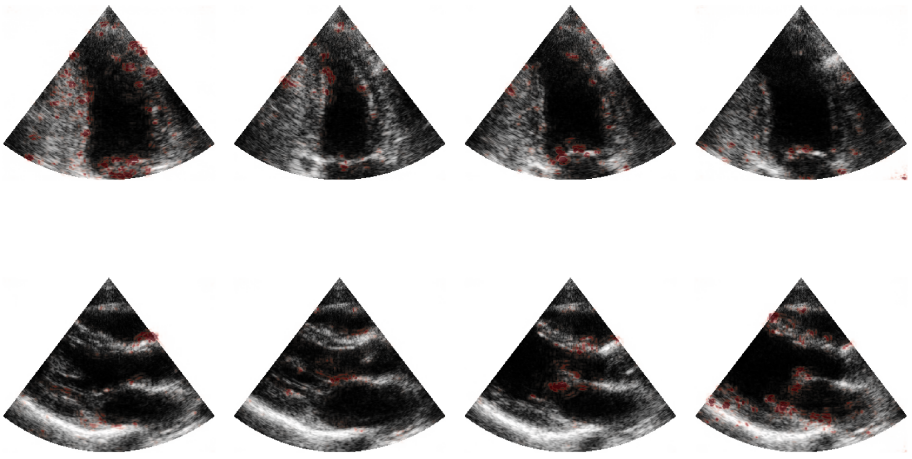
Master's thesis

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

Adrian Meidell Fiorito

Age Estimation from B-mode Echocardiography with 3D Convolutional Neural Networks

Master's thesis in Cybernetics and Robotics
Supervisor: Lasse Løvstakken
January 2019



NTNU
Norwegian University of Science and Technology

Adrian Meidell Fiorito

Age Estimation from B-mode Echocardiography with 3D Convolutional Neural Networks

Master's thesis in Cybernetics and Robotics
Supervisor: Lasse Løvstakken
January 2019

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

Abstract

Echocardiography is a noninvasive and safe imaging modality which uses ultrasound for assessment of the heart. As in many other fields, deep learning methods and convolutional neural networks (CNNs) in particular are being applied to tasks previously only performed by people. In comparison to many other fields, learning from echocardiography is difficult due to smaller data sets, higher noise levels and the complexity of the human heart. Methodologies for training CNNs in echocardiography are therefore typically based on results from other fields where more data is available. The effect is that domain-specific methodologies which might improve CNN performance have not been explored in depth.

The thesis is divided in two parts. First, a pipeline for automatic quality assurance of two-dimensional echocardiography is applied. Data quality is measured for each heart cycle using the output of CNNs trained for quantification in echocardiography. Samples with estimated poor quality are discarded.

Models are trained to predict the age from two-dimensional echocardiography. The benefit of this is that age is available for nearly all echocardiography studies, and the effects of aging on the heart is similar to several heart diseases. This means that methodologies for age estimation can be evaluated on almost all available echocardiography data, and that the acquired knowledge can be transferred to other approaches for automated quantification and disease detection. Heart cycles passing the automatic quality assurance are used. Several methodologies are attempted, including pretraining, optical flow, coordinate input channels and training with data from different standardized probe positions simultaneously.

Three studies are considered, one containing over a thousand patients from a normal population, and two data sets each containing more than two hundred patients with left ventricular dysfunction and coronary artery disease.

Visual inspection suggests that the prevalence of low quality data is higher among the discarded data, although objectively evaluating the automatic quality assurance step is difficult due to no available labels of data quality. More importantly, standardized data is generated containing heart cycles without requiring human intervention. The age estimation methodologies achieve at best a mean absolute error of 4.7 years per patient in the normal population. For comparison, linear regression using several clinical indices

achieves a mean absolute error of 7.2 years. The accuracy of the models are dependent on the distribution of ages in the training data, resulting in worse performance for data sets where ages are differently distributed. A model using optical flow input data performs most consistent on all data sets. Inspection of the optical flow model reveals that salient regions in the input cycles are known to be affected by aging. There are little or no observed difference between the estimates of healthy and diseased patients, suggesting that learned features are not affected by left ventricular dysfunction or coronary artery disease.

Sammendrag

Ekkokardiografi er en ikke-invasiv og sikker metode som bruker ultralyd for avbildning av hjertet. Som på mange andre felt, utfører metoder for dyp læring, spesifikt *Convolutional neural networks* (CNNs), oppgaver som tidligere kun ble oppnådd av mennesker. I forhold til flere andre felt er det vanskelig å lære fra ekkokardiografi på grunn av mindre datasett, høyere støynivåer og kompleksiteten til det menneskelige hjerte. Metodikker for trening av CNNs i ekkokardiografi er derfor vanligvis basert på resultater fra andre felt der mer data er tilgjengelig. Effekten er at domenespesifikke metoder som kan forbedre CNN-ytelsen, ikke er utforsket i detalj.

Avhandlingen er delt i to deler. Først blir det brukt en metode for automatisk kvalitetssikring av 2D ekkokardiografi. Datakvaliteten måles for hver hjertesyklus ved bruk av CNNs trent for kvantifisering i ekkokardiografi. Kvalitetsmålene brukes til å forkaste data som anslås å ha dårlig kvalitet.

Modeller er opplært til å estimere alder av mennesker i den vanlige befolkningen. Fordelen med dette er at alder er tilgjengelig for nesten alle ekkokardiografistudier, og virkningen av aldring i hjertet kan minne om flere hjertesykdommer. Dette betyr at metoder for aldersestimering kan evalueres på all tilgjengelige ekkokardiografidata, og at kunnskapen kan overføres til andre tilnærminger for automatisert kvantifisering og sykdomsdeteksjon. Hjertesykluser som passerer den automatiske kvalitetssikringen brukes til aldersestimering. Flere metoder blir forsøkt, inkludert pretraining, optical flow, egne inputkanaler med koordinater og trening med data fra forskjellige standardiserte probeposisjoner samtidig.

Tre datasett brukes, ett som inneholder over tusen pasienter fra en vanlig befolkning og to datasett som hver inneholder rundt to hundre pasienter med dysfunksjon i venstre ventrikkel og koronar hjertesykdom.

Visuell inspeksjon antyder at utbredelsen av data med lav kvalitet er høyere blant den forkastede dataen, men objektiv evaluering er vanskelig på grunn av manglende fasit for datakvalitet. Viktigere er det at standardiserte data blir generert som inneholder hjertesykluser av todimensjonal ekkokardiografi uten å kreve menneskelig inngrep. Aldersestimeringsmetodikkene oppnår i beste fall en gjennomsnittlig absolutt feil på 4.7 år per pasient i den vanlige befolkningen. Til sammenligning oppnår lineær regresjon med flere kliniske mål en gjennomsnittlig absolutt feil på 7.2 år. Modellens nøyaktighet er avhengig av

fordelingen av aldre i treningsdataen, noe som resulterer i dårligere ytelse for datasett hvor aldre er annerledes fordelt. En modell som bruker optical flow er mest konsistent på alle datasett. Inspeksjon av den optical flow-modellen viser at fremtredende områder i syklusene er regioner som er kjent for å bli påvirket av aldring. Det er liten eller ingen observert forskjell mellom estimatene for friske og syke pasienter, noe som tyder på at lærte egenskaper ikke påvirkes av dysfunksjon i venstre ventrikkel og koronar hjertesykdom.

Preface

Many thanks to my supervisor Lasse Løvstakken and co-supervisors Andreas Østvik and Erik Smistad at the Department of Circulation and Medical Imaging, NTNU, who came up with the idea of age estimation, gave access to data and computational resources, and have provided great guidance and feedback during the project and master thesis. A special thanks to Andreas for creating and presenting a poster of the project thesis at IUS 2018, as well as cooperation with writing the subsequent proceeding. Thanks to Amalie for great feedback and support throughout, as well as Shahrukh for proofreading. Finally, a thanks to my family, friends and fellow office mates for all the motivational support.

Table of Contents

List of Tables

List of Figures

Abbreviations

A2CH	=	Apical two chamber
A4CH	=	Apical four chamber
ALAX	=	Apical long axis
CAD	=	Coronary artery disease
CNN	=	Convolutional neural network
CVD	=	Cardiovascular disease
ECG	=	Electrocardiogram
GLS	=	Global longitudinal strain
LV	=	Left ventricle
MAPSE	=	Mitral annular plane systolic excursion
MAE	=	Mean absolute error
PLAX	=	Parasternal long axis
PSAX	=	Parasternal short axis
RV	=	Right ventricle
SGD	=	Stochastic gradient descent

1 Introduction

1.1 Motivation

Cardiovascular diseases (CVD) are the main cause of deaths in the world. In the United States, the cost of CVD was estimated to $330 billion dollars in 2013 [1]. In Norway approximately 515 000 people under the age of 74 consulted primary care due to CVD in 2016 [2]. One of the most important methods for assessing CVD and heart function in general is transthoractic echocardiography, further only referred to as echocardiography. Echocardiography is a noninvasive imaging modality that allows for safe and easy real-time imaging of the heart using ultrasound. To assist the echocardiographer, a variety of algorithms for computer processing and vision have been developed and applied. These algorithms allow for automated analysis and improved workflow for the practitioners.

Feature learning methods based on a hierarchy of transformations known as deep learning have had widespread success in computer vision in the recent years, and medical imaging is no exception. Disease detection with deep learning, especially using convolutional neural networks (CNN) have reached the level of specialists in multiple fields. Examples are detection of diabetic rhetinopathy in fundus imaging [3], skin cancer from dermoscopy and photographic images [4], pneumonia from X-rays [5], and arrhytmias in electrocardiogram (ECG) [6]. Deep learning has also been applied successfully to echocardiography in tasks such as segmentation of the myocardium [7], [8] and quantification of function and structure [8]–[10]. Detecting diseases automatically from echocardiography is difficult due to noise, limited data sets and variability in acquisition and pathologies. CNNs for disease detection in echocardiography therefore typically focus on diseases which can be detected with 2D CNNs using selected frames from the cardiac cycle and spatial features only. This includes CNNs for detection of hypertrophic cardiomyopathy, cardiac amyloid

and pulmonary arterial hypertension [8], all affecting the structure of the heart.

One of the most important factors for the success of deep learning is large amounts of available data. Although echocardiography data sets can be small and focus on a subset of diseases or measurements, patient information such as gender and age is present in most data sets. The effects of aging on the human heart shares many similarities to several types of heart disease, and aging is the dominant risk factor for the development of CVD [11]. By treating age as the dependent variable, more data can be used for training CNNs than what is available in single, specific data sets. The accuracy of models for age estimation can be used to evaluate the performance of methodologies aimed at performing automated quantification.

Every decade, the largest population based study of adults in Norway takes place in the North-Trøndelag area. The studies include echocardiography examinations of a sizable number of healthy patients, and are used to define normal ranges for clinical indices, divided into age group if necessary. A related idea is whether deep learning models trained to predict age on a healthy population data can be viewed as learning an estimate of "cardiovascular age", relative to the normal population. In this case, a CNN trained on healthy patients for estimating cardiovascular age might be used to define normal ranges for cardiovascular age. An estimated cardiovascular age relative to the chronological age outside normal ranges could warrant further inspection and/or lifestyle changes to improve cardiovascular health. One might also expect the difference between estimated and chronological age to increase for patients with signs of CVD related to aging, so called "unsuccessful" aging [12]. If the separation is large for healthy patients compared to unhealthy patients, the difference can in itself indicate disease. This method does not require explicitly labeled disease as training data, in contrast to supervised deep learning methods for disease detection.

Data cleaning and preprocessing is an important step for the quality of a deep learning model. An echocardiography exam can consist of several imaging modalities (e.g. M-mode, B-mode, Doppler imaging), taken from several probe postures (views). The duration, frame rate and image dimensions of each recording can also vary. Variations can also occur within a single recording, due to factors like changing acquisition settings and image plane. All of these effects makes deep learning a more challenging task. Standardization of the data is therefore beneficial to simplify the learning task. For many of the studies, data is unsufficiently labeled for standardizing all desired dimensions without human intervention. For example, the view, cycle phase and quality of a recording is usually not labeled, as an experienced echocardiographer can determine this by inspection.

Manual inspection of each recording is cumbersome in a machine learning setting, where data sets are large and manual inspection can have high variability for an inexperienced evaluator. Automating the standardization of echocardiography data sets will decrease the time required for performing deep learning experiments in echocardiography.

1.2 Contributions

The presented contributions are divided into two steps. First, a pipeline for automatically standardizing 2D echocardiography video is presented, denoted automatic quality assurance (QA). Secondly, methods for estimating the age directly from 2D echocardiography is presented and evaluated, using data extracted from automatic QA.

In automatic QA, recordings are divided into heart cycles, using accompanying electrocardiogram (ECG) measurements. The view of each cycle is classified by a CNN [13]. Measurements of the data quality for each cycle is generated using pretrained CNNs for quantification in echocardiography. Based on the distribution of quality measurements, thresholds are set for which cycles with worse estimated quality are discarded before being used for age estimation.

Given the rapid development in deep learning from video, the aim of proposed methods is to be compatible with the inevitable changes in state of the art deep learning models. Age is estimated from cycles using a state of the art CNN architecture known as I3D [14]. Compared to the standard 2D CNN, which only learns spatial features, the 3D CNN is also able to learn temporal features in all layers of transformation. This is well suited for the echocardiography domain, where not only still image features but also motion can provide useful information.

Parameters of the selected 3D CNN pretrained on photographic video is available. This allows evaluation of whether transfer learning from natural video to echocardiography is beneficial, or if the difference between domains is too large for successful transfer. To increase the ratio of training examples to the number of model parameters, training the models with several views simultaneously is attempted.

HUNT 3 [15], a fairly large data set in terms of echocardiography with 1266 patients is used for automatic QA and age estimation. Two newer data sets each containing approximately two hundred patients examined for left ventricular dysfunction (LVD) or coronary artery disease (CAD) are mainly used for testing.

1.3 Related Work

1.3.1 Automatic QA

As one of the most common methods for storing echocardiography data is in large databases, some level of preprocessing must usually take place. This involves conversion to formats more suited for machine learning. However, quality assessment has usually been performed by inspection [16] or not at all. CNNs for view classification have nearly reached levels of human annotators in the recent years [13], [17]. This allows for the use of view classification CNNs to replace human annotation as part of an automated pipeline. Østvik et. al [9] use a view classification CNN to extract apical four chamber frames. The confidence of the view classification network is used to evaluate whether the frame is applicable for further quantification. Smistad et al. [18] use a view classification network to acquire valid apical views for calculation of mitral annular systolic excursion (MAPSE). Zhang et. al [8] train CNNs for quantification, segmentation and disease classification trained using views classified by a CNN. The average confidence of the view classification network is also used to generate a measure of quality for each study, named the *View Probability Quality Score*. Abdi et. al [19] train a CNN for generating quality scores of apical four chamber end systolic frames. The model accurately learns quality as a number between 0 and 5 labeled by an expert echocardiographer. Although this method works well for the end-systolic apical four chamber view, further manual labeling is required to extend the method to other parts of the cycle or other views.

1.3.2 Age Estimation

Several calculators for "heart age" based on risk factors such as cholesterol, systolic blood pressure, smoking and diabetes have been made publicly available. Heart age prediction is typically a byproduct from tools intended to predict absolute risk of heart disease [20]. However, there is no universally agreed upon definition of the heart age concept. The heart age calculation based on the Framingham heart study [21] defines heart age as “the age of a person with the same predicted risk but with all other risk factor levels in normal ranges”. In countries such as the US, UK and New-Zealand the heart age is used for communicating the risk of heart disease [22]. However, the use of heart age is disputed, due to concerns of overmedication and difficulties in making treatment decisions based on heart age estimates.

Age estimation directly from echocardiography is a relatively unexplored concept. The medical ultrasound company *Quipu* provides an online vascular age calculator based on automated tracking in 2D ultrasound of the carotid intima media thickness [23]. Estimation is made by comparison to normal values in a healthy population. Age estimation from face images is a more common task, with applications in law enforcement. Deep learning methods have been proposed for this domain. Rothe et. al [24] collect 523,230 face images from IMDB and Wikipedia, along with corresponding age. Three problem formulations are proposed, regression, classification or expectation. Age expectation is given as a classification problem in $0, 1, ..., 100$ followed by a soft-max expected value operation over the 101 discrete outputs. This approach performs better than standard classification and slightly better than regression.

2 | Background

This section presents a brief introduction to the background material used throughout. Parts about the heart, echocardiography, deep learning and optical flow are adapted from the background section of the project thesis [25].

2.1 The Human Heart

2.1.1 Anatomy and Cardiac Cycle

The heart consists of four chambers. The upper two are named atria, and the bottom two are the ventricles. The chambers are surrounded by three layers of tissue, with the heart muscle known as myocardium being the largest. An illustration with chambers and valves indicated, together with the flow direction, is shown in Figure 2.1.

Venous, deoxygenated blood enters the right atrium. From there, it flows through the tricuspid valve into the right ventricle [27, Chapter 7]. Blood from the right ventricle exits through the pulmonary valve towards the lungs. Arterial blood from the lungs flow into the left atrium. From the left atrium, blood flows into the left ventricle through the mitral valve, and out through the aortic valve, to the aorta and into the body. This cycle, known as the cardiac cycle, is divided into two phases, diastole and systole. During diastole, higher pressure in the atria than the ventricles causes the mitral and tricuspid valves, together called the atrioventricular (AV) valves, to open. At the same time the aortic and pulmonary valves are closed, leading to filling and increased volume of the ventricles. In the first stage of the diastole, volume increases rapidly in the ventricles, before slowing down. In the second stage of the diastole the atria contract, leading to additional ventricular expansion before slowing down.

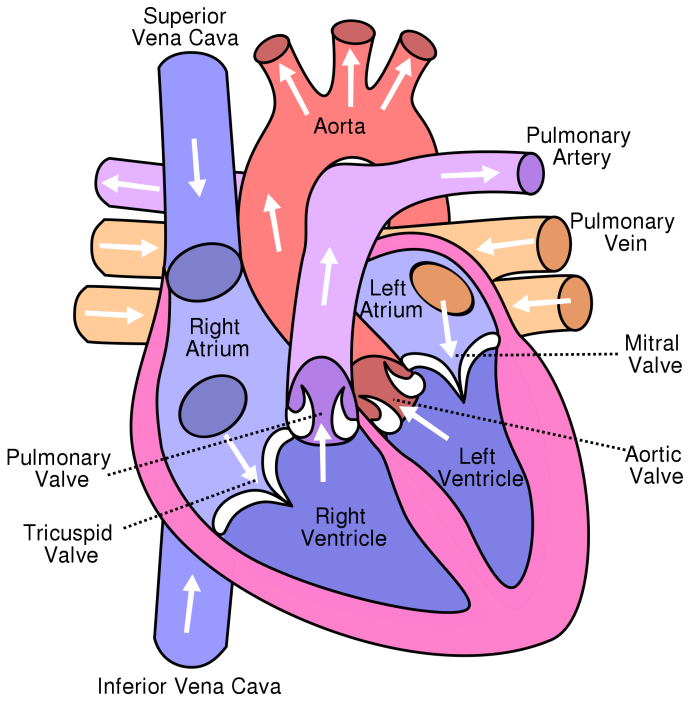


Figure 2.1: An illustration of the human heart [26].

The systole phase constitutes contraction of the ventricles. The contraction increases pressure rapidly in the ventricles, leading the AV valves to close. For a short while, the pressure in the aorta is still higher than in the left ventricle and the aortic and pulmonary valves are closed. This leads to a period of constant volume in the ventricles, known as isovolumetric contraction. When the pressure in the left ventricle is greater than that of the aorta, the aortic valve opens. This causes the left ventricular volume to rapidly decrease. Towards the end of the systole, the muscles in the ventricles relaxes, and the pressure in the ventricles decreases. A short while after this, the pressure from the aorta is much higher than the left ventricular pressure. This leads to the aortic valve closing. The AV valves are still closed at this point, as the pressure in the atria is lower than in the systole. This makes the volume remain the same even though the muscles are relaxing, named the isovolumetric relaxation. When the pressure in the ventricle decreases below that of the atria, the AV valves open and the cycle repeats.

2.1.2 The Effects of Aging

The structure and function of the heart changes with age in several ways. Changes at the cellular and molecular level are not easily visible using echocardiography, and are therefore ignored. The impact of aging on heart function is evident in the left ventricle during diastole [11]. The heart fills more slowly with age, in the early passive filling phase of the diastole. This is mainly caused by an increase in the isovolumic relaxation time [28]. At the age of 80 years, the reduction in LV filling rate is on average halved from that of the average 20 year old [29]. To compensate for reduced early filling, filling increases during the atrial contraction phase later in diastole. The difference can be seen as a decrease in the E/A ratio, which is the ratio of the filling rate at early diastole (E) to the filling rate during atrial contraction (A).

Systolic function is also affected during aging. The variability in heart rate between beats declines with age. There is on the other hand an increased prevalence of abnormal motion patterns by age, such as premature ventricular contractions [30]. Overall systolic function is unchanged at rest for healthy patients[28], as measured by the ejection fraction. The (left ventricular) ejection fraction is the difference in left ventricular volumes at end-diastole and end-systole, divided by the volume at end-diastole. During exercise, the overall function changes more noticeably. The maximum heart rate during exercise are both reduced with age. This impacts the ejection fraction as well as the cardiac reserve [29], which is the difference between the rate of pumping blood to the maximum capacity.

Several changes occur to the structure of the heart. Most noticeably, the thickness of the LV walls increase, and the prevalence of LV hypertrophy increases significantly with age [31]. A possible cause for this is reduced contractility and increased vascular stiffness during aging. However, the effects on total LV mass does not appear to change during aging for women, and either decreases or remains constant in men during aging [28]. Instead, cardiac muscle appears to be redistributed, with an asymmetrical increase near the interventricular septum compared to the free outer. The volume of the left ventricle decreases during aging [28], [32]. The left atrium also changes structure during aging, as the decrease in E/A ratio can cause atrial hypertrophy and enlargement.

2.1.3 Coronary Artery Disease

Coronary artery disease (CAD) is the build up of plaque inside the coronary arteries, supplying blood to the cardiac muscle. The plaque limits blood flow, which can cause blood

clots that further reduces the flow. This can lead to irreversible damage of the heart muscle (myocardial infarction). In addition, CAD can weaken the heart muscles (myopathy) and cause heart failure over time. Cardiomyopathy can cause a variety of changes to the heart, such as hypertrophy, enlarged or stiffened ventricles. Other effects of CAD are abnormal heart motion patterns (arrhytmias). Early diagnosis and treatment can significantly reduce the impact of CAD. Several procedures are available for diagnosis, including echocardiogram and coronary angiogram, the use of contrast dye and X-ray for imaging.

2.1.4 Left Ventricular Dysfunction

Left ventricular dysfunction (LVD) is a general term for several disorders involved in dysfunction of the LV and can be described as reduced ability of the LV to provide blood. It can be divided into systolic and diastolic dysfunction. Systolic dysfunction, or heart failure with reduced ejection fraction, can be defined as reduction of the ejection fraction below 50% [16]. Diastolic dysfunction, or heart failure with preserved ejection fraction, is dysfunction in the filling phase while the ejection fraction is above 50%. Diastolic dysfunction is typically caused by impaired LV relaxation and increased heart muscle stiffness. Other indicators than ejection fraction must therefore be used to determine diastolic dysfunction, especially markers of increased pressures over the left ventricle during filling [33]. The prevalence of diastolic dysfunction increases with age due to factors like vascular stiffening and increased left ventricular wall thickness.

2.2 Echocardiography

Echocardiography is a non-invasive method to study the heart using ultrasound. The method uses a probe known as the transducer which produces pressure waves, normally with frequencies between 1.5 - 10 MHz for a short period [34, Chapter 1]. The transducer then acquires the returning echoes. The velocity of sound in tissue is nearly constant, and by measuring the duration between transmit and receive, the distance to the reflection points is estimated.

To prevent previous transmitted pulses from affecting the current acquirement, a delay must be placed between the transmitted pulses. The delay is dependent on the depth of the object being imaged. The lower the frequencies, the deeper the ultrasound beam is able to penetrate. Spatial resolution increases with frequencies as objects smaller than

approximately half of the wave-length are not imaged. This results in a trade off between the sampling rate, resolution and maximal depth of an echocardiography recording.

The wavefront can be focused straight forward for a while before spreading, such that a majority of the reflecting objects coincide along the center-line of the wavefront. The position of the reflecting object can then assumed to be in the direction of the wave. When traveling through the body, the energy of the waves decrease due to absorption and reflection. Clearest reflections are produced when the waves travel between mediums of different density with a well defined boundary, such as the interface of tissue and blood.

Not all energy is reflected directly back to the transducer. The more well defined the reflection boundary is, the closer the reflection is to being specular. This means that the more perpendicular the wave front is to the reflecting surface, the less energy is reflected directly back to the transducer. In regions with more homogeneous tissue, the waves tends to scatter in several directions. Some of the energy reflect off multiple scatterers before reaching the transducer. This increases the time taken for the wave to reach the scanner, resulting in artifacts in the acquisition. Other common artifacts include the sound wave bending or reflecting sideways from the central beam, known as side lobes. Waves can also reflect multiple times back and forth in echo chambers, known as reverberations.

2.2.1 Views

Sound waves are rapidly dampened when traveling through bone and air. To circumvent this, echocardiograms are generally obtained through acoustic windows between bones. From these windows, different image planes through the heart, called views, can be generated by positioning the transducer appropriately. Standard views from the apical and parasternal windows are considered further.

The apical window is located at the apex (bottom) of the left ventricle. In the apical 4-chamber (A4CH) view, the ventricles can be seen in the top of the image, and the atria on the bottom. The right ventricle and atrium is in the left of the image, and left ventricle and atrium on the right. By rotating the transducer approximately 60° counter-clockwise from the A4CH view, the apical 2-chamber (A2CH) view is obtained. In A2CH, the left ventricle and atrium is seen, as well as the anterior and inferior walls. Further clockwise rotation of the transducer yields the apical long-axis view (ALAX), also known as the apical 3-chamber view. In addition to showing the left ventricle and atrium, the aortic valve and aortic root is visible.

The parasternal view is adjacent to the left side of the sternum, near the 4th or 5th inter-

costal space between the ribs. In the parasternal long axis view (PLAX), the left and right ventricle, left atrium, mitral and aortic valves, and the aortic root is seen. The parasternal short axis views (PSAX) are obtained by rotating the transducer 90° clockwise from PLAX. Different PSAX levels can be acquired by tilting the probe, as it is typically divided into three levels. In the aortic level, the aortic valve is seen in the middle, with the RV, RA, LA, PV and TV surrounding the aorta. On the mitral valve level, the right ventricle is still seen, and the mitral valve is in the center of the image. The midpapillary level shows the left and right ventricle, as well as the interventricular septum and the papillary muscles.

It is important to note that these are general categories of some views, and that variability exists within a single view. Differences can occur depending on which structure is imaged in the view, for example, by focusing on the right or left chamber versus the right. Modifications can also be made to work around to patient-specific difficulties [34].

2.2.2 Modes

There are multiple ways to gather and visualize the measurements. In M-mode, a wavefront with constant direction is imaged over time. The envelope of one reflected pulse yields a single scan-line of reflection amplitude at different depths. This is repeated in time, generating an image where the vertical axis represents depth and the horizontal axis is time. The benefit of M-mode is high frame rates. To instead achieve a 2D cross-section, scan lines are sampled in the plane in rapid succession. The scan-lines are sampled in rapid succession by sweeping the angle of the wavefront. The resulting scan-lines are spatially interpolated, and a 2D sector image is generated, with depth along one axis and the other axis in the plane of the wavefront sweep. The envelope of the reflecting echoes determines the brightness of the pixels in the image, known as brightness mode (B-mode). Resolution decreases with depth, due to the spread between each scan-line. In general, the resolution is higher in the axial direction than in the lateral direction, which generates non-square pixels. Other imaging modalities include 3D echocardiography, and Doppler imaging which uses the phase shift of generated waves to measure the velocity of blood or tissue.

2.3 Optical Flow

When objects in the real world moves relative to an imaging device, the projection of the objects on the image sensor also moves. This results in movement of the brightness patterns in the images. Optical flow is the apparent motion of these brightness patterns between consecutive images. This yields a vector field, where the optical flow vector $\mathbf{u}(\mathbf{x}, t)$ describes the displacement of the brightness pattern in consecutive images at pixel index $\mathbf{x}$. The estimated movement of certain key points is denoted the sparse optical flow. When motion is estimated for pixel, the result is denoted dense optical flow. Further, only two-frame dense optical flow in 2D is considered. Two consecutive frames are denoted $I_0(\mathbf{x})$, $I_1(\mathbf{x})$, where $\mathbf{x}$ are the image coordinates in the image space $\Omega \subseteq \mathbb{R}^2$ and $\mathbf{u} \in \mathbb{R}^2$. To calculate optical flow, assumptions must be made about the movement of the brightness patterns. The most common assumption is the *brightness constancy* assumption [35]. By assuming that the brightness patterns are constant during motion, the following holds:

$$I_0(\mathbf{x}) = I_1(\mathbf{x} + \mathbf{u}(\mathbf{x})). \tag{2.1}$$

Under the additional assumption that the motion field is smooth, a first order Taylor approximation can be made around some nearby point $\mathbf{u}_0$:

$$I_1(\mathbf{x} + \mathbf{u}) \approx I_1(\mathbf{x} + \mathbf{u}_0) + \langle \nabla I_1(\mathbf{x} + \mathbf{u}_o), \mathbf{u} - \mathbf{u}_0) \rangle, \tag{2.2}$$

where $\langle \cdot, \cdot \rangle$is the inner product. These assumptions result in an ill-posed problem, as there can be multiple or no points with the same intensities.

2.3.1 Variational Approach

One method for finding a solution is to formulate the task as an minimization problem, where the deviations from the assumptions are to be minimized. By minimizing a global loss function over all pixels, a dense flow field can be acquired. Additional regularizing terms can also be included in the loss function to introduce effects such as smoothing of the flow fields. Thus, the flow field is a minimizer of

$$min_{\vec{u}} \int_{\Omega} \{\lambda \phi(I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))) + \psi(\mathbf{u}(\mathbf{x}), \nabla \mathbf{u}(\mathbf{x}), ...)\} \, d\mathbf{x}, \tag{2.3}$$

where $\phi(\cdot)$ penalizes the deviation from brightness constancy, $\psi(\cdot)$ is a regularization term, and λ a weighting between the two terms [36].

2.3.2 Multiscale Approaches and Warping

The loss function is usually non-convex and highly nonlinear. In order minimize this function efficiently, a possible solution is linearization of the cost functional. The issue with linearization methods is that they are only accurate in a neighborhood of the linearization point. Therefore, the estimated flow will be inaccurate for large motions. To overcome this issue, optical flow can be calculated and combined at multiple scales. When downsampling, aliasing patterns can occur due to undersampling of high frequency content. Therefore, low-pass filtering is commonly applied before each downsampling step. The set of images at different scales is called an image pyramid.

A related approach is known as warping [37], [38]. Here, assuming an initial estimate of the flow field $\mathbf{u}_0$, iterative calculations of an increment $d\mathbf{u}$ to the flow estimate are found. Starting with $I_0(\mathbf{x})$ and $I_1((\mathbf{x})$, a first flow increment $d\mathbf{u}_0$ is found. The first flow estimate is then $\mathbf{u}_1 = \mathbf{u}_0 + d\mathbf{u}_0$, where $\mathbf{u}_0$ is commonly assumed to be zero. $\mathbf{u}_1$ is then used to warp one of the images towards the other image.

At the next iteration, optical flow is calculated between the first image and the warped second image. By repeatedly performing the warping operation, the differences between the first and the warped second image becomes smaller, such that smaller flow increments can be calculated at each step. In [39], it is shown that the warping step is equivalent to solving the non-linearized constancy equations.

When warping is combined with image pyramids, the algorithm is known as a coarse-to-fine approach. Here, large motions are found at the coarsest scales, between the maximally downsampled images. The optical flow estimate is then upsampled to the next scale and used as an initial flow estimate for that scale. As a result, large motions are detected at coarse scales, while smaller motions are detected at the finer scales.

2.4 Deep Learning

2.4.1 Supervised Deep Learning

Supervised learning is the task of learning a mapping between inputs and outputs from examples. Inputs are often called data, samples or x, and corresponding outputs are known as targets, labels or y. Formally, the goal of supervised learning is to find values for the parameters θ of the model $f(x;\theta)$, such that the output $\hat{y} = f(x;\theta)$ approximates the relationship between x and y. For supervised learning to be applicable, the trained model should approximate the input-output relationship for unseen data, and perform well on relevant tasks. Inputs and outputs can be in arbitrary dimension, and dimensionality will further be described as the shape. Shape is denoted as $dim_1 \times dim_2 \times ... \times dim_n$. For example, video data has the shape $length \times height \times width \times channels]$, where the channel axis are omitted when it is implicit.

Mathematical operations of a model are commonly arranged sequentially in layers. When the model consists of many layers, the task can be denoted deep learning. Some of the operations have parameters, such as matrix multiplication, while others are parameter-free, such as the sigmoid activation function. A separation is made between trainable parameters of a model (weights) and the parameters describing the set up of the model and learning task (hyperparameters). Parameters or weights refers to the values inside the model that are learned.

Loss Functions

The learning task is framed as an minimization problem over the loss function J

$$\arg\min_{\theta} J(y, f(x;\theta)). \tag{2.4}$$

Which loss function to minimize depends on the task at hand. When the output y is in $\mathbb{R}^1$ a common loss function is the mean absolute error (MAE)

$$J_{MAE}(y, \hat{y}) = \frac{1}{N}\sum_{n=1}^{N} |y_n - \hat{y}_n|. \tag{2.5}$$

Here, N is the number of samples, y_n is the target for sample n and $\hat{y}_n$ is the model prediction for sample n, i.e. $\hat{y}_n = f(x_n;\theta)$. The smallest loss occurs when $y_n = \hat{y}_n$ for

all n.

Gradient Based Optimization

Deep learning models are typically nonlinear and only piecewise differentiable, making optimization non-convex. This means that the learning task is not guaranteed to find optimal parameters for the samples. Additionally, deep learning models can consist of millions of parameters, such that using global or high order solvers is intractable for most problems. For these reasons, most deep learning algorithms use gradient based optimization.

Gradient based optimization depends on the gradient of a function pointing in the direction of steepest ascent. When the loss is nonzero and differentiable in a region around the current parameters, the loss is non-increasing along the direction opposite of the gradient, for a small step size in parameter change. Evaluating the gradient of the loss function with respect to the model parameters yields a direction in parameter space along which the loss generally decreases. The gradient descent rule for updating parameters can then be formulated,

$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta J(y, f(x; \theta)). \tag{2.6}$$

θ_k is the value of θ after k iterations of gradient descent. The hyperparameter α is the learning rate, controlling the magnitude of the weight update.

Calculating $\nabla_\theta J$ of a large model is feasible when each layer used in the model has a known derivative. The functions in the model can be considered nodes in a computational graph. This allows for finding the gradient as a sequence of repeated applications of the chain rule. Gradients with respect to parameters in a layer are found by traversing the graph backwards from the loss to the layer. The algorithm for doing this is called back-propagation.

The loss should be minimized over all possible data generated from the considered process. For most real world tasks, only a limited number of samples are available. Under the assumption that the data set represents the underlying process fairly well, the true gradient over the distribution can be approximated by calculating it on the data set. Evaluating the gradient over the entire data set for each update is too computationally expensive for most applications. In practice one calculates the gradient using a subset of the data for each iteration. This is known as stochastic gradient descent (SGD), and the number of samples used per update is called the batch size. Increasing the batch size will produce a more accurate estimate of the true gradient, but for non-convex problems, noisy gradient estimates can allow the training algorithm to step out of a local minima.

A common modification to SGD is to change the gradient update to a weighted sum of the previous update and the current gradient, denoted momentum. The updated SGD formulation becomes

$$\begin{aligned} v_{k+1} &= \beta v_k + \alpha \nabla_\theta J(y, f(x;\theta)), \\ \theta_{k+1} &= \theta_k - v_{k+1}. \end{aligned} \tag{2.7}$$

Here, β is the momentum coefficient.

Overfitting and Regularization

If a model is accurate on data used for training, while performing worse on new, unseen data, the model is said to be overfitting. The more parameters a model contains the more likely it is to overfit, as the model has more parameters that can be used to recognize each sample in the training data. A model might base its predictions on some uncorrelated feature in the samples of the data set, which is not present in general. In the context of ultrasound imaging, a model might learn to recognize an image artifact, and memorize the target output for the sample containing the artifact. On the other hand, if a model has not learned the relationship from input to output well, the model is underfitting. This can occur if a model has too few parameters to represent the desired mapping, or if the parameters generated by the learning process are not good enough.

Selecting suitable hyperparameters for a deep learning model is difficult. Problems are often high dimensional and models consists of multiple layers each with their own hyperparameters. Therefore, a good choice is often to use a larger model than necessary and enforce constraints on the parameters using regularization.

Regularization is a collective term for methods that increase the generalizability of the model. Regularization often makes the model perform worse on the data set used for training, while retaining accuracy on unseen data. Instead of only minimizing loss as a function of target and predicted values, an extra term to the loss function,

$$J(\theta; X, y)_{total} = J(\theta; X, y) + \sum_k \alpha^{(k)} \Omega^{(k)}(\theta^{(k)}). \tag{2.8}$$

Here, $\Omega^{(k)}$ is a regularization loss penalizing the parameters $\theta^{(k)}$ in layer k, and $\alpha^{(k)}$ weights the regularization loss for this layer. The most common way to perform regularization for deep learning models is to penalize the norms of the parameters in the model.

Most common is using the L^2 norm, called L^2 regularization:

$$\Omega^{(k)}(\theta^{(k)}) = ||\theta^{(k)}||_2 = \sqrt{\sum_i {\theta_i^{(k)}}^2} \tag{2.9}$$

Due to large weights being heavily penalized by the square term, the parameters of the model are pushed closer to zero. For computational efficiency, taking the square root in the L^2-norm is usually omitted.

The models should not be evaluated on the same data that is used for training. To evaluate the generalizability of a model, a portion of the data set can be held out from the learning process until the final evaluation is performed. One should also avoid evaluating the model on the test data when searching for model hyperparameters. This will increase the risk of overfitting, and result in overestimation of the performance on unseen data. To evaluate how well a model is performing during training, a third part of the data set can be left out for evaluation and validating hyperparameters.

2.4.2 Deep Learning Layers

Activation Functions

An activation function is a nonlinear function designed to threshold the input, only keeping parts of its domain unchanged. Activation functions are typically applied to each scalar in the input. Sigmoidal functions are typical activation functions. The disadvantage of sigmoidal functions is that the gradient approaches zero for large negative and positive inputs. In these flat areas of the sigmoid curve the derivative of the sigmoid function approaches zero. This means that the gradient can become diminishingly small for earlier layers in the model, and a gradient update does not modify the model noticeably. This is called the vanishing gradient problem and makes training the networks harder. A simple activation function avoiding the vanishing gradient problem is the rectified linear unit (ReLU),

$$ReLU(x) = max(0, x). \tag{2.10}$$

Convolutional Layers

Convolutional layers are important operations to consider when locally connected data points are correlated, such as images. The output of the convolutional layer at a position is a combination of a few samples in the input. A model with several convolutional layers as the main component is denoted a CNN.

The convolutional operation can be applied to inputs of arbitrary dimensionality. The simplest form is 1D convolution. For 1D convolution, the operation will consist of sliding matrices called kernels along one axis in the 2D input. An explanation to why the N-dimensional operates on (N+1) dimensional inputs, is that convolutional layers are commonly applied to images, where it is often implicit that the data has an additional channel dimension (e.g. RGB). Consider an input x of shape $W_X \times C$ and convolutional kernel K of shape $W_K \times C$. The formula for 1D-convolution can be written as

$$y_i = \sum_{w=1}^{W_K} \sum_{c=1}^{C} x_{i+w,\,c} \cdot K_{w,\,c}. \tag{2.11}$$

The output of a convolutional operation is referred to as a feature map. Notice from Equation 2.11 that the output at index i is the sum of the input at all channels. This results in the operation removing the channel axis. By performing the convolutional operation with different kernels, a set of feature maps are acquired. Each feature map is stacked, such that the output of a convolutional layers has number of channels equal to the number of kernels in the layer.

Most deep learning software also center the cross-correlation. For simplicity this is omitted in the equations, but it is important to know when the operation is centered because it leads to non-causal filtering if one of the dimensions represents time.

In order for the convolutional operation to be valid, the kernel must fully overlap with the input. For data points at the borders of the input there are two options. One is to only perform convolution where the kernel and input is overlapping. This reduces the input data by $W_K - 1$ in each dimension, where W_K is the width of the kernel. The other option is to pad the input data at the borders such that the shape of the output is equal to the input shape.

It is also possible have larger strides for a convolutional layer. A stride of s means that the convolutional kernel skips s steps in the input for every output, as in Equation 2.12

$$y_i = \sum_{w=1}^{W_K} \sum_{c=1}^{C} x_{i \cdot s+w,\, c} \cdot K_{w,\, c}. \tag{2.12}$$

This is a common way to downsample the data. The spatial extent of the original input affecting a convolutional layer is called the field of view of the layer. Striding is an efficient way to increase the field of view of later layers.

Extending the convolution to an arbitrary dimensionality is trivial. 3D convolution operates on four dimensional data of shape $L_X \times H_X \times W_X \times C$ and slides the 4D-convolutional kernel K of shape $L_K \times H_K \times W_K \times C]$ along three axes in the input. This is often applied to video data and to volumetric data, where the fourth axis represents either another spatial dimension or time. The noncentered formula with unit strides is

$$y_{i,j,k} = \sum_{l=1}^{L_K} \sum_{h=1}^{H_K} \sum_{w=1}^{W_K} \sum_{c=1}^{C} x_{i+l,\, j+h,\, k+w,\, c} \cdot K_{l,\, h,\, w,\, c}. \tag{2.13}$$

Convolution is a linear transformation, such that repeating the convolutional operation still results in a linear transformation. In order to approximate more complex functions, convolutional layers are also followed by an activation function.

Pooling Layers

A method for downsampling the data is to use pooling layers. Pooling is similar to convolutional layers as a function is slid along the axes of the input. The difference is that the pooling function typically does not have learnable weights and the main purpose of pooling is downsampling. Similar to convolutional layers, pooling is usually centered, can be extended to inputs with any number of axes, and padding can be performed along the borders. Most common is the max-pooling operation. For 2D input data of shape $W_x \times C$, with a stride of s and width of W_p, 1D max-pooling can be written as

$$y_{i,c} = \max\{x_{i \cdot s+w,\, c}\}_{w=1}^{W_p} \tag{2.14}$$

The advantage of max pooling is that the largest inputs are kept, while weaker activations are discarded. This enhances the position invariance of a model with convolutional layers followed by pooling. For example, if one convolutional kernel implements an edge detection filter, performing max pooling on the feature map zeroes pixels in the output containing less edges than neighbouring pixels.

Another commonly used method is average pooling. This returns an average of inputs, which is equivalent to strided convolution with a box filter.

Dropout

Dropout is a layer used for regularization. Dropout intends to approximate the averaging all possible parameters of a model [40]. During training, each unit in the input to the dropout layer has the probability p of being set to zero,

$$\begin{aligned} r_i &\sim \text{Bernoulli}(p) \\ y_i &= r_i \cdot x_i \end{aligned} \tag{2.15}$$

At test time, the output is scaled by p.

$$y_i = p \cdot x_i \tag{2.16}$$

The effect of dropout is that models must learn a mapping from input to output even when some features are missing.

Batch Normalization

When a layer is updated, the distribution of its outputs changes. This makes training deep neural networks difficult, because as the distribution of inputs for the next layer changes, the next layer must adapt to the new distribution. To address this problem, the Batch Normalization (BN) layer normalizes the data to zero mean and unit variance [41]. Each dimension of the input is normalized individually. During training, normalization is performed by estimating mean and variance on each data batch. This is given by

$$\begin{aligned} \mu_{\mathcal{B}} &= \frac{1}{m}\sum_{i=1}^{m} x_i, \\ \sigma_{\mathcal{B}}^2 &= \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2, \\ \hat{x}_i &= \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}, \\ y_i &= \gamma\hat{x}_i + \beta. \end{aligned} \tag{2.17}$$

Here, x_i is the scalar i-th input for some layer of the network from the mini-batch $\mathcal{B} = \{x_{1...m}\}$, and y_i is element i in the output. As can be seen in Equation 2.17, inputs are passed through an affine transformation defined by γ, β. The affine transform is added such that the BN layer can represent the unit transformation. The parameter ϵ is added to avoid singularities if the variance $\sigma^2_{\mathcal{B}}$ is small.

When the model is used at test time, the batch normalization layer is modified to use estimates of mean and variance over the training set. This is shown in Equation 2.18. $E[\cdot]$ takes the the expected value.

$$\begin{aligned} E[x] &= E_{\mathcal{B}}[\mu_{\mathcal{B}}] \\ Var[x] &= \frac{m}{m-1} E_{\mathcal{B}}[\sigma^2_{\mathcal{B}}] \\ y &= \frac{\gamma}{\sqrt{Var[x] + \epsilon}} \cdot x + (\beta - \frac{\gamma E[x]}{\sqrt{Var[x] + \epsilon}}) \end{aligned} \tag{2.18}$$

2.4.3 Inception Blocks and Inception-v1

When designing a CNN, the kernel size of convolutional layers must be determined. Instead of at each layer setting only one filter size, [42] propose to apply several convolutional layers of different sizes in parallel. 2D convolutional layers of size 1×1, 3×3 and 5×5, in addition to a Max pooling layer with a kernel of 3×3 and unit strides are all applied to the same input. The feature maps are concatenated into a single output with an increased number of channels. To be able to concatenate the feature maps, padding is applied at the borders to keep the feature map size constant.

The large number of feature maps in the output results in an increasing number of floating point operations (flops) at the next layer. Therefore, before the 3×3 and 5×5 convolutions, and after the Max pooling layer, 1×1 convolutions with fewer output feature maps than the input is applied. The result is a substantial decrease in number of flops, which is important because image recognition models can contain millions of parameters. The resulting operation is named the Inception block (Figure 2.2). The network consisting of Inception blocks was denoted GoogLeNet/Inception v1. Several variations of the original Inception network were later proposed.

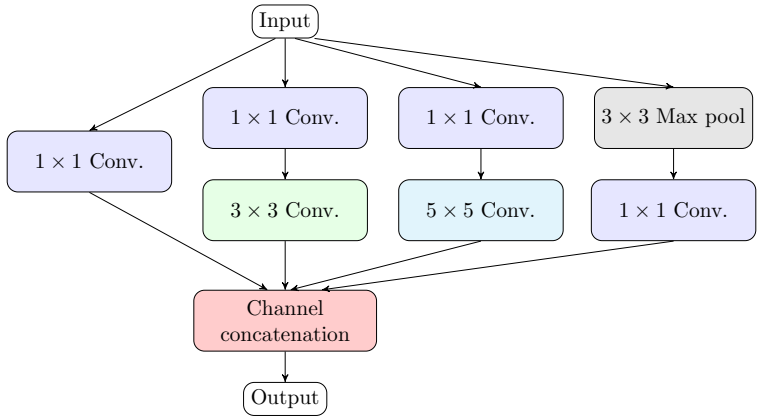


Figure 2.2: Inception-block as defined in [42]

2.5 Statistical Evaluation

This section describes plots and measurements used for evaluation in later chapters.

2.5.1 Coefficient of Determination

The coefficient of determination, also known as R^2, is a measure of how much of the variance in the dependent variable is described by the predictions. It is an indicator of how well a model has learned the distribution of the data. The coefficient of determination is defined as

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2},$$

where $\bar{y}$ is the mean value of y. An R^2 score of one corresponds to zero error, and all variance of the data is explained. On the other hand, when predicting the mean of the dependent variable for all samples, i.e. $\hat{y}_n = \bar{y}_n \forall n \in N$, R^2 becomes zero. This means that an R^2 score of zero corresponds to performing equal to random guessing based on the prior distribution without the use of any predictors. A negative R^2 score could even occur, for example due a large bias between measurements and predictions.

2.5.2 Pearson Correlation Coefficient

Pearson correlation (Pearson's r) measures the linear relationship between two variables. It is given by

$$r = \frac{\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_{n=1}^{N}(x_n - \bar{x})^2}\sqrt{\sum_{n=1}^{N}(y_n - \bar{y})^2}} \tag{2.19}$$

This is between $[-1, 1]$, where 1 represents a perfect positive linear relationship, -1 represents a perfect negative linear relationship, and 0 represents no linear increase.

2.5.3 Bland-Altman Plot

The Bland-Altman plot [43] is common in medicine and biology, where it is used to visually assess the agreement between two measurements as a function of magnitude. The horizontal axis shows the mean of the two variables, and the vertical axis shows the difference between the two variables. When one of the variables is a reference value, one might consider replacing the mean value with the reference on the horizontal axis. However [44] show that this can lead to a correlation between the difference and the magnitude, even when there is no correlation.

3 | Data Sets

Three different data sets are considered, containing patients in rest. All data generated from the scanners is stored in the Digital Imaging and Communications in Medicine (DICOM) format, a standard for storage of medical images. B-mode scan-lines are stored in the $\theta \times d$ space, where the θ axis is the angle of the beam to the transducer head, and d is depth. Custom software developed at the Department of Circulation and Medical Imaging, NTNU, is used to load the data in Python. Linear interpolation with *RegularGridInterpolator* from the scipy-package [45] is used to transform from $\theta \times d$ to spatial $d \times w$ coordinates. The DICOM files contain several attributes (tags) in addition to the pixel data. This includes the patient ID and acquisition settings such as sector size, sample times, and ECG recordings. Labels and measurements that are not recorded on the scanners is stored either in comma separated value (csv) format or in Excel-files and exported to csv. This includes information such as patient data, echocardiographic indices and disease status.

3.1 The Nord-Trøndelag Health Study 3 (HUNT 3)

HUNT 3 is a population based study, containing adults from Nord Trøndelag in Norway. Out of 93210 invited, 49827 participated in the study [46]. Randomly selected participants without known cardiovascular disease, diabetes or hypertension were invited for echocardiography examinations, which 1296 consented to participate to. For 30 of these participants significant pathologies were found, resulting in exclusion and a total of 1266 examinations of healthy patients. Acquisitions were made between 2006-2008.

One experienced echocardiographer performed the examinations. A Vivid 7 scanner (GE Vingmed Ultrasound, Horten, Norway) were used. All patients are examined during quiet respiration in the left lateral position. Examinations consist of parasternal long- and short axis views, as well as three apical views (two-chamber, four-chamber, apical long axis).

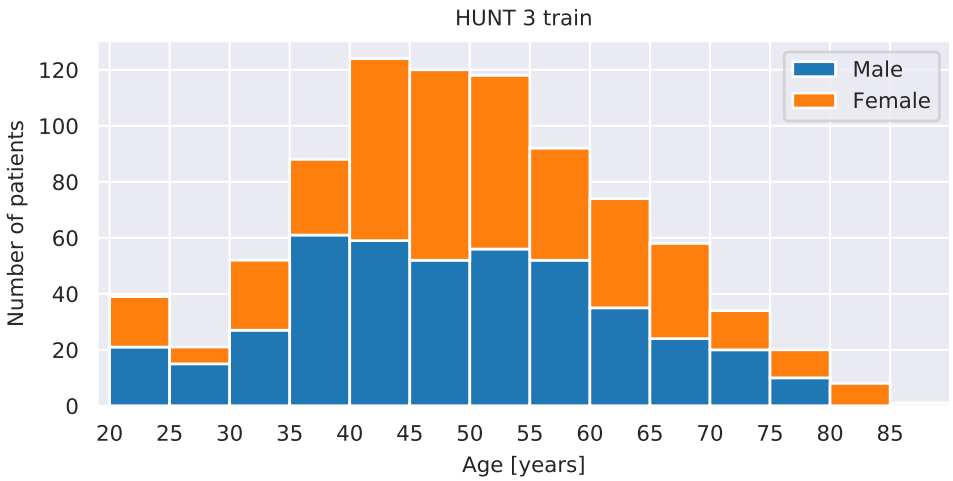


Figure 3.1: Age of participants in the training split of the HUNT 3 echocardiography study.

B-mode second harmonic images and color tissue Doppler are acquired, as well as ECG alongside the recordings. B-mode recordings are optimized for evaluation of the left ventricle. Each recording contains three heart cycles.

Several publications have been made on the HUNT 3 echocardiography study. Dalen et. al [46] studied segmental and global longitudinal strain using both speckle tracking and tissue doppler imaging, and found that strain decreases with increasing age. They also showed changes in mitral and tricuspid annular velocities with age and sex [47]. Støylen [48] et. al studied the relationship between global longitudinal strain (GLS) and mitral annular plane systolic excursion (MAPSE) and showed that MAPSE is also negatively correlated with age. Støylen et. also studied left ventricular geometry (length, diameter, and relative wall thickness) using M-mode and showed that ventricular dimensions vary with age and gender [49].

Out of the 1266 patients, 673 are female and 623 are male. Age distributions in HUNT3 is close to normally distributed, with a low age of 19 and a high age of 88. The mean and standard deviation is 47.8 ± 13.6 years for women and 50.6 ± 13.7 years for men. HUNT 3 is randomly split into a training, validation and test set containing $70\%/15\%/15\%$ of the patients respectively. Age distributions for the data splits is seen in Figures 3.1, 3.2 ,3.3.

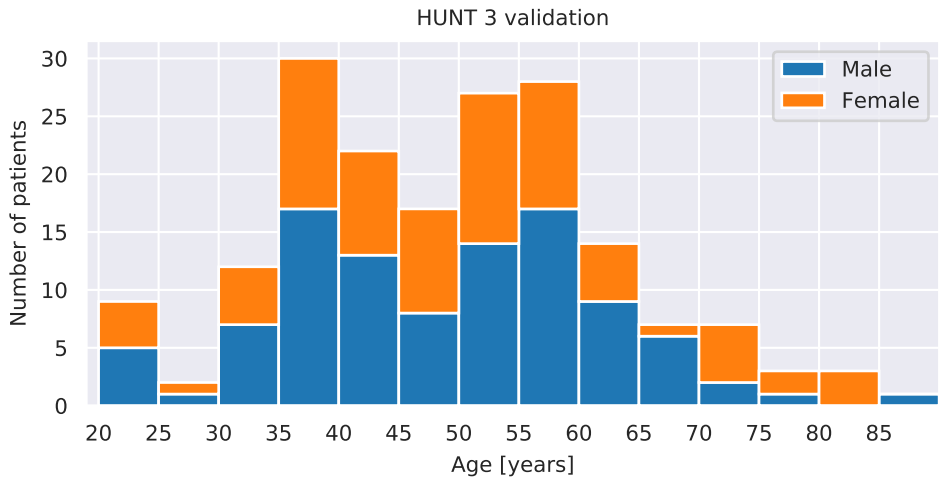


Figure 3.2: Age of participants in the validation split of the HUNT 3 echocardiography study.

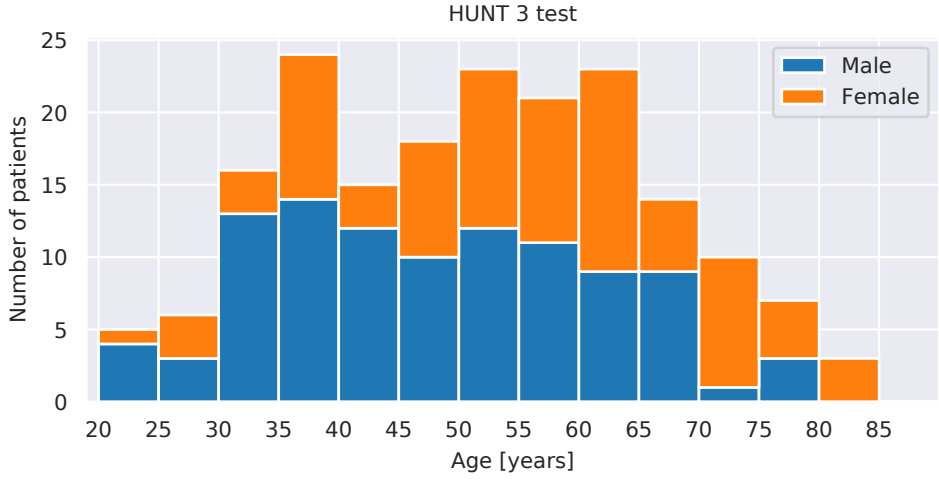


Figure 3.3: Age of participants in the testing split of the HUNT 3 echocardiography study.

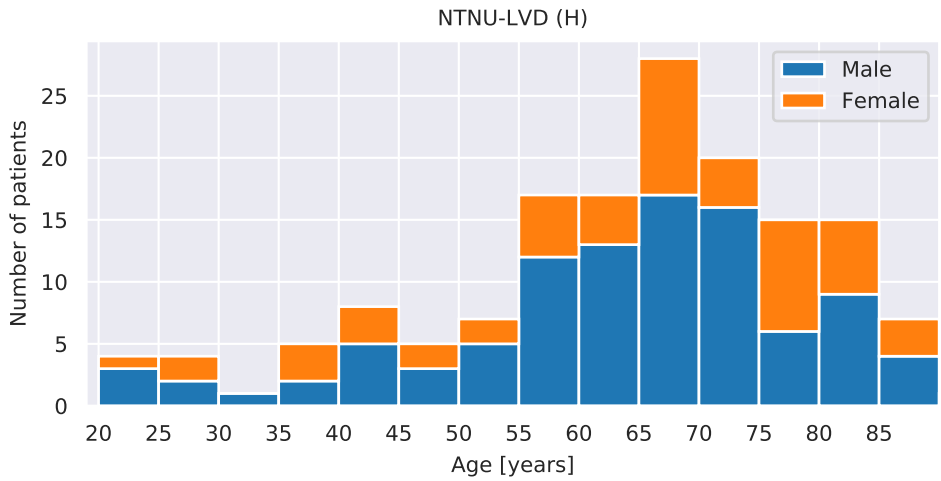


Figure 3.4: Age of patients in the NTNU-LVD (H) data set.

3.2 Left Ventricular Dysfunction Study (NTNU-LVD)

The study [16] consists of patients above 18 years old, referred for echocardiographic examination at the Department of Cardiology, St. Olavs hospital. Acquisition was performed between 2013 and 2015. Vivid E7 and Vivid E9 systems were used for acquisition. Patients were excluded if the image quality of the recordings are too poor for evaluation by cardiologists. Reference examinations are performed on all patients by experienced cardiologists or experienced sonography techicians. 204 patients are included in the study (62%/38% male/female). 47 patients were diagnosed with left ventricular dysfunction, whereof 47 were systolic and 11 were diastolic. Systolic dysfunction is defined as ejection fraction < 50. Diastolic dysfunction is determined by ejection fraction ≥ 50 along with findings of increased LV filling pressure. The data set is split into healthy and diseased patients. A patient is considered to be diseased if either systolic or diastolic dysfunction is labeled. The data split consisting of healthy patients is denoted NTNU-LVD (H), while the diseased split is denoted NTNU-LVD (D). Age distributions for the patients in NTNU-LVD is seen in Figures 3.4 and 3.5.

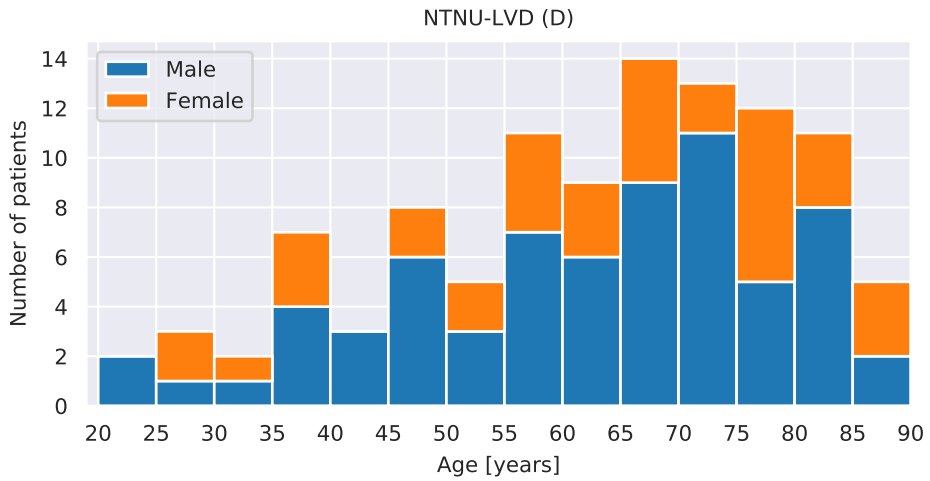


Figure 3.5: Age of patients in the NTNU-LVD (D) data set.

3.3 Tromsø Coronary Arterty Disease Study (UNN-CAD)

The study was performed at the University Hospital of North Norway (UNN), and consists of patients referred to a CT-scan due to symptoms of ischemic heart disease. All patients underwent coronary angiography to detect narrowing of the blood vessels (stenosis). The data set is denoted UNN-CAD. Out of 250 patients, 48 had findings significant enough to undergo stenting, the placement of metal tubes to expand the arteries. Acquisition is performed by an experienced cardiologist and a doctoral research fellow between 2016 and 2017, using a Vivid E9 system. The aim of the study is to determine whether regional measurements of heart function from echocardiography in rest can be used to identify CVD.

The age distribution of men is similar to that of HUNT 3 with a mean age of 56.2 years, while female ages are skewed to the right and has a mean age of 61.2 years.

Equivalently to the NTNU-LVD dataset, the data set is split into healthy and diseased patients, denoted (H) and (D). Age distributions of the data sets are seen in Figure 3.6 and 3.7.

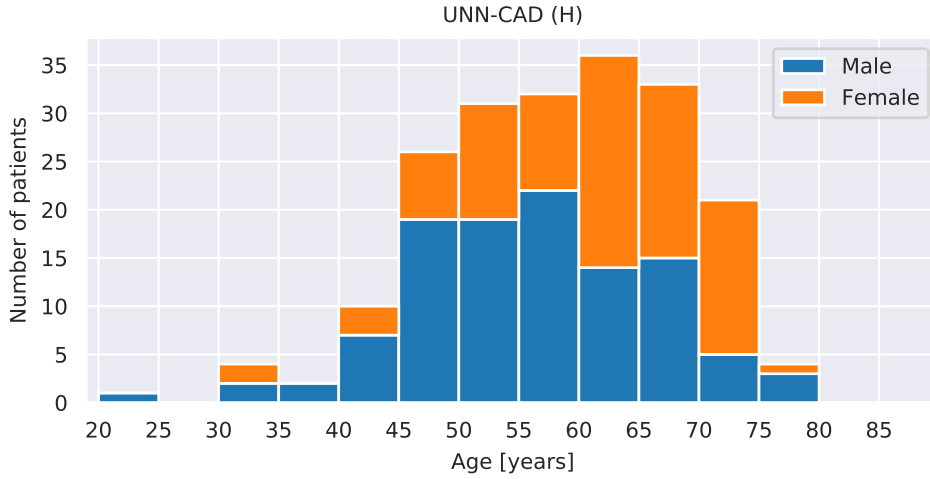


Figure 3.6: Age of patients in the UNN-CAD (H) data set.

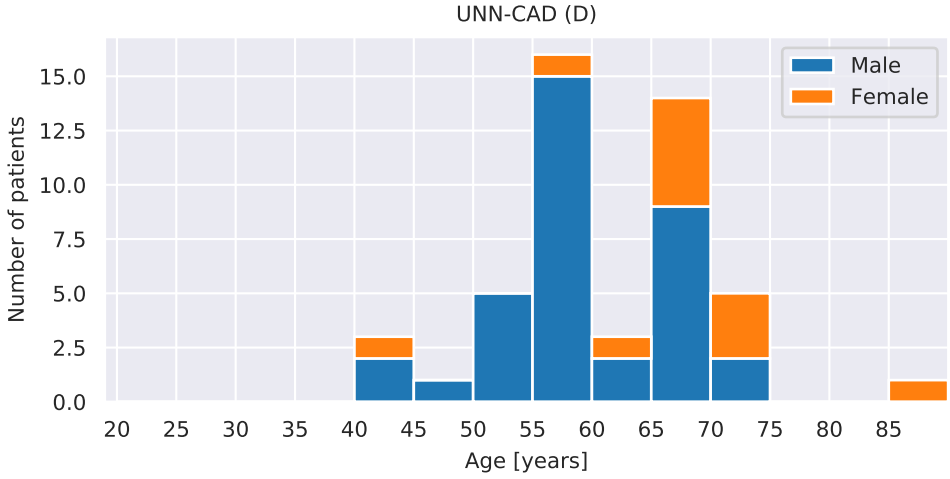


Figure 3.7: Age of patients in the UNN-CAD (D) data set.

4 | Automatic Quality Assurance

An echocardiography exam includes a variety of different views and imaging modalities, with different measurements being taken in each modality and view. Even for the same view and mode, there are large variations in data. This can be due to differences in transducer positions, image settings, noise-characteristics and artefacts, echocardiography hardware and software as well as patient variations. The more variability in the data unrelated to the task in consideration, the more difficult the learning task becomes. Learning from data with little diagnostic value can result in an inferior model, as the model might instead fit to other features such as noise. From a practical point of view, the time usage for training the models decrease when fewer samples are used.

Defining the diagnostic value of a sample is difficult, as it is dependent on the task. Global image statistics such as signal to noise (SNR) ratios might indicate a poor acquisition, but SNR also differs with imaging settings and patient conditions and is not directly related to the information attainable from a recording. Although high noise levels in general makes interpretation more difficult, samples with high noise levels can still contain useful information, and samples with low noise levels can contain no useful information at all.

Supervisedly training a model to generate quality measurements from pixel data is a possible way to perform quality assessment, but it requires a large set of labeled data covering different views, imaging settings and patient, which is unavailable. The difficulty of learning quality scores is seen in [19]. Although the model learns quality well for most samples, atypical cases are difficult. An example of this is that patients with artifacts caused by prosthetics were marked as acceptable quality by the expert, but poor by the classifier which had not learned to ignore the effect of prosthetics on imaging.

To achieve some coherence between the selected quality measures and the value of a sample for echocardiography assessments, CNNs trained for tasks in echocardiography are considered. As the models are fairly accurate, the model outputs should be dependent on

whether useful features are present.

4.1 Initial Removal

Only recordings optimized for 2D B-mode are considered further. This includes removal of modalities such as M-mode, color Doppler and 3D imaging. Although many of these modalities also contain accompanying B-mode recordings, imaging properties are altered compared to standard 2D B-mode recordings. For HUNT3, 59 Patients with pathologies within healthy ranges were removed, with the aim of increasing the difference between estimates for healthy and diseased patients. DICOM-files causing errors during loading or scan-converting with *pydicom* are discarded. This is for example caused by erroneous tags or missing beam data.

4.2 Cycle Separation

Several heart cycles are stored in a recording, although often only some are intended for quantification. To avoid the quality measurements being affected by variation in quality between cycles, the quality assurance scheme is applied to each cycle. Cycle separation is performed using ECG, where an event in the ECG signal known as the QRS-complex is easily detectable in most cases. The time of the QRS complex can be automatically detected by the scanners and included in the DICOM-files, which is the case for the NTNU-LVD and UNN-CAD datasets, but it was not with pydicom for HUNT 3. Therefore, QRS-complexes for HUNT 3 are detected using an algorithm inspired by the Pan-Tompkins algorithm [50]. Automatically detected QRS were used when available.

The algorithm is based on detecting large slopes in the ECG signal during the QRS-wave, and proceeds as follows. First, the ECG signal is differentiated and squared, which attenuates the large differences between the Q, R and S peaks. Secondly, a moving average smooths out the squared difference signal, to produce a single peak for each QRS. A rectangular window with a size of 50 samples is used for this, which is approximately equal to the width of QRS complexes encountered in the HUNT 3. The signal is normalized between zero and one, and only peaks above 0.5 are included. A minimum distance between detected peaks is set to 60% of the average heart cycle time, as given in the DICOM-data. The Python package *PeakUtils* [51] is used for the peak detection.

Figure 4.1 shows an ECG signal from HUNT 3, the filtered signal and the detected QRS

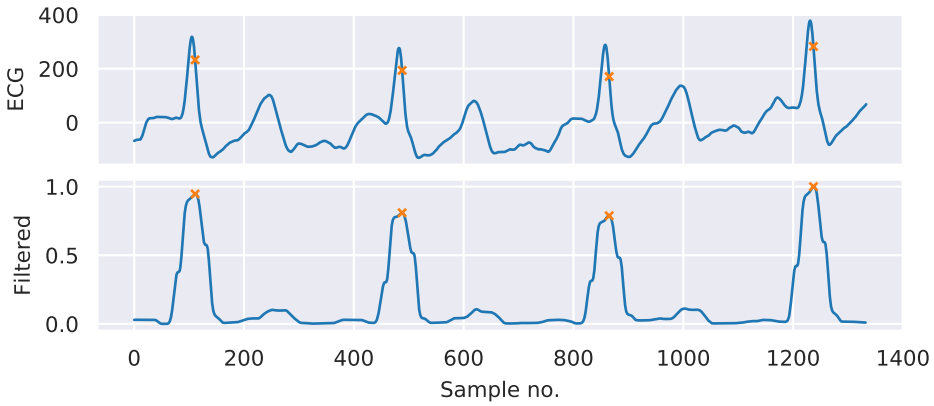


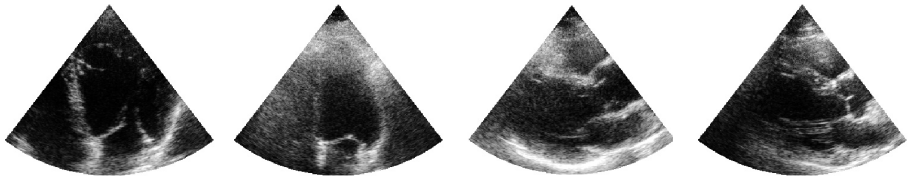
Figure 4.1: Peak detection algorithm from ECG inspired by the Pan-Tompkins algorithm [50].

peaks. Notice that there is a slight delay between peak R values to the detected peaks due to time lag from the moving average. This can be compensated for by subtracting the time lag, which is not considered here as finding the exact time of end-diastole is unimportant. The echocardiography and ECG signal are sampled at different frequencies. To acquire a frame for each detected QRS, the timestamps of the echocardiography recording is considered. The frame with time stamp closest in time to a detected QRS is selected.

Cycles containing fewer than 5 frames are discarded, as this is likely caused by false positives from the QRS detector. Cycles longer than 150 frames are discarded as they are caused by false negatives of the QRS detector.

4.3 View Classification and the View Error

In the data sets considered, no information is available about the view of each recording. To label the view of each cycle, a CNN for view classification is used. The model is trained to classify apical two, four and long-axis, parasternal long and short-axis, and an unknown class. Any recording not belonging to the apical or parasternal window should therefore be classified as unknown. The accuracy of the view classifier is reported to be $(96 \pm 0.9)\%$. The view classification network operates on individual frames, giving the probability of the frame belonging to each of the six classes. Formally, the model output is $P(V|I \sim f(I;\theta_{view}) \in \mathcal{R}^6$ and $\sum_v P(V = v|I) = 1$, where I is an image, θ_{view} is the parameters of the view classification CNN, and v is one of the fiew views or unknown.



(a) A4CH: 0.00 **(b)** A4CH: 0.12 **(c)** PLAX: 0.00 **(d)** PLAX: 0.14

Figure 4.2: Classification of views for frames in HUNT 3, with probabilities.

To reduce variability in the view predictions, probabilities are averaged over a cycle. The view maximizing the average probability of a cycle is selected as the view of the cycle. This is given as

$$\arg\max_v \sum_{t=1}^{T} P(V = v|I(\mathbf{x}, t)), \tag{4.1}$$

where T is the length of the cycle.

The view classification network is also used to generate an automatic view quality measure. This probability should decrease as a sample deviates more and more from a standard view. Therefore, if the model cannot be certain of the view, it is possibly caused by data with poor quality.

This is used to generate a measure of deviation from quality of a cycle. The measure is denoted the *view error*, e_v:

$$e_v = 1 - \max_v \sum_{t=1}^{T} P(V = v|I(\mathbf{x}, t)/n. \tag{4.2}$$

The smaller e_v the better, with 0 being the lowest possible error.

Figure 4.2 shows examples of views and view errors for four randomly selected patients from the data. By averaging the probabilities for each frame in a cycle, one view and corresponding view error is acquired.

4.4 The Timing Error

The view classification network operates on each frame separately, and does therefore not take motion into account. Poor quality due to motion can occur because of excessive movement of the patient or the transducer during acquisition. For a quality measure based

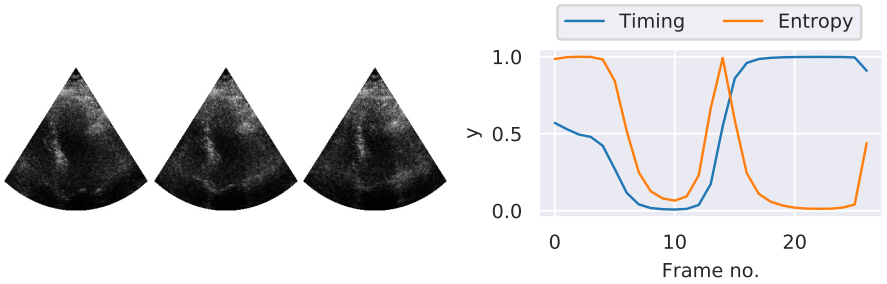


Figure 4.3: Frames from a cycle in HUNT 3 alongside the output of the timing model and the corresponding entropy per frame. The timing error (average entropy) for the cycle is 0.38.

on spatio-temporal features, the 3D CNN for detecting cardiac cycle phase from the project thesis is used [52]. The model gives one output for each frame, where 0 represents systole and 1 represents diastole. The model was trained to minimize the binary cross-entropy between the model output and the label for each frame, where the binary cross-entropy is given by

$$L_ce(y) = -y \log_2(\hat{y}) - (1-y) \log_2(1-\hat{y}). \tag{4.3}$$

A2CH and A4CH views were used for training. During the project thesis, it was noticed that the output of the cardiac cycle phase models were less certain for samples with poor quality. To measure the uncertainty of the model output, the entropy function is used. The entropy H at timestep t is defined as

$$H_t = -\hat{y}_t \log_2(\hat{y}_t) - (1-\hat{y}_t) \log_2(1-\hat{y}_t), \tag{4.4}$$

where $\hat{y}$ is the prediction of the timing model for a given frame. The timing error is averaged over all frames in a cycle, given by

$$e_t = \frac{1}{T} \sum_{t=1}^{T} H_t, \tag{4.5}$$

where T is the number of frames in the cycle. Figure 4.3 shows the output of the timing model on a cycle. The timing error is averaged over all frames in the cycle.

4.5 Automatic QA Results

As no ground truth labels are available to evaluate the automatic QA measurements, no quantitative results of the steps are given. Instead, statistics of the measurements are supplied and visual inspection is performed. For an objective analysis of automatic data selection, reference values of data quality should be used. More examples of timing and view errors are given in Appendix 8.1.

When using the proposed cycle separation algorithm alongside automatically generated QRS triggers from the scanner, a bias in detected QRS could make a difference for the deep learning models trained using a different start time. To evaluate the difference between the custom QRS algorithm and the QRS-detector of the scanner, the time of the detected QRS complexes from both methods are compared using the NTNU-LVD dataset. Comparison is difficult due to the possibility of false positives and negatives in both methods. False positives occur due to noise, while negatives occur most frequently due to recordings being cut at the time of QRS. To be able to compare QRS measurements, only recordings with the same number of detected QRS complexes are compared. If this is satisfied, each QRS complex of the custom algorithm is assigned to the closest QRS complex from the scanner. This excludes false positives or negatives that occur only for one of the methods.

6831 out of 7966 recordings in NTNU-LVD have an equal amount of detected QRS-complexes. The resulting Bland-Altman plot is seen in Figure 4.4. 96 out of 19880 detected QRS complexes are outside the plot boundaries. The bias between the methods is low, and the variance of 0.02 seconds is fairly low compared to the length of a heart beat.

The distribution of view errors for all B-mode cycles in HUNT 3 is shown in Figure 4.5. As the model is trained for classification, the timing error is heavily skewed towards 0, with only some larger errors. To be able to visualize the distribution, a histogram with logarithmic scale on the vertical axis and uneven bin sizes is used. From inspection, view scores appear to be highly dependent on how clearly visible the heart walls are.

Figure 4.6 shows the distribution of the timing errors for HUNT 3. The timing errors is less skewed, and a regular histogram is used for visualization. Unsurprisingly, the model has lower timing error for apical views compared to parasternal, as it is trained on apical views. Almost no cycles have below 0.1 timing error. The reason for this is that the output must switch between 0 to 1 when going from systole to end-diastole, as seen in Figure 4.3. The switch usually requires a few steps, which results in increasing the error slightly.

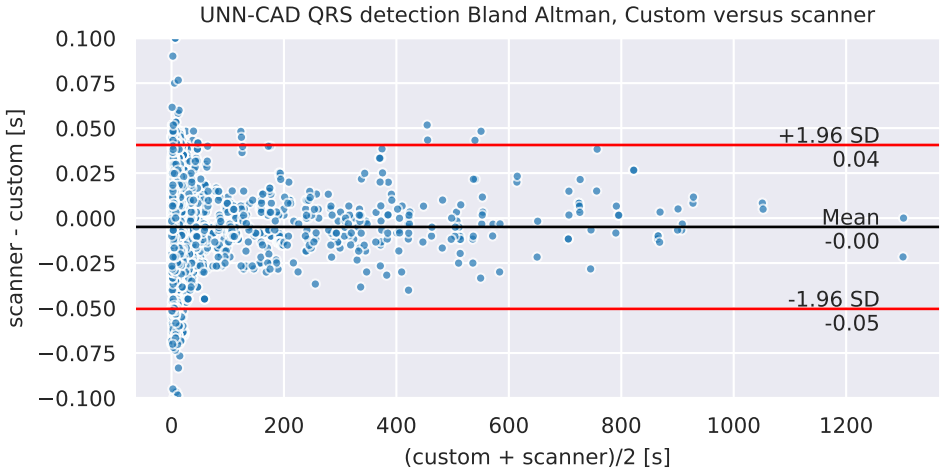


Figure 4.4: Bland-Altman plot comparing the times of QRS detection by the custom algorithm based on Pan-Tompkins [50] to the QRS detection algorithm on the scanners.

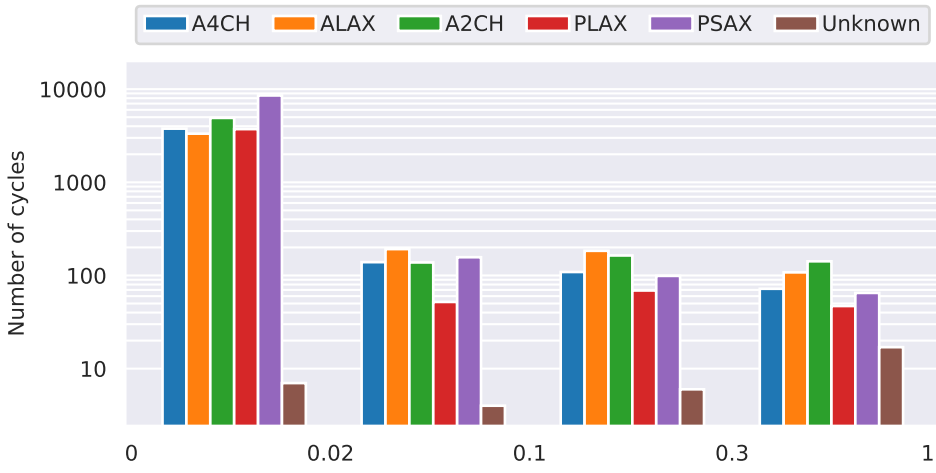


Figure 4.5: Distribution of view errors for cycles in the HUNT 3 dataset, as measured by a CNN for view classification. Uneven bin sizes and a logarithmic scale on the y-axis is used, as an excess of errors are close to zero.

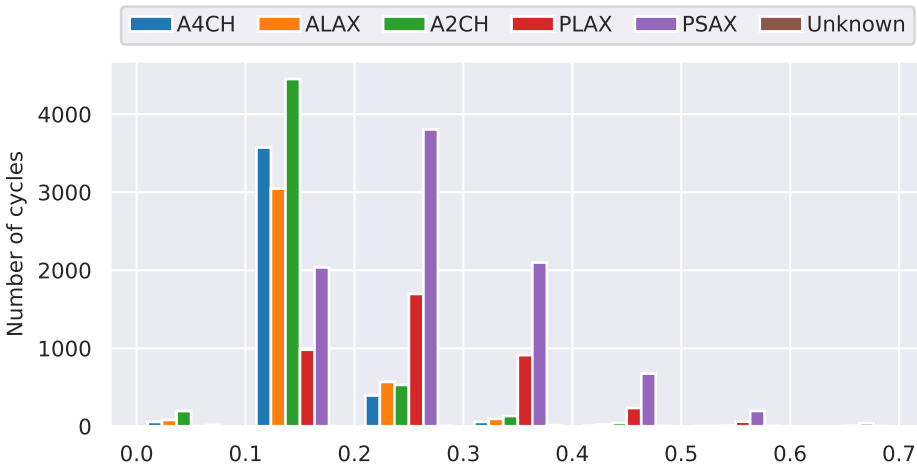


Figure 4.6: Distribution of timing errors (entropy of the timing model output) over all cycles in HUNT 3.

From inspection, the timing error appears to be mostly dependent on the visibility of the contraction of the chambers and movement of various valves. This is shown in more detail in the Appendix 8.1.

To discard cycles based on timing and view error, a simple approach is used. All cycles with timing errors above the 90th percentile of timing errors or above the 90th percentile of view errors errors is discarded. The thresholds and number of cycles kept versus discarded is shown in Table 4.1.

To mitigate data leakage from automatic QA, only HUNT 3 is used for generating quality score thresholds. The HUNT 3 thresholds are then applied to the other data sets. Some leakage still occurs due to using all HUNT 3 data splits for calculating the quality score distributions. This is because the QA scores were calculated before the HUNT 3 data splits were generated. Due to the large size of HUNT 3 -train, the effect of including the validation and test sets in the distributions should be small.

If quality scores differ between healthy and diseased patients, discarding data based on fixed thresholds from HUNT 3 result in discarding a larger fraction of data from one of the groups. A large separation between the distributions of quality measures can also be to classify disease directly. To evaluate this, view and timing scores are evaluated for healthy and diseased patients in the UNN-CAD-dataset. Normalized histograms are plotted as there are fewer diseased patients than healthy patients. The results are summarized in Figure 4.7. A two-sample T-test for the healthy versus diseased patients result in a p-value

Table 4.1: Statistics for automatic QA on B-mode cycles in HUNT 3

View	Kept	Discarded	View error threshold	Timing error threshold
A4CH	3418	671	0.0030	0.21
ALAX	3177	637	0.052	0.24
A2CH	4480	879	0.0060	0.22
PLAX	3252	641	0.00010	0.39
PSAX	7338	1542	0.00020	0.41
Unknown	0	34	0	0

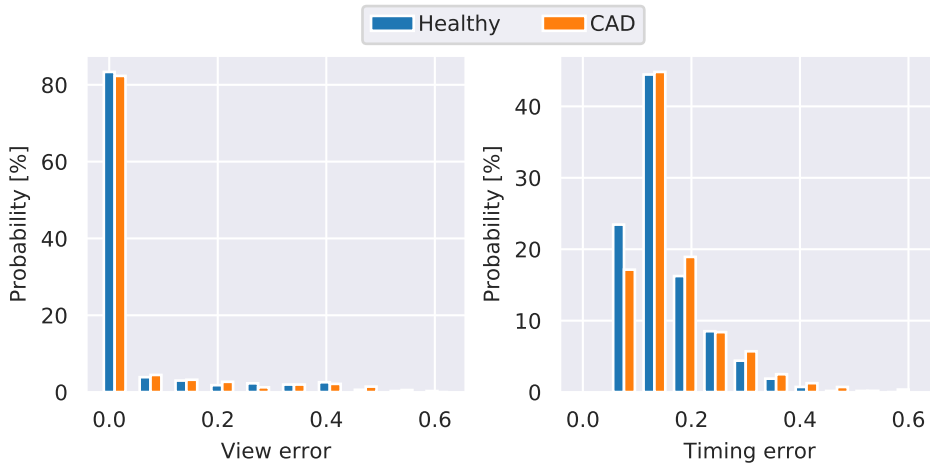


Figure 4.7: QA scores for patients with CAD vs healthy patients from the Tromsø dataset.

for different means of 0.75, meaning that there likely is no difference. For the timing error, the mean is different ($p = 2.7 \times 1e^{-5}$). The mean error for healthy patients is 0.17 and the mean error for diseased patients is 0.18, and the difference is too small to separate between healthy and diseased in practice.

Another concern is whether timing or view errors vary with age. In this case, data is discarded differently based on age. The Pearson correlation between timing error and age is 0.0089. The Pearson correlation between view error and age is 0.050. This indicates little or no difference with age. Interestingly, there is also little correlation between view errors and timing errors, with a Pearson correlation of 0.0259. An interpretation for this is that the errors does not capture a shared concept of quality. However, the low correlation is most likely caused by the uneven distribution of the view error, which result in small differences between most kept and discarded cycles.

5 | Estimation of Age

5.1 Problem Formulation

Age estimation can be formulated both as a regression and a classification problem. The time since birth is a continuous measure, and age estimation can therefore be considered a regression problem. Age can also be categorized, for example as the integer number of years since birth or into age groups. It is common practice to specify normal ranges in echocardiography by age, such as <40 years, 40-60 years, and >60 years [46], [48]. In this sense, age prediction is a classification problem.

Each way of posing the problem has its own of advantages and disadvantages. From the modeling perspective, there is little difference between the regression and the classification task. Classification can be achieved by normalizing the output of a CNN by a logistic activation function such as soft-max or sigmoid. In other words, a classification and a regression model can be separated only by a single parameter-free layer. In [24], where a CNN for age estimation from human faces is proposed, classification of each year followed by a soft-max results in slightly improved performance. However the authors note that this increases the possibility of overfitting.

An easier classification problem can be achieved by discretizing age into larger categories. At the coarsest level, age can be divided into young versus old, with some boundary between young and old. With one output category for each year, perfect accuracy can be achieved when age is labeled in integer years. This results in a large number of classes, making the classification problem more difficult. With the large and complex models used, finding a suitable number of classes and its boundaries increases the number of hyperparameters to search for. Finally, categorical loss functions do not capture the relationship between classes, such that modification of the loss must be performed, e.g. with the soft-

max approach. For example, predicting an age of 50 when the truth is 51 should result in a lower loss than predicting an age of 20 for the same patient.

For simplicity, only regression of age is performed.

5.2 Weighted Averaging for Patient Estimates

Multiple cycles from several views are available for most patients, resulting in multiple estimates. To utilize this information the estimates for a patient are averaged. Averaging is commonly performed in echocardiography to reduce variability between cycles and measurements.

An unweighted average is simple, but does not take into account the difference in accuracy for the views, or the dependence between samples from the same recording and view. This is likely to be suboptimal in echocardiography, where some views are more appropriate for a given task. Instead, estimates can be combined based on this knowledge. This can be considered a new estimation problem, where optimal averaging weights can be found using any desired optimization algorithm, for example ordinary least squares [53]. Regression models typically require a constant number of regressors. In the age estimation case, there is a variable number of estimates to be combined. This makes the regression averaging less straightforward as a fixed number of regressors must be generated from a variable number of cycles.

Instead, an approach inspired by *boosting* [54] is presented. In boosting, multiple estimates for a sample is generated by multiple models. The estimates are weighted using a function decreasing with the error of each model. Although most commonly applied to classification, boosting methods can also be applied to regression models. Granitto et. al [55] propose the following weighting for regression:

$$w_i = \frac{e_i^{-\alpha}}{\sum_j e_j^{-\alpha}}. \tag{5.1}$$

Where w_i is the weight for estimator i, e_i is the error for estimator i, and α is a parameter set to 2. In other words, the weights are inversely proportional to the square of the error.

To also take into account that estimates from the same recording and view are less independent, a weighting inversely proportional to the number of cycles in the recording and the number of recordings for the given view is also introduced. The resulting unnormalized

weight w_c for a cycle is given by

$$w_c = \frac{1}{e_v^{\alpha_1} n_r^{\alpha_2} n_c^{\alpha_3}} \tag{5.2}$$

Here, e_v is the MAE of the validation set for the view of the cycle, n_r is the number of recordings of the same view for the patient and n_c is the number of cycles in the recording of the cycle. The weights for a patient are then normalized to one by dividing by the sum of the weights.

The larger α becomes, the smaller the weight becomes. Setting α_2 or α_3 to zero corresponds to regular averaging. Setting α_2 or α_3 to one corresponds to one cycle having the same influence as n cycles. α_2 and α_3 should therefore be set to a value between zero and one. For simplicity, α_2 and α_3 are simply set to 0.5, indicating that the influence of multiple samples from the same view or recording decreases by the the square root of the number of samples. As in [55], α_1 is set to two.

5.3 CNN Architectures

A variety of approaches have been proposed for machine learning from video inputs. Traditional machine learning models for video classification have mostly followed the same pipeline as in image classification. First, spatial or spatio-temporal features are extracted from the videos, sparsely or densely. These features are then pooled into a fixed size representation of each video, which is used to train a shallow classifier. As in many other computer vision fields, CNNs have surpassed shallow counterparts in accuracy. Common architectures are Long-term Recurrent Convolutional Networks [56], Two-Stream CNNs [57] and 3D CNNs [58].

Video learning is significantly more difficult than learning from still images because of the increased dimensionality, requiring both spatial and temporal features, and 2D CNNs are inherently unable to learn spatio-temporal features. If motion is an important aspect of the problem, such as is often the case in echocardiography, including the temporal dimension in the model is preferable. This is seen from the domain of human action recognition [14], [59]. The disadvantage of 3D CNNs is that with increased model size, even more data and computational resources are required for successful learning.

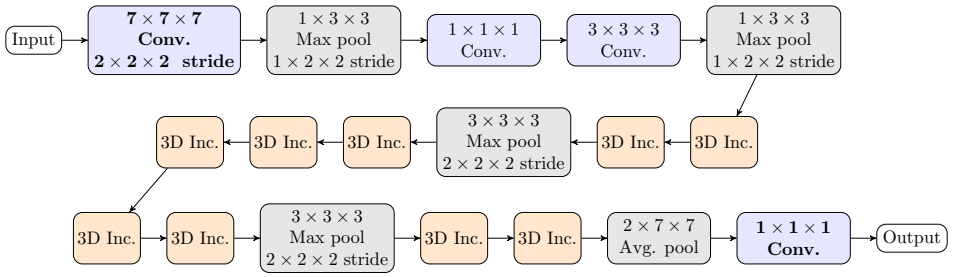


Figure 5.1: One stream of the I3D Inception model [14]. Layers in bold are modified for echocardiography, while other layers are equal. The first convolutional layer has 1 input channel with weights equal to the sum over RGB input channels of the pretrained layer weights, but is unchanged between the optical flow models. Random initialization of the weights are used for the last layer, and the final sigmoidal activation function is removed.

5.3.1 Base Model: I3D

Selecting the ideal model for age estimation from echocardiography is time-consuming. Instead of attempting different architectures, a state-of-the-art architecture from human action recognition is selected as a base model, the Two-Stream Inflated 3D ConvNet (I3D) [14]. When published, it outperformed LRCN, 2D Two-Stream and plain 3D CNNs in a comparison on large video data sets. The architecture is a result of *inflating* filters of the Inception-v1 model into 3D. Inflation refers to stacking 2D convolutional filters of shape $N \times N \times C$ along a temporal dimension and dividing weights by N, resulting in 3D convolutional filters of shape $N \times N \times N \times C$. A benefit of this inflation is the possibility of pretraining on image datasets such as ImageNet [60].

Drawing from two-stream networks, two equal models (streams) are defined, one using standard video inputs and the other using optical flow inputs. Each stream is trained separately, and at testing time the two streams are averaged. The only difference between the two streams is the first convolutional layer, which has 3 channels (kernel size of $7 \times 7 \times 7 \times 3$) for the RGB stream, and 2 channels (kernel size of $7 \times 7 \times 7 \times 2$) for the flow stream. The overall layout of a single stream is seen in Figure 5.1, where 3D Inc. refers to an inflated Inception block (Figure 2.2). Each stream consists of 3 convolutional layers at the input, 9 Inception modules, and a fully connected layer at the end. Max pooling layers are distributed throughout, where the first two pooling layers only apply along spatial axes to avoid discarding temporal structure early on. Unlike Inception-V1, there is no local response normalization, and each convolutional layer is followed by batch normalization before the activation function. Additionally, 5×5 convolutions are replaced by $3 \times 3 \times 3$

convolutions. The resulting stream has 12 million parameters.

5.3.2 Modifications for Age Estimation

Three input modalities are attempted: B-mode cycles, B-mode cycles with additional coordinate channel input data, and optical flow inputs.

To reduce the amount of data required for training, model weights pretrained on the Kinetics dataset containing 240 000 videos of human actions are used as initialization. Although there is a considerable difference between human actions and echocardiography, visualization of the first convolutional layer reveals that the pretrained model has learned generic features like edges, rectangles and blobs. Some filters are changing along the temporal dimension, while others are fairly constant in time, suggesting that both spatial and spatiotemporal feaures are utilized. These filters are not specific to human action recognition and hopefully applicable to echocardiography as well.

The original model has 3 input channels, while standard B-mode only has one brightness channel, such that the convolutional operation cannot be applied. One solution is to repeat the channel axis of the B-mode cycles three times, resulting in the correct input shape. While this requires no modifications to the pretrained network, unneccessary computation is performed. To reduce the number of operations, the kernels of the first convolutional layer are reduced into one channel by summation along the channel axis. Reducing convolutional kernels by summing along the channel axis is equivalent to the original convolution operation the channels are constant, due to the linearity of the convolutional operation. By this modification, B-mode images can be input directly to the network. Color information contained in the pretrained I3D is discarded by this step, but B-mode does not contain color information regardless.

To perform classification, the final layer of I3D uses a sigmoidal activation function. This is not applicable for age regression, and the activation function is removed.

The model is implemented in Keras 2.2.4 [61] with the TensorFlow 1.10 backend [62], expanding from a Keras implementation of I3D [63].

5.3.3 Using Multiple Views

Two options are explored for training with data from multiple views. The first is to train a stream for each considered view. resulting in five models and $5 \cdot 12 = 60M$ million

parameters.

Alternatively, the same stream is reused for all views. In previous work [25], accuracy was improved by using both A2CH and A4CH data. The simplest way of reuse is to apply the same operations to all inputs. This might be suboptimal as there are distinct differences in the structures that are visible in each view.

To achieve a difference in the operations while reusing early layers, the final layer of the model is set have five outputs, one for each view. The output corresponding to the view of a cycle is then used as the age estimate for the cycle. To achieve this, a one-hot encoding of the view is input to the model along the video data, e.g. the encoding is $[1, 0, 0, 0, 0]$ for an A4CH cycle and $[0, 0, 0, 1, 0]$ for an ALAX cycle. A dot product between the encoding and the outputs is taken to select the corresponding output.

5.3.4 Coordinate Channels

The physical sector width and depth of a recording varies between recordings. This results in a variable pixel spacing when resizing every recording to 224×224 pixels. Knowledge of the physical spatial dimensions is required to perform quantification, unless dimensions are constant for all samples. Inspired by [64], physical dimensions are input into the network alongside the images, by appending two coordinate channels to the B-mode channel. The first channel represents the horizontal positions of each pixel, and the second position along the depth direction.

I3D is not pretrained with input coordinates. Therefore, channels corresponding to coordinate inputs in the first convolutional layer are randomly initialized, while the channel corresponding to B-mode inputs is still the sum of the pretrained RBG channels. Appending coordinate channels results in inputs deviating further from the data used for pretraining, and the advantage of pretraining is likely to diminish. To evaluate whether pretraining is useful in this case, a smaller model is also trained with randomly initialized weights for all layers. The smaller I3D is defined by removing the last 3 Inception blocks, reducing the number of kernels in the Inception blocks, and removing one branch of the inception block consisting of 1×1 followed by 3×3 convolutions. The resulting smaller I3D has $2.5M$ parameters. The initialization of Glorot [65] is used.

5.3.5 Saliency

This part is adapted from previous work [25]. An issue with deep learning is that understanding the path from input to output is difficult due to the millions of parameters and operations. Understanding the model is especially important in echocardiography.

In order to improve understanding of the model, saliency maps [66] are generated by taking the derivative of the model output with respect to the input video. This yields a video where intensities correspond to the change in output when each individual pixel is changed.

Normal saliency has a tendency to produce noisy results. This is because both pixels that increases and decreases the output the most are found. To obtain more visually pleasing results, the method of guided backpropagation [67] sets negative gradients to zero when backpropagating through ReLU-layers. This removes gradients that changes the output towards the saturated region of the ReLU layers. The effect is that pixels that does not contribute to a change in the output are suppressed.

5.4 Preprocessing

After automatic quality assurance, a set of recordings remain. There is still variability between recordings that is unrelated to aging. For example, videos are sampled at different frame rates and with different resolutions. To reduce this variability, preprocessing of the data is performed.

5.4.1 Input Shape

Choice of input shape for each sample has a major impact on memory and computational costs, as well as the performance of the model. The optimal input shape depends on the task and the model of choice. Downsampling reduces the memory and computational costs at the expense of discarding potentially useful information. Down to a certain spatial size, downsampling mostly suppresses high frequency content, e.g. noise and speckles. A higher resolution is useful for data augmentations, as there are more data to interpolate from. In Madani et al. [17], studies of the effect of input size are performed. Here, the accuracy of a CNN for LV hypertrophy classification increases with input size up to 120×160, where improvements does not occur when doubling the dimensions. For

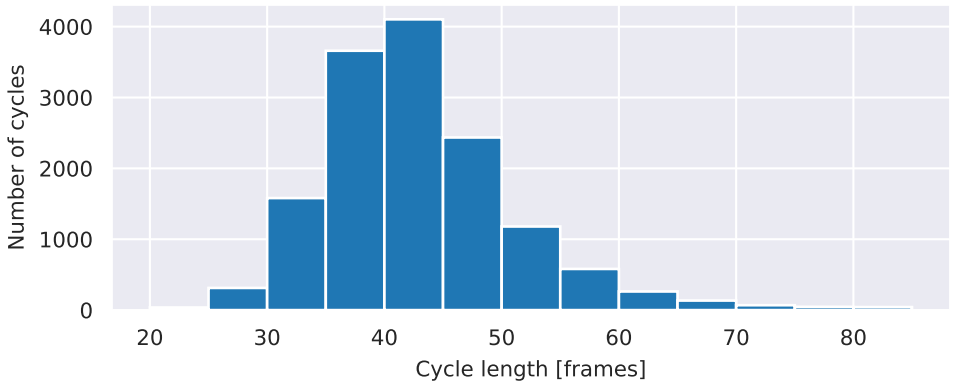


Figure 5.2: Length of cycles in HUNT3

Zhang et al. [8] an input size of 224×224 is selected to accurately detect hypertrophic cardiomyopathy and cardiac amyloidosis.

Following this, spatial dimensions of 224×224 pixels are selected, which by visual inspection preserves the structure most tissue.

The temporal dimension is treated separately from the spatial dimensions. The input length is instead set to a fixed number. Cycles longer than the input length are cut at the end, and videos shorter than the input length are padded with zeros. To determine an appropriate batch length, the lengths of cycles in HUNT 3 is considered, seen in Figure 5.2. Setting a fixed batch length of 50 frames keeps the number of padded cycles low, with most cycles being fully included, reducing computational and GPU memory usage. With a mean frame rate of 44 Hz in HUNT 3, cycles with heart rate above $60 \cdot 44 \cdot 50 = 52.8$ BPM are fully included in a batch.

The number of cycles in a batch is set to 6, as is done for the original I3D.

5.4.2 Temporal Normalization

In HUNT 3, the frame rate has a low variability, but the mean frame rate is different from that of NTNU-LVD and UNN-CAD. For the age estimation model to learn features related to velocity, the sample time should be known or fixed.

Having data with variation in sampling rates also introduces some practical issues. A difference in sampling rate results in variations in the duration of each recording, such that the length of each cycle deviates more from 50 frames. This results in an increased

temporal padding or cropping.

Two approaches are considered for resolving these issues. The least computationally expensive solution is to downsample by an integer factor, whenever the resulting sample rate is closer to the reference sampling rate by downsampling. For a video with sampling rate f, the integer downsampling ratio r and new sample rate f_{new} is given by

$$r = \text{nint}(\frac{f}{f_{ref}}) \qquad f_{new} = \frac{f}{r} \tag{5.3}$$

where nint$(\cdot)$ rounds to the nearest integer. Cycles with a downsampling ratio of zero did can be discarded. The start frame of downsampled frames is randomized to avoid discarding $r - 1$ out of r frames from cycles during training.

The alternative is to perform video frame interpolation, allowing for setting an exact sample rate at the expense of possible errors from interpolation and increased preprocessing time. By considering a video as a 3 dimensional volume ($t \times h \times w$), new temporal values can be interpolated at a fixed time step. Linear interpolation is performed using the *RegularGridInterpolator* in *scipy* [45]. Automatic QA is reapplied to the interpolated values, resulting in minor differences in quality scores.

For both approaches, a target sample rate is set to the mean sample rate in HUNT 3, 44 Hz. This results in downsampling 859 cycles in HUNT 3 by a factor of 2, while no cycles had a downsampling ratio of zero.

5.4.3 Optical Flow

Just as optical flow inputs frequently appear as inputs to CNNs in various domains, echocardiography is no exception. Gao et. al [68] apply the variational optical flow of Brox et. al [39] twice to achieve the apparent acceleration of pixels in the B-mode cycles. The acceleration images are input into a 2D two-stream network performing view classification. Østvik et. al [9] acquire the motion of the myocardium using a CNN to estimate optical flow, and filter inaccurate motions by masking estimates with a segmentation of the left ventricle. The estimates are used as part of a pipeline to automatically calculate left ventricle strain. Optical flow from 2D echocardiography were also used in [25]. Here, a 2D CNN for automatic timing of cardiac cycle phase was greatly improved by using optical flow inputs.

With a dataset of over 1×10^5 frames, calculating optical flow is costly computationally

and storage wise. One optical flow algorithm is therefore selected for further usage. Both accuracy and computational cost is considered. The accuracy is determined visually by inspection of a subset of the available data, giving a coarse estimate of the quality of the different algorithms. To keep processing time reasonable, optical flow is calculated on the resized frames. The time to calculate optical flow is benchmarked using a $30 \times 256 \times 256$ video, using 4 Intel(R) Xeon E5-2637.

In this work, two coarse-to-fine variational algorithms are considered. The first is TV-L1 optical flow [36], which is used in the original I3D flow stream. The second is a Python version of [69] named *pyflow*, a closely related algorithm. pyflow improved performance of a 2D CNN for timing of ED and ES in the project thesis.

The properties of the optical flow estimates depends on the selected parameters. Optimal parameters depends on the scale of the motions involved in the images, and the noise level of the data. With a target frame rate of 44 Hz, motions in echocardiography are small relative to the image size. By increasing the size of the smallest pyramid and using a low downsampling ratio between pyramids, smaller motions can be detected in favour of larger motions.

The heart is nonrigid, and motion estimates should vary along tissue regions for healthy patients. This is seen in strain imaging, where akinetic motion can indicate an infarction. This is different from many data sets used to evaluate optical flow, which contains large and rigid motion. Using parameters proposed for these data sets can result in too regularized flow estimates. Therefore lower regularization than proposed might be more suitable. On the contrary, too low regularization results in noisy and discontinuous optical flow due to the noise present in echocardiography. A tradeoff between noisy and small displacement estimates must be found.

TV-L1

TV-L1 is a variational optical flow formulation with the aim of minimizing

$$\int_{\Omega} \{\lambda |(I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))| + |\nabla \mathbf{u}(\mathbf{x})|\} \, d\mathbf{x}. \tag{5.4}$$

To find a solution, the brightness constancy assumption is linearized, and the equation is approximated by a convex function. The convex function is minimized using fixed-point iteration. Due to the linearization, a coarse-to-fine warping approach is used to find solutions at the coarsest scales first. The downsampling factor is fixed at 0.5 per pyramid

level, i.e. spatial dimensions are halved at each pyramid.

The Python bindings for OpenCV [70] contains a CPU implementation of TV-L1. To better estimate small-scale motions, the number of pyramids is decreased from the default of five to three. The default value of $\lambda = 0.15$ is kept, resulting in more visually pleasing results than larger or smaller values. Calculating optical flow takes $33.7s \pm 132ms$ using all CPU cores on the test sequence.

pyflow

pyflow is a variational optical flow algorithm based on [69]. The energy functional to minimize is

$$\int_\Omega \{\phi(|I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))|^2) + \alpha\psi((\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 + (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial x})^2)\}\, d\mathbf{x}. \quad (5.5)$$

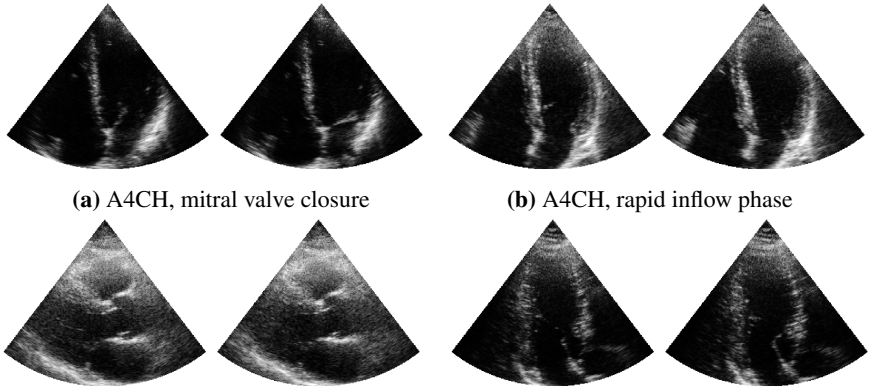
Here, $\mathbf{u} = (u, v)$ and $\psi(x) = \phi(x) = \sqrt{x^2 + 10^{-5}}$. The brightness constancy equation is then linearized, and the optical flow estimates are found using coarse-to-fine warping. Unlike TV-L1, multiple warping steps are applied at each scale, and a different numerical approach is used for solving.

A downsampling ratio of 0.8 is used, and three pyramids are constructed. The regularization parameter α is set to 0.02, much lower than the default value of 1. Calculation time of Pyflow is $7.27s \pm 25.2ms$.

Comparison

Four pairs of consecutive frames are presented for visualization of the optical flow algorithms (Figure 5.3). Frames in Figure 5.3a (A4CH) is from the time of mitral valve closure, while Figure 5.3b (A4CH) shows the rapid inflow phase. Figure 5.3c (PLAX) shows the ejection phase of systole. Figure 5.3d (ALAX) shows atrial systole phase during diastole. These frames contain different issues for optical flow, such as large movements of small regions, small movements of larger regions and noisy pixel intensities.

Optical flow for the frames is shown in Figure 5.4. The estimates are quite similar in appearance, which is not surprising given the similarities in the algorithms. It is seen that both methods capture the general motion of heart walls accurately. This is most clearly visible in Figure 5.3b, where the left ventricle expands. Estimated motions also points in right direction in general for the noisy PLAX view. The motion of the mitral valve



(a) A4CH, mitral valve closure **(b)** A4CH, rapid inflow phase

(c) PLAX, ejection phase **(d)** ALAX, atrial systole

Figure 5.3: Pairs of consecutive frames used to evaluate optical flow.

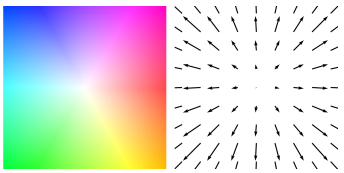
is not as accurate, especially for the TV-L1 algorithm (Figure 5.4b), first column). This can be expected, as the mitral valve is small in the A4CH view and has a relatively high velocity. The TV-L1 algorithm also has flow estimates outside the sector. This likely is due to the larger downsampling for TV-L1, such that estimates from the coarse levels are set as initial values outside the sector. Decreasing the number of pyramids reduces the leakage flow.

Due to the slight improvements of pyflow compared to TV-L1 in terms of accuracy, and considerably lower processing times, pyflow is selected for further use. To reduce storage and preprocessing further, only valid cycles after automatic QA are preprocessed, as determined by thresholds in Table 4.1. Preprocessing of these cycles took 2 days for HUNT 3 using 14 CPU-cores. Storage of 8145 recordings in HUNT3 which passed automatic QA uses 409 GB using gzip compression.

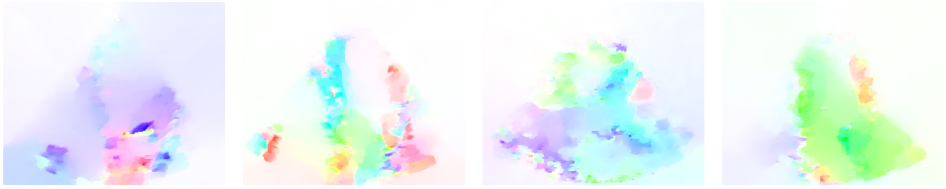
5.4.4 Intensity Normalization

The original I3D normalizes the intensities of the input data. To minimize the difference between the original input data and B-mode cycles, the same intensity normalization is used here. This constitutes normalizing B-mode intensities and optical flow vectors to $[-1, 1]$.

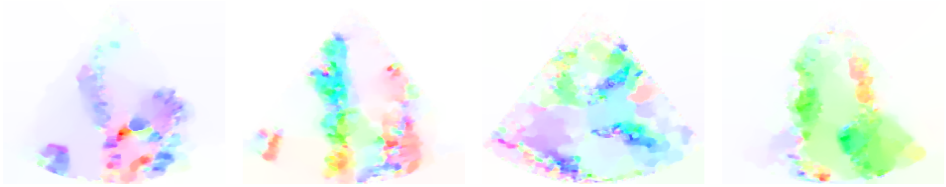
The origin of the coordinate system for the coordinate channels is set 5 cm from the transducer tip along the vertical axis, and centered on the horizontal axis. Both dimensions are measured in decimeters. This results in the domain of coordinates close to $[-1, 1]$ in the



(a) Optical flow color coding, hue indicates direction while saturation indicates magnitude.



(b) openCV implementation of TV-L1 [36].



(c) pyflow, implementation of [69].

Figure 5.4: Optical flow calculations of the frames in Figure 5.3 using two different algorithms. Images from left to right are calculated on 5.3a, 5.3b, 5.3c and 5.3d respectively.

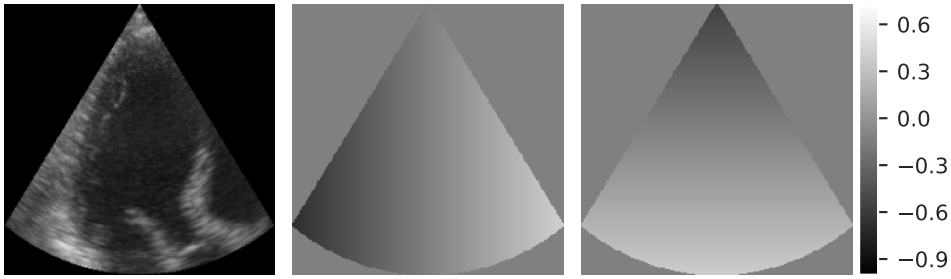


Figure 5.5: Normalized B-mode and coordinate channels.

same order of magnitude as the B-mode intensities. The normalized B-mode intensities and coordinate channels are seen in Figure 5.5.

5.5 Learning Details

5.5.1 Loss Function

Regression methods commonly minimize the sum of the squared errors. One of the motivations for the squared error is that has a simple derivative. Squaring the error results in a large loss for outliers. Outliers can occur both due to bad data and due to large variations in "heart age". To avoid outliers having too large impact, MAE is used instead such that errors are no longer squared.

5.5.2 Optimizer

Similar to the original I3D, the model is trained using SGD with momentum of 0.9. L^2 regularization is enforced in the optimizer by setting a weight decay of 10^{-7}. Different learning rates are attempted, but a final learning rate of 10^2 is used.

5.5.3 Training and Model Selection

Training is performed for 100 epochs, where an epoch is defined as an iteration through all the training data. This allows the model to converge on the training data. Checkpointing is done at the end of each epoch if the MAE is reduced on the validation data, such that

the model with the lowest MAE on the validation data is selected as the final model. The training data is shuffled at the end of each epoch.

Two nVidia Titan V GPUs with 12 GB ram are used.

5.6 Data Augmentations

The purpose of data augmentations is to increase the training data set size by modifying the data. This is especially useful when the number of samples is small. A set of augmentations is performed, some of which resulted in improved performance in the project thesis.

Augmentations are performed on the CPU in parallel to training the model on the GPUs. Care is taken to reduce the time usage of augmentations, to avoid this becoming a bottleneck during training. The augmentations are:

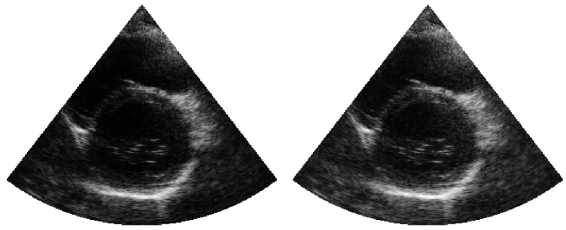
- Random additive noise
- Random rotation
- Random pad or crop

Augmentations are applied in the given order. Finally, the augmented data is resized to the model input size ($224 \times 224 \times c$).

Random Additive Noise

Echocardiography data has a high noise level compared to photographic video. With the aim making the models more robust to noise, and to reduce the chance of overfitting to speckle patterns, random noise is added during training. Speckle noise has been modeled in several ways, both additive and multiplicative, from different distributions such as the Rayleigh distribution [71]. Here, an ad hoc method for generating noise is used.

In an attempt to produce both temporally stationary noise caused by various scatterers, and completely random noise in all dimensions, the augmentation consists of adding two noise layers. One is stationary random noise, equal for all frames in a cycle. The other changes for each frame. Both are sampled from a Rayleigh distribution. The Rayleigh scale parameter is randomly selected from an exponential distribution with scale $\lambda = 0.05$, and clipped to $[0, 0.3]$. This ensures that low noise levels are most frequently produced. Each



(a) Before **(b)** After

Figure 5.6: Augmenting the training data with noise.

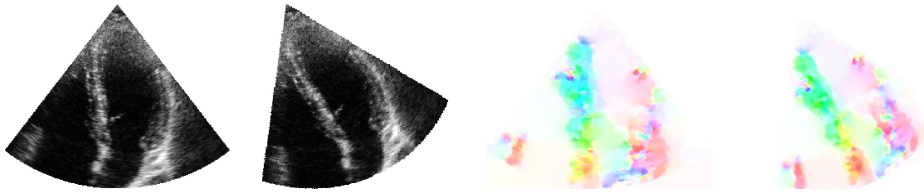
pixel is sampled individually from the Rayleigh distribution. To introduce spatial correlation in the noise, a Gaussian filter is applied. The standard deviation of the Gaussian filter is sampled randomly in $[0.8, 1.5]$, which determines the spatial spread of each speckle. The stationary and changing noise is then added to regions within the sector.

Noise is only added to B-mode inputs, as optical flow frames does not contain the same high frequent noise. An example of a frame before and after noise is shown in Figure 5.6.

Random Rotation

CNNs are not rotation equivariant, meaning that a rotation of the input does not result in equal rotation of the output, while changes in viewpoint occur naturally in echocardiography. The tilt of the transducer can vary slightly in relation to the patient, resulting in different orientations of B-mode image. To make the model more robust to changes in orientation, recordings are rotated by a random angle during training, a common augmentation in computer vision. The angle is randomly sampled uniformly from ± 25 degrees. The image size is increased to fit the rotated image, and avoid discarding pixels at the edges.

When rotating optical flow frames, extra steps must be taken to ensure that the flow vectors still describe the same displacement. Image rotation rotates the content of each frame, while preserving intensities. This results in the optical flow vectors pointing in the same direction as before rotation. To adjust for the rotation, each optical flow vector is rotated by the same angle as the image. The formula for counterclockwise image rotation with angle θ is given in Equation 5.6.



(a) Before **(b)** After **(c)** Before **(d)** After

Figure 5.7: Example of rotation by 25 degrees of B-mode and optical flow. The optical flow vectors are also rotated by 25 degrees in the same direction, as seen by changes in color.

$$\begin{bmatrix} u_{rot} \\ v_{rot} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin(\theta) & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} \tag{5.6}$$

This is a transposed rotation matrix compared to the standard rotation matrix used counter-clockwise rotations, due to the optical flow axes $[u, v]^T$ forming a left handed coordinate system.

Example of rotated frames for B-mode and flow is shown in Figure 5.7. Rotation is a computationally expensive operation involving interpolation. To reduce training time, the Python *pillow* [72] library is used for B-mode, with nearest neighbour interpolation. This package does not handle images with two channels, and expects the range of each frame to be in floating points with range $[0, 1]$ or unsigned integers in $[0, 256]$. This option is therefore unsuited for optical flow. For optical flow, the scikit-image package [73] is used with bilinear interpolation.

Random Padding or Cropping

The final augmentation considered is cropping followed by zooming. The purpose of this augmentation is twofold; to enforce focus on different features in a cycle by removing regions, and to improve robustness to variations in input dimensions.

The random pad or crop augmentation randomly removes or pads pixels at the borders of each frame. The pad or crop amount is sampled uniformly in $[-20, 20]$ for each edge, where negative equals padding, and positive equals cropping. The pad/crop amount is constant for all frames in a cycle.

Padding or cropping is followed by resizing to the fixed model input size. Resizing is performed in *pillow* with nearest neighbour interpolation for B-mode, and *skimage* with Bilinear interpolation for optical flow. An example is shown in Figure 5.8.

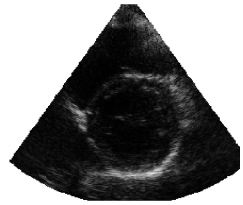


Figure 5.8: Random cropping followed by resizing.

When resizing optical flow images, the optical flow magnitudes must be corrected by the changing scale. The correction is given by

$$\begin{bmatrix} u_{res} \\ v_{res} \end{bmatrix} = \begin{bmatrix} \frac{w_{res}}{w} u \\ \frac{h_{res}}{h} v \end{bmatrix}, \tag{5.7}$$

where h, w are the height and width of the images, and $(_{res})$ indicates the new values after resizing $(_{res})$.

5.7 Comparison Model: OLS Linear Regression

HUNT 3 has been thoroughly analyzed, and a large number of measurements are available for each patient. To have a reference for comparing the accuracy of the deep learning models, a multiple linear regression is fit to predict age from these measurements. Ordinary least squares (OLS) with *statsmodels* [74] is used to fit the regression, i.e. minimizing the sum of squared residuals. A subset of the available variables is manually selected, for which the effects of age is well documented. To ensure fair comparison with the deep learning model, only variables acquired through echocardiography are used. The selected variables with units and image modalities used for measurements are:

IVSs Systolic interventricular septum thickness (mm), M-mode

IVSd Diastolic interventricular septum thickness (mm), M-mode

IVRT Average isovolumic relaxation time (ms), PW-Doppler

LVPWd Diastolic LV posterior wall thickness (mm), M-mode

LVPWs Systolic LV posterior wall thickness (mm), M-mode

E/A Mitral valve early E/A ratio (1), PW-Doppler

GLS 16 segments model absolute longitudinal systolic global strain (%), speckle tracking and PW tissue Doppler

DT Mitral valve deceleration time (ms), PW-Doppler

e' 4 wall mean early diastolic mitral annular velocity (cm/s), PW tissue doppler

a' 4 wall mean late diastolic mitral annular velocity (cm/s), PW tissue doppler

MAPSE 4 wall mean Mitral annular plane systolic excursion (cm), M-mode

All measurements except global strain have been calculated previously using EchoPAC (GE Vingmed, Horten, Norway). Global strain was calculated in [46] using customized software. The Pearson correlation between different variables is shown in Figure 5.9. All features correlate with age to some extent, with the largest absolute correlations being the E/A ratio and the e' velocity. The E/A ratio is also highly correlated with the e' and a' velocities. Measurements of the same dimension at diastole or systole (IVSs, IVSd and LVPWs, LVPWs) are also highly intercorrelated. The correlations between the independent variables results in multicollinearity, such that the effect of the correlated variables becomes entangled. However, this is not an issue for the accuracy of the regression.

The HUNT 3 training is used for model fitting, while the testing set is used for evaluation. Patients missing one or more of the variables are removed. Each feature is scaled to zero mean and unit variance as calculated on the training set.

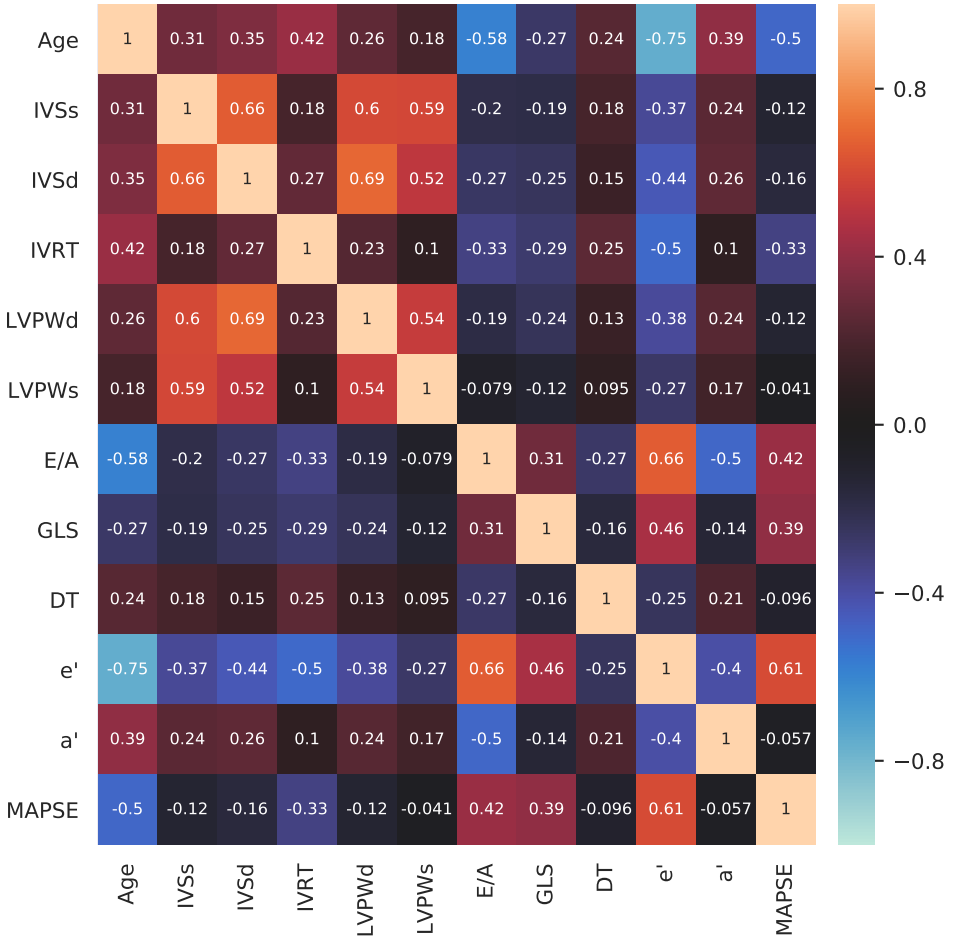


Figure 5.9: Pearson correlation between variables used for linear regression of age in HUNT 3.

6 | Age Estimation Results

The error is defined as chronological age minus estimated age, i.e. $e = y - \hat{y}$. The mean and standard deviation of the error is denoted μ_e, σ_e respectively. Errors (MAE, μ_e, σ_e) are given in years. MAE and R^2 is calculated using *scikit-learn* [75].

Results per patient for different approaches are presented. For comparison, mean value prediction using ages from from HUNT 3 represents the lower bound on performance, as all models are trained on HUNT 3. Although estimates are generated from individual cycles, reporting results for each view and model produces too many results to feasibly compare. Results are therefore only presented for the averaged estimates as described in Section 5.2. One model is selected for further analysis. Results for each individual cycle are presented in the Appendix.

6.1 Model Descriptions

The approaches with abbreviations are as follows:

HUNT3-Mean Predicting the mean age of cycles in the HUNT 3-training set.

OLS Ordinary least squares linear regression from clinical indices in HUNT 3.

Single Five single view I3D models (one for each view), B-mode inputs and integer temporal downsampling.

Multi Multiple view I3D, B-mode inputs and integer temporal downsampling.

Multi-Random Multiple view I3D with randomly initialized weights, B-mode inputs and integer temporal downsampling.

Multi-Flow Multiple view I3D, optical flow inputs, integer temporal downsampling.

Multi-Coords Multiple view I3D, B-mode and xy-coordinate channel inputs, temporal interpolation.

Multi-Coords-Small Smaller multiple view I3D, randomly smaller weights, B-mode inputs and xy-coordinate channel inputs, temporal interpolation.

All models are trained using HUNT 3-train only. The MAE on the validation set of HUNT 3 for each view are given in Table 6.1. These are not used for further analysis, but are intended for weighing each estimate as described in Section 5.2.

Table 6.1: MAE of models on HUNT 3-validation for the different views. This is used to weight each prediction when averaging estimates for a patient.

Model	A4CH	ALAX	A2CH	PLAX	PSAX
Single	5.42	5.58	5.74	4.96	6.43
Multi	5.15	5.71	5.58	4.89	6.45
Multi-Random	5.90	6.21	5.90	5.30	6.67
Multi-Flow	5.62	6.04	5.60	5.00	6.30
Multi-Coords	5.16	5.57	5.49	5.33	6.36
Multi-Coords-Small	6.02	6.37	6.06	6.44	6.94

6.2 Model Comparisons on all Datasets

Here, results for each model and each dataset is presented.

Table 6.2 shows how the models performs on the HUNT 3-test split, including mean value prediction and OLS regression.

Table 6.2: Results for different models on the HUNT3-test split. μ_e is the mean of the error, σ_e is the standard deviation of the error.

Model	N	MAE	μ_e	σ_e	R^2
HUNT3-Mean	186	12.5	2.11	14.6	-0.0212
OLS	172	7.21	0.826	9.03	0.605
Single	186	4.93	1.85	6.25	0.801
Multi	186	4.89	0.802	6.25	0.815
Multi-Random	186	5.29	2.61	6.48	0.77
Multi-Flow	186	4.77	**0.0665**	6.12	**0.825**
Multi-Coords	186	**4.66**	1.80	**5.94**	0.820
Multi-Coords-Small	186	5.88	0.943	7.30	0.747

All models perform significantly better than mean value prediction, i.e. using only prior information. The HUNT 3-mean MAE of 12.5 is reduced to 7.21 with linear regression using age-affected indices. Further reductions in MAE are seen in all deep learning models. The non-pretrained small I3D performs significantly better than the OLS regression model, and an R^2 of 0.747 already suggests a good fit, considering the variation in the data. Pretrained models perform even better than the randomly initialized models. This suggests that there is a benefit in using weights pretrained on photographic videos for echocardiography, even though there are large differences between the domains. Interestingly, the small randomly initialized model with coordinate channel inputs performs noticeably worse than the large randomly initialized model, while the large pretrained coordinate channel model appears to improve performance. Even though echocardiography recordings are sparse in information and spatio-temporally correlated, 12M parameters are not too much to represent aging.

Results for NTNU-LVD are given in Table 6.3 and 6.4. All models perform significantly worse for this dataset. This can in part be explained by the difference in age distribution between the datasets. With an older distribution of patients in NTNU-LVD, models are interpolating with little data for some patients, and extrapolating for patients older than the highest age in HUNT 3. Models have similar MAE and standard deviations for NTNU-LVD (H) and (D), while the mean error is increased and the R-squared is lower for the healthy patients with LVD. This can be explained by the differences in age distributions seen in Figures 3.4 and 3.4. The healthy patients has a higher mean age than the diseased patients, increasing the difference between the HUNT 3 train split. In addition, there is a smaller standard deviation in the age of the healthy patients, further reducing the R^2. However, considering the significantly higher standard deviation of models on NTNU-LVD, there might be some other differences between the data sets resulting in higher errors. HUNT 3 and NTNU-LVD has a large overlap in the scanners used and the sector widths and depths, such that the coordinate channel models should perform similar for the two data sets. The fact that the coordinate channel models perform worst of all models on this data backs the suspicion of differences in the input data.

Also note that the number of patients used are different for the coordinate channel models. This is most likely caused by the automatic QA scores being different because of temporal interpolation, resulting in differences in automatically discarded data. The differences in test data makes comparison less straight forward, and training a model using coordinate channels on the same data as the other models would be better for comparison.

All models still perform significantly better than predicting the mean value of the training

Table 6.3: Patient results on NTNU-LVD (H).

Model	N	MAE	μ_e	σ_e	R^2
HUNT3-Mean	82	19.9	17.2	14.2	-1.46
Single	82	11.2	7.88	11.2	0.06
Multi	82	**9.54**	5.79	**10.5**	**0.294**
Multi-Random	82	10.6	5.33	12.0	0.14
Multi-Flow	82	9.90	**4.87**	11.7	0.206
Multi-Coords	73	11.4	8.44	12.0	0.00756
Multi-Coords-Small	73	12.8	9.73	13.3	-0.250

data. The multiple view B-mode and optical flow models continue to be the better performing models on NTNU-LVD. Again, pretrained models perform better than randomly initialized models. This suggests that models have learned features that describe aging to some extent for NTNU-LVD.

Table 6.4: Patient results on NTNU-LVD (D).

Model	N	MAE	μ_e	σ_e	R^2
HUNT3-Mean	94	17.8	12.6	16.3	-0.599
Single	94	10.3	5.67	11.1	0.414
Multi	94	**8.91**	2.85	**10.3**	**0.572**
Multi-Random	94	10.3	3.45	12.0	0.406
Multi-Flow	94	9.54	**2.84**	11.2	0.500
Multi-Coords	80	10.6	6.72	11.2	0.360
Multi-Coords-Small	80	12.3	8.11	12.8	0.133

Results for UNN-CAD is shown in Table 6.5 and 6.6. All models perform better predicting the HUNT3-mean value. However, standard B-mode models perform much worse than the coordinate channel and optical flow models here. This might be due to the differences in scanners used and sector sizes between the data sets. The optical flow input is more regularized, leaving less room for overfitting to a specific data set. Also, note that most methods perform worse than predicting the mean value of UNN-CAD (D) on UNN-CAD (D), with a negative R^2. In addition, the MAE of models is smaller than NTNU-LVD. The negative R^2 is again partially explained by a small variation in the age of UNN-CAD (D).

Table 6.5: Patient results on UNN-CAD (H).

Model	N	MAE	μ_e	σ_e	R^2
HUNT3-Mean	198	11.4	9.21	10.1	-0.825
Single	198	7.85	6.49	6.88	0.131
Multi	198	8.19	6.68	7.21	0.0624
Multi-Random	198	9.11	7.64	7.86	-0.166
Multi-Flow	198	5.89	**3.20**	6.88	**0.441**
Multi-Coords	193	**5.87**	3.84	**6.59**	0.421
Multi-Coords-Small	193	6.03	3.26	7.32	0.360

Table 6.6: Patient results on UNN-CAD (D).

Model	N	MAE	μ_e	σ_e	R^2
HUNT3-Mean	48	13.0	12.3	8.78	-1.96
Single	48	8.04	7.35	7.85	-0.49
Multi	48	8.00	7.44	**7.34**	-0.415
Multi-Random	48	9.33	9.06	7.43	-0.779
Multi-Flow	48	**6.55**	**5.05**	7.39	**-0.0363**
Multi-Coords	47	6.88	5.26	7.90	-0.144
Multi-Coords-Small	47	6.87	5.71	7.61	-0.150

6.3 Further Analysis of Selected Model

The Multi-flow I3D is selected for further analysis, which performed well on most datasets. Other models were also inspected, but similar trends as the ones presented here were observed.

6.3.1 Results per View

Measurements for each cycle of the data sets are shown in Table 6.7 to 6.11. The most consistently performing view is PLAX, which maintains a positive R^2 for all data sets. This is followed by A4CH and A2CH views. Interestingly, the ALAX view performs well on HUNT 3, but poorly on all other datasets. The opposite occurs for the PSAX view, performing fairly well on the UNN-CAD datasets. It must be noted that results reported on individual cycles do not have sample independence due to multiple cycles coming from each recording and patient. This could be mitigated by only selecting one measurement per patient or cycle, at the expense of fewer samples for evaluation.

The averaging method for each patient improves performance when considering the ac-

curacy for most cycles. However, averaging performs slightly worse than using PLAX esimates for NTNU-LVD and UNN-CAD, due to poor performance of other views.

Table 6.7: Multi-flow model on cycles in the HUNT 3-test set

View	Num. cycles	MAE	μ_e	σ_e	R^2
A4CH	477	5.57	-0.393	6.86	0.771
ALAX	475	5.50	0.224	7.04	0.751
A2CH	655	5.84	-0.592	7.41	0.716
PLAX	481	4.82	-0.482	6.24	0.816
PSAX	1078	6.01	0.407	7.57	0.707

Table 6.8: Multi-flow model on cycles in NTNU-LVD (H)

View	Num. cycles	MAE	μ_e	σ_e	R^2
A4CH	484	10.3	3.02	13.3	0.202
ALAX	143	11.6	7.19	12.8	-0.0914
A2CH	373	10.1	4.66	12.5	0.158
PLAX	119	8.84	2.27	11.3	0.276
PSAX	237	12.5	6.2	14.18	0.0707

Table 6.9: Multi-flow model on cycles in NTNU-LVD (D)

View	Num. cycles	MAE	μ_e	σ_e	R^2
A4CH	481	9.26	2.25	11.1	0.476
ALAX	140	10.5	3.95	11.9	0.301
A2CH	339	10.5	4.36	12.2	0.421
PLAX	130	8.32	-0.703	10.4	0.516
PSAX	220	10.8	4.15	13.0	0.317

Table 6.10: Multi-flow model on cycles in UNN-CAD (H)

View	Num. cycles	MAE	μ_e	σ_e	R^2
A4CH	340	6.59	3.12	7.71	0.293
ALAX	423	7.27	4.70	7.98	0.134
A2CH	538	7.12	3.65	8.24	0.190
PLAX	453	6.09	1.40	7.95	0.372
PSAX	12	4.91	-1.26	5.35	0.325

Table 6.11: Multi-flow model on cycles in UNN-CAD (D)

View	Num. cycles	MAE	μ_e	σ_e	R^2
A4CH	76	7.78	6.56	8.50	-0.727
ALAX	87	7.44	6.17	8.31	-0.390
A2CH	121	7.63	6.12	8.53	-0.370
PLAX	90	5.79	3.30	6.32	0.502
PSAX	3	2.13	-0.962	2.44	0.00

6.3.2 Bland-Altman for Patient Estimates

Bland-Altman plots are generated for patient estimates, shown in Figures 6.1 to 6.5. It can be seen that chronological age correlates with estimated age, and that the model has a low bias for patients aged between 50 and 60. However, the bias increases with age, a trend seen in all Bland-Altman plot, as the correlation is not strong enough to accurately predict age for older patients.

It is difficult to say how much of the error is due to cardiovascular differences between patients and how much is caused by differences in data acquisition or by modeling errors. The resolution of imaging that can be achieved is limited, resulting in an upper bound on age estimation accuracy, but the increasing error with age seen in the Bland-Altman plots show that this is not reached yet.

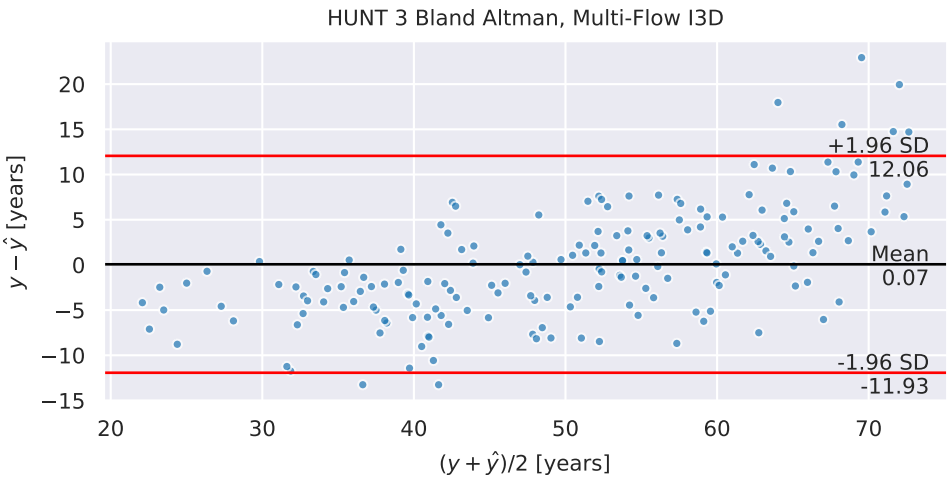


Figure 6.1: Bland-Altman plot for each patient in the HUNT 3 test-set

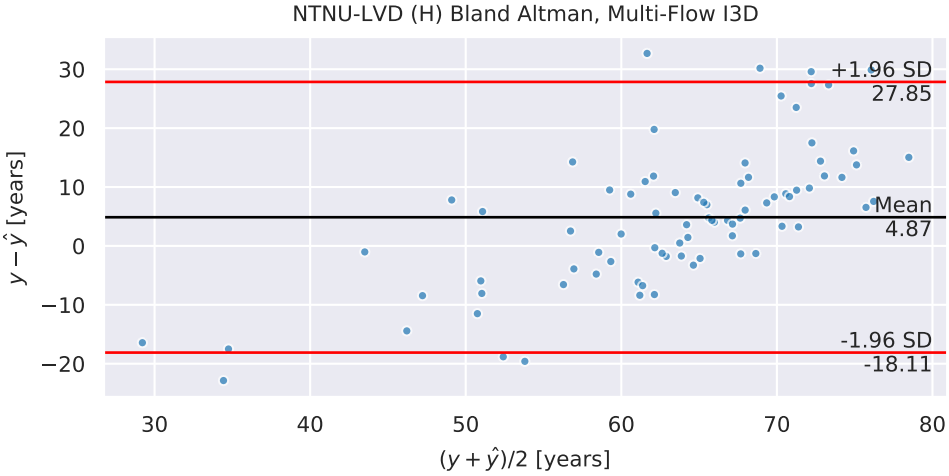


Figure 6.2: Bland-Altman plot for each patient in NTNU-LVD (H)

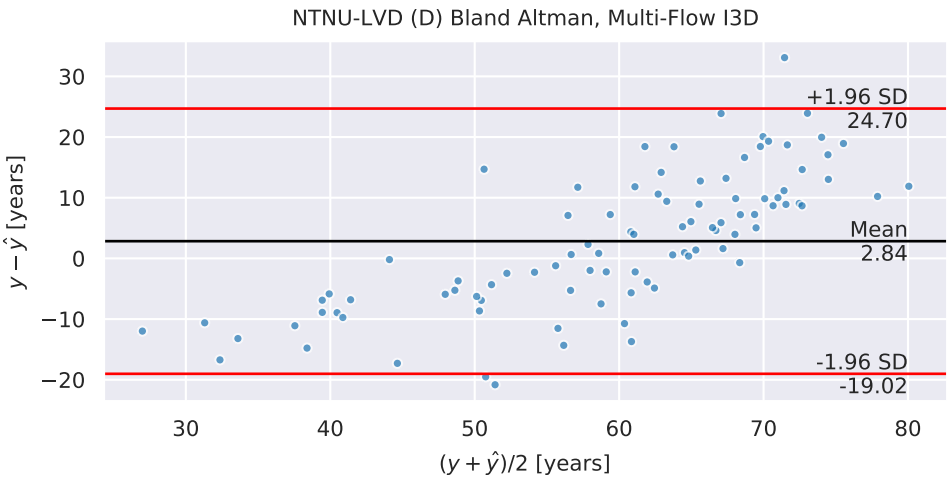


Figure 6.3: Bland-Altman plot for each patient in NTNU-LVD (D)

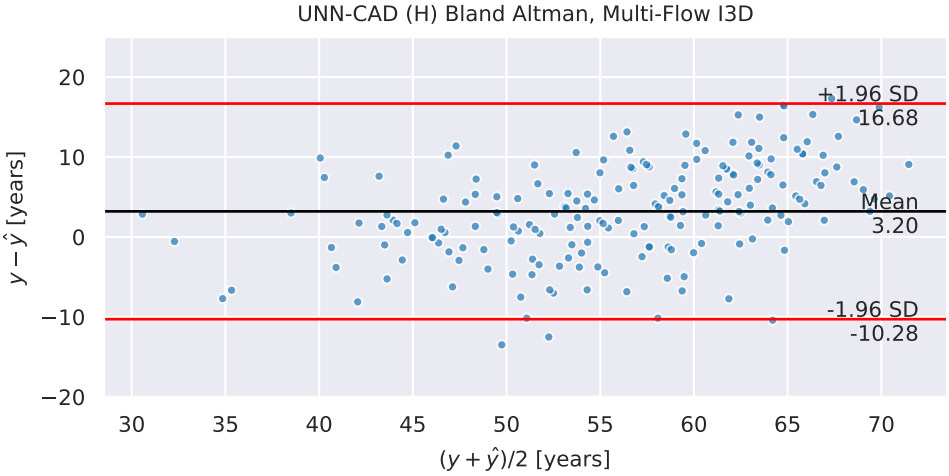


Figure 6.4: Bland-Altman plot for each patient in UNN-CAD (H)

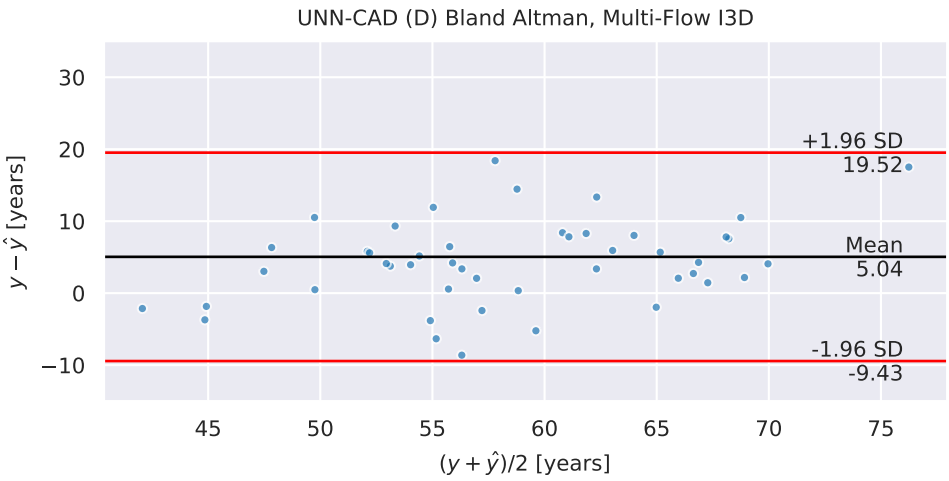


Figure 6.5: Bland-Altman plot for each patient in UNN-CAD (D)

6.3.3 Guided Backpropagation

Examples of sequences with saliency and age estimates are shown in Figure 6.6. Guided backpropagation (red) is overlayed on the B-mode frames, with saturation indicating saliency magnitude. Saliency reveals that the regions of heart valves are important. Especially salient are the regions near the mitral valve, as can be seen in the plots of A4CH, ALAX, PLAX and to a lesser extent A2CH. This corresponds to the well established effects of aging on the mitral region, which is also seen in the correlation between age and measurements in the mitral region (E/A, DT, e', a', MAPSE, Figure 5.9). The mitral valve is not clearly visible in the aortic and midpapillary level of the PSAX view, which might help explain why the models perform worse in this view. The tricuspid valve is slightly highlighted in the A4CH sequence, while the same is true for the aortic valve in the ALAX and PLAX views. For the PSAX view, which to the authors knowledge is from the midventricle level, the IVS is the salient region. This corresponds with the effects of aging on the IVS (IVSs, IVSd, also seen in Figure 5.9). Similar saliency patterns were also observed for other models. Appendix 8.3 shows guided backpropagation for the Multi-I3D.

Guided backpropagation, especially saliency magnitudes must not be overinterpreted. As deep CNNs are highly nonlinear, the results are only valid for a small increase in pixel intensities. As saliency is the gradient of the output with respect to every input pixel, saliency shows the effect of changing each individual pixel independently from the effect on neighbouring pixels. The effect of changing multiple pixels simultaneously is not re-

vealed. Saliency is therefore likely to be larger for small objects such as the valves, where a change in pixels results in a larger relative change in the image compared to a pixel change in larger regions such as the walls.

6.3.4 Sex Differences

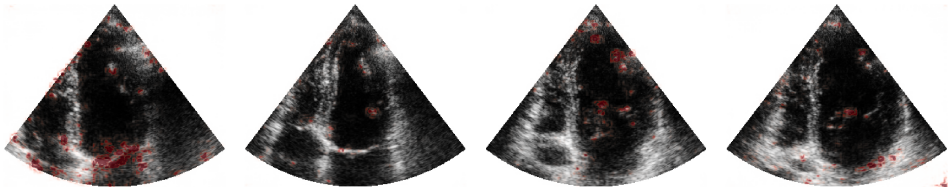
Due to not inputting any other information than the pixel data and cycle view, sex has not been considered as a feature. To assess whether sex related differences occur in the age estimates, estimated age versus chronological age of patients in HUNT 3-test are plotted and colored by sex. The result is shown in Figure 6.7. There is little difference in estimates between men and women for most age groups. This makes sense, as the models does not have information about sex during training, and must therefore learn features that work for both men and women It could also suggest that differences in aging between sexes are not prominent. For the oldest patients, consisting solely of women, an underestimation of chronological age is made. There are few male patients in this age group for comparison, such that assessing whether this is a random pattern or due to some sex-specific differences is difficult.

6.3.5 Healthy vs. Diseased Patients

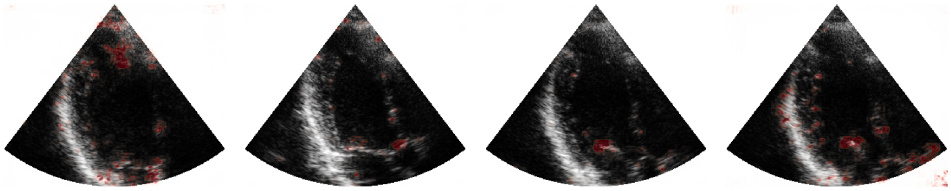
To explore whether there is a difference between predicted age and chronological age for patients with diagnosed LVD or CAD, the estimates for healthy versus diseased patients is plotted. This is seen in Figures 6.8 and 6.9.

The large deviation in estimates for NTNU-LVD in comparison to HUNT 3 and UNN-CAD also suggest that there is a difference in terms of the input data for NTNU-LVD. The differences might be methodological, for example a focus on the left ventricle in NTNU-LVD compared to the other data sets. The differences in data is apparent in the results for each cycle, Tables 6.7 to 6.8, where each data set has a different distribution of views. For NTNU-LVD there is an overweight of A2CH and A4CH views, as labeled by the view classifier. This is not to say that the difference in distributions of views is the reason for the decline in performance, but it indicates that there are substantial differences in the acquisition of the different different data sets.

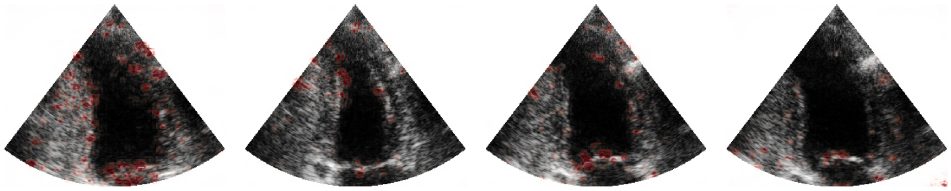
Both plots show little or no separation between healthy and diseased patients in terms of estimated age. The hypothesis of differences in estimated age between healthy and diseased patients is therefore not correct with the data and setup used here. This means



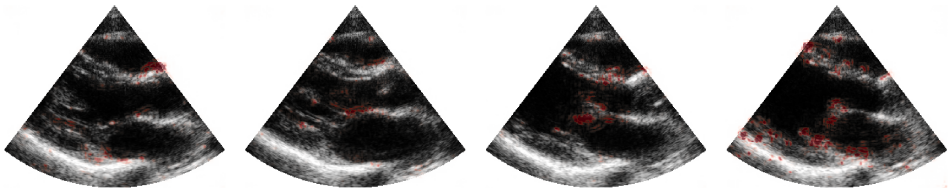
(a) View: A4CH, Age: 66 years, Predicted age: 65.9 years



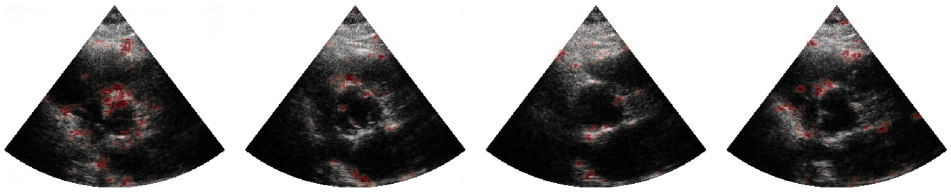
(b) View: ALAX, Age: 24 years, Predicted age: 31.4 years



(c) View: A2CH, Age: 70.5 years, Predicted age: 75 years



(d) View: PLAX, Age: 20 years, Predicted age: 23.9 years



(e) View: PSAX, Age: 56 years, Predicted age: 57.2 years

Figure 6.6: Predictions and guided saliency (red) for the Multi-flow I3D on cycles in the HUNT3 test set. B-mode frames are shown instead of the optical flow input frames.

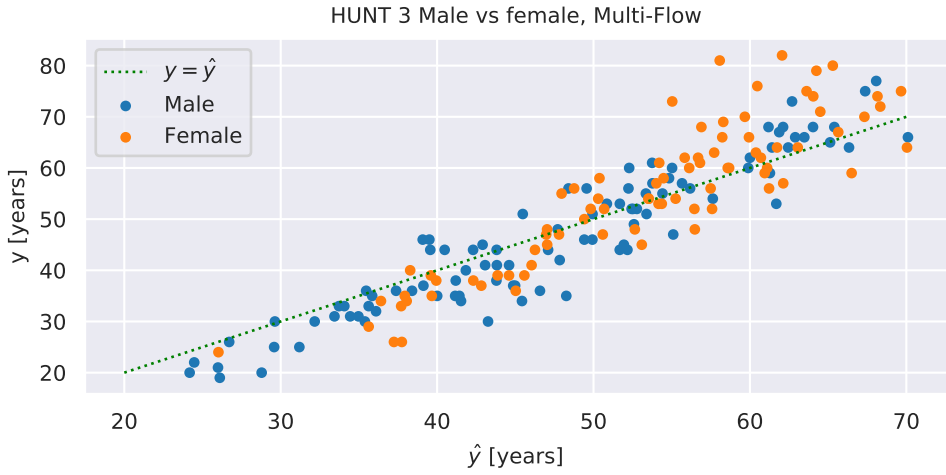


Figure 6.7: Chronological vs. estimated age by sex in HUNT 3 for the multi-flow model.

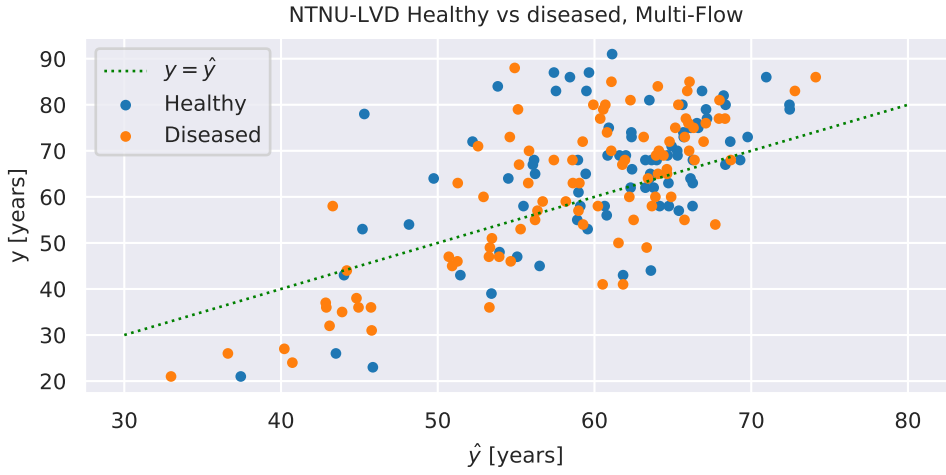


Figure 6.8: Chronological vs. estimated age in NTNU-LVD.

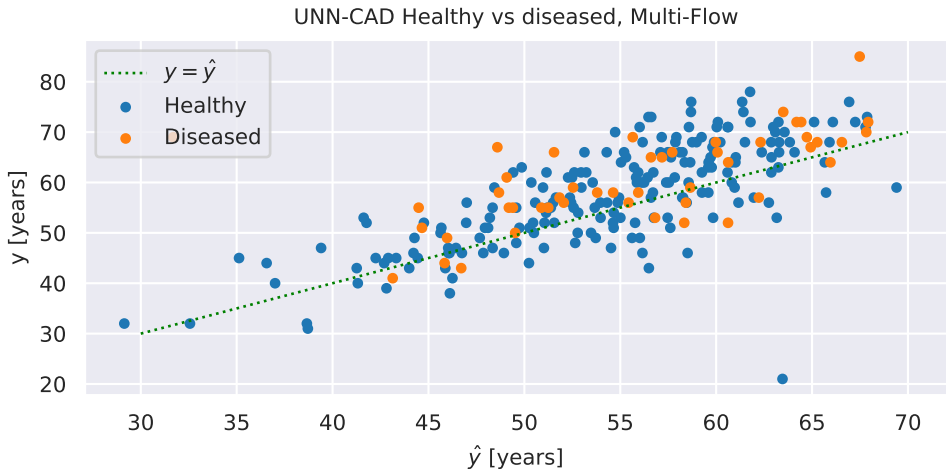


Figure 6.9: Chronological vs. estimated age in UNN-CAD.

that the models have learned mappings to age that are not impacted by disease. For systolic dysfunction, the largest group of diseased patients, this can be explained by the inclusion criteria. As ejection fraction does not change with age, there is no difference in diagnosed patients based on the inclusion criteria alone. This does not explain the lack of distinction for patients with diastolic dysfunction. Key points to be made here is that the model is either using other features to learn age than used for diagnosing diastolic dysfunction (a', e', etc.) or that the variance of the NTNU-LVD estimates is too large to accurately separate the groups.

For UNN-CAD, the lack of separation between healthy and diseased patients can be explained by signs of stenosis not being apparent enough in the echocardiography recordings to make a difference between healthy and diseased patients. This is supported by the aim of UNN-CAD, being to determine exactly if the diseased patients can be accurately diagnosed from echocardiography.

Due to little separation between the patients, no further analysis is performed to quantify the separability between healthy and diseased.

7 | Discussion

7.1 Data Sets

Differences in age distributions between the data sets impacts the results substantially. Results, especially R^2, is highly sensitive to the distribution of the chronological age contained in the data sets. As models are trained to minimize the MAE over the age distributions of HUNT 3-train, the models are less accurate for patients far from the mean age. The explanation for this is that with a close to normally distributed age distribution, a lower emphasis is placed on younger and older patients.

Differences in data acquisition between the data sets can be caused by several factors. HUNT 3 is highly standardized, with one echocardiographer performing all examinations using the same scanner, each recording containing 3 cycles, and the five views considered here being recorded for all patients. Acquisition in NTNU-LVD was performed by multiple experts, using both Vivid E9 or Vivid 7 scanners, although all data were approved by a single cardiologist. In addition, the study was designed to detect left ventricular dysfunction, which might change the regions imaged compared to a general examination. UNN-CAD is acquired using a Vivid E9, a newer scanner than used in HUNT 3, likely resulting in differences in data quality. In addition, UNN-CAD contains different frame rates and sector sizes. This affects the pixel spacing, likely contributing in worse performance. In HUNT 3, the general population is invited to participate, selecting an approximately random sample of the population in North Trøndelag. For the other data sets, patients are not a random sample of the population, but are referred to echocardiography examinations due symptoms of disease. In other words there is a bias in the data sets, even between healthy patients in HUNT 3 and healthy patients in NTNU-LVD and UNN-CAD.

Both disease studies contain general diseases, with many possible symptoms. In NTNU-

LVD, patients are also labeled with more specific types of dysfunction, such as systolic, diastolic, or high blood pressure related effects. A more detailed separation of patients could be performed, for example by only considering patients with diastolic dysfunction. This however reduces the number of patients considered, while increasing the number of tested hypotheses tested.

7.2 Automatic QA

As automatic QA is the first step of data selection, the results of the age estimation models is evidently affected by the choices here. The method presented has several possibilities for alteration.

7.2.1 Cycle Separation

Only ECG was considered for cycle separation. For recordings where ECG is unavailable, the cycle separation algorithm will not work. Instead, methods for cardiac cycle phase separation from echocardiography can be used, such as the timing model used to generate a timing score. Developing an algorithm without the use of ECG is not required for the data sets considered, but will likely introduce more variance in the cycle separation.

7.2.2 View Classification

Performing view classification is a well justified step, as it is done intuitively by the practitioners and as a data selection step in other deep learning approaches. View classification likely positively affected the results, considering the increased accuracy of the multiple view models and differences in accuracy by view. On the contrary, the view classification model is small compared to the I3D models, such that the age estimation models might be able to determine the view of the input as part of the computation. This also removes errors caused by misclassified views. One can therefore argue that the view classification step is unnecessary. However, the misclassification rate is low, and the view classification step makes the learning task easier as the age estimation model does not have to learn view classification simultaneously. This is important given limited amounts of data.

7.2.3 Quality Measures

The models used are only two out of many possible domain-specific models that can be used to extract quality measures. Other examples include segmentation CNNs, or even the error of the age estimation models. With any quality score based on machine learning, the usefulness of the quality score depends on how accurate the model is. This in turn is dependent on the model architecture, and the size and quality of the training data.

The view classifier quality score is unevenly distributed, due to the model being trained to minimize cross-entropy. In this case, basing scores on percentiles resulted in small thresholds for the quality scores, e.g. 0.00010 for the ALAX view. A more evenly distributed view classification quality score might be more suitable for increasing the separation between good and poor quality. Another interpretation of the skewness of the quality scores is that most cycles in HUNT 3 already have a high quality, which is why the confidence of the view classifier is high.

The use of the timing model for generating quality scores is a novel idea. The timing quality score is more evenly distributed. Saliency for the timing model in [25] also show similar regions as the models for age estimation, especially the mitral valve and inner heart walls. Therefore, using the timing model as a quality score is likely to remove samples where the mitral valve or wall motion is poorly represented, which increases the focus of the age estimation model on these regions.

Quantile Thresholds

Using fixed thresholds at the 90th percentiles in both quality measures might seem overly cautious. As discarding is performed if any error measure is above the given threshold, the fraction of discarded data lies between 0.1 and $1 - 0.9^2 = 0.19$ depending on the correlation of the measures. The fraction for HUNT 3 is approximately sixteen percent, as the quality scores are uncorrelated. This is a significant reduction in sample size, and is likely to remove many useful samples, especially when considering that HUNT 3 is already standardized. However, finding optimal thresholds is difficult without data where the quality is labeled. The upside is that a high discard rate reduces the time required for training, allowing for faster experimentation. Thresholds can then be relaxed once an appropriate model has been found. The importance of removing data with lower quality is more apparent for optical flow where computational and storage constraints increase, and flow estimates are worsened by noisy data.

An alternative to discarding samples at a fixed threshold is to weight the estimates and losses as a function of the quality score. This way, all data can be used, but data with an estimated high quality has a larger impact on the loss than poor data. This increases data set size, and can possibly improve the performance and generalizability of the models using the data. In this case, an appropriate weighing function must be found from the quality scores. The time usage of training and testing would also increase, and it is not guaranteed that the additional data contains enough useful information to yield any significant improvements in performance.

7.3 Age Estimation Setup

A large number of alternatives are available to the methods presented here. Although the data sets are fairly large, the number of hypotheses tested must be limited to reduce the chance of making hypothesis testing errors. The differences between the models are intentionally very small to only evaluate the effect of the modification, but optimal hyperparameters will also change as a result of the modifications. This means that the observed differences in performance can be partially caused by the selected hyperparameters. To exemplify, randomly initialized models might achieve better performance than the pretrained models when using another optimizer, learning rate or batch size. This makes it harder to draw conclusions, especially if the differences in performance is small. Given the time and resources, each change could be cross-validated evaluated over a wide range of hyperparameters.

7.3.1 OLS Linear Regression Model

Due to aging being a complex process which likely does not affect the measurements linearly, nonlinear feature transformations or more complex regression models could have improved the results of the OLS model. In addition, only a small set of measurements are used. More measurements correlating with age could also have been included to increase the accuracy. These options increase the chance of overfitting.

7.3.2 Averaging Patient Estimates

Averaging reduces the impact of outliers, which can occur due to a number of factors such as poor data quality, errors in the automatic QA step, and inability to accurately model the

given sample. As seen in the various plots of the multi-flow model, there are few considerable outliers. The value of weighted averaging is seen in the results for the different views, for example Table 6.7. There are more than twice as many cycles in the PSAX view for HUNT 3, while the errors are significantly larger. With unweighted averaging, the PSAX cycles would affect the estimates more due to quantity. The averaging method however has several parameters which could be optimized. Further averaging could also have been used, for example by averaging estimates from multiple models.

7.3.3 Multiple View Models

It might not be surprising that the multiple view models performs better than the five single view models, considering the fivefold reduction in parameters. For a smaller model, where the number of parameters are too few to learn a mapping for all views, or the number of samples from a single view is sufficient, it might be better to only use a single view.

A middle ground between the single model per view and the multiple view models was also considered. By splitting the model into separate *branches* for each view somewhere in the network, more parameters are available for learning view-specific transformations while reusing early layers. A split was attempted after the 5th Inception block, i.e. the weights until the last row of Figure 5.1 are shared between views. Unfortunately, the whole model with five branches was not able to fit into GPU memory during training. To avoid computing the output for all branches and thereby reducing memory consumption, only the output of the correct branch could be dynamically calculated. To avoid switching to a dynamic graph framework such as PyTorch or Eager Tensorflow or writing custom training loops, this was not performed.

The disadvantage of the multiple view model is that it does not extend nicely to inputting multiple features, such as gender, age or weight. As discussed previously, this issue can be avoided by simply having a single output for all views as in [25]. Another alternative is to input the view as a feature somewhere in the network. 3D convolutional layers are not ideal for scalar features, such that in this case the features should likely be input into a more suitable layer such as a fully-connected layer.

7.3.4 Optical Flow Models

The optical flow models perform better than B-mode, although optical flow is generated from B-mode. In other words, no new information is present in the optical flow data.

This show that data representation does matter in deep learning for echocardiography, and that using raw inputs are not always the best option. One might however argue that the optical flow models perform better due to the data being more similar to the task of the pretrained weights, compared to B-mode. Considering the success of the optical flow model, a possible option for improvement is using strain imaging instead of optical flow inputs.

The task is close to speckle tracking, which is commonly used in echocardiography quantification. Speckle tracking typically involves human interaction and manual finetuning over the regions of interest, and could therefore result in better motion estimates compared to unsupervised optical flow algorithms. However, speckle tracking typically requires high quality data for accurate tracking, and manual interaction is cumbersome for large data sets.

7.3.5 Coordinate Channels

Convolutional layers are position equivariant, such that translation of an input results in equal translation of the output. This equivariance might not be optimal for standardized views in echocardiography, where structures are normally found in a limited image region. By including coordinates, the convolutional layers can learn position dependent transformations. For example, in the A4CH view, the septum is most likely contained within a small region. This means that convolutional layers can learn features specific septum earlier by using the position to infer that the pixels in the input is from the right wall, instead of requiring a large field of view to determine position.

A temporal coordinate channel can also be appended. This enables each convolutional layer to easier determine which part of the cycle the current data belongs to. This is likely even more useful for data that is not temporally interpolated. This is left to further work.

In addition to coordinate channels, resizing recordings to a fixed pixel spacing was also considered. In this case, a target pixel spacing must be set. With a fixed spacing, a trade-off occurs between the amount of cropping of data with larger sector sizes and the amount of padding to apply to data smaller than the fixed input shape. For smaller sector sizes where padding is used, less information than what is possible is input to the network. This is also true for larger sector sizes where cropping is used such that data is removed. Secondly, data augmentations involving resizing can no longer be applied, as it would break the fixed spacing assumption.

7.3.6 Checkpointing

It might have been useful to apply a similar weighting as performed in testing while training. This would result in improving accuracy per patient in the validation set, as models would be selected based on minimizing the error per patient, instead of improving average performance over all cycles. However, batches consist of randomly selected cycles such that more than one cycle from the same patient in a batch is not the norm.

Although checkpointing results in the best performance for cycles in the validation set, it likely reduces generalizability as 100 evaluations of the validation data are taken for each model. In fact, the optical flow model performing best on average has higher validation error than the other pretrained models. This might not have been the ideal way to select model parameters in retrospect. Hopefully, with nearly 200 patients for validation, the effect is mitigated.

7.3.7 Data Choices

Input Length

An alternative to zero padding shorter cycles is to loop the input data. This is appropriate given that the start and end frame of the input to be looped are the most similar. Looping still results in a discontinuity between the start and end frames, due to cycle variability and the motion of the transducer and patient. Alternatives which prevents temporal discontinuities is to use other padding operations common in image processing, such as reflection or replication. Finally, given that more frames are available before the start of or after the end of the cycle, these frames can be included. However, as the convolutional layers of I3D already use zero padding to obtain equal input and output shapes, the discontinuity will occur in the model either way.

Optical Flow Calculations

Although the optical flow model performs better than the other approaches, the computational cost of estimating optical flow increases the latency significantly. If computational cost or time usage is important, one of the B-mode alternatives might be preferable.

Whether the optical flow magnitudes are correct is difficult to evaluate. The selection of optical flow algorithm and parameters is based on visual inspection. A more rigorous

but time consuming method would be to acquire ground truth motion of some regions for comparison, either by manual labeling or speckle tracking.

The optical flow assumptions of smooth motion does not hold for echocardiography due to the nonrigid movements. This might make an optical flow algorithm without these assumptions work better. Recently, deep learning approaches have also promised state-of-the-art results for optical flow estimation. Therefore, PWC-Net, one of the best performing deep optical flow algorithms was also briefly tested [76]. This algorithm yielded promising results in terms of computational time, but the optical flow estimates were highly regularized. The differences in performance might be explained by the model being trained on the *Flying-Chairs* data set, purely rigid motion at large scales, highly different from 2D echocardiography. On the other hand, the pretrained I3D uses variational optical flow such that a variational algorithm might be more suitable when using the pretrained I3D.

7.3.8 Augmentations

Discussion of the rotate and crop augmentations are adapted from the project thesis.

While hopefully improving the invariance to rotated features in the CNNs, rotation of the transducer does not yield the same result as rotating the B-mode image. Rotating the transducer yields a rotated view of the heart, by including features in the direction of the rotation, and discarding features at the direction opposite of the rotation. Rotating the images themselves yields an image with the same features, but the sector rotated in relation to the image axes. For small rotation angles the discrepancy is small.

Cropped versions of the images is another augmentation that generates unrealistic data, as the ultrasound images always contains the whole sector. On the other hand, cropping followed by resizing generates features of slightly different size and aspect ratio, corresponding to how objects vary from patient to patient and transducer positioning. Cropping also makes the model focus on different parts of the videos at each epoch. For example, one crop might exclude parts of the heart walls or the AV valves. This should make the models more robust to unseen data from other scanners or acquisition methods. For the models performing quantification, padding or cropping means that there is more variation in the dimensions of the input.

Although noise augmentation have not seen much use in echocardiography, similar augmentations have been used frequently in deep learning, for example the color augmentation of [77]. Compared to rotation, padding or cropping, the noise augmentation generates data that is difficult to visually distinguish from real data. One could argue that adding random

noise degrades the data, making the models less capable of learning finer details. This is where the exponentially decreasing distribution of the added noise intensity is useful. As low intensity noise is most probable, there is little chance of data degradation resulting in worsened overall performance. The noise model is however still quite simplistic, and more realistic noise could have been generated for example by taking into account the changes in noise properties by frame rate, depth and width.

7.4 Conclusion

Whether a deep learning model would be able to learn the concept of aging directly from echocardiography was unknown, but models proposed were all able to learn age estimation to some degree of accuracy. This is valid for the normal population which the models were trained on, and to some extent for patients referred to examinations due to suspicion of heart disease. The models were not able to learn an accurate mapping for patients far from the mean age of HUNT 3. This is to be expected, as interpolation and extrapolation with a nonlinear model and high dimensional features is difficult.

There were little or no separation between healthy and diseased patients in terms of estimated age in the data sets considered. Possible reasons are that healthy patients referred to examination are more similar to the diseased patients than patients in the normal population used for training, or simply inaccurate models with too high variation in estimates. However, models were accurate for two out of three data sets, which suggest that the models have learned age estimation using features not affected by the diseases in consideration. Supporting this view is the inclusion criteria for systolic dysfunction, and that the aim of the UNN-CAD study is to determine how applicable echocardiography is for detecting the disease of the patients.

Deep learning models trained for echocardiography tasks can be used to estimate quality in a semi-supervised manner. Inspection suggest that samples with poor quality are found at a higher rate in the removed data than what is otherwise observed. These results are only indicative, and labels of data quality should be generated before quantifiable results can be provided.

Key takeaways from the results of age estimation are as follows:

- Small displacement optical flow is a viable input modality, on average outperforming models operating on standard B-mode.
- Appending coordinate channels to the inputs improved results for two out of three data sets, but decreased performance on the third. This suggests possibly added value, but that more investigation into how to best use coordinate channels is needed.
- Reusing pretrained parameters from the photographic domain improves accuracy significantly compared to random initialization.
- Training a model with data from several views is beneficial for age estimation, as the amount of data available for training is increased. This is in contrast to most current models, training with only one or a few views at a time.

- Deep learning performed better than clinical measurements followed by linear regression. Here, it must be noted that the clinical measurements are not originally intended for age estimation.
- Saliency studies suggest that models are looking at regions of the heart known to change with age, such as the atrioventricular valves and interventricular septum.

7.5 Further work

Further research into data quality assurance will enable faster experimentation without manual inspection. Unsupervised or semi-supervised methods are beneficial as little or no labeled data is required, but labeled data is still desireable for evaluating the feasibility of the methods.

Inputting physical dimensions to the CNNs important for automated quantification. Coordinate channels or interpolation to fixed dimensions are possible approaches which can be developed further. Additionally, embedding models with other features such as gender, height and age in a suitable way allows for more accurate models with access to other information available to the practitioner. In this work, a simple approach was proposed to embed the view of each cycle, but this method does not scale well to several features without modifications.

When it comes to age estimation, finding a way of enforcing models to learn features affected by disease can still be a viable option for disease classification without requiring a large data set of diseased patients for training. Comparison between other types of CVD is needed to determine if the approach is applicable for classifying any kinds of disease. The methods proposed can also be used as a starting point for automated disease classification from B-mode cycles.

The next HUNT study is currently in the finalizing stage. Taken 10 years later and inviting HUNT 3 patients for participation, HUNT 4 likely contains more old patients than HUNT 3. Evaluation of the effects of estimated age can therefore be evaluated, by comparing estimated age and patient health ten years later.

Bibliography

[1] E. J. Benjamin, S. S. Virani, C. W. Callaway, et al. “Heart disease and stroke statistics 2018 update: a report from the American Heart Association”. In: *Circulation* 137.12 (2018), e67–e492.

[2] I. Ariansen, R. Selmer, S. Graff-Iversen, et al. *Hjerte- og karsykdommer i Norge [Cardiovascular disease in Norway]*. 2018. URL: https://www.fhi.no/nettpub/hin/ikke-smittsomme/Hjerte-kar/ (visited on 01/14/2019).

[3] G. V, P. L, C. M, et al. “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs”. In: *JAMA* 316.22 (2016), pp. 2402–2410. DOI: 10.1001/jama.2016.17216. eprint: /data/journals/jama/935924/joi160132.pdf. URL: +%20http://dx.doi.org/10.1001/jama.2016.17216.

[4] A. Esteva, B. Kuprel, R. A. Novoa, et al. “Dermatologist-level classification of skin cancer with deep neural networks”. In: *Nature* 542.7639 (2017), p. 115.

[5] P. Rajpurkar, J. Irvin, K. Zhu, et al. “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning”. In: *arXiv preprint arXiv:1711.05225* (2017).

[6] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, et al. “Cardiologist-level arrhythmia detection with convolutional neural networks”. In: *arXiv preprint arXiv:1707.01836* (2017).

[7] E. Smistad, A. Østvik, et al. “2D left ventricle segmentation using deep learning”. In: *Ultrasonics Symposium (IUS), 2017 IEEE International*. IEEE. 2017, pp. 1–4.

[8] J. Zhang, S. Gajjala, P. Agrawal, 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.

[9] A. Østvik, E. Smistad, T. Espeland, et al. "Automatic Myocardial Strain Imaging in Echocardiography Using Deep Learning". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Ed. by D. Stoyanov, Z. Taylor, G. Carneiro, et al. Cham: Springer International Publishing, 2018, pp. 309–316. ISBN: 978-3-030-00889-5.

[10] D. Behnami, C. Luong, H. Vaseli, et al. "Automatic Detection of Patients with a High Risk of Systolic Cardiac Failure in Echocardiography". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Ed. by D. Stoyanov, Z. Taylor, G. Carneiro, et al. Cham: Springer International Publishing, 2018, pp. 65–73. ISBN: 978-3-030-00889-5.

[11] M. Steenman and G. Lande. "Cardiac aging and heart disease in humans". In: *Biophysical reviews* 9.2 (2017), pp. 131–137.

[12] E. G. Lakatta and D. Levy. "Arterial and cardiac aging: major shareholders in cardiovascular disease enterprises: Part I: aging arteries: a set up for vascular disease". In: *Circulation* 107.1 (2003), pp. 139–146.

[13] A. Østvik, E. Smistad, S. A. Aase, et al. "Real-time classification of standard cardiac views in echocardiography using neural networks". In: *2017 IEEE International Ultrasonics Symposium (IUS)*. Sept. 2017, pp. 1–1. DOI: 10.1109/ULTSYM.2017.8092375.

[14] J. Carreira and A. Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE. 2017, pp. 4724–4733.

[15] S. Krokstad, A. Langhammer, K. Hveem, et al. "Cohort profile: the HUNT study, Norway". In: *International journal of epidemiology* 42.4 (2012), pp. 968–977.

[16] J. F. Grue, S. Storve, H. Dalen, et al. "Automatic Measurements of Mitral Annular Plane Systolic Excursion and Velocities to Detect Left Ventricular Dysfunction". In: *Ultrasound in medicine & biology* 44.1 (2018), pp. 168–176.

[17] A. Madani, R. Arnaout, M. Mofrad, et al. "Fast and accurate view classification of echocardiograms using deep learning". In: *npj Digital Medicine* 1.1 (2018), p. 6.

[18] E. Smistad, A. Østvik, I. M. Salte, et al. "Fully automatic real-time ejection fraction and MAPSE measurements in 2D echocardiography using deep neural networks". In: *Ultrasonics Symposium (IUS), 2018 IEEE International*. IEEE. 2018.

[19] A. H. Abdi, C. Luong, T. Tsang, et al. “Automatic quality assessment of echocardiograms using convolutional neural networks: Feasibility on the apical four-chamber view”. In: *IEEE transactions on medical imaging* 36.6 (2017), pp. 1221–1230.

[20] K. Groenewegen, H. Den Ruijter, G. Pasterkamp, et al. “Vascular age to determine cardiovascular disease risk: a systematic review of its concepts, definitions, and clinical applications”. In: *European journal of preventive cardiology* 23.3 (2016), pp. 264–274.

[21] R. B. D’Agostino, R. S. Vasan, M. J. Pencina, et al. “General Cardiovascular Risk Profile for Use in Primary Care”. In: *Circulation* 117.6 (2008), pp. 743–753. DOI: 10.1161/CIRCULATIONAHA.107.699579. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCULATIONAHA.107.699579. URL: https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.107.699579.

[22] C. Bonner, K. Bell, J. Jansen, et al. “Should heart age calculators be used alongside absolute cardiovascular disease risk assessment?” In: *BMC Cardiovascular Disorders* 18.1 (2018), p. 19. ISSN: 1471-2261. DOI: 10.1186/s12872-018-0760-1. URL: https://doi.org/10.1186/s12872-018-0760-1.

[23] *Vascular Age Calculator*. URL: http://www.quipu.eu/vascular-age-calculator/ (visited on 01/13/2018).

[24] R. Rothe, R. Timofte, and L. Van Gool. “Deep expectation of real and apparent age from a single image without facial landmarks”. In: *International Journal of Computer Vision* 126.2-4 (2018), pp. 144–157.

[25] A. M. Fiorito. “Automatic classification of cardiac events from ultrasound images using deep learning”. In: (2018).

[26] Wapcaplet. *Diagram of the human heart*. License: CC-BY-SA-3.0. 2006. URL: https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_(cropped).svg.

[27] O. Sand, Ø. V. Sjaastad, and E. Haug. *Menneskets fysiologi*. Gyldendal Norsk Forlag, 2011.

[28] J. B. Strait and E. G. Lakatta. “Aging-associated cardiovascular changes and their relationship to heart failure”. In: *Heart failure clinics* 8.1 (2012), pp. 143–164.

[29] E. G. Lakatta and D. Levy. “Arterial and cardiac aging: major shareholders in cardiovascular disease enterprises: Part II: the aging heart in health: links to heart disease”. In: *Circulation* 107.2 (2003), pp. 346–354.

[30] Y.-M. Cha, G. K. Lee, K. W. Klarich, et al. “Premature Ventricular Contraction-Induced Cardiomyopathy”. In: *Circulation: Arrhythmia and Electrophysiology* 5.1

(2012), pp. 229–236. DOI: 10.1161/CIRCEP.111.963348. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCEP.111.963348. URL: https://www.ahajournals.org/doi/abs/10.1161/CIRCEP.111.963348.

[31] D. Levy, K. M. Anderson, D. D. Savage, et al. "Echocardiographically detected left ventricular hypertrophy: prevalence and risk factors: the Framingham Heart Study". In: *Annals of internal medicine* 108.1 (1988), pp. 7–13.

[32] S. Cheng, V. R. Fernandes, D. A. Bluemke, et al. "Age-related left ventricular remodeling and associated risk for cardiovascular outcomes: the Multi-Ethnic Study of Atherosclerosis". In: *Circulation: Cardiovascular Imaging* (2009), CIRCIMAGING–108.

[33] S. F. Nagueh, O. A. Smiseth, C. P. Appleton, et al. "Recommendations for the Evaluation of Left Ventricular Diastolic Function by Echocardiography: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging". In: *Journal of the American Society of Echocardiography* 29.4 (2016), pp. 277–314. ISSN: 0894-7317. DOI: https://doi.org/10.1016/j.echo.2016.01.011. URL: http://www.sciencedirect.com/science/article/pii/S0894731716000444.

[34] S. D. Solomon. *Essential Echocardiography. A Practical Handbook With DVD*. Humana Press, 2007.

[35] B. K. Horn and B. G. Schunck. "Determining optical flow". In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203.

[36] C. Zach, T. Pock, and H. Bischof. "A Duality Based Approach for Realtime TV-L1 Optical Flow". In: *Pattern Recognition*. Ed. by F. A. Hamprecht, C. Schnörr, and B. Jähne. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 214–223. ISBN: 978-3-540-74936-3.

[37] A. Bruhn, J. Weickert, and C. Schnörr. "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods". In: *International journal of computer vision* 61.3 (2005), pp. 211–231.

[38] M. J. Black and P. Anandan. "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields". In: *Computer vision and image understanding* 63.1 (1996), pp. 75–104.

[39] T. Brox, A. Bruhn, N. Papenberg, et al. "High accuracy optical flow estimation based on a theory for warping". In: *European conference on computer vision*. Springer. 2004, pp. 25–36.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, et al. “Dropout: A simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[41] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[42] C. Szegedy, W. Liu, Y. Jia, et al. “Going deeper with convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[43] J. M. Bland and D. Altman. “Statistical methods for assessing agreement between two methods of clinical measurement”. In: *The lancet* 327.8476 (1986), pp. 307–310.

[44] J. Bland and D. Altman. “Comparing methods of measurement: why plotting difference against standard method is misleading”. In: *The Lancet* 346.8982 (1995), pp. 1085–1087. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(95)91748-9. URL: http://www.sciencedirect.com/science/article/pii/S0140673695917489.

[45] E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed 22. Nov. 2018]. 2001. URL: http://www.scipy.org/.

[46] H. Dalen, A. Thorstensen, S. A. Aase, 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. DOI: 10.1093/ejechocard/jep194. eprint: /oup/backfile/content_public/journal/ehjcimaging/11/2/10.1093_ejechocard_jep194/1/jep194.pdf. URL: http://dx.doi.org/10.1093/ejechocard/jep194.

[47] H. Dalen, A. Thorstensen, L. J. Vatten, et al. “Reference Values and Distribution of Conventional Echocardiographic Doppler Measures and Longitudinal Tissue Doppler Velocities in a Population Free From Cardiovascular Disease”. In: *Circulation: Cardiovascular Imaging* 3.5 (2010), pp. 614–622. DOI: 10.1161/CIRCIMAGING.109.926022. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCIMAGING.109.926022.

[48] A. Støylen, H. E. Mølmen, and H. Dalen. “Relation between Mitral Annular Plane Systolic Excursion and Global longitudinal strain in normal subjects: The HUNT study”. In: *Echocardiography* 35.5 (), pp. 603–610. DOI: 10.1111/echo.13825. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.

1111/echo.13825. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/echo.13825.

[49] A. Støylen, H. E. Mølmen, and H. Dalen. "Importance of length and external diameter in left ventricular geometry. Normal values from the HUNT Study". In: *Open Heart* 3.2 (2016). DOI: 10.1136/openhrt-2016-000465. eprint: https://openheart.bmj.com/content/3/2/e000465.full.pdf. URL: https://openheart.bmj.com/content/3/2/e000465.

[50] J. Pan and W. J. Tompkins. "A real-time QRS detection algorithm". In: *IEEE Trans. Biomed. Eng* 32.3 (1985), pp. 230–236.

[51] L. H. Negri and C. Vestri. *lucashn/peakutils: v1.1.0*. Sept. 2017. DOI: 10.5281/zenodo.887917. URL: https://doi.org/10.5281/zenodo.887917.

[52] A. M. Fiorito, A. Østvik, E. Smistad, et al. "Detection of Cardiac Events in Echocardiography using 3D Convolutional Recurrent Neural Networks". In: *Ultrasonics Symposium (IUS), 2018 IEEE International*. IEEE. 2018.

[53] S. Hashem. "Optimal Linear Combinations of Neural Networks". In: *Neural Networks* 10.4 (1997), pp. 599–614. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(96)00098-6. URL: http://www.sciencedirect.com/science/article/pii/S0893608096000986.

[54] Y. Freund, R. E. Schapire, et al. "Experiments with a new boosting algorithm". In: Citeseer. 1996.

[55] P. M. Granitto, P. F. Verdes, and H. A. Ceccatto. "Neural network ensembles: evaluation of aggregation algorithms". In: *Artificial Intelligence* 163.2 (2005), pp. 139–162.

[56] J. Donahue, L. Anne Hendricks, S. Guadarrama, et al. "Long-term recurrent convolutional networks for visual recognition and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

[57] K. Simonyan and A. Zisserman. "Two-stream convolutional networks for action recognition in videos". In: *Advances in neural information processing systems*. 2014, pp. 568–576.

[58] D. Tran, L. Bourdev, R. Fergus, et al. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[59] C. Feichtenhofer, H. Fan, J. Malik, et al. "SlowFast Networks for Video Recognition". In: *arXiv preprint arXiv:1812.03982* (2018).

[60] J. Deng, W. Dong, R. Socher, et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.

[61] F. Chollet et al. *Keras*. https://keras.io. 2015.

[62] M. Abadi, P. Barham, J. Chen, et al. "TensorFlow: A System for Large-scale Machine Learning". In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. OSDI'16. Savannah, GA, USA: USENIX Association, 2016, pp. 265–283. ISBN: 978-1-931971-33-1. URL: http://dl.acm.org/citation.cfm?id=3026877.3026899.

[63] dlpbc. *keras-kinetics-i3d*. 2018. URL: https://github.com/dlpbc/keras-kinetics-i3d (visited on 10/30/2018).

[64] R. Liu, J. Lehman, P. Molino, et al. "An intriguing failing of convolutional neural networks and the coordconv solution". In: *Advances in Neural Information Processing Systems*. 2018, pp. 9627–9638.

[65] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Y. W. Teh and M. Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256.

[66] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *CoRR* abs/1312.6034 (2013). arXiv: 1312.6034. URL: http://arxiv.org/abs/1312.6034.

[67] J. T. Springenberg, A. Dosovitskiy, T. Brox, et al. "Striving for simplicity: The all convolutional net". In: *arXiv preprint arXiv:1412.6806* (2014).

[68] X. Gao, W. Li, M. Loomes, et al. "A fused deep learning architecture for viewpoint classification of echocardiography". In: *Information Fusion* 36 (2017), pp. 103–113. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2016.11.007. URL: http://www.sciencedirect.com/science/article/pii/S1566253516301385.

[69] C. Liu et al. "Beyond pixels: exploring new representations and applications for motion analysis". PhD thesis. Massachusetts Institute of Technology, 2009.

[70] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[71] O. V. Michailovich and A. Tannenbaum. "Despeckling of medical ultrasound images". In: *ieee transactions on ultrasonics, ferroelectrics, and frequency control* 53.1 (2006), pp. 64–78.

[72] E. Soroos, A. Clark, Hugo, et al. *Pillow: 3.1.0*. Jan. 2016. DOI: 10.5281/zenodo.44297. URL: https://doi.org/10.5281/zenodo.44297.

[73] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, et al. “scikit-image: image processing in Python”. In: *PeerJ* 2 (2014), e453.

[74] S. Seabold and J. Perktold. “Statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference*. 2010.

[75] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

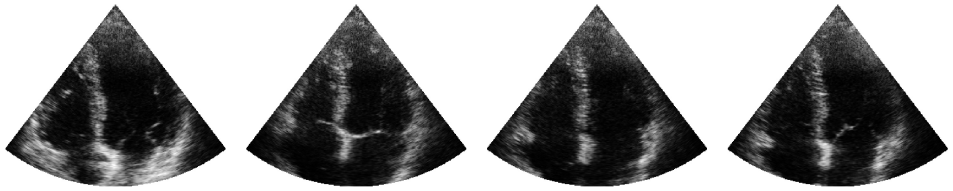
[76] D. Sun, X. Yang, M.-Y. Liu, et al. “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[77] K. Simonyan and A. Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

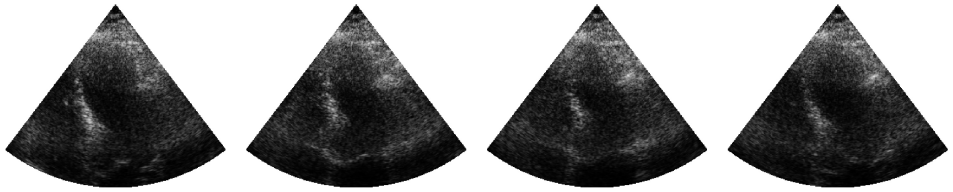
8 | Appendix

8.1 Examples of Cycles With Quality Measurements

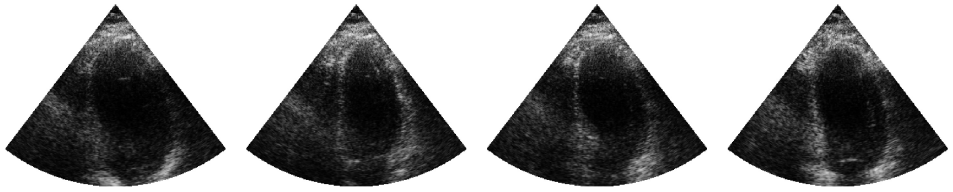
Here, randomly extracted cycles from HUNT 3 with automatic QA measures are presented. Numbers in green indicates errors below automatic QA thresholds, and numbers in red indicating errors above the QA thresholds.



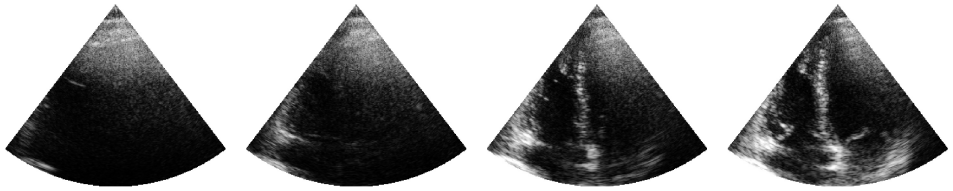
(a) View error: 0.00, Timing error: 0.099



(b) View error: 0.00, Timing error: 0.40

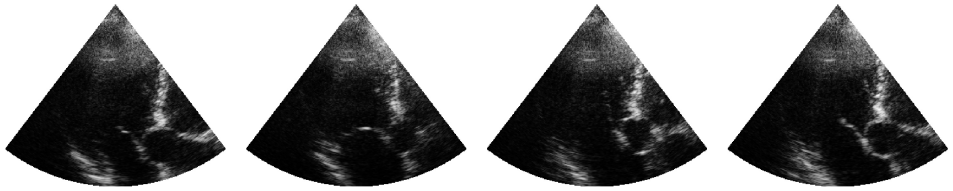


(c) View error: 0.40, Timing error: 0.19

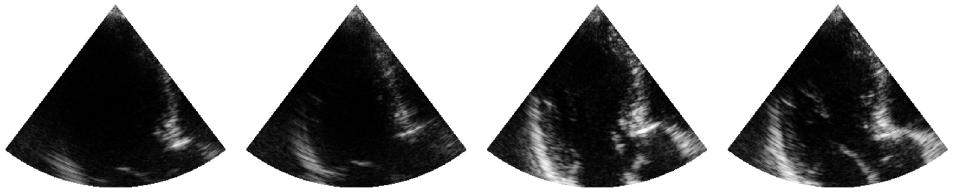


(d) View error: 0.52, Timing error: 0.33

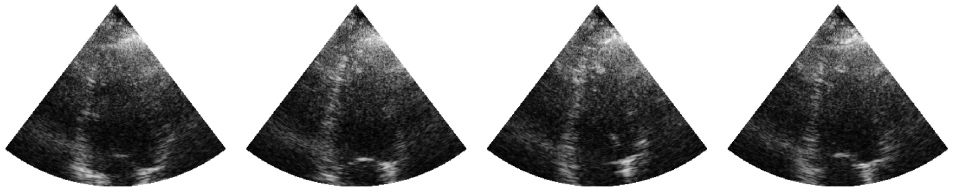
Figure 8.1: Automatic quality assurance scores for cycles predicted to be the A4CH view.



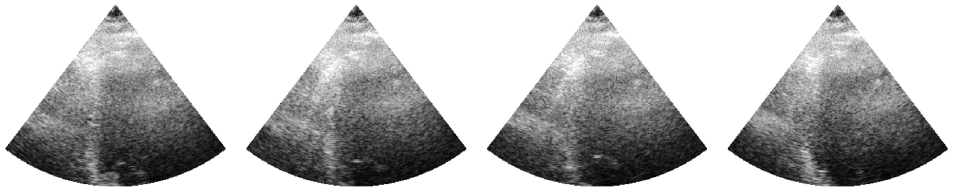
(a) View error: 0.00, Timing error: 0.17



(b) View error: 0.00, Timing error: 0.32

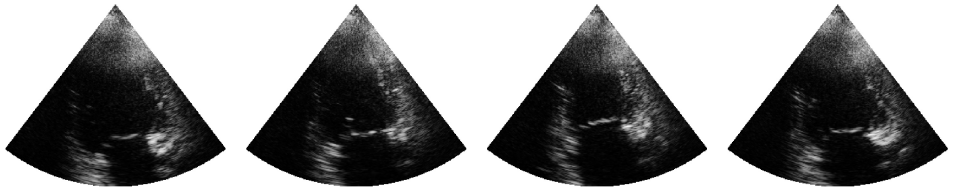


(c) View error: 0.44, Timing error: 0.14

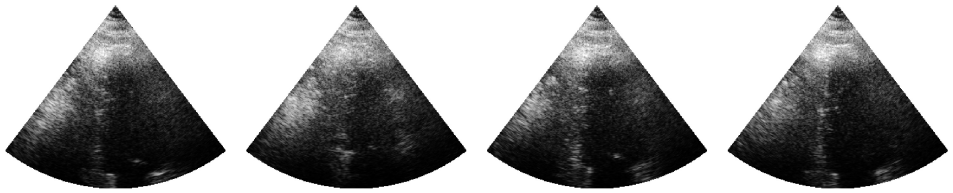


(d) View error: 0.40, Timing error: 0.31

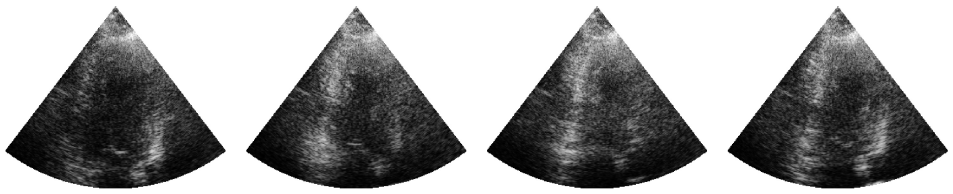
Figure 8.2: Automatic quality assurance scores for cycles predicted to be the ALAX view.



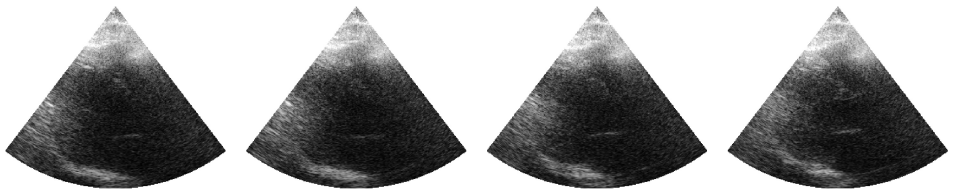
(a) View error: 0.00, Timing error: 0.18



(b) View error: 0.00, Timing error: 0.35

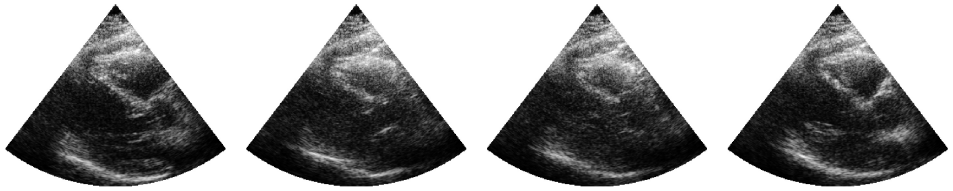


(c) View error: 0.45, Timing error: 0.15

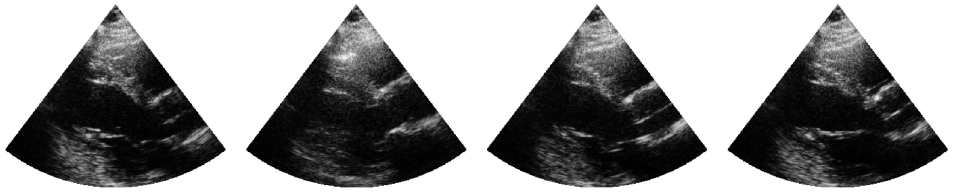


(d) View error: 0.61, Timing error: 0.37

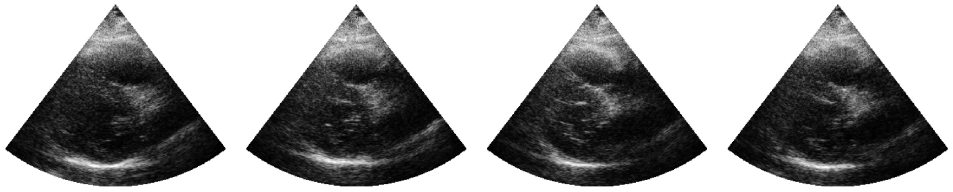
Figure 8.3: Automatic quality assurance scores for cycles predicted to be the A2CH view.



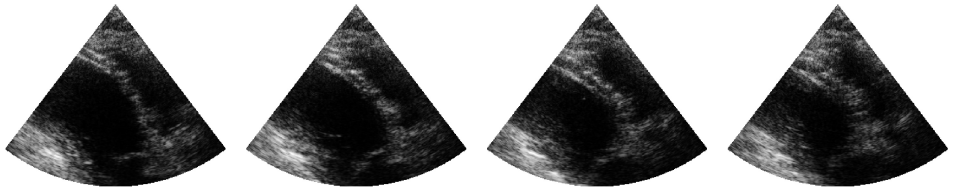
(a) View error: 0.00, Timing error: 0.17



(b) View error: 0.00, Timing error: 0.55

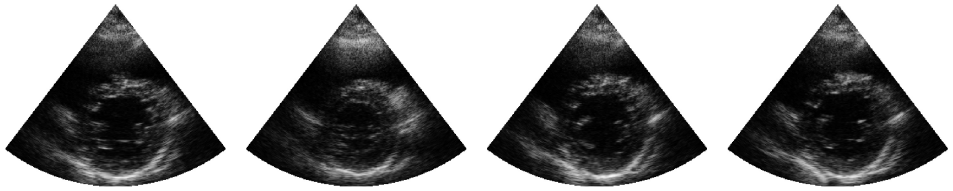


(c) View error: 0.48, Timing error: 0.15

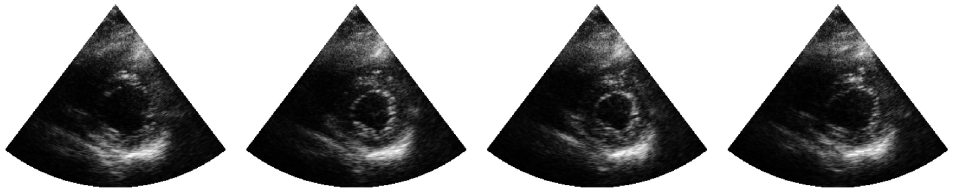


(d) View error: 0.39, Timing error: 0.54

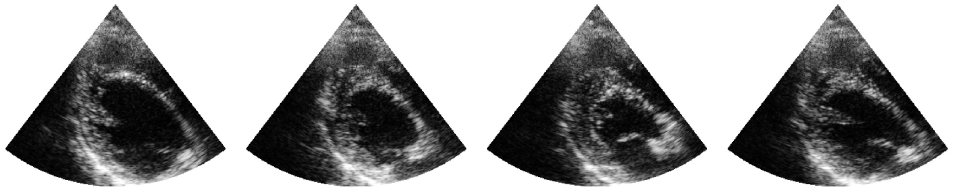
Figure 8.4: Automatic quality assurance scores for cycles predicted to be the PLAX view.



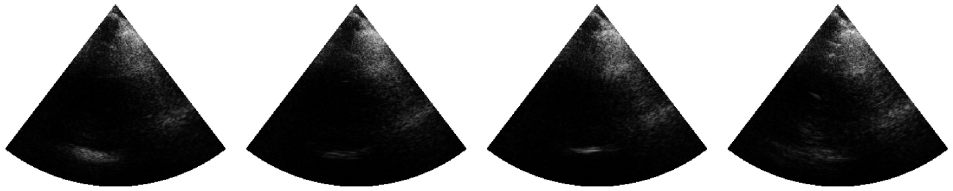
(a) View error: 0.00, Timing error: 0.20



(b) View error: 0.00, Timing error: 0.75

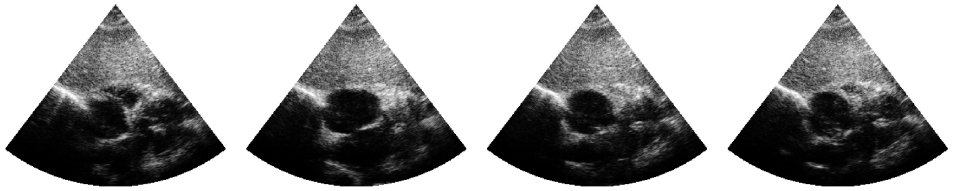


(c) View error: 0.41, Timing error: 0.17

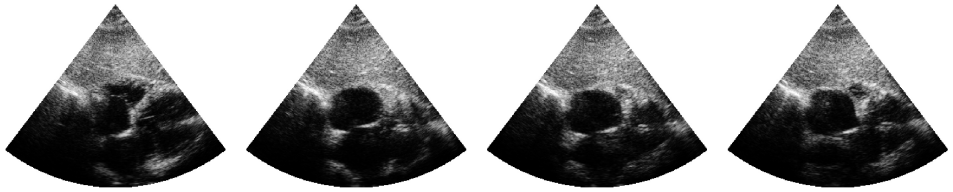


(d) View error: 0.47, Timing error: 0.56

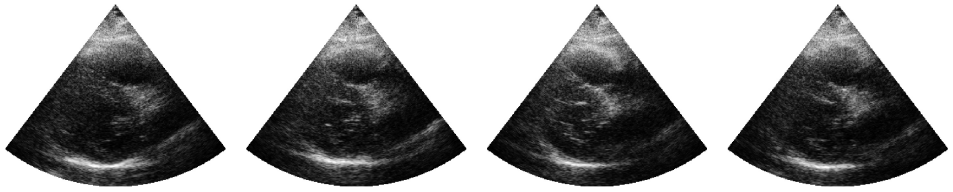
Figure 8.5: Automatic quality assurance scores for cycles predicted to be the PSAX view.



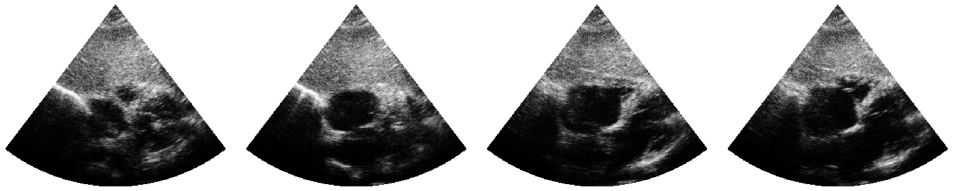
(a) View error: 0.01, Timing error: 0.39



(b) View error: 0.02, Timing error: 0.53



(c) View error: 0.48, Timing error: 0.15



(d) View error: 0.45, Timing error: 0.60

Figure 8.6: Automatic quality assurance scores for cycles predicted to be the unknown view. All cycles from the unknown view are discarded.

8.2 Results for Each View

Descriptions of the models are found in Section 6.1.

8.2.1 HUNT 3 - Mean

The mean ages are calculated from cycles in the HUNT 3 training set, separately for each view. The mean ages for training splits of all data sets are given in Table 8.1. The differences in mean age by view are caused by unevenly distributed number of cycles in each view for a patient.

Table 8.1: Mean chronological age [years] for cycles in the HUNT 3-train split and NTNU-LVD and UNN-CAD datasets for each view.

Dataset	A4CH	ALAX	A2CH	PLAX	PSAX
HUNT 3	48.7	48.2	48.4	48.6	48.2
NTNU-LVD (H)	64.5	67.7	66.0	65.85	62.6
NTNU-LVD (D)	61.9	64.4	62.7	60.2	58.5
UNN-CAD (H)	56.7	56.8	57.2	57.3	55.7
UNN-CAD (H)	61.0	59.5	60.3	61.0	58.0

Results for predicting the HUNT 3 mean cycle age for patients in HUNT3-test, NTNU-LVD and UNN-CAD is given in Tables 8.2 to 8.6.

Table 8.2: Predicting the HUNT 3-train mean cycle age on the HUNT 3-test

View	N	MAE	μ_e	σ_e	R^2
A4CH	477	12.3	1.28	14.4	-0.00800
ALAX	475	12.1	2.22	14.1	-0.0248
A2CH	655	11.8	0.620	13.9	-0.00198
PLAX	481	12.5	1.46	14.6	-0.00999
PSAX	1078	11.7	0.728	14.0	-0.00271

Table 8.3: Predicting the HUNT 3-train mean cycle ages on NTNU-LVD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	484	19.5	15.8	15.3	-1.08
ALAX	143	21.5	19.5	14.1	-1.91
A2CH	373	20.2	17.6	14.5	-1.47
PLAX	119	19.8	17.3	13.5	-1.64
PSAX	237	19.1	14.4	16.1	-0.808

Table 8.4: Predicting the HUNT3-train mean cycle ages on NTNU-LVD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	481	17.8	13.2	15.7	-0.709
ALAX	140	19.8	16.2	15.0	-1.17
A2CH	339	19.7	14.3	17.0	-0.703
PLAX	131	16.2	11.6	14.9	-0.608
PSAX	220	16.8	10.3	16.5	-0.391

Table 8.5: Predicting the HUNT3-train mean cycle ages on UNN-CAD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	340	10.7	7.99	9.89	-0.652
ALAX	423	10.9	8.59	9.95	-0.746
A2CH	538	11.1	8.75	10.0	-0.763
PLAX	453	11.1	8.72	10.2	-0.734
PSAX	12	7.83	7.47	6.69	-1.25

Table 8.6: Predicting the HUNT3-train mean cycle ages on UNN-CAD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	76	12.8	12.3	8.17	-2.27
ALAX	87	12.4	11.3	8.78	-1.65
A2CH	121	12.7	11.9	8.97	-1.75
PLAX	90	13.5	12.4	10.1	-1.49
PSAX	3	9.80	9.80	0.0	0.0

8.2.2 Single I3D models

Table 8.7: Results of the single I3D models on cycles in the HUNT 3 test-set.

View	N	MAE	μ_e	σ_e	R^2
A4CH	477	6.28	1.73	7.73	0.695
ALAX	475	5.28	1.74	6.67	0.762
A2CH	655	5.83	2.65	7.10	0.710
PLAX	481	5.32	0.863	6.67	0.787
PSAX	1078	6.34	1.00	7.92	0.675

Table 8.8: Results of single view B-mode I3D models on cycles in NTNU-LVD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	484	12.8	8.79	14.0	-0.166
ALAX	143	12.5	8.50	13.9	-0.339
A2CH	373	13.3	10.2	11.9	-0.173
PLAX	119	8.09	1.49	10.2	0.414
PSAX	237	15.6	10.5	15.4	-0.341

Table 8.9: Results of the single I3D models on cycles in NTNU-LVD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	481	11.8	7.34	12.5	0.143
ALAX	140	10.8	6.89	10.9	0.259
A2CH	339	12.4	9.15	12.0	0.218
PLAX	131	7.18	-0.356	9.09	0.626
PSAX	220	11.8	6.57	13.3	0.191

Table 8.10: Results of the single I3D models on cycles in UNN-CAD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	340	9.73	8.70	7.73	-0.383
ALAX	423	8.49	6.13	8.80	-0.0162
A2CH	538	8.49	6.63	7.86	-0.054
PLAX	453	7.38	4.69	7.73	0.212
PSAX	12	4.89	-1.39	5.73	0.223

Table 8.11: Results of the single I3D model on cycles in UNN-CAD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	76	11.6	11.0	9.62	-2.18
ALAX	87	7.31	5.89	8.98	-0.497
A2CH	121	8.89	7.78	9.00	-0.760
PLAX	90	6.77	5.62	6.15	0.3282
PSAX	3	8.88	-8.88	3.39	0.00

8.2.3 Multi I3D

Table 8.12: Results of the multi I3D on cycles in the HUNT 3-test set.

View	N	MAE	μ_e	σ_e	R^2
A4CH	477	5.42	0.204	6.96	0.765
ALAX	475	5.29	2.21	6.36	0.772
A2CH	655	5.38	0.158	6.87	0.757
PLAX	481	5.50	0.441	7.07	0.764
PSAX	1078	6.36	0.0203	7.90	0.682

Table 8.13: Results of the multi I3D on cycles in NTNU-LVD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	484	10.4	4.15	12.2	0.289
ALAX	143	11.6	9.07	11.3	-0.0553
A2CH	373	11.1	7.11	11.2	0.166
PLAX	119	8.75	2.67	10.9	0.313
PSAX	237	11.9	4.79	13.7	0.179

Table 8.14: Results of the multi I3D on cycles in NTNU-LVD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	481	8.76	2.64	10.2	0.547
ALAX	140	10.1	5.28	10.8	0.351
A2CH	339	10.1	5.93	10.9	0.466
PLAX	131	7.23	0.0405	9.17	0.620
PSAX	220	10.1	1.88	12.3	0.433

Table 8.15: Results of the multi I3D on cycles in UNN-CAD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	340	8.58	6.66	7.88	-0.0881
ALAX	423	10.0	8.64	8.41	-0.469
A2CH	538	8.99	7.24	8.18	-0.190
PLAX	453	6.96	4.28	7.71	0.250
PSAX	12	5.83	1.10	6.62	-0.00702

Table 8.16: Results of the multi I3D on cycles in UNN-CAD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	76	10.4	9.70	9.00	-1.63
ALAX	87	9.58	8.79	8.97	-1.05
A2CH	121	8.88	7.86	7.84	-0.533
PLAX	90	6.53	5.09	6.54	0.328
PSAX	3	2.26	-2.26	1.63	0.00

8.2.4 Multi-Coords I3D

Table 8.17: Results of the Multi-coords I3D on cycles in the HUNT 3-test set

View	N	MAE	μ_e	σ_e	R^2
A4CH	479	5.78	1.46	7.22	0.739
ALAX	478	5.73	2.67	6.91	0.726
A2CH	662	5.46	0.801	6.92	0.752
PLAX	481	5.57	2.07	7.11	0.745
PSAX	1081	6.19	1.09	7.52	0.704

Table 8.18: Results of the Multi-coords I3D on cycles in NTNU-LVD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	470	12.75	8.32	14.2	-0.120
ALAX	120	15.6	13.4	14.4	-0.931
A2CH	331	11.1	7.65	12.8	-0.00173
PLAX	112	10.14	6.69	11.6	0.0633
PSAX	208	12.45	7.68	13.4	0.0411

Table 8.19: Results of the Multi-coords I3D on cycles in NTNU-LVD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	460	11.9	8.16	12.6	-0.104
ALAX	137	14.1	10.2	12.8	-0.0676
A2CH	325	12.3	7.68	12.6	0.304
PLAX	117	8.59	3.14	10.2	0.473
PSAX	201	10.7	4.37	12.4	0.369

Table 8.20: Results of the Multi-coords I3D on cycles in UNN-CAD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	354	6.04	3.71	6.99	0.341
ALAX	430	6.67	4.46	7.67	0.226
A2CH	523	6.54	3.49	7.83	0.246
PLAX	460	6.15	3.30	7.30	0.349
PSAX	10	2.58	-1.49	3.69	0.651

Table 8.21: Results of the Multi-coords I3D on cycles in UNN-CAD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	83	8.04	6.14	9.63	-0.714
ALAX	84	7.81	6.58	8.76	-0.680
A2CH	117	7.34	4.76	9.17	-0.337
PLAX	91	6.42	4.11	6.96	0.361
PSAX	0	-	-	-	-

8.2.5 Multi-Coords-Small I3D

Table 8.22: Results of the Multi-coords-small I3D on cycles in the HUNT 3-test

View	N	MAE	μ_e	σ_e	R^2
A4CH	479	6.28	0.00833	7.79	0.709
ALAX	478	6.66	2.52	8.01	0.648
A2CH	662	6.11	0.380	7.65	0.699
PLAX	481	6.52	-1.03	8.14	0.687
PSAX	1081	7.24	1.95	8.78	0.587

Table 8.23: Results of the Multi-coords-small I3D on cycles in NTNU-LVD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	470	14.0	8.63	15.8	-0.343
ALAX	120	17.3	15.01	16.0	-1.41
A2CH	331	12.7	8.21	15.2	-0.342
PLAX	112	11.7	7.39	13.1	-0.18
PSAX	208	15.2	11.3	14.8	-0.395

Table 8.24: Results of the Multi-coords-small I3D on cycles in NTNU-LVD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	460	12.6	8.72	12.9	-0.197
ALAX	137	17.0	13.0	14.7	-0.518
A2CH	325	13.9	8.99	14.7	0.0550
PLAX	117	9.90	3.43	11.8	0.284
PSAX	201	12.9	8.17	13.6	0.0750

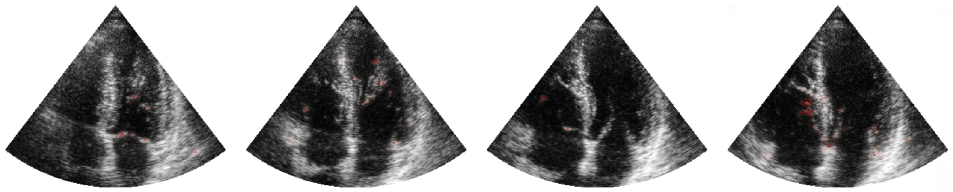
Table 8.25: Results of the Multi-coords-small I3D on cycles in UNN-CAD (H)

View	N	MAE	μ_e	σ_e	R^2
A4CH	354	5.88	1.92	7.61	0.352
ALAX	430	7.71	5.33	8.59	-0.00475
A2CH	523	6.76	3.31	8.13	0.207
PLAX	460	6.12	1.70	7.70	0.369
PSAX	10	4.44	-0.08	5.14	0.399

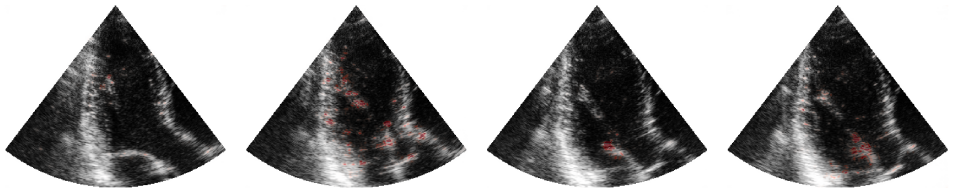
Table 8.26: Results of the Multi-coords-small I3D on cycles in UNN-CAD (D)

View	N	MAE	μ_e	σ_e	R^2
A4CH	83	6.84	4.97	8.27	-0.220
ALAX	84	8.61	7.48	8.78	-0.864
A2CH	117	7.90	6.42	8.56	-0.434
PLAX	91	7.16	3.63	7.95	0.252
PSAX	0	-	-	-	-

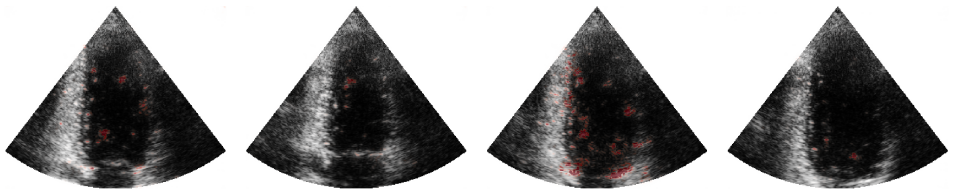
8.3 Examples of Predictions and Saliency: Multi-I3D



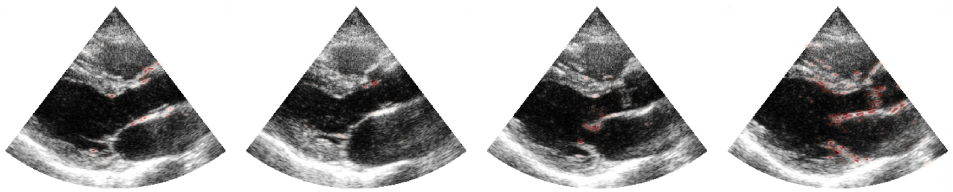
(a) View: A4CH, Age: 59 years, Predicted age: 62.6 years



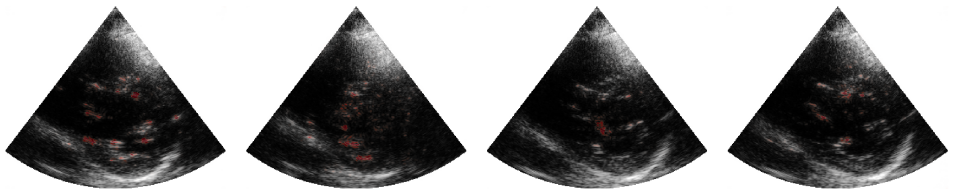
(b) View: ALAX, Age: 37 years, Predicted age: 38.8 years



(c) View: A2CH, Age: 30 years, Predicted age: 27.1 years



(d) View: PLAX, Age: 48 years, Predicted age: 47.9 years



(e) View: PSAX, Age: 66 years, Predicted age: 59.1 years

Figure 8.7: Predictions and guided saliency (red) for the multi-I3D on cycles in the HUNT3 test set.

Adrian Meidell Fiorito

Age Estimation from B-mode Echocardiography with 3D Convolutional Neural Networks

NTNU
Norwegian University of
Science and Technology