatial derivatives of the displacements. This ensures that the transformation is locally affine, meaning that the transformation is globally smooth[54]. The scaling factor λ controls the amount of regularization.

$$P = \lambda \sum_{x,y \in I} \left(\frac{\partial^2 \vec{D}^d}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \vec{D}^d}{\partial y^2} \right)^2 + 2 \left(\frac{\partial^2 \vec{D}^d}{\partial xy} \right)^2 \tag{3.1}$$

B-splines were chosen as the interpolation method because they are controlled locally. That means that a change in one control point only affects the neighborhood around it in the resulting image. As regional strain is a measure of local deformation, this is a significant advantage compared to other popular interpolation methods, such as thin plate splines. Another advantage is that the k-th derivative of a B-spline of degree n is simply a B-spline of degree $n - k$. This means that the differentials needed for the bending penalty can be calculated by interpolating $\vec{D}$ using linear and quadratic B-splines[32, 55].

Image registration is often done in multiple stages in a coarse-to-fine manner. This makes the registration less sensitive to local optima and image folding at high resolutions[56]. In their work, de Vos et al. propose that a similar multi-stage strategy be employed for their

method[32], and this was implemented for the motion estimator. By daisy-chaining networks with decreasing downsampling rates (i.e., number of pooling layers), as illustrated in Figure 3.6, the warped image of each stage can be propagated to the next. In this way, each network estimates increasingly fine motions. The training of such daisy-chained networks should be done sequentially, optimizing each stage on the warped images from the preceding stage, keeping the weights fixed in the preceding stages, similar to boosting of tree-based models[57].

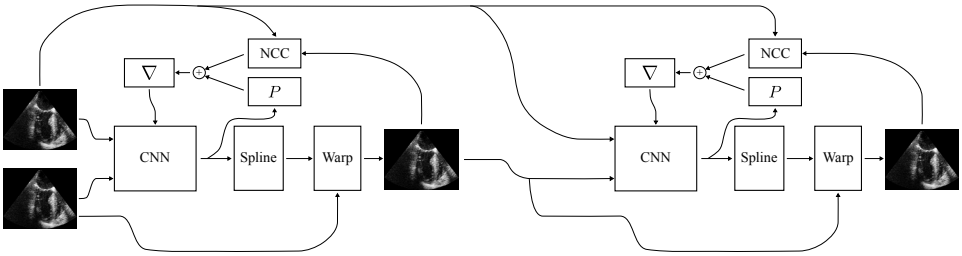


Figure 3.6: Daisy chaining of motion estimators. The warped frame of each consecutive estimator is propagated through to the next and is compared to frame i in every step.

The motion estimation was implemented in the Python programming language, using Tensorflow version 1.10[1] with eager execution enabled. As no source code was made available by de Vos et al.[32], the networks were implemented from scratch. Readers interested in the source code are referred to the Github repository for this thesis[2]. All convolutional layers except for the output layer consisted of 32 filters, and batch normalization and ReLU activations were applied. The last 1×1 convolutional layer consisted of two filters and was left unconstrained with no activation to freely estimate the displacements. Average pooling was done over 2×2 neighborhoods with a stride of 2, thus downsampling the features by 2 in each pooling layer. Zero-padding was performed for all convolution and pooling layers to ensure the correct dimensions of the dense displacement field.

The B-spline interpolation was implemented using fractionally strided convolutions, in which zeros are placed between each pixel in the image to achieve the desired dimensions. A B-spline kernel may then be constructed that weigh the original samples appropriately to produce an interpolated image through convolution. Because the downsampling factor is determined by the number of pooling layers in the network, the B-spline kernels for both the interpolation and the computation of the bending penalty may be precomputed, leaving only the convolution to be performed at runtime. For a more thorough explanation of this procedure, the reader is referred to Appendix B.

Three models for estimating motion were trained and evaluated: one model with four downsampling layers and one model with two downsampling layers, hereafter referred to as the high-downsampling-rate and low-downsampling-rate model, respectively, and a daisy-chained model combining the two. In all models, the regularization parameter was

[1] `https://www.tensorflow.org`
[2] `https://github.com/torjush/Strain_estimation`

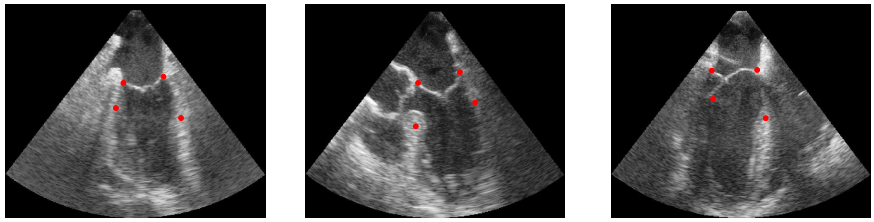
set to $\lambda = 5 \cdot 10^{-6}$, chosen empirically from results on the validation set. All weights were initialized using the Glorot Uniform initializer[58] and optimized using the Adam optimizer[51] with a learning rate $\alpha = 10^{-4}$. The batch sized used was 16 pairs of consecutive frames. The training was performed on a Tesla K80 GPU (Nvidia, Santa Clara, California) rented through the Floydhub cloud service for deep learning[3].

[3] `https://www.floydhub.com`

4 Results

4.1 Landmark detection

Evaluation of the detected landmarks was done by visual inspection by the author. Each initial frame of the test set was inspected, and the points detected were classified as either unsuitable, suitable, or highly suitable for strain estimation, based on the desired properties stated in Section 3.2. The criterium for being suitable was that the landmarks were visible and on the myocardium. The points that were near perfectly placed along the myocardium edge were classified as highly suitable. Points that were either invisible, noise, or not on the myocardium were classified as unsuitable for tracking. One example of each of these classes is shown in Figure 4.1. In Figure 4.1a, the detected points are located on the myocardium in a bright region near the endocardium. In Figure 4.1b, the detected points are placed on the myocardium, but the left point is a bit far from the edge. Lastly, in Figure 4.1c, one of the points is placed on the mitral valve, not on the myocardium, making it unsuitable for strain estimation.



(a) Highly suitable **(b)** Suitable **(c)** Unsuitable

Figure 4.1: Examples of detected points (lower red dots) that are highly suitable, suitable and unsuitable for strain estimation.

Table 4.1 summarizes the results of this visual inspection for each view present in the dataset and overall. For the four- and two-chamber views, the majority of the points are highly suitable, and suitable and highly suitable points make up around 95% of all detections. On the long-axis samples, the performance was slightly worse, with 87% of the detected points being suitable or highly suitable. Of these, only 29% were deemed highly suitable.

Table 4.1: Evaluation of detected landmarks on the test set, sorted by view.

4-chamber

	Highly Suitable	Suitable	Unsuitable	Total
# of detections	41	24	6	71
Percent	58%	34%	8%	100%

2-chamber

	Highly Suitable	Suitable	Unsuitable	Total
# of detections	57	13	5	75
Percent	76%	17%	7%	100%

Long-axis

	Highly Suitable	Suitable	Unsuitable	Total
# of detections	21	42	9	72
Percent	29%	58%	13%	100%

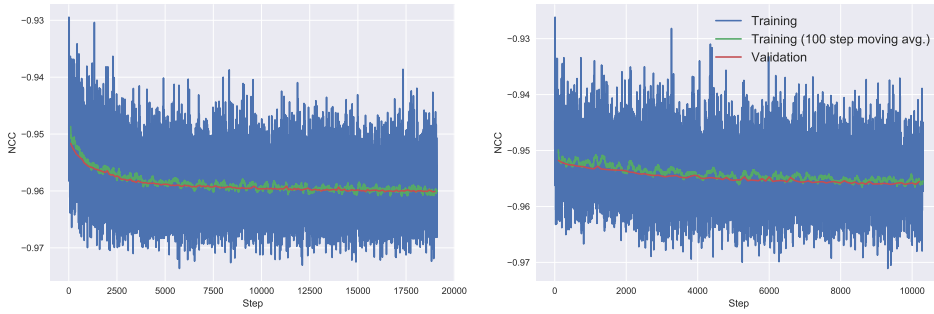
Overall

	Highly Suitable	Suitable	Unsuitable	Total
# of detections	119	79	20	218
Percent	55%	36%	9%	100%

4.2 Motion estimation and landmark tracking

4.2.1 Model training

During training, the negated normalized cross-correlation (NCC) was monitored for the training and validation sets. Validation was done every 100 steps, and training was ended when the validation NCC seemed to converge. Figure 4.2 shows the NCC on the training and validation sets throughout the training for the low-downsampling-rate (LDR) and high-downsampling-rate (HDR) models. As the inter-batch variance of the training NCC is high, a 100 step moving average (MA) of the training NCC is included to ease the comparison to the validation NCC.



(a) LDR model **(b)** HDR model

Figure 4.2: Learning curves showing the negated normalized cross-correlation for the LDR (left) and HDR (right) models on the training and validation sets. Training was ended when validation cross-correlation converged. Note that the scales are different in the two graphs.

Initially, the validation NCC of the two models are almost equal, but for the LDR model, it decreases more rapidly over the first 5000 steps. After 5000 steps, validation NCC decreases more slowly and eventually converges around -0.960 for the LDR model and -0.956 for the HDR model. Throughout the training, the validation NCC seems to follow the MA of the training NCC closely for both models.

When training the daisy-chained model, the resulting weights from training the HDR and LDR model separately were used to initialize the networks. Then, the LDR model was fine-tuned on the output of the HDR model. The learning curves for the daisy-chained model is shown in Figure 4.3. The dashed line indicates where fine-tuning of the LDR model begins. The training NCC seems to drop more than the validation NCC, as the validation curve does not follow the MA curve as closely as before, with validation NCC ending at -0.961.

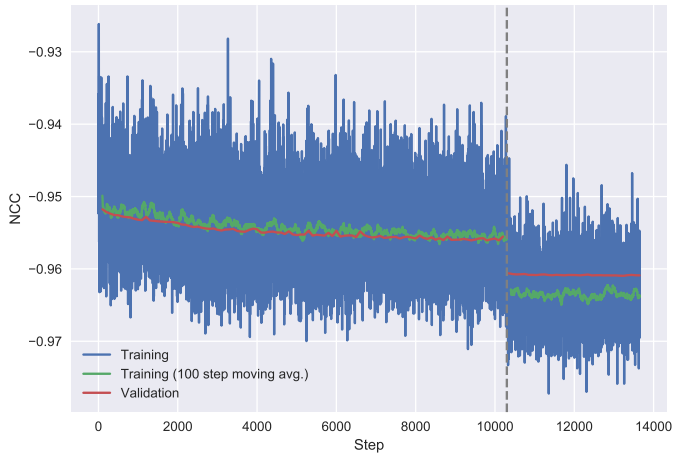


Figure 4.3: Learning curves showing the negated normalized cross-correlation for the HDR and LDR models daisy-chained and trained sequentially. The dashed line indicates where fine-tuning of the LDR model begins. Training was ended when validation cross-correlation converged.

The inference time for the dense displacements was estimated for all models when running on a MacBook Pro (Apple, Cupertino, California) with a 2.9GHz Core i5 processor (no GPU) by averaging over 100 random samples. For the two single-networks, inference time was found to be $(232 \pm 11)\,\mathrm{ms}$ and $(230 \pm 22)\,\mathrm{ms}$ for the LDR and HDR model, respectively. For the daisy-chained model: $(451 \pm 15)\,\mathrm{ms}$.

4.2.2 Visual inspection

The results of the landmark tracking were also inspected qualitatively by the author. For each basal segment in each view, the tracking of the mitral annulus and the detected landmark was classified as either successful or unsuccessful. For a tracking to be successful, the landmarks should be followed perfectly (as precisely as could be determined by visual inspection). No lag or drifting was allowed. Figures 4.4 and 4.5 show the percentage of successful trackings for the LDR model and the HDR model, respectively. The same diagrams are shown in Figure 4.6 for the daisy-chained model. The raw numbers used to produce these figures are available in Appendix C.

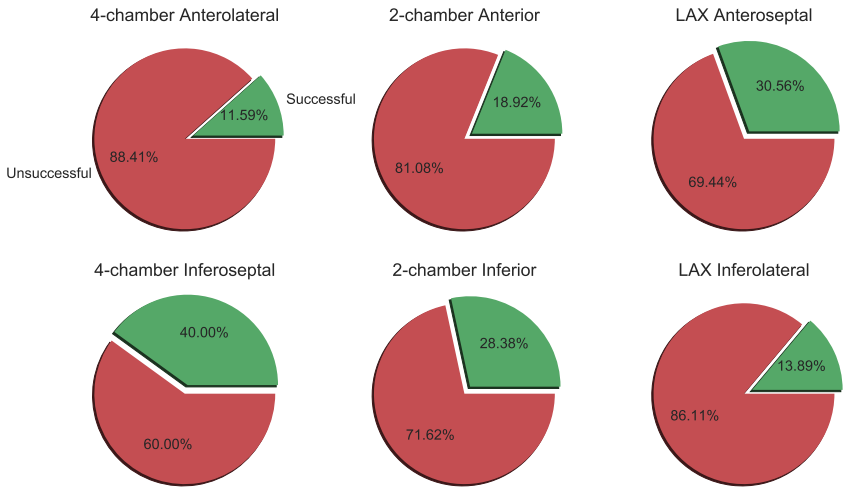


Figure 4.4: Results from visual inspection of the landmark tracking using the LDR model. Tracking is classified as either successful or unsuccessful.

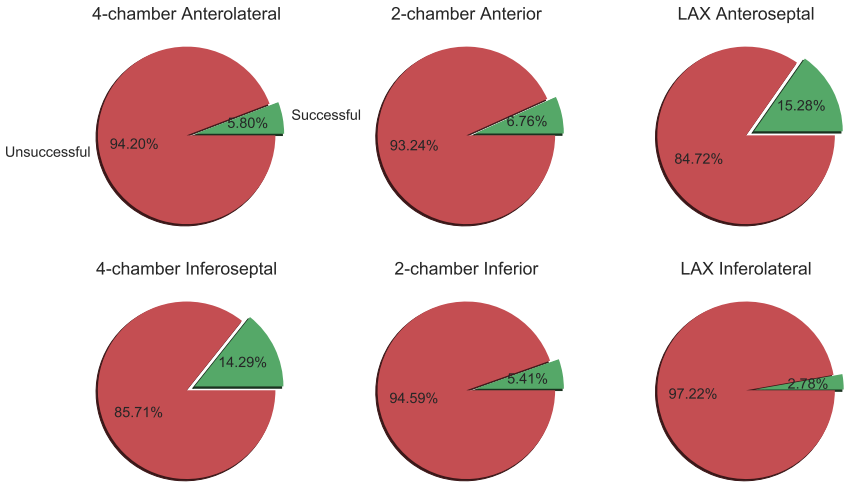


Figure 4.5: Results from visual inspection of the landmark tracking using the HDR model. Tracking is classified as either successful or unsuccessful.

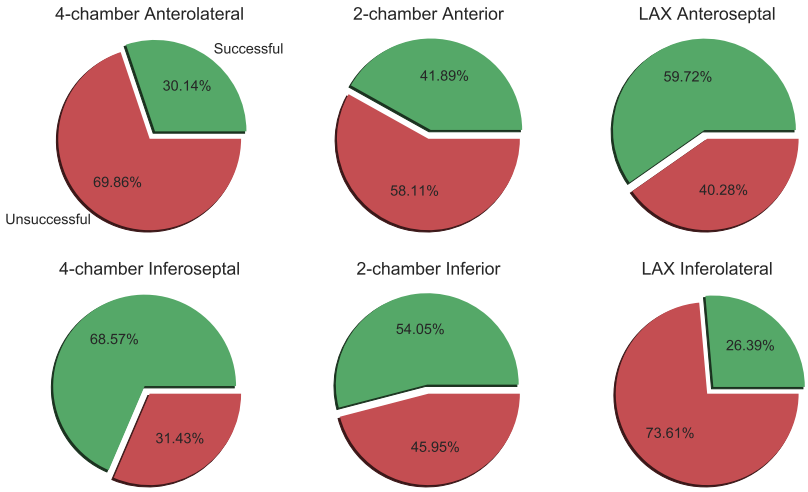


Figure 4.6: Results from visual inspection of landmark tracking for the daisy-chained networks. Tracking is classified as either successful or unsuccessful.

From Figures 4.4 and 4.5, it is clear that the LDR model outperforms the HDR model in both basal segments in all views. In many of the unsuccessful samples, the LDR model was able to track the landmarks quite well when displacements were small, but lag was observed for larger displacements due to underestimating the motion. The HDR model consistently underestimated the motion of slow-moving landmarks. However, in some samples captured at low frame rates, or where the patient's heart rate was high, the overall motion between frames was larger, and the HDR model was able to track the landmarks perfectly.

The daisy-chained model outperformed both single-network models and tracked the landmarks of one basal segment perfectly in more than 50% of the samples. Like the LDR model, large displacements caused lag due to underestimation, but to a lesser degree and in fewer of the samples.

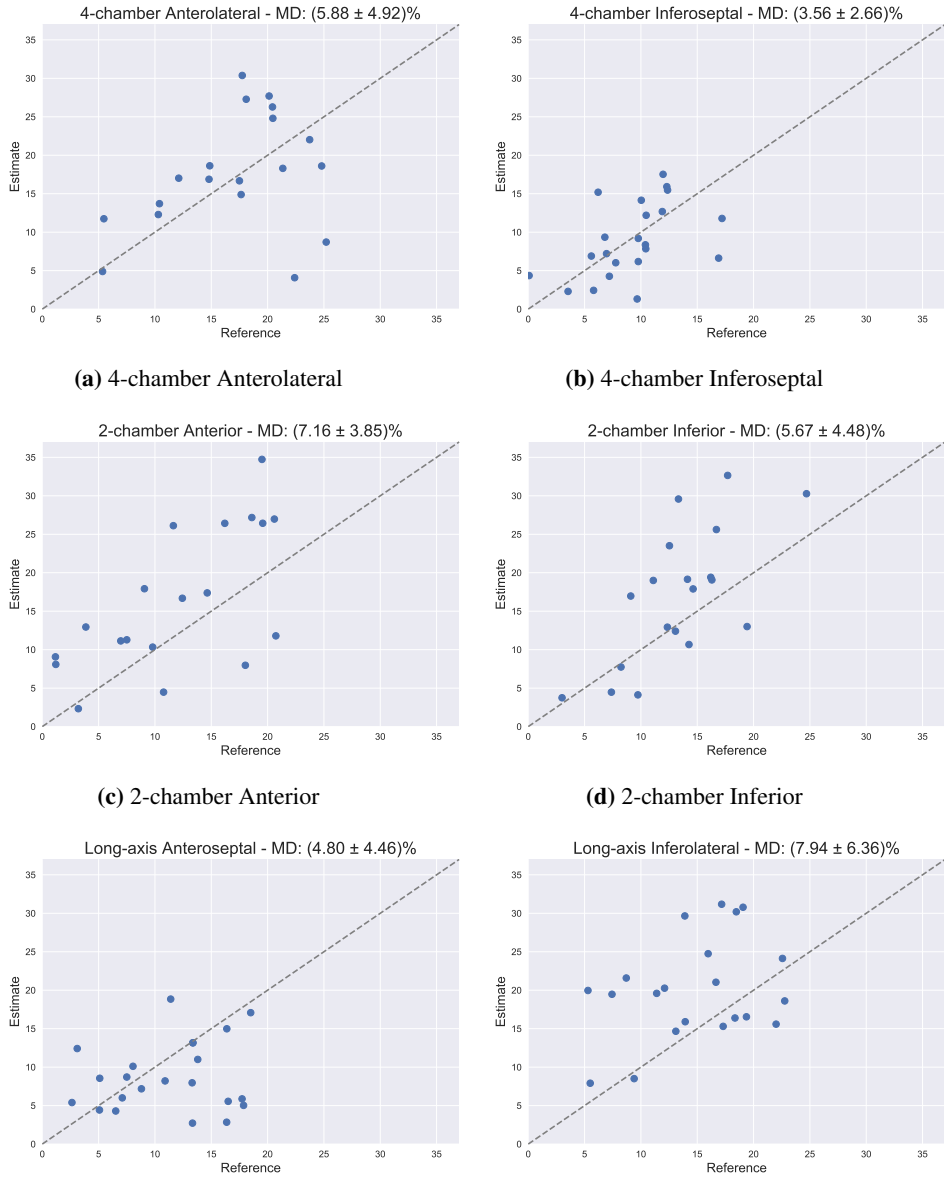
In the digital appendix accompanying this thesis, some videos showing the tracking done by the daisy-chained model are included. The files `out-of-plane.mp4`, `air_pocket_noise.mp4`, and `uniform_region.mp4` correspond to Figure 5.2, 5.3, and 5.4, respectively. `successful_4c.mp4`, `successful_2c.mp4`, and `successful_lax.mp4` show examples of successful trackings in the 4-chamber, 2-chamber, and long-axis view, respectively.

4.3 Strain estimation

Basal end-systolic strain was estimated for each cardiac cycle in the recordings and averaged using both the proposed pipeline and commercially available speckle tracking software with operators being blinded to each other's results. For the anteroseptal segment in the long-axis view, the strain estimates produced by the CNNs are made by tracking two points on either side of the left ventricular outflow track, whereas the reference values are made tracking points further down on the myocardium. Due to the low frame rate in some samples, and low image quality in others, ten segments out of the total 138 present in the test set were deemed unsuitable for tracking in EcoPAC, and were removed from the comparison.

Figures 4.7, 4.8, and 4.9 show the comparisons between the reference and the LDR, HDR, and daisy-chained model, respectively. Each basal segment is shown separately, and the mean absolute differences (MD) between the references and the CNN estimates in percent are included. Table 4.2 summarizes the MD-values, along with their estimated standard deviation σ. For each segment, the correlations ρ between the reference values and the CNN estimates are also included.

The daisy-chained model estimates are closest to those from the speckle tracking method overall, with the lowest MD in three out of the six segments and the highest correlation in four. The HDR model produces the least similar estimates, with the highest MD and lowest correlation in five out of the six segments. All models consistently underestimated the strain in the anteroseptal segment in the long-axis view.



(a) 4-chamber Anterolateral

(b) 4-chamber Inferoseptal

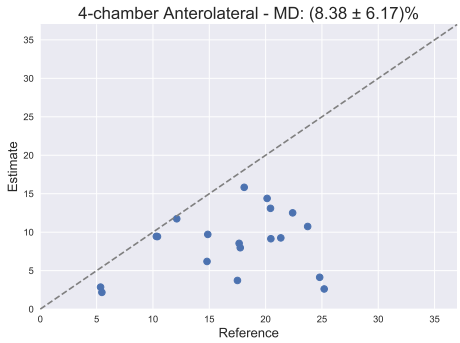
(c) 2-chamber Anterior

(d) 2-chamber Inferior

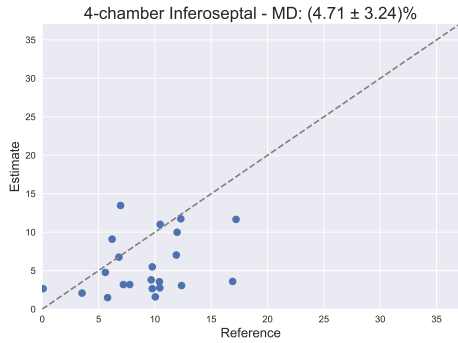
(e) Long-axis Anteroseptal

(f) Long-axis Inferolateral

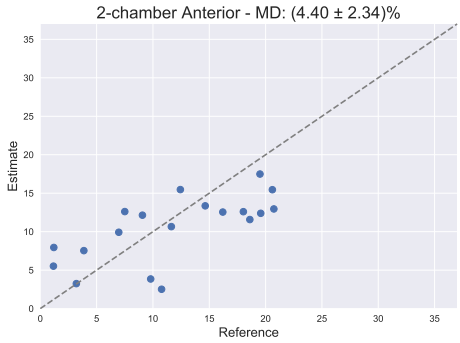
Figure 4.7: Strain estimates from the LDR model plotted against the reference values.



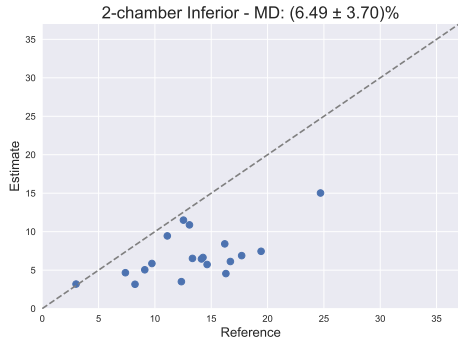
(a) 4-chamber Anterolateral



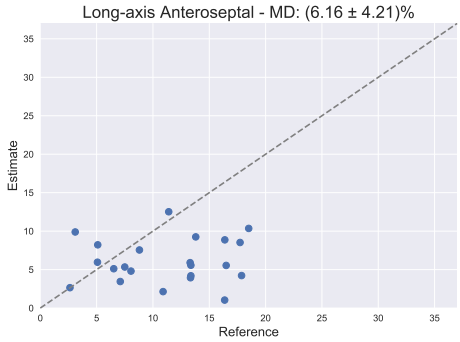
(b) 4-chamber Inferoseptal



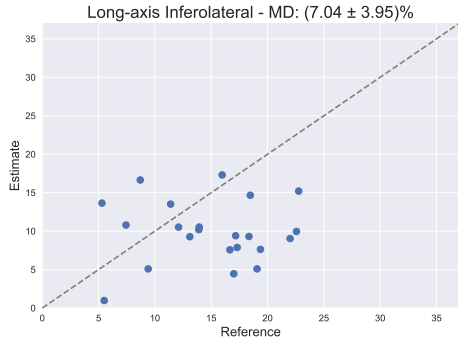
(c) 2-chamber Anterior



(d) 2-chamber Inferior

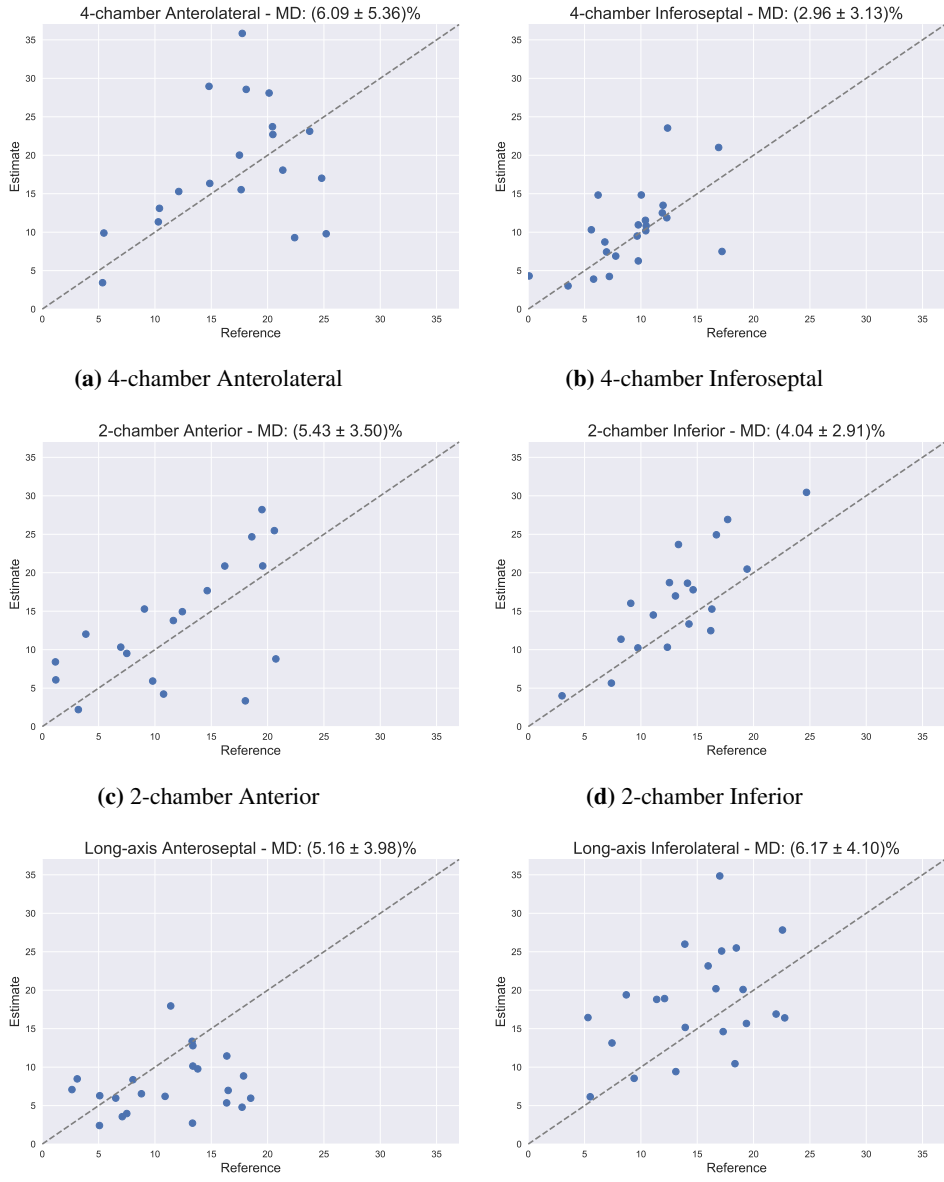


(e) Long-axis Anteroseptal



(f) Long-axis Inferolateral

Figure 4.8: Strain estimates from the HDR model plotted against the reference values.



(a) 4-chamber Anterolateral

(b) 4-chamber Inferoseptal

(c) 2-chamber Anterior

(d) 2-chamber Inferior

(e) Long-axis Anteroseptal

(f) Long-axis Inferolateral

Figure 4.9: Strain estimates from the daisy-chained model plotted against the reference values.

Table 4.2: Comparison between the reference values and the CNN estimates in percent. Mean absolute difference and standard deviation σ are provided, along with the correlation coefficient ρ. The best results are indicated in bold.

4-chamber

	Anterolateral			Inferoseptal		
	MD	σ	ρ	MD	σ	ρ
LDR	**5.88**	4.92	0.34	3.56	2.66	0.47
HDR	8.36	6.17	0.26	4.71	3.24	0.30
Daisy-chained	6.09	5.36	**0.38**	**2.96**	3.13	**0.59**

2-chamber

	Anterior			Inferior		
	MD	σ	ρ	MD	σ	ρ
LDR	7.16	3.85	0.65	5.67	4.48	0.69
HDR	**4.40**	2.34	**0.68**	6.49	3.70	0.61
Daisy-chained	5.43	3.50	0.62	**4.04**	2.91	**0.82**

Long-axis

	Anteroseptal			Inferolateral		
	MD	σ	ρ	MD	σ	ρ
LDR	7.94	6.36	0.15	**4.80**	4.46	0.32
HDR	7.04	3.95	0.083	6.16	4.21	0.042
Daisy-chained	**6.17**	4.10	**0.21**	5.16	3.98	**0.42**

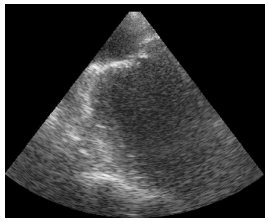
5 Discussion

5.1 Landmark detection

The visual inspection of the detected landmarks showed that the method was able to detect a visible point on the myocardium in the majority of the samples in the test set. The detector performs best on the 2-chamber view and worst on the long-axis view. This is to be expected, due to the trade-offs that are made in each view.

Two of the main applications of the 4-chamber view is to diagnose atrioventricular valve dysfunction and atrial septal defects. As a consequence, these images focus on the mitral valve and the septum, with the anterolateral segment being less important. The long-axis view is often focused on the aortic valve and the left ventricular outflow tract. To achieve this, the inferolateral segment is given less priority. These trade-offs negatively impact the landmark detection, as the frequency of out-of-plane and blurry basal segments is higher in these views. In the 2-chamber view, there is less of a trade-off from a strain imaging perspective. Here, priority is given to the left ventricle (LV) in its entirety in addition to the mitral valve, and both basal segments should be well depicted.

A few common factors were found to negatively impact the landmark detection. The first and most obvious is the case where the myocardial segment is missing from the images entirely, as shown in Figures 5.1a and 5.1b. In these cases, the detection of suitable landmarks is entirely infeasible. A slightly less sinister case is when the boundary between the basal segments and the ventricle is unclear or obfuscated by noise, as in Figure 5.1c. These unclear boundaries make the detection of a highly suitable landmark difficult, but in most cases, a somewhat suitable point is still found. While there is not much to do with the cases where the segments are not visible in end-diastole, the method proved to be robust against a variety of other issues, including random noise, variation in the mitral plane angle with respect to the probe, and slightly decreased visibility of the myocardium.



(a) Inferior segment is outside of the scanning sector.

(b) Anterolateral segment is outside of the scanning plane.

(c) Inferolateral segment is visible, but has an unclear boundary to the ventricle.

Figure 5.1: Examples of images where landmark detection is difficult or infeasible

Increasing the ratio of highly suitable points detected would be desirable. The main difference between the suitable and highly suitable points was that the suitable points were not close enough to the endocardium edge. Thus, to improve these detections, the detector should be motivated to choose points closer to the center of the images. This could be done by weighting such points more when searching for the brightest pixel. Another more computationally expensive approach would be to employ an automated LV contouring algorithm and detect suitable points along this contour. This would most likely require training a machine learning model using manually annotated examples, making it a costly improvement.

5.2 Motion estimation and landmark tracking

The motion estimation task was performed by a fully convolutional neural network (CNN), which was trained in an unsupervised manner. This has several advantages over other approaches. As the network contains only convolutional layers, it can predict motion in samples of any resolution. This is particularly useful in ultrasound, where the view and sample rate impacts the spatial resolution, so that samples acquired using the same equipment on the same patient may have different dimensions. Unsupervised training does not require manual annotation of the training set, which is a time-consuming process. This makes it easy to retrain a model with different data. While hybrid approaches, such as supervised training on synthetic datasets have shown promise[21], it was assumed that it would be advantageous to train the model using the same type of data as for testing.

During the training of the CNNs, no evidence of overfitting was observed. For both single-network models, the validation loss follows a moving average of the training loss closely on a decreasing trend, indicating successful learning. When daisy-chaining the two single-network models and fine-tuning the low-downsampling-rate (LDR) model, the training loss decreased faster than the validation loss, indicating that little gain could be made by fine-tuning. Still, the daisy-chaining alone gave lower validation loss. The training set loss fluctuates a lot, meaning that there will be large variations between consecutive update steps. This could make convergence to an optimum slower and more difficult, and would normally be countered by increasing the batch size or lowering the learning rate. Due to memory limitations, an increase in batch size was not possible, and a lower learning rate

did not improve the end result in the presented experiments.

The inference times estimated are quite high, about twice the time needed using Østvik et al.'s method for global longitudinal strain estimation[20]. This is an unfair comparison, however, as a laptop with a GPU was used in their experiments. Inference times can likely be significantly reduced if the experiments are run using a GPU, due to their excellent performance on convolutions. Furthermore, the implementation used in this work relied on processing by the Python interpreter in addition to the Tensorflow library. If the method was to be implemented in a compiled language, such as C++, further performance gains might be achieved. Another way of reducing inference time would be to crop the frames before inference. As the landmarks are located in the top half of a frame, it is wasteful to estimate the frame-to-frame displacements for the entire image.

As for the landmark detection, it is clear that the performance of the landmark tracking depends on the view in all models. As expected, tracking performance in the 4-chamber and long-axis views is best in the inferoseptal and anteroseptal segment, respectively, as these segments are prioritized when acquiring samples in these views. In the 2-chamber view, there is less of a difference in performance between the two basal segments.

Out of the three models evaluated, the daisy-chained model performs best. In total, 46% of the segments are successfully tracked. This compares well to current methods. In their population study, Dalen et al. report that tracking was infeasible in 60% of the segments in samples acquired specifically for the purpose of strain estimation[11].

All models underestimated the motion of the landmarks in certain situations. The high-downsampling-rate (HDR) model underestimated almost all motion, except for very large movements. This indicates that the receptive field of the CNN is too large. The LDR model shows the opposite behavior. It consistently underestimated large motion, while following smaller movements quite well, indicating that the receptive field is too small. Combining the two in a daisy-chained model improved tracking overall. This is somewhat expected, as the sum of two underestimated motion vectors can be closer to the true motion. However, underestimation was still an issue in some samples. An interesting experiment could be to daisy-chain several networks with few downsampling layers. This would estimate the larger motions incrementally while keeping the advantage on smaller motions. As the bending penalty, in its effort to ensure spatial smoothness, also penalizes the magnitude of the displacements, it is also possible that a decrease in the regularization parameter could alleviate the underestimation.

Some issues common to all models were identified. Out-of-plane movement, where the motion of the myocardium is not parallel to the scanning plane of the ultrasound beam, makes the myocardium disappear in some frames and while being present in others, causing the tracked point to drift or get stuck. This is sometimes a result of the 4-chamber view not being recorded at exactly 0°. These improper 4-chamber images can be identified by the presence of the left ventricular outflow tract (LVOT) in the images. Out-of-plane motion is illustrated in Figure 5.2. In Figure 5.2a, the anterolateral segment is clearly visible, but a few frames later, the segment has moved out of the scanning plane and becomes invisible in Figure 5.2b.

(a) Anterolateral segment visible

(b) Anterolateral segment not visible

Figure 5.2: Example of out-of-plane movement of the myocardium. The anterolateral segment's motion is not parallel to the scanning plane, causing it to disappear.

Another issue is noise caused by pockets of air between the probe and the heart. As air has a higher acoustic impedance than tissue, these pockets cause strong reflections, resulting in a bright spot in the images which is constant between frames. Figure 5.3 shows a frame where such noise is covering the mitral annulus above the anterolateral segment, indicated by a red rectangle. This causes the tracking to get stuck. Lastly, in some samples, the landmarks are found in a near uniform, bright region. In these cases, the tracking follows the region quite well but tends to drift within the region. Figure 5.4 shows a frame where the mitral annulus above the inferior segment is found in such a region.

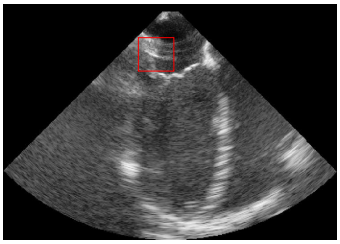


Figure 5.3: Example of noise caused by a pocket of air. The bright spot indicated in red is present in all frames in the sample and causes the tracking to get stuck.

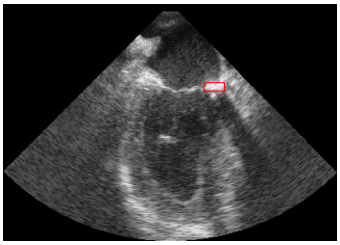


Figure 5.4: Example of a uniform, bright region around the landmark. The region is indicated in red. Such uniformity causes the tracking to drift within the region.

Out-of-plane movement and constant noise in the samples ares issues that are difficult

to overcome. Efforts were made to keep the tracking from failing in these situations, including smoothing the motion vectors using exponential moving averages (EMAs) and Kalman filtering without success. Using such smoothing methods also induce latency in the tracking, which may lead to failed tracking in later frames or introduce bias in the strain estimates.

Drifting in uniform regions can be attributed to the way the motion estimators are trained. As the normalized cross-correlation was optimized, the trained models do not differentiate which pixels go where in the warped frames, as long as they have similar brightness. That means that the pixels within a uniform region may be shuffled, causing folding in the warped frame. To reduce folding more networks could be daisy-chained, or the regularization parameter could be increased. Both of these approaches come at a cost, as more networks would increase the inference time, and increased regularization would cause the estimates to be more biased towards small displacements.

It should be noted that the performance of both the landmark detection and tracking was evaluated by visual inspection by the author. This inspection may be subject to biased opinions, and intra-observer variability must be assumed. This is a consequence of using unsupervised methods and should be taken into consideration when interpreting the results.

5.3 Strain estimation

In line with expectations from the visual inspection, the strain estimates produced by the daisy-chained model from the inferoseptal segment are closest to the reference values in terms of mean absolute difference (MD). Looking at correlation, the inferior segments compare best. One would expect better performance on the anteroseptal segments from the visual inspection, but as the CNN estimates on this segment are taken across the LVOT, and the references are measured on the myocardium, some underestimation is expected, and can in fact be observed in Figures 4.7e, 4.8e, and 4.9e. Strain is not normally measured across the LVOT, and the fact that it was done in this implementation was a result of the assumption of known mitral annulus landmarks, and the particular landmark detection algorithm.

In Figures 4.7, 4.8, and 4.9, a large spread around the diagonal can be observed for all models. This is not surprising, as the proposed pipeline includes several sources of error. Firstly, 55% of the detected landmarks used for tracking were highly suitable; the remaining 45% are expected to cause decreased precision during tracking. Secondly, the landmark tracking was deemed unsuccessful in the majority of the segments in the test set for all models except for the daisy-chained model. Drifting and underestimation of motion when tracking a landmark were the leading causes of unsuccessful tracking, both of which may cause large variation in the resulting strain estimate.

The reference values that the strain estimates were compared against were acquired using a commercial speckle tracking method. Speckle tracking, while commercially available and widely used, can not be viewed as an absolute truth. As Knackstedt et al. demonstrated, these methods suffer from significant inter- and intra-observer variation, in addition to inter-vendor variability[17]. Thus, some deviation from the reference was expected. In

addition, many of the samples used for strain estimation were acquired at very low frame rates, challenging the feasibility of accurately measuring strain using traditional methods.

5.4 Limitations of study and future work

Mitral annulus landmark detection was not included in this thesis, as efforts have shown this task to be learnable by deep learning models, and it is the subject of ongoing work at our department. This is nevertheless a major shortcoming of the proposed estimation pipeline, and to achieve the end-goal of fully automating basal strain estimation from raw TEE images it would be necessary to extend the landmark detection algorithm to also detect the mitral annulus. Furthermore, the algorithm should be extended to detect the edge of the myocardium below the LVOT in the long-axis view so that the right landmarks are tracked.

To improve the landmark tracking, more experiments should be done with different downsampling rates and different combinations of daisy-chained networks. Each hyperparameter should also be carefully tuned for each model, most importantly the regularization parameter, learning rate, and the number of filters in the convolutional layers. It is also possible that training a separate model for each view would be beneficial. As the learning problem becomes less comprehensive, each model would be able to specialize more, perhaps increasing performance. Alternatively, the networks could be provided with information about the view. It would also be interesting to perform more experiments using the current setup. Cherry picking high-quality samples with high frame rates for testing could more precisely determine the feasibility of estimating strain using the proposed pipeline. Then, if the results are satisfactory, efforts could be concentrated towards increasing robustness.

For a more accurate assessment of the strain estimates, reference values could be acquired using several commercially available methods for comparison. The proposed method should also be compared to other recent approaches, such as Østvik et al.'s flownet based approach[20, 21], evaluated on the same test set. Such comparisons could provide a more detailed view of the strengths and weaknesses of the method proposed in this thesis.

The long inference times are problematic, as continuous monitoring would require estimation in real- or at least near real-time. Though a GPU is believed to improve performance significantly, experiments should be run on the actual hardware available in the operating theater to accurately determine wether or not real- or near real-time strain estimation is feasible using the proposed method.

6 Conclusion

In this thesis, a novel approach to automatic regional strain estimation in transesophageal echocardiographic images was presented, consisting of landmark detection, landmark tracking, and strain calculation. The landmark detection algorithm assumed the location of the mitral annulus to be known and used traditional linear filtering to detect a second landmark on the myocardium at a suitable distance from the mitral annulus. The landmark tracking was done using an adapted implementation of the Deep Learning Framework for Unsupervised Affine and Deformable Image Registration introduced by de Vos et al.[32].

Three deep learning models were trained and evaluated for the landmark tracking task: two single-network models and one daisy-chained network combining them. Both training and evaluation were done using unselected transesophageal echocardiographic images in 4-chamber, 2-chamber, and long-axis views. The daisy-chained model achieved the best performance, both when subject to visual inspection and when compared to a commercially available method. In particular, the results from the 4- and 2-chamber views show promise, while performance decreased in the long-axis samples. Several sources of error were identified, and improvements suggested. The observed inference times rules out real-time applications. However, several readily available measures may be taken that are believed to reduce the time needed to estimate the frame-to-frame displacements significantly.

In conclusion, the results show that the method can accurately estimate strain in the basal segments in the 4- and 2-chamber views in samples where image quality is high. This indicates that strain estimation is indeed a learnable task using deep learning methods and that full automation is feasible. In samples where the images are of lower quality, the method performs variably, or fails entirely to produce a reasonable estimate. Thus, further efforts should be focused towards improving robustness and increasing performance in the long-axis view.

References

[1] Sebastian Zaunseder et al. “Impact of cardiac surgery on the autonomic cardiovascular function”. In: *Journal of Computational Surgery* 1.1 (2014), p. 9. ISSN: 2194-3990. DOI: 10.1186/2194-3990-1-9. URL: https://doi.org/10.1186/2194-3990-1-9.

[2] John G. Laffey, John F. Boylan, and Davy C. Cheng. “The Systemic Inflammatory Response to Cardiac Surgery: Implications for the Anesthesiologist”. In: *Anesthesiology: The Journal of the American Society of Anesthesiologists* 97.1 (July 2002), pp. 215–252. ISSN: 0003-3022. eprint: http://anesthesiology.pubs.asahq.org/jasa/content_public/journal/jasa/931215/0000542-200207000-00030.pdf. URL: https://doi.org/.

[3] Lawrence L. Creswell et al. “Intraoperative Interventions*: American College of Chest Physicians Guidelines for the Prevention and Management of Postoperative Atrial Fibrillation After Cardiac Surgery”. English. In: *Chest* 128.2 (Aug. 2005). Copyright - Copyright American College of Chest Physicians Aug 2005; Last updated - 2014-05-17; CODEN - CHETBF, 28S–35S. URL: https://search.proquest.com/docview/200452102?accountid=12870.

[4] Scott T. Reeves et al. “Basic Perioperative Transesophageal Echocardiography Examination: A Consensus Statement of the American Society of Echocardiography and the Society of Cardiovascular Anesthesiologists”. In: *Journal of the American Society of Echocardiography* 26.5 (2013), pp. 443 –456. ISSN: 0894-7317. DOI: https://doi.org/10.1016/j.echo.2013.02.015. URL: http://www.sciencedirect.com/science/article/pii/S0894731713001399.

[5] Dheeraj Arora and Yatin Mehta. “Recent trends on hemodynamic monitoring in cardiac surgery”. In: *Annals of Cardiac Anaesthesia* 19 (Oct. 2016), pp. 580–583. DOI: 10.4103/0971-9784.191557.

[6] Jean-Louis Vincent et al. “Perioperative cardiovascular monitoring of high-risk patients: a consensus of 12”. In: *Critical Care* 19.1 (2015), p. 224. ISSN: 1364-8535. DOI: 10.1186/s13054-015-0932-7. URL: https://doi.org/10.1186/s13054-015-0932-7.

[7] Peter R. Hoskins, Kevin Martin, and Abigail Thrush. *Diagnostic Ultrasound : Physics and Equipment*. 2nd ed. Cambridge University Press, 2010.

[8] Kayvan Najarian and Robert Splinter. *Biomedical Signal and Image Processing*. 2nd ed. CRC Press, 2012. ISBN: 978-1-4398-7034-1.

[9] Roberto M. Land et al. “Recommendations for Cardiac Chamber Quantification by Echocardiography in Adults: An Update from the American Society of Echocardiography and the European Association of, Cardiovascular Imaging”. In: *European Heart Journal - Cardiovascular Imaging* 17.4 (Mar. 2016), pp. 412–412. ISSN: 2047-2404. DOI: 10.1093/ehjci/jew041. eprint: http://oup.prod.

sis.lan/ehjcimaging/article-pdf/17/4/412/7464119/jew041.pdf. URL: https://doi.org/10.1093/ehjci/jew041.

[10] Constant L.A. Reichert et al. "Prognostic value of biventricular function in hypotensive patients after cardiac surgery as assessed by transesophageal echocardiography". In: *Journal of Cardiothoracic and Vascular Anesthesia* 6.4 (1992), pp. 429 –432. ISSN: 1053-0770. DOI: https://doi.org/10.1016/1053-0770(92)90008-U. URL: http://www.sciencedirect.com/science/article/pii/105307709290008U.

[11] Havard Dalen et al. "Segmental and global longitudinal strain and strain rate based on echocardiography of 1266 healthy individuals: the HUNT study in Norway". In: *European Journal of Echocardiography* 11.2 (2010), pp. 176–183. ISSN: 1525-2167.

[12] Asbjørn Støylen. *Basic Doppler ultrasound for clinicians*. URL: http://folk.ntnu.no/stoylen/strainrate/Basic_Doppler_ultrasound#Tissue_Doppler (visited on 05/24/2019).

[13] Thomas H. Marwick. "Measurement of Strain and Strain Rate by Echocardiography: Ready for Prime Time?" In: *Journal of the American College of Cardiology* 47.7 (2006), pp. 1313 –1327. ISSN: 0735-1097. DOI: https://doi.org/10.1016/j.jacc.2005.11.063. URL: http://www.sciencedirect.com/science/article/pii/S0735109706001628.

[14] Brage H. Amundsen et al. "Noninvasive Myocardial Strain Measurement by Speckle Tracking Echocardiography: Validation Against Sonomicrometry and Tagged Magnetic Resonance Imaging". In: *Journal of the American College of Cardiology* 47.4 (2006), pp. 789 –793. ISSN: 0735-1097. DOI: https://doi.org/10.1016/j.jacc.2005.10.040. URL: http://www.sciencedirect.com/science/article/pii/S0735109705027506.

[15] Brage H. Amundsen et al. "Regional myocardial long-axis strain and strain rate measured by different tissue Doppler and speckle tracking echocardiography methods: a comparison with tagged magnetic resonance imaging". In: *European Heart Journal - Cardiovascular Imaging* 10.2 (July 2008), pp. 229–237. ISSN: 2047-2404. DOI: 10.1093/ejechocard/jen201. eprint: http://oup.prod.sis.lan/ehjcimaging/article-pdf/10/2/229/13801377/jen201.pdf. URL: https://doi.org/10.1093/ejechocard/jen201.

[16] Dan Rappaport et al. "Assessment of myocardial regional strain and strain rate by tissue tracking in B-mode echocardiograms". In: *Ultrasound in Medicine & Biology* 32.8 (2006), pp. 1181 –1192. ISSN: 0301-5629. DOI: https://doi.org/10.1016/j.ultrasmedbio.2006.05.005. URL: http://www.sciencedirect.com/science/article/pii/S0301562906016048.

[17] Konstantinos Farsalinos et al. "Head-to-Head Comparison of Global Longitudinal Strain Measurements among Nine Different Vendors: The EACVI/ASE Inter-Vendor Comparison Study". In: *Journal of the American Society of Echocardiography : official publication of the American Society of Echocardiography* 28 (July 2015). DOI: 10.1016/j.echo.2015.06.011.

[18] B. Heyde et al. "Elastic Image Registration Versus Speckle Tracking for 2-D Myocardial Motion Estimation: A Direct Comparison In Vivo". In: *IEEE Transactions*

on Medical Imaging 32.2 (2013), pp. 449–459. ISSN: 0278-0062. DOI: 10.1109/TMI.2012.2230114.

[19] Christian Knackstedt et al. "Fully Automated Versus Standard Tracking of Left Ventricular Ejection Fraction and Longitudinal Strain". In: 66.13 (2015), pp. 1456–1466. DOI: 10.1016/j.jacc.2015.07.052.

[20] A. Østvik et al. "Automatic myocardial strain imaging in echocardiography using deep learning". In: vol. 11045. Springer Verlag, 2018, pp. 309–316. ISBN: 9783030008888.

[21] E. Ilg et al. "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1647–1655. DOI: 10.1109/CVPR.2017.179.

[22] Bryan Catanzaro. *Deep Speech: Accurate Speech Recognition with GPU-Accelerated Deep Learning*. Feb. 2015. URL: https://devblogs.nvidia.com/deep-speech-accurate-speech-recognition-gpu-accelerated-deep-learning/ (visited on 05/10/2019).

[23] Yoav Goldberg. "A Primer on Neural Network Models for Natural Language Processing". In: *CoRR* abs/1510.00726 (2015). arXiv: 1510.00726. URL: http://arxiv.org/abs/1510.00726.

[24] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: *arXiv* (2018).

[25] Kaiming He et al. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870.

[26] Holger Haenssle et al. "Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists". In: *Annals of oncology : official journal of the European Society for Medical Oncology* 29 (May 2018). DOI: 10.1093/annonc/mdy166.

[27] Thijs Kooi et al. "Large Scale Deep Learning for Computer Aided Detection of Mammographic Lesions". In: *Medical Image Analysis* 35 (Aug. 2016). DOI: 10.1016/j.media.2016.07.007.

[28] Younghak Shin and Ilangko Balasingham. "Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification". In: vol. 2017. July 2017, pp. 3277–3280. DOI: 10.1109/EMBC.2017.8037556.

[29] Mojtaba Akbari et al. "Classification of Informative Frames in Colonoscopy Videos Using Convolutional Neural Networks with Binarized Weights". In: (Feb. 2018).

[30] Guha Balakrishnan et al. "VoxelMorph: A Learning Framework for Deformable Medical Image Registration". In: *CoRR* abs/1809.05231 (2018). arXiv: 1809.05231. URL: http://arxiv.org/abs/1809.05231.

[31] Bob D. de Vos et al. "End-to-End Unsupervised Deformable Image Registration with a Convolutional Neural Network". In: *CoRR* abs/1704.06065 (2017). arXiv: 1704.06065. URL: http://arxiv.org/abs/1704.06065.

[32] Bob D. de Vos et al. "A Deep Learning Framework for Unsupervised Affine and Deformable Image Registration". In: *CoRR* abs/1809.06130 (2018). arXiv: 1809.06130. URL: http://arxiv.org/abs/1809.06130.

[33] Andreas Østvik et al. “Real-Time Standard View Classifcation in Transthoracic Echocardiography Using Convolutional Neural Networks”. In: *Ultrasound in Medicine & Biology* 45 (Nov. 2018). DOI: 10.1016/j.ultrasmedbio.2018.07.024.

[34] Jeffrey Zhang et al. “Fully Automated Echocardiogram Interpretation in Clinical Practice”. In: *Circulation* 138.16 (2018), pp. 1623–1635. DOI: 10.1161/CIRCULATIONAHA.118.034338. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCULATIONAHA.118.034338. URL: https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.118.034338.

[35] Mohammad H. Jafari et al. “A Unified Framework Integrating Recurrent Fully-Convolutional Networks and Optical Flow for Segmentation of the Left Ventricle in Echocardiography Data”. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Ed. by Danail Stoyanov et al. Cham: Springer International Publishing, 2018, pp. 29–37. ISBN: 978-3-030-00889-5.

[36] Erik Smistad et al. “Fully automatic real-time ejection fraction and MAPSE measurements in 2D echocardiography using deep neural networks”. eng. In: (2018). ISSN: 1948-5719. URL: http://hdl.handle.net/11250/2588226.

[37] Richard Drake, A. Wayne Vogl, and Adam W. M. Mitchell. *Gray’s Anatomy for Students*. 3rd ed. Churchill Livingstone, 2015. ISBN: 9780702051333.

[38] Lindsay M. Biga et al. *Anatomy & Physiology*. Open Oregon State, Oregon State University, 2019.

[39] John Gorcsan and Hidekazu Tanaka. “Echocardiographic Assessment of Myocardial Strain”. In: *Journal of the American College of Cardiology* 58.14 (2011), pp. 1401–1413. ISSN: 0735-1097. DOI: 10.1016/j.jacc.2011.06.038. eprint: http://www.onlinejacc.org/content/58/14/1401.full.pdf. URL: http://www.onlinejacc.org/content/58/14/1401.

[40] Anders Opdahl et al. “Myocardial strain imaging: how useful is it in clinical decision making?” In: *European Heart Journal* 37.15 (Oct. 2015), pp. 1196–1207. ISSN: 0195-668X. DOI: 10.1093/eurheartj/ehv529. eprint: http://oup.prod.sis.lan/eurheartj/article-pdf/37/15/1196/24121129/ehv529.pdf. URL: https://doi.org/10.1093/eurheartj/ehv529.

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

[42] Junfei Qiu et al. “A survey of machine learning for big data processing”. In: *EURASIP Journal on Advances in Signal Processing* 2016.1 (2016), p. 67. ISSN: 1687-6180. DOI: 10.1186/s13634-016-0355-x. URL: https://doi.org/10.1186/s13634-016-0355-x.

[43] Isha Salian. *SuperVize Me: What’s the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?* Aug. 2, 2018. (Visited on 05/05/2019).

[44] Jürgen Schmidhuber. “Deep Learning in Neural Networks: An Overview”. In: *CoRR* abs/1404.7828 (2014). arXiv: 1404.7828. URL: http://arxiv.org/abs/1404.7828.

[45] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314. ISSN: 1435-568X.

DOI: 10.1007/BF02551274. URL: https://doi.org/10.1007/BF02551274.

[46] Balázs Csanád Csáji. "Approximation with Artificial Neural Networks". In: (2001).

[47] John A. Bullinaria. *Applications of Multi-Layer Perceptrons*. 2015. URL: http://www.cs.bham.ac.uk/~jxb/INC/l11.pdf (visited on 05/03/2019).

[48] François Chollet. *Deep Learning with Python*. 20 Baldwin Road, Shelter Island, New York: Manning, 2018.

[49] Rafael C Gonzalez. *Digital image processing*. eng. 4th ed. New York: Pearson, 2018. ISBN: 9781292223049.

[50] Andrej Karpathy. *Convolutional Neural Networks: Architectures, Convolution / Pooling Layers*. URL: https://cs231n.github.io/convolutional-networks/ (visited on 05/05/2019).

[51] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014).

[52] Torjus Haukom. "End-Systole and End-Diastole Frame Detection in Cardiac Ultrasound". 2018.

[53] Baichuan Yuan et al. "Machine learning for cardiac ultrasound time series data". In: *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*. Vol. 10137. International Society for Optics and Photonics. 2017, p. 101372D.

[54] Marius Staring, Stefan Klein, and Josien P. W. Pluim. "A rigidity penalty term for nonrigid registration". In: *Medical Physics* 34.11 (), pp. 4098–4108. DOI: 10.1118/1.2776236. eprint: https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1118/1.2776236. URL: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.2776236.

[55] WU Zufeng et al. "Medical Image Registration Using B-Spline Transform." In: *International Journal of Simulation–Systems, Science & Technology* 17.48 (2016).

[56] Julia A. Schnabel et al. "A Generic Framework for Non-rigid Registration Based on Non-uniform Multi-level Free-Form Deformations". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001*. Ed. by Wiro J. Niessen and Max A. Viergever. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 573–581.

[57] Gareth James et al. "Tree-Based Methods". In: *An Introduction to Statistical Learning: with Applications in R*. New York, NY: Springer New York, 2013, pp. 303–335. ISBN: 978-1-4614-7138-7. DOI: 10.1007/978-1-4614-7138-7_8. URL: https://doi.org/10.1007/978-1-4614-7138-7_8.

[58] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

A | IEEE IUS 2019 Abstract

Basal Strain Estimation in Transesophageal Echocardiography (TEE) using Deep Learning based Unsupervised Deformable Image Registration

Torjus Haukom[1], Erik Andreas Rye Berg[23], Gabriel Hanssen Kiss[24], [1]Department of Electronic Systems, NTNU, Trondheim, Norway, [2]Center for Innovative Ultrasound Solutions (CIUS), NTNU, Trondheim, Norway, [3]Clinic of Cardiology, St. Olavs hospital, Trondheim, Norway, [4]Operating Room of the Future, St. Olavs hospital, Trondheim, Norway

Background, Motivation and Objective

Major surgery and interventions may impact cardiac performance. As of today, per-operative monitoring is based on vital signs and clinical observations by the anesthesiologist. This, however, does not offer a complete monitoring of left ventricular function throughout the intervention. We hypothesize that functional monitoring of the heart can be performed automatically based on TEE images.

Statement of Contribution/Methods

Aim: compute the non-linear deformation between subsequent images in a TEE sequence of the left ventricle and estimate basal longitudinal strain to assess regional myocardial function via a deep learning approach.

An unsupervised approach based on a convolutional neural network was implemented (code available), similar to the work of De Vos et al. The output of the CNN network is a dense vector field that describes the non-linear deformation required to maximize the similarity (normalized cross correlation) between two images. A B-spline based smoothing function was implemented and optimized in order to regularize the deformation. Manually selected points on the basal segments can be tracked from end-diastole to end-systole and strain derived.

Recordings from 42 consecutive complete TEE exams from the Echocardiography Unit were anonymized and used for training. Recordings from 5 consecutive TEE exams performed during heart surgery, also anonymized, were used as test set and the frame order kept. All recordings were made using GE Vivid E95 or E9 systems with a 6T probe (GE Vingmed, Ultrasound, Horten, Norway). All recordings of a patient were captured within a limited time gap, and no patient selection was performed. The captures include 3 heart cycles of standard 4C, 2C, and LAX views.

Results/Discussion

For the test set patients, the basal strain was manually annotated in EchoPac by an expert echocardiographer. In this preliminary experiment, 19 heart cycles were randomly selected from the test set and checked to ensure visibility of the points to be tracked. Overall when estimating strain (Fig. 1), there was a mean difference of 7.25% (±4.56%).

This research is ongoing, and more tests will be performed with more data. Still, the point tracking is working as expected in most low noise scenarios, where the myocardium is well depicted. However, dropouts, noise generated by implants or air bubbles after surgery, confuse the tracker and drifting occurs.

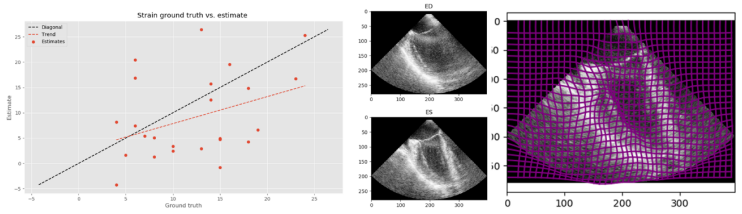


Fig. 1. Left: strain ground truth vs. deep learning estimates for 19 test heart cycles; center: sample end-diastolic and end-systolic TEE frames; Right: deformed grid overlaid over the end-systolic frame

B | B-spline interpolation

Given a digital image Ψ with width W and height H, it can be represented as a matrix with dimensions $W \times H$[1]. Let Ω be the image domain $\Omega = \{(x, y) : 0 \leq x < W, 0 \leq y < H\}$. Then, a continuous function approximating the image can be defined on Ω using B-spline interpolation, as given in Equation (B.1).

$$T(x,y) = \overbrace{\frac{1}{6}\begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}^T \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}}^{\vec{u}^T} \tilde{\Psi}_{i,j} \overbrace{\frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}}^{\vec{v}} \tag{B.1}$$

$$\tilde{\Psi}_{i,j} = \begin{bmatrix} \Psi_{i-1,j-1} & \Psi_{i-1,j} & \Psi_{i-1,j+1} & \Psi_{i-1,j+2} \\ \Psi_{i,j-1} & \Psi_{i,j} & \Psi_{i,j+1} & \Psi_{i,j+2} \\ \Psi_{i+1,j-1} & \Psi_{i+1,j} & \Psi_{i+1,j+1} & \Psi_{i+1,j+2} \\ \Psi_{i+2,j-1} & \Psi_{i+2,j} & \Psi_{i+2,j+1} & \Psi_{i+2,j+2} \end{bmatrix}$$

In Equation (B.1), $u = x - \lfloor x \rfloor$ and $v = y - \lfloor y \rfloor$ ($u, v \in [0, 1)$) are the relative x and y positions on the local spline, $i = \lfloor x \rfloor$ and $j = \lfloor y \rfloor$ are the pixel indices of the closest pixel to x and y in the original image that satisfy $i \leq x, \quad j \leq y$. $\tilde{\Psi}_{i,j}$ is the 4×4 neighborhood of pixels closest to (x, y), referred to as the B-spline control points. $\vec{u}$ and $\vec{v}$ contain the basis functions for the splines. We see that for $x = i$ and $y = j$, $u = v = 0$, and the resulting value is the original pixel value. Furthermore, it is clear that changing the brightness of one pixel only affects the neighborhood of this pixel in the interpolated image.

It can be shown that Equation (B.1) can be rewritten to a sum over a Hadamard product of two matrices as shown in Equation (B.2).

$$T(x,y) = \sum_{k=1}^{4}\sum_{l=1}^{4} \left[\left(\vec{u}\vec{v}^T\right) \circ \tilde{\Psi}_{i,j} \right]_{k,l} \tag{B.2}$$

Equation (B.2) looks a lot like a convolution. In fact, we can get the original image back by convolving Ψ with $\vec{u}\vec{v}^T$ by setting $u = v = 0$. But, more interestingly, we can also get the interpolated pixel values halfway between the original pixels by setting $u = v = 0.5$. Similarly, we can get values at a quarter of the way between the original pixels with

[1] This results in a transposed image. This is done for notational purposes.

$u = v = 0.25$ and so on. Thus, if we want to upsample an image by a factor N_{up}, we can construct expanded $\vec{u}$ and $\vec{v}$ vectors $\vec{u}_{int}$ and $\vec{v}_{int}$ by interleaving $\vec{u}$ and $\vec{v}$ vectors with N_{up} evenly spaced values on $[0, 1)$ for u and v. If we now insert $N_{up} - 1$ zeros between each pixel in the original image, this image can be convolved with $\vec{u}_{int}\vec{u}_{int}^T$ to form the upsampled image. This process of inserting zeros before convolution is known as fractionally strided convolution or transposed convolution.

C | Landmark tracking

Tables C.1, C.2, and C.3 contain the results from the visual inspection of the landmark tracking performed by the low downsampling rate, high downsampling rate and daisy-chained model, respectively.

Table C.1: Number of successful and unsuccessful landmark trackings for the low-downsampling-rate model.

	4-chamber			
	Lateral		Septal	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	8	61	28	42
Percent	11.59%	88.41%	40.00%	60.00%

	2-chamber			
	Anterior		Inferior	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	14	60	21	53
Percent	18.92%	81.08%	28.38%	71.62%

	LAX			
	Anteroseptal		Posterior	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	22	50	10	62
Percent	30.56%	69.44%	13.89%	86.11%

Table C.2: Number of successful and unsuccessful landmark trackings for the high-downsampling-rate model.

	4-chamber			
	Lateral		Septal	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	4	65	10	60
Percent	5.80%	94.20%	14.29%	85.71%

	2-chamber			
	Anterior		Inferior	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	5	69	4	70
Percent	6.76%	93.24%	5.41%	94.59%

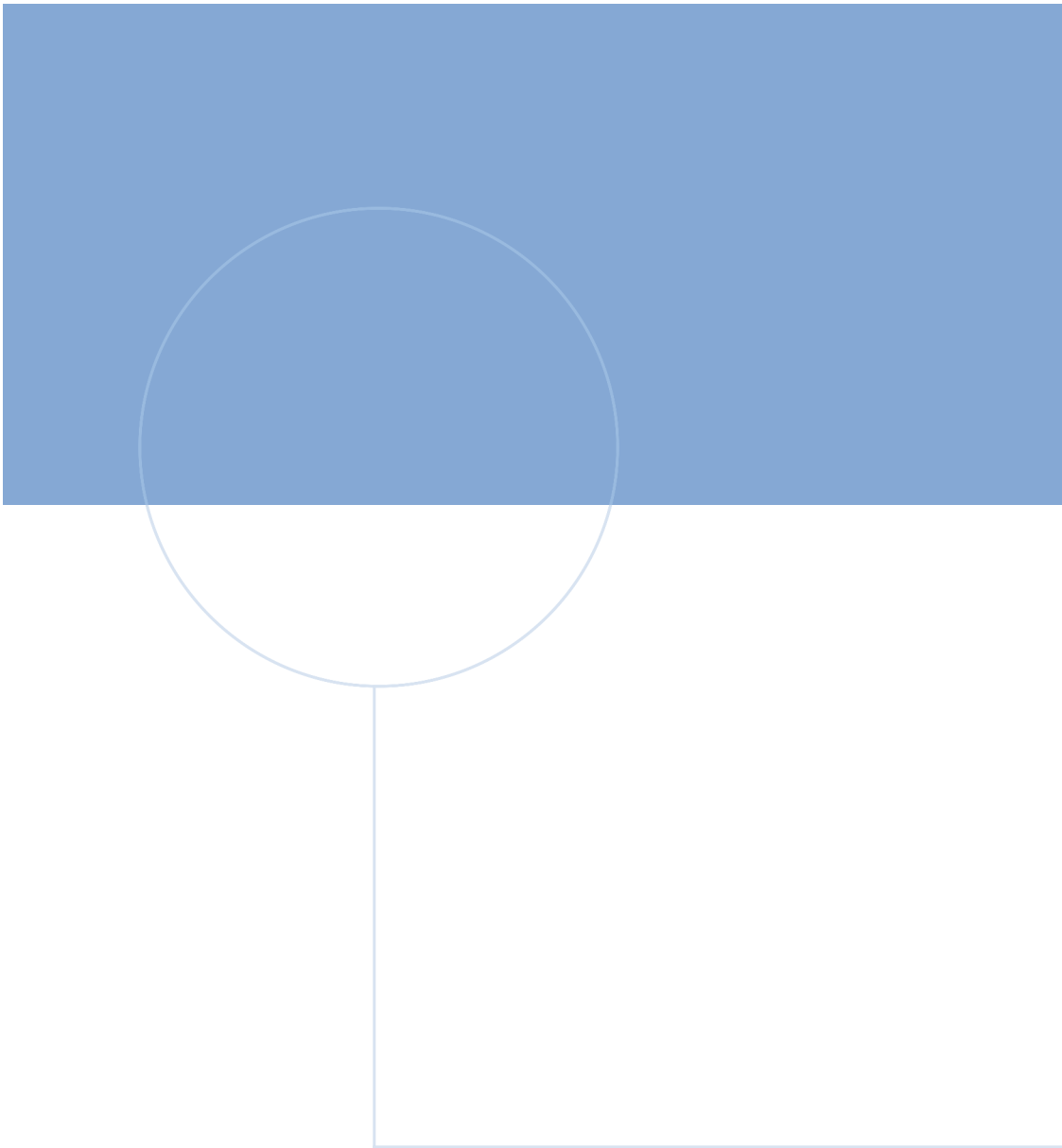
	LAX			
	Anteroseptal		Posterior	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	11	61	2	70
Percent	15.28%	84.72%	2.78%	97.22%

Table C.3: Number of successful and unsuccessful landmark trackings for the daisy-chained model.

	4-chamber			
	Lateral		Septal	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	18	51	48	22
Percent	26.09%	73.91%	68.57%	31.43%

	2-chamber			
	Anterior		Inferior	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	31	43	40	34
Percent	41.89%	58.11%	54.05%	45.95%

	LAX			
	Anteroseptal		Posterior	
	Successful	Unsuccessful	Successful	Unsuccessful
# of trackings	43	29	19	53
Percent	59.72%	40.28%	26.39%	73.61%



NTNU
Norwegian University of
Science and Technology