

Confidentiality Disclaimer

Due to confidential agreement, the content of the master thesis of Zhuangzhuang Yu, will not be made publicly available to third parties, neither electronically, in print, nor by any other means of publication until 25/11/2023 (12 months after defense).

Machine Learning of Ultrasound Data: Cardiovascular Parameters Detection Using Carotid Artery Measurements

Machine Learning of Ultrasound Data: Cardiovascular Parameters Detection Using Carotid Artery Measurements

Dissertation

to obtain the degree of Master of Science
in Electrical Engineering
at Technische Universiteit Delft,
to be defended on 25th November 2022

by

Zhuangzhuang YU

Born in Canton, China

This thesis has been approved by
promotor: Prof. dr. B. Hunyadi

Thesis committee:

Prof. dr. B. Hunyadi,	Technische Universiteit Delft
Prof. dr. C Strydis,	Technische Universiteit Delft
Dr. F. Beutel,	Imec NL
Dr. M. Sifalakis,	Imec NL

The work in the thesis has been carried out in collaboration with Imec NL.



Style: TU Delft House Style, with modifications by Moritz Beller
<https://github.com/Inventitech/phd-thesis-template>

An electronic version of this dissertation is available at
<http://repository.tudelft.nl/>.

Simplicity is the highest goal, achievable when you have overcome all the difficulties. After one has played a vast quantity of notes and more notes, it is simplicity that emerges as the crowning reward of art.

Frédéric Chopin

Acknowledgments

It is only at the end of this thesis when I can finally recollect the things and ideas that have happened and thought in the past 10 months. Without a doubt, things I have learned in this work and the exploration itself is the most important experience in my past life in academics and for which, I would like to express my deepest gratitude to my supervisors: Dr. Fabian Beutel, Dr. Manolis Sifalakis from Imec NL and Prof. Dr. Borbála Hunyadi from TU Delft, who have guided me through this project with great attainments in academics and professions, and great patience for all those prolonged meetings or the scheduled discussions. Your supports are very essential to the path of the exploration and the completion of this thesis. I would also express my heartfelt appreciation to peer students working in Imec and TU Delft, it is an absolutely enjoyable experience to work and chill with you.

It is not a very easy decision for me to choose further education abroad, for which I would also express my special gratitude and deep miss towards my parents, family and friends in China, you have been always supported me unconditionally for the two years I have been away from home. The feeling of homesickness and isolation could be strong, but thanks to my friends I have here, Daniel, Loucas, Carlos, Stephan, Yubin, Yongdian, Yuxiang...that there are countless memorable moments and experience that cheers me up and motivates me through this stressful pandemic times, and especially my dearest girlfriend, Jia, who always understands and supports me with a lot of efforts and faith, which I would definitely cherish for my lifetime.

*Zhuangzhuang Yu
Rotterdam, 18 November 2022*

Abstract

Background & Objective:

Cardiovascular diseases (CVDs) are the leading cause for death globally nowadays. Pulse wave velocity (PWV), a marker of arterial stiffness, is an important predictor of CVD risk. In precedent work, carotid artery data was collected with ultrasound to estimate the PWV with a digital signal processing (DSP) pipeline. As a potential alternative to the DSP-based approach, this thesis studies the applicability of machine learning(ML) for the estimation of carotid artery motion (diameter, distension, etc.) and explores to what extent neural networks can exploit the ultrasound data to extract relevant biomarker information.

Methods:

This thesis proposes a ML pipeline that processes the ultrasound data in a different perspective than the DSP approach. The ML pipeline consists of four modules (neural networks post-processing) to: 1) segmentation based on cardiac cycle (CC), 2) detect the region of interest (ROI) of artery in the ultrasound data, 3) tracking the artery diameter and 4) post processing to estimate cardiac parameters e.g. pulse arrival time (PAT), an essential part of PWV estimation. Exploiting the features of the artery-lumen structure and time-evolving characteristics of collected ultrasound signal, the designed ML pipeline can acquire cardiac markers spatially and temporally with irregular kernels and sliding mechanism, decompose the complicated estimation into compact sub-modules.

Results:

The results show that the ML approach can successfully estimate the artery diameters and reserve important waveform features (max-slope moment) of the artery diameter, and can infer the CC markers without ECG data as a segmenting event for heart cycles. Thus, the PAT can be computed as the time difference of the max-slope moments of inferred artery diameter and detected CC markers. According to the numerical results, the PAT can be estimated on the average for an ultrasound data recording (120s), and the correlation coefficient of label PAT (computed from estimated parameters of DSP pipeline and ECG data) and estimated PAT (ML pipeline) is 0.8250. This indicates a good correlation and hence proves the effectiveness of the mean PAT estimation.

Conclusion:

In conclusion, the proposed ML pipeline can effectively estimate mean PAT, and demonstrate the feasibility to estimate PWV as a relevant cardiovascular marker using only ultrasound data of carotid arteries. Apart from the PAT, the heart rate can also be possibly tracked via intermediate results of the ML pipeline (CC markers). From the future perspective, the potential of phase information in the raw ultrasound data and further optimization are worth exploring, and the extension to hardware (e.g. chips, embedded system) can be implemented as a practical application.

Contents

Acknowledgments	vii
Abstract	ix
1 Introduction	1
1.1 Cardiovascular Disease	1
1.1.1 Background & Context	1
1.1.2 Pulse Wave Velocity	2
1.2 Ultrasound Imaging	3
1.2.1 Medical Ultrasound Imaging	3
1.3 Machine Learning Approach	4
1.3.1 Neural Network	5
1.3.2 Variants and Applications	6
1.4 Related Researches	7
1.5 Objectives & Thesis Outline	8
1.5.1 Research Problems & Objectives	8
1.5.2 Thesis Outline	9
2 Data Preparation	11
2.1 Ultrasound Data	11
2.1.1 Data Acquisition	11
2.1.2 Data Format	12
2.2 Digital Signal Processing Pipeline	14
2.2.1 Carotid Artery Model for Ultrasound.	14
2.2.2 Design of Digital Signal Processing Pipeline	14
2.3 Data Processing	17
2.3.1 Carotid Artery Parameters	17
2.3.2 Example of the Result of DSP Approach	18
3 Methods	19
3.1 Problem Formulation.	19
3.1.1 Input & Output.	19
3.1.2 Challenges & Heuristics	21
3.1.3 Machine Learning Processing Pipeline Design	21
3.2 Neural Networks.	22
3.3 Part 1: ROI Detection	27
3.3.1 Problem Formulation.	27
3.3.2 ROI Detection Network Design.	28

3.4	Part 2: Time Domain Segmentation	31
3.4.1	Problem Formulation.	32
3.4.2	Segmentation Network Design	34
3.4.3	Post-processing for Time Domain Segmentation	35
3.5	Part 3: Diameter Tracking	37
3.5.1	Problem Formulation.	38
3.5.2	Tracking Network Design	39
3.5.3	Post-processing	40
3.6	Cardiac Parameter Estimation: Pulse Arrival Time	41
4	Results & Evaluation	43
4.0.1	Validation Scheme	43
4.1	Intermediate Results & Analysis	44
4.1.1	ROI Detection	44
4.1.2	Time Domain Segmentation	45
4.1.3	Diameter Tracking	45
4.2	Numerical Results & Analysis	48
4.2.1	Intra-subject Results	48
4.2.2	Inter-Subject Results	49
4.3	Concluding Remarks	50
5	Conclusion & Discussion	53
5.1	Discussion	53
5.1.1	Research Questions	53
5.1.2	Limitation	54
5.1.3	Recommendation.	54
5.2	Conclusion.	55
5.3	Future Perspective	56
	Bibliography	57
	Appendix	63

List of Figures

1.1	Carotid-femoral Artery, cfPWV and Central Arterial Waveform Assessment. P1: first systolic pressure feet; P2: second systolic pressure feet; PP: pulse pressure[1]	3
1.2	Ultrasound Wave Propagation	4
1.3	An Example of Artificial Neural Networks(Multilayer Perceptron)	5
1.4	Example of the Architecture of CNN, Extracted Features in the Convolution Kernels[2].	6
1.5	Recurrent Neural Network Architecture	7
2.1	Collected Data Format (3D) and the Collection of Ultrasound Data	12
2.2	An Example of A-mode and Envelope of Ultrasound Data	13
2.3	B-mode, M-mode Ultrasound Data	14
2.4	Carotid Artery Model for Ultrasound [3]	15
2.5	Flow Diagram of DSP Pipeline, proposed in [3]	15
2.6	Result of DSP Approach: All Parameters	18
3.1	Artery Walls, Lumen and Unwanted Reflection (noise)	21
3.2	Machine Learning Processing Pipeline	22
3.3	Illustration of a Neuron in a Neural Network	23
3.4	Illustration of 2D Convolution	24
3.5	<i>Dropout</i> Mechanism During One Weight Update, Figure by [4]	26
3.6	Response Vector(Labels for ROI Detection). The left figure shows the original envelope of A-mode data, the right figure shows the response vector calculated by the artery wall positions	29
3.7	ROI Detection Neural Network Structure	30
3.8	Example of Detected CC Marker on the Ultrasound Image (White Vertical Line implies Detected CC Markers)	31
3.9	Example of Shift-sensitive Network and Translation-invariant Network	33
3.10	An Illustration of the $l1$ distance Labeling	35
3.11	Convolution Neural Network for Heartbeat Segmentation	36
3.12	Illustration of the Motion of Artery Wall Movement	38
3.13	Artery Wall Diameter Tracking Neural Network Structure	40
3.14	An Illustration Reference Points on the Artery Diameter for PAT Computation, Partly Adapted from [5]	41
3.15	An Illustration of PAT Computation	42
4.1	Validation Scheme	43

4.2	Comparison of the Original Ultrasound Image, Response Vectors and Lumen Centers (Label, output). The Images Display the Intensities in Gray Scale	46
4.3	Time Domain Segmentation: Result (excerpt). In the Label/Output image, the label is plotted in red line, the regressed distance to a CC marker is plotted in blue	47
4.4	Diameter Tracking: Result (excerpt)	47
4.5	Histogram of Error from CC Markers, Regressed Diameter, Split 1	48
4.6	Estimated, Ground Truth PAT, Split 1 (Mean Label PAT: 129.9106, Mean Estimated PAT: 127.0579)	49
4.7	Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 1	50
4.8	Correlation between Estimated & Ground Truth PAT	50
5.1	On-chip Neural Network Applications	56
2	Histogram of Error from CC Markers, Regressed Diameter, Split 2	63
3	Histogram of Error from CC Markers, Regressed Diameter, Split 3	63
4	Histogram of Error from CC Markers, Regressed Diameter, Split 4	64
5	Histogram of Error from CC Markers, Regressed Diameter, Split 5	64
6	Histogram of Error from CC Markers, Regressed Diameter, Split 6	64
7	Estimated, Ground Truth PAT, Split 2	65
8	Estimated, Ground Truth PAT, Split 3	65
9	Estimated, Ground Truth PAT, Split 4	65
10	Estimated, Ground Truth PAT, Split 5	66
11	Estimated, Ground Truth PAT, Split 6	66
12	Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 2	66
13	Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 3	67
14	Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 4	67
15	Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 5	67
16	Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 6	68

List of Tables

3.1	A comparison between different data representations for machine learning-based approach	20
3.2	Hyperparameters and Settings of the ROI Detection Network	30
3.3	Hyperparameters and Settings of the Segmentation Network	37
3.4	Hyperparameters and Settings of the Tracking Network	41
4.1	Error in ROI Detection (response vector and lumen center)	44
4.2	Error in Time Segmentation: The Detection of CC Markers	45
4.3	Error in Diameter Tracking	46
4.4	Estimated & Label Mean PAT	49

1

Introduction

In this chapter, the background and context of the cardiovascular diseases and the importance of medical interventions or health assessments are discussed. The pulse arrival time (PAT), an essential quantity of the pulse wave velocity (PWV) is introduced as one of the main objective of this work. In the following the ultrasound sound image technology and machine learning are introduced and the motivation behind their applications. Covering the general background of this topic, the objective and thesis outline is proposed, together with the research questions.

1.1 Cardiovascular Disease

1.1.1 Background & Context

As a slow-progressing and rather common disease in the population, Cardiovascular Disease (CVD) has been already identified as the leading cause of death globally, according to World Health Organization. It occurs when fatty deposits clog the blood vessels that deliver blood to the brain or organs. As an estimation, in 2019 there are 17.9 million people passed away due to CVD, representing 32% of the global deaths (World Health Organization, 2021), and in EU there are more than 6 million new cases of CVD, with almost 49 million people living with the disease, the cost to the EU economics is high at €210 billion a year [6]. Cardiovascular Disease, as a group of disorders of heart and vessels, contains different types and symptoms. Common symptoms such as stroke, heart attacks and sudden numbness happening in different part of the human body. The threat of CVD to human not only lies in the fatal risks, but also severe consequences such as damage to the nerve system and the brain also has been seen quite often on the patients after CVD attacks[7].

Usually identified as a chronic process, the forming of CVD is highly related to the risk factors such as smoking, diet, stress, alcohol, age etc.. The commonality of such factors in the population has made the forming of cardiovascular disease more easily, results in higher risk in the global population, especially in aged groups. Under such circumstances, approaches that can detect the risk of CVDs and intervenes early are very important to preventing the symptoms and consequences. Over the decades, the focus of early prevention of CVD is mainly hypertension(high blood pressure, BP), while recent researches has

shown great interest into the pulse wave velocity (PWV), as a marker of the stiffness of artery, and an independent predictor of CVD risk complementary to BP. While the BP is usually regarded as the strongest evidence for causations and it has a high prevalence of exposure [8], the PWV is identified as an independent predictor of CVD and all-cause mortality beyond classical risk factors like BP, and is suggested to be one of the best biomarkers available to calculate the prospective cardiovascular risk and mortality risk of an individual [9]. Hence, the estimation of PWV becomes very important in monitoring the situation of artery, and practical approaches that can measure central artery PWV are studied and developed by researchers. Ideally, such techniques would be widely applicable for both the practitioners and patients [10].

1.1.2 Pulse Wave Velocity

Pulse wave velocity (PWV) is the velocity with the blood pressure pulse propagates through the circulation system, usually an artery or a combined length of arteries. In practice, PWV is measured across two distinct vessel positions over a large arterial trajectory (regional, or global PWV), and from a single artery site or locally across piecewise segments of individual target arteries [11]. It is used clinically as a measure of arterial stiffness and can be readily measured non-invasively in humans.

In practice, the carotid-femoral pulse wave velocity (cfPWV) is regarded as the gold standard method for determination of arterial stiffness [12]. The carotid and femoral artery directly extend from the aorta, as shown in figure 1.1, and the artery distension or diameter waveform (time series signals) can be acquired by a piezoelectric ultrasound probe. With the measured data, the cfPWV can be computed as follow

$$PWV = \frac{L}{\Delta T} [m/s] \quad (1.1)$$

Here, L is the distance travelled between the arterial sites and ΔT is the time delay between the systolic foot (SF) fiducial marker on simultaneously acquired waveforms. Researches indicate that the cfPWV is increasing with grows in age and more risk factors, meaning the stiffness of artery is positively correlated to overall and long-term cardiovascular risk [13].

In practice, the acquisition of pulse wave velocity needs to compute pulse arrival time (PAT), it is the time duration of travel time from the hearts to periphery arteries (e.g. carotid artery, femoral artery, etc.) [5]. It can be measured as the interval from the R-peak on an electrocardiogram (ECG) to the peripheral arrival of the pulse wave. [5]. Therefore, two kinds of information are needed: artery movements collected by different sensors (Ultrasound, PPG, etc.) and time information (ECG R-peak moments). The details of PAT computation is further introduced in 3.6.

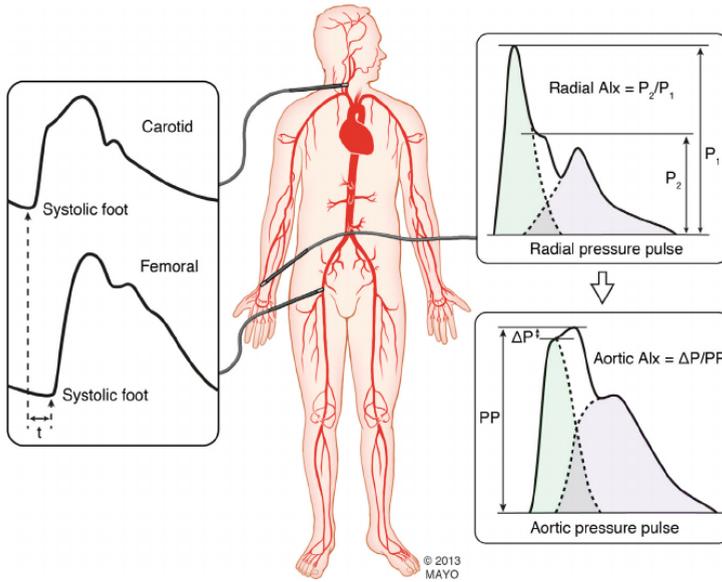


Figure 1.1: Carotid-femoral Artery, cfPWV and Central Arterial Waveform Assessment. P1: first systolic pressure feet; P2: second systolic pressure feet; PP: pulse pressure[1]

1.2 Ultrasound Imaging

Ultrasound is the acoustic wave that has a frequency higher than 20,000 Hz and it is not perceivable by the hearing ability of humans. The ultrasound technologies that apply such acoustic wave to detect and measure things has been developed very well over the past few decades. It is used in many fields, especially in medical imaging.

1.2.1 Medical Ultrasound Imaging

In practice, the ultrasound sensors are usually piezoelectrical crystals that convert electrical signal into mechanical signal, hence generates the oscillation of ultrasound signals and vice versa. The acoustic sound wave is applied as a propagating signal to detect information of the medium by the reflected wave. With its concentrated energy and high frequency, the ultrasound signal is able to detect fine details in the human organs or tissues with high resolutions in a non-invasive manner. Also, since ultrasound signal can be captured in real time, it can reveal the movement of blood vessels or internal organs.

The propagation of ultrasound can be seen in the figure 1.2 Without loss of generality, the ultrasound propagation can be formulated as follows: with its propagation speed determined c by the compressibility κ and density ρ of a material,

$$c = \sqrt{\frac{1}{\kappa\rho}} \tag{1.2}$$

the propagation speed c in the air is 330 m/s, while it is assumed to be 1540 m/s in soft tissues. When propagating in the medium, the amplitude of ultrasound signal also atten-

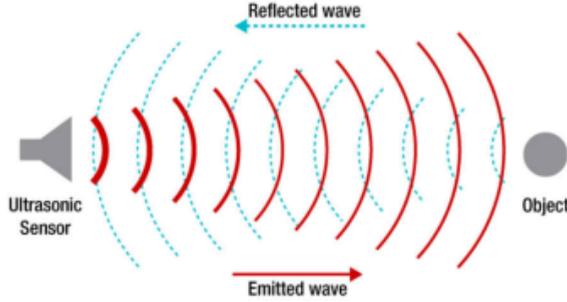


Figure 1.2: Ultrasound Wave Propagation

uated by a factor (amplitude attenuation factor, units cm^{-1}) μ_A , with axial depth z , the amplitude of ultrasound signal can be expressed as

$$A(z) = A_0 e^{-\mu_A z} \quad (1.3)$$

As ultrasound signal propagates and reflects back, it is expected to have a decrease in the average amplitude. The information of certain structure inside the medium is detected by the pulse echo of transmitted ultrasound signal. At one element of the transmitter, a short pulse of its characteristic frequency is emitted into the medium. Upon a change in the acoustic impedance, the circular propagating wavefront would be partly reflected and partly transmitted. Hence, the distance of an object or surface(acoustic impedance change) can be calculated by the delay of transmitted signal and reflected signal(time of flight).

$$d = \frac{\Delta t}{2} c \quad (1.4)$$

Apart from the reflection and attenuation, ultrasound signal generally shares other properties as waves, e.g. scattering, interference etc.. These properties also have an impact on the quality of ultrasound signal (noise, interference etc.).

1.3 Machine Learning Approach

Machine learning is the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data. As one of the most progressive and popular techniques recently, it is broadly applied in statistics, computer vision and different engineering fields. As a very general terminology, it has a broad range of different algorithms and structure, e.g. neural networks, support vector machines (SVM), decision trees, random forest, Boltzmann machine, etc.. It has been discovered that the machine learning approach can be very effective because of its ability to generalize in high dimension with increasing availability of training data. Based on learning strategies, machine learning can be categorized into different kinds: supervised learning(with labels), unsupervised learning(without labels) and reinforcement learning(learning by rules). Numerous

in-depth literature and researches have been conducted to discover different applications of all these types of machine learning.

1.3.1 Neural Network

Neural networks, also known as artificial neural network (ANNs), is the backbone of machine learning (deep learning) algorithms that enables the generalizing ability over high dimensional space. Inspired by the human brain structure, the neural network usually consists of multiple nodes and layers, with each node representing a neuron in the brain, the nodes connect each other by a certain weight, a bias and an activation function. With many of this nodes and connections, the nodes that are in the same depth (same connection numbers from the beginning) form a layer, such structure gives this model a very high complexity and hence can be approximate many functions that are not easily approximated by linear equations or polynomials.

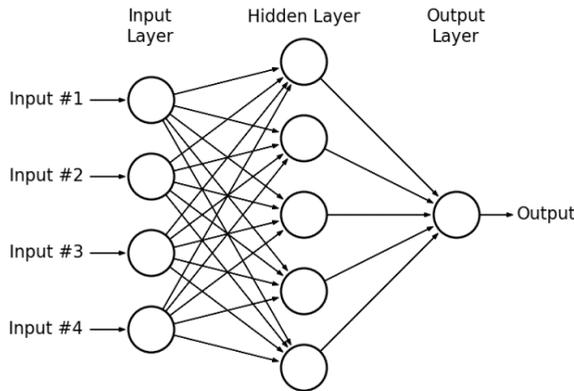


Figure 1.3: An Example of Artificial Neural Networks (Multilayer Perceptron)

Without loss of generality, a neural network can be formulated as a linear regression model on each node, for a j -th neuron in the hidden layer (net_j), its input can be expressed as

$$net_j = \sum_{i=1}^m w_i x_i + b \quad (1.5)$$

w_i is the weight assigned to the connection, x_i is the element corresponding to i -th in the input vector and b is the bias. While the output is an outcome of an activation or threshold (usually non-linear),

$$y_j = f(y_j) = f\left(\sum_{i=1}^m w_i x_i + bias\right) \quad (1.6)$$

In this way, the connection of neurons can approximate both linear and non-linear functions because the neural network is combined with linear inputs and non-linear activation functions.

In the training of neural network, a cost function has to be defined as the objective the neural network need to learn. Common cost functions such as mean square error (MSE), cross entropy loss, negative log-likelihood loss (NLL), etc. are designed to be minimized by an optimizer, using gradient descent method, It backpropagates the loss through the neurons and update the weights of the neural network. With rapid technical advance in computational power of computing units (GPUs, CPUs, etc.), the time and difficulties to train a neural network is reducing over recent years, making it possible to train very deep and large neural network that can handle highly abstract problems.

1.3.2 Variants and Applications

Based on the basic principles of how the neural network structure is organized, many variants of the ANNs have been studied and put into practice. These variants share the same optimization process and part of the architecture.

Convolution Neural Network (CNN): This network alters the first few layers of the neural network with convolution kernels that only focusing on part of the data at one time. This architecture is inspired by the natural visual perception mechanism of the living creatures[2]. Its framework was established in the 1990s and later advanced rapidly in 21th century. The structure of CNNs can be illustrated as 1.4 It is obvious that CNNs

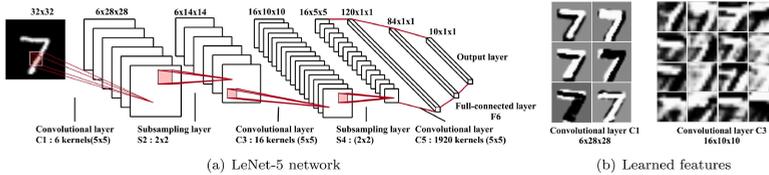


Figure 1.4: Example of the Architecture of CNN, Extracted Features in the Convolution Kernels[2].

are powerful with the feature extraction process that can synthesis geometrically-related information, the original (sub)image information is transform into a lower dimension representation, while the fully connected layers will learn the combinations of these low-dimension image features to regress the objective output.

CNNs are very popular in the visual-pattern-recognition related tasks, like image classification, object detection and tracking. To increase its capability to learn more abstract, high-level representations, modifications are made to the CNNs, some of these CNN models turned out to be very successful in the accuracy and robustness, e.g. ZFNet[14], VGGNet[15], and ResNet[16]. With very large scale of the parameters and pre-trained model, these neural networks can easily fit into different image-related tasks.

Recurrent Neural Network (RNN) In recurrent networks, rather than give only the output that corresponding to the current input, the output of the network is the integration of the inputs over time. Different from the feed forward network, RNN has recurrent connections that allows the history over certain neuron to be integrated.

For each timestep t , the activation $a^{<t>}$ and the output $y^{<t>}$ can be expressed as follows:

$$\begin{aligned} a^{<t>} &= g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \\ y^{<t>} &= g_2(W_{ya}a^{<t>} + b_y) \end{aligned} \quad (1.7)$$

where W_{ax} , W_{aa} , W_{ya} are the weights of the connections, b_a , b_y is the bias, these are the coefficients that are shared temporally. g_1 , g_2 are activation functions. In this sense, the RNN can be illustrated as 1.5: Such recurrent connections make RNN can possibly process

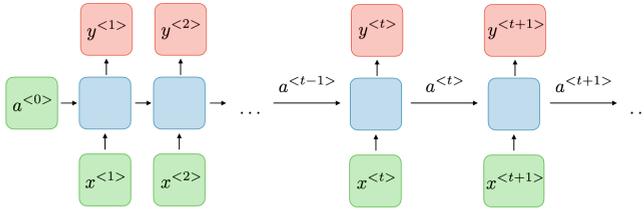


Figure 1.5: Recurrent Neural Network Architecture

input of any length, and have a fixed size that does not increasing with the size of the input. However, it also turns out to be slow at the computation, especially backpropagation through time mechanism, and also difficult to accessing information from a long time ago. As a special neural network that can make time dependency over the input, it has a relatively more complicated control logic over the updating of parameters and memory usage. Based on the original RNN structure, many new modifications that can solve the disadvantages of RNN and some of these networks become successful for their performance. Gated recurrent unit (GRU)[17] and long short-term memory units (LSTM)[18] are the most used RNN structure currently. These network structures are widely applied in sequential sequence problem, e.g. speech recognition, sequence to sequence translation, price prediction, etc..

1.4 Related Researches

The high demand for reliable and high-resolution algorithms and signals in medical ultrasound imaging has encouraged the neural network to play an increasingly important role in ultrasound imaging. The data-driven neural network approach becomes more and more popular because of the increasing data availability and its robustness. Generally, the ML approaches applied on ultrasound data can be summarized into the following categories.

Classification: One of the most popular scenarios where ML approach is applied on ultrasound data is lesion detection & classification. Mostly focus on human organs like breast, liver and lung, these works usually apply the ML as a classifier or detector for lesions, and part of the computer-aided disease diagnosis of the whole system. In [19], a computer-aided diagnosis(CAD) system is developed for breast ultrasound, where different machine learning algorithms are applied for feature extraction. With deep neural network increasingly involved, in [20], deep convolution neural network (DCNN) is applied to reduce false positive(FP) based on an automated breast volume scanner. And in [21], transfer learning is applied with DCNN model (CaffeNet and VGGNet) for classification, supporting the conclusion that transfer learning with CNN can be used to construct effective classifiers for abdominal ultrasound images. When fusing hand-crafted features and ones extracted by deep learning, [22] proposes a methodology that can improve the performance, circumvent the need for image pre-processing and also computationally efficient.

Segmentation: In segmentation tasks, the structural boundaries of certain objects are to

automatically extracted by the ML algorithms. ML-based segmentation usually extract a structural region by doing pixel-wise classification followed by a smoothing step. Such procedures are similar in the segmentation task for images by using a fully convolution neural network[23].

In [24], a segmentation scheme for 2D ultrasound images is proposed, with discrete wavelet transform (DWT) is used to build the input feature space of the network. In [25], a fully automatic segmentation technique based on ML and statistical pattern recognition to measure Intima-Media Thickness (IMT) from ultrasound Common Carotid Artery (CCA) images is proposed. The automated US segmentation is regarded as challenging for the speckles, shadow, missing boundaries that usually accompany US images.

Other Applications: Some of the works proposed recently focus on the registration of ultrasound images. In [26], neural network, together with code mutual information are combined to perform registration of CT and ultrasound images of the spine. Instead of applying ML approaches for image processing tasks, some work has been proposed to focus on higher-level problems, in [27], a deep convolution neural network is designed for content interpretation in ultrasound videos, [28] proposes a framework to process US videos, accurately acquires fetal ultrasound standard planes by a transferred recurrent neural network.

Within the past few years, deep learning (DL) approaches have been shown to significantly improve performance when compared with classifiers operating on handcrafted features in the classical ways[29]. It is predictable that with more available data and in-depth researches over the ML processing US data, ML or DL algorithms can perform interpretable processing and efficient computation over the ultrasound data, achieving similar progress as in the computer vision domain.

1.5 Objectives & Thesis Outline

Based on the discussion in section 1.4, there are already many researches over the ML approaches processing US data, indicating the effectiveness of ML and DL algorithms in image processing domain, where the classification or segmentation can readily help the process of computer-aid diagnosis system. Most of these works apply large scale, complicated neural networks that specialize in high-level abstract problem such as lesion detection, region of interest (ROI) detection on 2D or 3D image level and other similar problems that involve deep feature extractions and highly non-linear localizations or abstractions. However, for low-level parameters tracking and estimation, most approaches apply deterministic signal processing techniques that are highly demanding computation and memory resource for processing. Hence, the discovery of possible ML approaches that process the ultrasound data to estimate carotid artery parameters in an end-to-end, half-black-box and low-computation-cost manner can facilitate the development of compact devices that specialize at monitoring or detecting pulse wave velocity, and hence make contributions to the research of cardiovascular disease.

1.5.1 Research Problems & Objectives

Following the precedent work[3], where carotid artery parameters are estimated through a digital signal processing (DSP) pipeline and novel pulse wave velocity methods are pro-

posed (introduced in section 2.2), this thesis will focus on the ML approaches that can perform tracking movement of the carotid artery, estimating parameters through a compact, effective and computationally efficient process. Hence, in this thesis, following problems are researched and the objectives are expected:

1. How machine learning algorithms can obtain the same artery motion parameters through proper training, without heavy computation and processing?
2. Can we re-interpret the existing DSP pipeline in the perspective of machine learning? What is the approach to assemble/decompose the subparts in the DSP pipeline in Machine Learning?
3. Is ultrasound data enough for estimation of pulse arrival time (PAT) without ECG signal?

1.5.2 Thesis Outline

This thesis is organized in the following way:

- chapter 1 Introduction: The motivation that focuses on cardiovascular disease, background context, application of ultrasound and machine learning are discussed, laying the theoretical foundation of the thesis; A list of research problems and objectives, an outline is given for the organization of the thesis;
- chapter 2 Data Preparation: Analysing ultrasound data and parameters and the DSP pipeline. Explaining the organization of available ultrasound data for further design of ML approach.
- chapter 3 Methods: From basic principles to the detailed design of the neural networks applied in the ML processing pipeline, containing 4 parts (ROI detection, Time Domain Segmentation and Diameter Tracking, Post-processing)
- chapter 4 Results & Evaluations: Evaluate the result from the ML-based approaches in different aspects, e.g. losses/errors in the desired parameters, numerical analysis of the results to quantitatively assess the cardiac parameter estimation
- chapter 5 Conclusions & Discussions: Conclude the thesis with evaluation results, provide insights for future and possible applications, evaluate the possible extensions with the reference of other applications.

2

Data Preparation

In this chapter, the details of the collection process and configuration of ultrasound data, ultrasound data structure and different modes of displaying ultrasound data are introduced in section 2.1, followed by the introduction of the prior work, a digital signal process pipeline (DSP) that lays the foundation and provides available data and labels for the machine learning approach, in section 2.2. Finally, examples are given for the parameters available extracted by the DSP pipeline.

2.1 Ultrasound Data

2.1.1 Data Acquisition

The ultrasound data provided for the machine learning task is collected by ultrasound transducers that record the movement of the common carotid artery (CCA) of 7 different subjects (healthy humans). These test subjects (38 ± 10 years) were repeatedly measured in 3 sessions over 3 weeks (interval of 1 week). To include diversity in the ultrasound waveform of different features, 3 different posture/state the test subjects were asked to stay: resting condition (2 min), paced breathing (2 min) and hand grip (1 min). In order to detect the evolution of artery diameters, positions and other parameters, the transducer is placed horizontally perpendicular to the CCA of the test subjects, resulting the maximum detection response in the radial direction. Apart from the ultrasound transducer, an electrocardiogram (ECG) is also deployed to record subject's heart activity. The digital processing pipeline needs both the ECG data and ultrasound data to perform parameters tracking, covered in section 2.2.

The ultrasound transducer deployed in the data acquisition is the L11-5v, and the ultrasound system is a Verasonics Vantage 64 model (Verasonics Inc., USA). The transducer consists of 128 elements in a linear array, working on a center frequency (f_c) of 7.8MHz, acquiring a 19.2 mm wide segment with a custom plane wave sequence of sampling frequency 500Hz from the center 64 transducer elements. Based on the fact that the CCA diameter increases approximately 0.01 mm per year from about 4 mm in young adults [30], the above ultrasound processing system should be able to capture all the movement of the artery in the transverse plane, assuming that the artery of the test subject stays

approximately in the same place during the measurement process. Furthermore, with the center frequency of the RF ultrasound signal (7.8 MHz) and the assumed travel speed of ultrasound (1540 m/s in soft tissue, 1580-1630 m/s in the arterial wall and 1570 m/s in blood), the wavelength λ is approximately 0.2 mm for a single pulse and hence, the spatial resolution ($\lambda/2$, 0.1 mm) satisfies requirements of artery wall detection in the axial resolution. With further RF data sampling at 31.2MHz, a spatial sampling distance of 50 μ m can fully represent the original signal with sufficient accuracy. The ECG was recorded simultaneously with the ultrasound data by a Biopac MP-160 base module and ECG100C module (Biopac Inc., USA) with sampling frequency (f_s) 500Hz, which is synchronized with ultrasound signal through an external trigger signal.

2.1.2 Data Format

The acquired data is organized with 3 dimension: depth, time and scanlines (width). The depth dimension corresponds to the short interval where a single pulse is transmitted and received, recording the reflecting pulse wave as depth information, with spatial resolution as 0.1mm, defined by the RF frequency (7.8 MHz). In time dimension, the ultrasound pulses emit with temporal sampling frequency 500 Hz, the frames of each reflected pulses can show the temporal information of the artery, e.g. respiratory movement, distension movement. The scanlines (width) dimension is the width of the elements in the transducer, as a linear array, it detects the information on the transverse plane of the artery. In ultrasound imaging, these three dimensions also correspond to A-mode, M-mode and B-mode.

A-mode: A-mode is the amplitude mode, it shows the amplitude of the reflected pulse

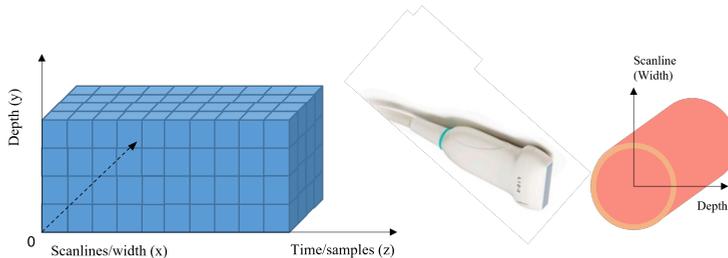


Figure 2.1: Collected Data Format (3D) and the Collection of Ultrasound Data

wave in a short time interval (depth dimension). As shown in figure figure 2.2, A-mode ultrasound data only contains depth information over one scanline and one time frame.

From the A-mode data figure (example from the collected ultrasound data), two peaks at the middle with very low amplitude signal in the middle can be observed, this structure is the detected artery with artery walls on both sides and lumen in between. As the envelope shows, the features of detected artery can be very well represented by the envelope of RF ultrasound signal.

M-mode: M-mode is motion mode, it is the sequence of A-mode data frames recorded for certain duration on a single transducer element (scanline). It is shown on the right of figure 2.3. From the evolving of the waveform, the distension of the artery walls is clearly visible.

B-mode: B-mode is brightness mode, it's a two-dimensional ultrasound image display

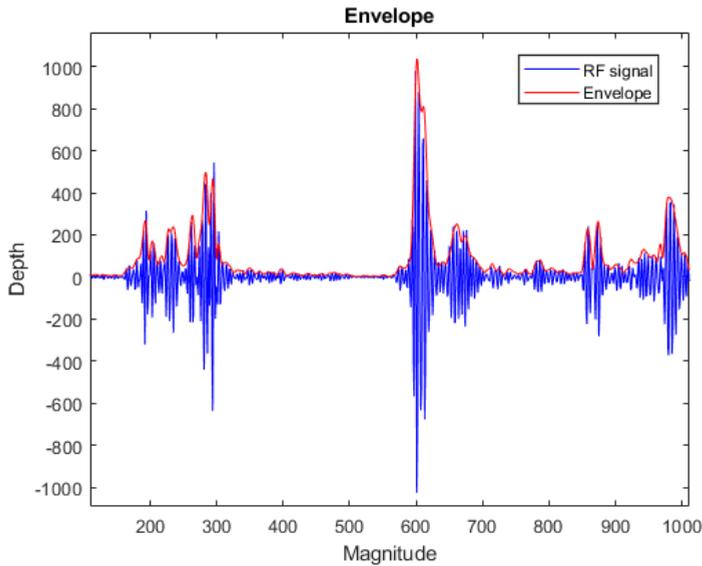


Figure 2.2: An Example of A-mode and Envelope of Ultrasound Data

composed of bright dots representing the ultrasound echoes. It contains spatial information on depth and width dimensions. As shown to the left of figure 2.3, the spatial structure of the artery walls can be seen (the curve of the artery is not obvious as only 8 of 128 scanlines are synthesized and the width of each scanline is scaled for visualization here). In clinical practice, B-mode ultrasound images are widely used as they can show the anatomical structure of tissues and organs.

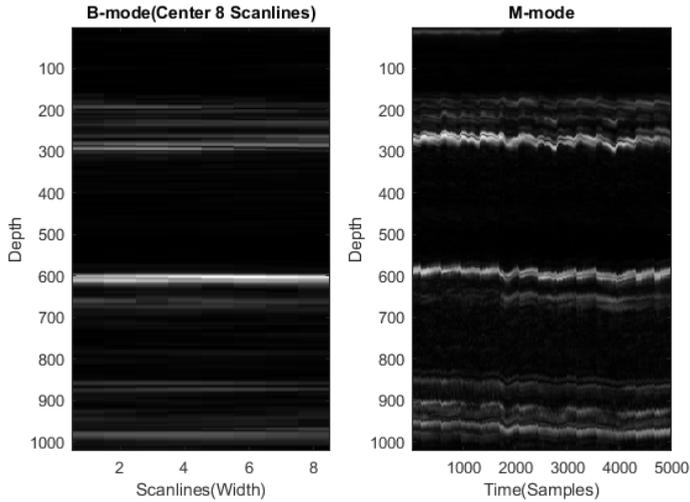


Figure 2.3: B-mode, M-mode Ultrasound Data

2.2 Digital Signal Processing Pipeline

In this section, the digital signal processing pipeline is discussed in parts, and the model where these algorithms/heuristics are based is also explained.

2.2.1 Carotid Artery Model for Ultrasound

Like all arteries, the carotid arteries consist of three layers of tissue: intima (the smooth innermost layer), media (the muscular middle layer), and adventitia (the outer layer)[31]. For simplicity, an assumption can be made on the diameter (5mm) and width of the artery wall (0.5mm). The following scheme figure 2.4 shows how the ultrasound pulses propagate in the z axis and have high reflections upon arriving at the adventitia, the outer layer, while during the transition in the intima (inner layer), low reflectivity is expected due to a minor negative impedance mismatch, resulting in a blur minor peak slightly after the major peak of reflection from the outer layer. In between the inner layer of both artery walls lies the lumen, it is hyperechoic, which means that very few reflections are expected for the transition of ultrasound pulses in such medium. Based on this characteristic, high contrast between the transition from the artery wall (mainly adventitia) and the lumen is expected in the A-mode data and is a very important feature for the DSP pipeline to locate the artery. In addition to the artery, in ultrasound there are other tissues and organs that are visible, e.g. veins, tendon, etc., that are considered as noise and can affect the algorithm to confuse the choice of the actual artery wall.

2.2.2 Design of Digital Signal Processing Pipeline

In precedent work[3], a Digital Signal Processing(DSP) pipeline was designed to process ultrasound data and obtain desired artery parameters (diameter, wall position, distensions, etc.). These parameters will further become the ground-truth labels for the neural net-

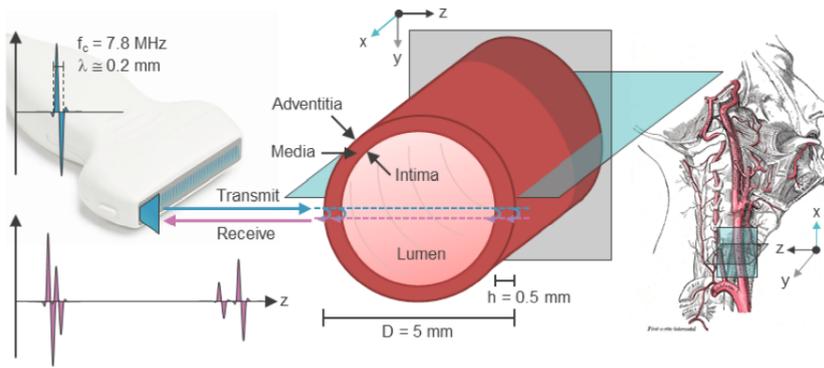


Figure 2.4: Carotid Artery Model for Ultrasound [3]

works. A flow diagram of the DSP pipeline can be shown as figure 2.5.

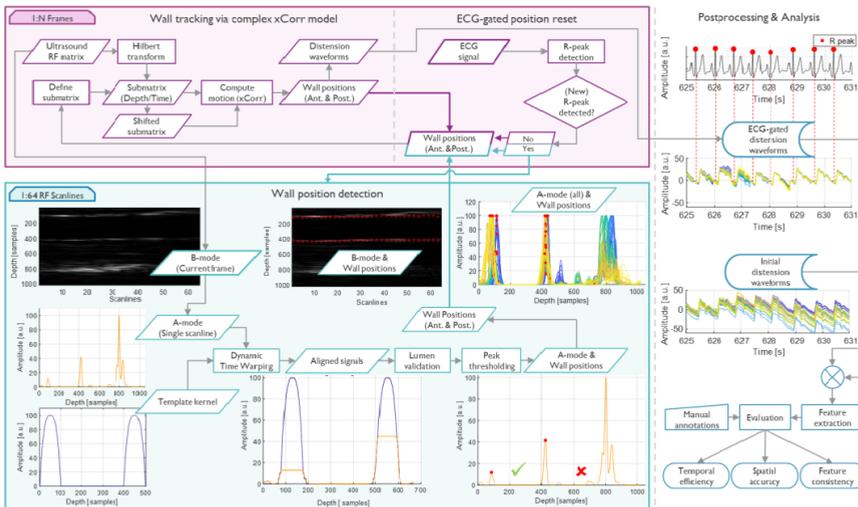


Figure 2.5: Flow Diagram of DSP Pipeline, proposed in [3]

The DSP pipeline consists of 3 main parts: Wall position detection, wall tracking (correlation model) and ECG-gated reset mechanism. After obtaining the raw ultrasound data in the 3D data cube, the data are first processed by Hilbert Transform on the depth dimension for every single time sample and scanline, where an analytic complex signal is acquired. Depending on the process, this analytic signal is fed into the following parts to further process. The preprocessing (Hilbert Transform) and the processing blocks mentioned above are explained below.

Hilbert Transform Hilbert transform is a specific linear operator that takes a function $S(t)$ of a real variable and produces another function of a real variable $H(S)(t)$. This lin-

ear operation is a convolution with the function $h(t) = \frac{1}{\pi t}$, known as Cauchy kernel. The definition of Hilbert transform is given by

$$\begin{aligned} H(S)(t) &= S(t) * \frac{1}{\pi t} \\ &= \frac{1}{\pi} p.v. \left[\int_{-\infty}^{\infty} \frac{S(\tau)}{t - \tau} d\tau \right] \end{aligned} \quad (2.1)$$

where $p.v.[\cdot]$ denotes the Cauchy *principle value* of the improper integral[32]. Hilbert transform can turn the original real signal into a complex signal, where the imaginary part is a version of the original real sequence with a 90° phase shift and the real part is the original signal. Therefore, the energy of the analytic signal (envelope) can be extracted by taking the magnitude of the analytic signal.

Wall Position Detection: Based on the carotid model introduced in section 2.2, a template kernel is designed to match the A-mode signal with the approximate shape of the artery-lumen structure. By averaging all region of interest(ROI)s of the artery from all subjects, the kernel represents all generalized features of the artery-lumen structure. In order to find the ROI in each A-mode frame, dynamic time warping (DTW) is applied to align the original signal with the kernel by a correlation(i.e. a minimization in Euclidean distances). After identifying the location of the ROI that contains the CCA, lumen validation is implemented based on the hyperechoic property(low amplitude and noise levels in the lumen) to detect whether the lumen lies in the middle of selected ROI. Subsequently, the algorithm tries to find the peak on both side of the lumen with thresholding. The location of the peak is further extended as a window to include the waveform of the artery wall, which is further processed in the wall tracking correlation model to calculate the minor phase shift as displacement of the artery wall. The locations of both artery walls are also directly taken into account in the wall positions when an ECG reset signal is detected(covered in section 2.2.2) to avoid drifts in the estimated parameters.

Wall Tracking(correlation model): In this part, a sub-matrix ($x = 2$ scanlines(0.6mm), $y = 50$ depth samples(1.2mm), $z = 5$ frames(10 ms)) is selected from the raw data cube(complex signal, after Hilbert transform) to estimate the phase shift via complex cross-correlation. By shifting the sub-matrix one sample in time, and keeping the rest to overlap, the phase difference can be estimated in the following way[33].

ECG-gated Position Reset In the electrocardiogram, the R wave is the most dominant wave characteristic of the QRS complex (three graphical deflections seen in a typical ECG), and the peaks in the R wave could indicate left ventricular hypertrophy. As a reference to segment each heart cycle in time, such R-peaks are detected as an easy flag event in the ECG.

The ECG-gated position reset applies another sensor measuring the ECG waveform of the subject. Provided the ECG data, this algorithm block checks if a new ECG R-peak is detected; if so, the tracking block stops processing the frame-to-frame phase changes as movement and the wall position resets to the position of this moment(the continuity in phase with previous samples is abandoned), this will result in a discontinuity in the estimated parameters(wall positions, distension, diameter, etc.), to prevent the estimated wall positions gradually drifting from the right ones.

2.3 Data Processing

In the DSP pipeline, the following parameters are estimated by direct tracking / detection or calculated as a linear combination of previously acquired parameters: artery wall positions (anterior, posterior), artery diameter, artery motions (anterior, posterior) and distension waveform.

2.3.1 Carotid Artery Parameters

The target parameters to be estimated are listed and explained below.

Artery Wall Positions: These are the absolute locations where the anterior (posterior) artery wall lies in each A-mode frame. Both walls move in the opposite direction to each other at the same time, representing the distension of the artery. In the anatomic structure of human's neck, the artery is positioning in front of multiple neck muscles and the cervical vertebrae, resulting in a smaller range of movement of the posterior artery wall than that of anterior artery wall. As essential independent parameters, the positions of the artery wall (anterior / posterior) are used to derive other parameters.

Artery Wall Motion: The artery wall motion is the relative movement of each wall with reference to the beginning of one heart cycle. Assuming at time t_0 an ECG R-peak is detected, in the following heart cycle time t , the artery wall motion can be calculated as

$$M_t = P_t - P_{t_0} \quad (2.2)$$

Once the ECG reset event takes place, the motion of the anterior and posterior walls is set to 0, and the new heart cycle that begins takes this point as a reference. For simplicity, the ECG R-peak will be denoted as cardiac cycle marker (CC markers).

Artery Diameter: This is the diameter of the current A-mode frame of the artery. As one of the relative quantities, the diameter can be calculated by the difference in wall positions.

$$Diam = P_{Post.} - P_{Ante.} \quad (2.3)$$

Since the diameter evolves over time in one heart cycle, it becomes a very important parameter for estimating the pulse wave velocity (PWV), and hence can provide information for arterial stiffness assessment. Explanations are given in section 3.6

Artery Distension: Similar to the relation between wall motion and wall position, the distension parameter is the relative movement of the artery diameter, with reference to the ECG reset flag (CC markers). By subtracting the reference diameter, the distension starts at 0 when a new heart cycle begins, and the systolic and diastolic phases are shown more clearly. Assuming at time t_0 an ECG R-peak is detected, in the following heart cycle time t , the distension can be calculated as

$$Dist_t = D_t - D_{t_0} \quad (2.4)$$

Due to the ECG resetting mechanism, in all these estimated parameters, a discontinuity exists in every heart cycle when the ECG R-peak is detected. This is inevitable in the current DSP approach. While certain averaging over time can improve the signal quality close to a resetting point, the data prepared for further machine learning algorithm is the original one.

Cardiac Cycle Marker: These markers are detected ECG R-peaks and synchronized with the ultrasound data. As the estimation of pulse arrival time (PAT) requires the information of ECG R-peak, the CC markers are extracted from the ECG signals with only the indices of R peaks.

2

2.3.2 Example of the Result of DSP Approach

For the purpose of better illustration, the result of the estimated parameters over time of a measurement (test subject 2, third session, hand grip condition) in section 2.3.2.

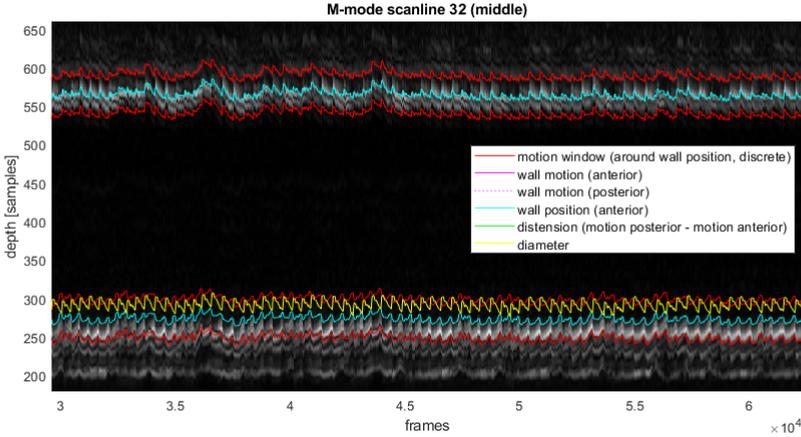


Figure 2.6: Result of DSP Approach: All Parameters

As the parameters set for the RF ultrasound, the scaling factor of the samples-to-SI units is defined by the ratio of the propagation speed of ultrasound pulses and the RF sampling frequency (4 times as the RF center frequency).

$$\alpha = \frac{c}{2f_{RF,s}} \quad (2.5)$$

Such setting gives a scaling factor of 2.4648×10^{-5} , in m , from the example given above section 2.3.2, diameter usually has around 280-300 samples in the depth dimension, corresponding to 6.9 - 7.4 mm in the actual length. This conclusion is consistent with statistics where a study implied that the mean diameters of CCA of women are 6.10 ± 0.80 mm and in men 6.52 ± 0.98 mm [34].

3

Methods

In this chapter, three parts will be introduced covering the basic theories and principles of the whole design. First, an overview and formulation on the problem and an explanation of the challenges, followed by the general design of the whole pipeline, is explained. In the second part, the foundations and principles of neural network design are provided to theoretically support the design of neural networks and machine learning (ML) pipeline. In the third part, a detailed formulation of the ML processing pipeline and post-processing pipeline for pulse arrival time (PAT) estimation is given; with this, the pulse arrival time can be estimated from the parameters obtained at the outputs of each module. The ML processing pipeline consists of three subparts: ROI detection, time domain segmentation and diameter tracking. With these sub-modules combined, the artery parameters can be estimated similar to the previous introduced digital processing (DSP) pipeline. Furthermore, post-processing is introduced to extract time-related information from the first derivative of the detected diameter. By identifying the max slope in the first derivative every heart beat, the PAT can be estimated combined with cardiac cycle (CC) markers.

3.1 Problem Formulation

The formulation of the ML-based approach has 2 main parts, the decision of the input-output pair for the ML pipeline (data, labels), and the challenges that need to be solved and inspiring heuristics that can provide detailed insights.

3.1.1 Input & Output

To decide which form of the ultrasound data is suitable for a machine learning-based approach, raw ultrasound data, analytical signal (via Hilbert Transform), envelope (magnitude of the analytical signal) and phase signal (phase of the analytic signal) are inspected and compared to select a suitable form for neural network learning, their characteristics and perspectives from the neural network are listed in table 3.1. Similar to many ultrasound data processing tasks, the envelope (image) of ultrasound data is suitable for machine learning tasks (e.g. [35], [36], etc.); in this work, it is clear that the neural network should use the envelope of ultrasound data as input based on the reason in the table 3.1.

Table 3.1: A comparison between different data representations for machine learning-based approach

Data Representation/Pre-processing	Characteristics	Neural Network Perspective
Raw Ultrasound Data (None)	Oscillating sinusoidal signal, absolute amplitude w.r.t. physic meaning	Difficult for neural network (hard to extract a pattern in noisy oscillating signals), usually processed by FFT for later processing
Analytic Signal (Hilbert Transform)	Complex signal, a 90° shifted imaginary part, oscillating sinusoidal signal on real and imaginary part	Not a conventional signal form (complex) for neural network, complicated back-propagation and gradient computation, usually processed by correlation
Envelope (Hilbert Transform, magnitude computation)	Positive envelope signal, smooth transitions	Proper regional feature for neural network, can composite image signal,
Phase (Hilbert Transform, phase computation (<i>argtan</i>))	Phase of the analytic signal, sawtooth-like linear signal ranging from $[-\pi, \pi]$	Tiny-scale movement between depth/time samples, no position/object information, can only be used as an augmentation

After several experiments, we conclude that only using envelope data can a neural network successfully converge.

From the available parameters of the DSP pipeline (artery diameters, positions of artery walls, distension waveforms, CC markers), the *diameteris* chosen for the following reason: **Relative Quantity:** Unlike the positions of artery walls (anterior, posterior), the diameter does not have the absolute position information about the artery walls, with the correct selection of the artery, this is directly available without inferring the position of each artery wall.

Direct Estimation: As explained in section 3.6, the diameter is the key parameter that contains information to estimate PATs, estimating directly from the diameter can avoid detecting both artery walls, which could introduce more errors and artifacts. Additionally, the diameter waveform has a wider range compared to other parameters.

As the estimation of pulse arrival time (PAT) also requires ECG R-peaks, the CC markers are required to train the neural networks. In section 3.4, the details of the labeling scheme of CC markers are introduced.

Therefore, the input-output pair for the ML approach is selected as the envelope of ultra-

sound data (image) and the diameter of the artery, and modified CC markers.

3.1.2 Challenges & Heuristics

Based on the observation of the envelope of ultrasound data, there are a few challenges that should be taken care of in the ML processing pipeline.

Unwanted Reflection (Noise): As we can perceive in the figure 3.1, apart from the region of interest (ROI) that consists of the artery wall and lumen in between, there are obvious reflections of tendons or other tissues in the vicinity of the artery, which could cause problems for the neural network to ‘understand’ what exactly is the artery and its diameter. It can be expected that without proper pre-processing, it would be very hard for the neural network to train a ‘black box’ to interpret full-depth signal into a single diameter value. Therefore, prior to diameter tracking, a selection of the ROI will definitely be necessary.

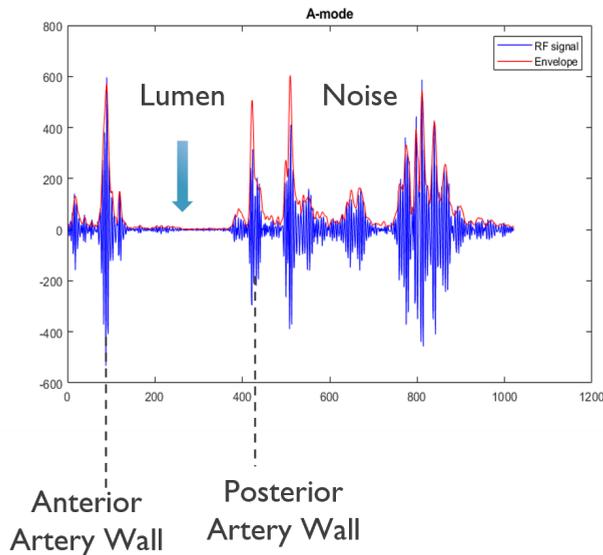


Figure 3.1: Artery Walls, Lumen and Unwanted Reflection (noise)

Temporal & Spatial Processing: Since in the ML pipeline the diameter (as well as other carotid artery movement parameters) and the CC markers (temporal event) are expected, it is necessary that the designed ML pipeline has both temporal and spatial (depth dimension) processing. The spatial information of the artery wall should be computed over each time frame, while the CC markers should be derived from a time series. Consequently, in the ML pipeline, there are multiple sub-networks that are specializing these different tasks.

3.1.3 Machine Learning Processing Pipeline Design

To tackle the challenges in the previous problem formulation section 3.1.1, in the ML pipeline structure, the entire cardiac parameter estimation task would be divided into 3 consecutive parts. The first part is the ROI detection that crops the full-depth input

data to a part only contains artery walls and lumen. The second part consists of a neural network or a ML related detection algorithm that segments the input sequence as single heartbeats, and the third part takes care of the tracking of artery wall movement, has a structure similar to convolution neural network (CNN) or recurrent neural network (RNN). This approach is based on the assumption that during one heartbeat the statistics of ultrasound data does not vary much, and the low-frequency patterns(e.g. breathing) can also be cleared out as interference during the tracking process. The following diagram shows the machine learning processing pipeline. Eventually, the machine learning processing

3

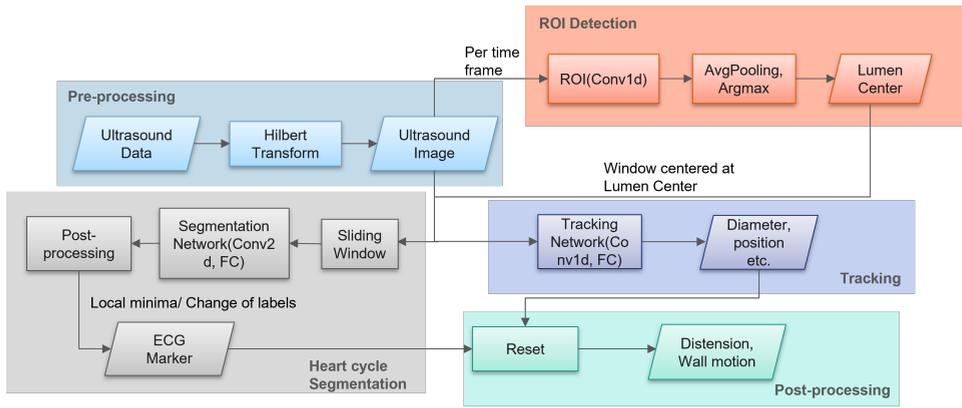


Figure 3.2: Machine Learning Processing Pipeline

pipeline should be able to identify each heart cycle and infer the artery diameters (artery wall positions), which can be further processed to estimate cardiac parameters (e.g. pulse wave velocity).

3.2 Neural Networks

In the machine learning processing pipeline, the neural networks applied in different parts are mainly the convolution neural network (1D, 2D) and the artificial neural network(ANN). The working process of neural networks is explained below. **Fully Connected Layers:** As introduced in chapter 1, the fully connected layer consists of multiple neurons, and the connections of these neurons form a neural network. When all inputs from one layer are connected to every activation unit of the next layer, then it is called a fully connected layer. As an element of neural networks, a neuron is essentially a mathematical function that models the functionality of a biological neuron, illustrated in figure ???. The mathematical expression of a single neuron is given in section 1.3.1. Activation functions are nonlinear, according to the Universal Approximation Theorem, a two-layer neural network with a nonlinear activation function can be a universal function approximator [37]. In practice, activation functions such as *sigmoid*, *Rectified linear unit (ReLU)*, *Hyperbolic tangent (tanh)*, etc. are commonly applied in practice. In this work, *ReLU* is applied as the main activation function for neural networks in the machine learning pro-

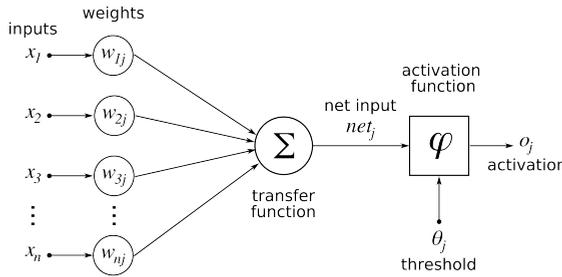


Figure 3.3: Illustration of a Neuron in a Neural Network

cessing pipeline. The *ReLU* is defined as:

$$ReLU(net) = \max(0, net) = \begin{cases} 0 & \text{if } net \leq 0 \\ net & \text{if } net > 0 \end{cases} \quad (3.1)$$

Empirically, different choice for activation function leads to difference performance in terms of convergence and accuracy. The *ReLU* and its variants (*Leaky ReLU*, *GeLU*) have become very popular for many neural network applications.

Convolution Layers: Commonly seen in image processing or pattern recognition, convolution layers can spatially correlate pixels or samples of certain topology and geometry. To ensure the invariance of some degree of shift, scaling, and distortion, convolution neural networks combine three architectural ideas: *local receptive fields*, *shared weights* and *spatial sub-sampling* [38]. Following these ideas, convolution kernels that contain shared weights are applied to extract the features of the image, which results in *feature maps* in the next layer, describing how these features, represented by the convolution kernels are distributed in the space. Generally, the convolution operation can be expressed as

$$g(x, y) = w * f(x, y) = \sum_{dx=-a}^a \sum_{dy=-b}^b w(dx, dy) f(x - dx, y - dy) \quad (3.2)$$

where $g(x, y)$ is the output of convoluted image (*feature maps*), w is the weighted kernel, $f(x, y)$ is the input image. In the context of convolution neural networks, this boils down to a weight kernel doing pixel-wise product with a moving windowed ROI in the image, illustrated as figure 3.4. Similarly, in 1D situations, the convolution kernel is aggregating the features into 1D feature maps.

$$g(x) = w * f(x) = \sum_{dx=-a}^a w(dx) f(x - dx) \quad (3.3)$$

Due to low computation complexities and compact configurations, 1D convolution neural networks also widely applied in signal processing without involving images or videos [39].

To ensure that the extract features are insensitive to the small variance of shift or distortion, pooling layers usually applied after convolution layers, after or before the activation

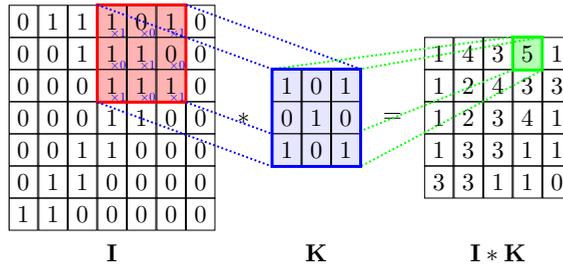


Figure 3.4: Illustration of 2D Convolution

function. *Max pooling* is one that is often applied in CNNs, it calculates the maximum value for patches of a feature map, creating a downsampled feature map at its output.

Stochastic Gradient Descent & Back-propagation In order to minimize the objective function (loss function), stochastic gradient descent (SGD) is used to update the weights and biases of the neural network, given a subset of the training sample. Considering dividing the whole dataset into subsets (batches), a single subset containing m training samples $[x^{(1)}, x^{(2)}, \dots, x^{(m)}]$ with respective labels $[y^{(1)}, y^{(2)}, \dots, y^{(m)}]$, the weights in the neural network can be updated as [40]

$$W^{i+1} = W^i - \alpha \frac{1}{m} \sum_{i=1}^m \nabla_w L(x^{(i)}, y^{(i)}, W^i) \quad (3.4)$$

where α is the learning rate, it is the step in optimization, controls how quickly the model is adapted to the problem. ∇ is the gradient and L is the loss for a specific sample and its label.

As stochastic gradient descent only takes a subset of all samples in the datasets, the result in the loss function will converge with an oscillate and noisy pattern as batches are trained. While in gradient descent, the weights only update after the whole dataset is trained. As compared to gradient descent, where model parameters only update when the entire dataset is fed in, the stochastic gradient descent converges faster and is not as computationally expensive as gradient descent. In practice, the batch size (the size of the subset of the dataset) is a tunable hyperparameter that has impacts on convergence and performance of the neural network.

Back-propagation is a widely used algorithm for training feed-forward neural networks. It computes the gradient of the loss function with respect to the weights of the network for a single input-output instance. By calculating the gradient through the neural network, the influence of each weight and bias towards a certain neuron is determined, allowing the adjustment of the weights and biases via optimization (gradient descent). Using chain rule, the partial derivative of the error with respect to a weight w_{ij} can be computed as

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial y_j} \frac{\partial y_j}{\partial w_{ij}} \quad (3.5)$$

where o_j is the activation function of the output of j -th neuron.

Loss Function In machine learning, the loss function is the mapping of the inference

decisions with their associated costs. It measures how far an estimated value is from its true value. For neural networks, loss function is the objective that the network training process tries to optimize the neural network.

Depending on the application, the choice of the loss function differs from quadratic cost functions, exponential cost functions, divergence, etc.. For a neural network to be trained, the loss function should be able to compute its gradient with respect to its input term. In regression tasks, quadratic cost like *mean square error (MSE)* is widely applied, while in classification tasks, cross-entropy loss is more often used. In this work, *MSE* and *Huber loss* are used to train the neural networks. Given the number of samples N , label y_i , and the output of the neural network x_i , *MSE* can be expressed as

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (3.6)$$

and *Huber loss* is given below

$$l_i = \begin{cases} 0.5(x_i - y_i)^2 & \text{if } |x_i - y_i| < \text{delta} \\ \text{delta} \cdot (|x_i - y_i| - 0.5 \cdot \text{delta}) & \text{otherwise} \end{cases} \quad (3.7)$$

$$L = \frac{1}{N} \sum_{i=1}^N l_i \quad (3.8)$$

As a combination of *l1 loss* and *MSE*, *Huber loss* has the advantages of both loss functions, it is more sensitive to outliers than *MSE*, and also guarantees the smoothness for the absolute difference smaller than the *delta*.

Optimization: As introduced in *Stochastic gradient descent*, the optimization of the neural network is done via *Stochastic gradient descent* and gradient variants algorithms, e.g. *Adagrad* [41], *RMSprop* [42], *Adam* [43], etc.. As all of these algorithms are iterative, they differ in the updating steps of calculating and applying the gradient and momentum. The in-depth discussion of these optimizing algorithms is beyond the scope of this thesis. In this work, the optimizer chosen for the optimization of neural networks is *Adam* for its good performance over large-scale datasets, convergence rate and popularity. The *Adam* algorithm is described as follows (Algorithm 1).

Regularization: In machine learning, regularization is a process that is used to calibrate the models in order to minimize the adjusted loss function, it is often used to obtain results for ill-posed problems or to prevent overfitting [44]. Regularizations are usually applied in the loss function as an extra term that tries to constrain the weights of the layers in the neural network. This can be expressed as

$$L_r(\hat{x}, x, w) = L(\hat{x}, x) + \lambda R(w) \quad (3.9)$$

where \hat{x} is the output of the neural network, x is the respective label, and w is the weights of the neural network. and λ is the coefficient that determines the importance of the term of regularization error with respect to the data-dependent error. The term of regularization error usually applies *l1-norm* (LASSO problem) or *l2-norm*. In this work, the regularization

Algorithm 1 Adam Algorithm**Require:** Step size α , exponential decay rates β_1, β_2 , $f(\mathbf{w})$ Initialize: parameter vector \mathbf{w}_0 , first moment vector $\mathbf{m}_0 = \mathbf{0}$, second moment vector $\mathbf{v}_0 = \mathbf{0}$, timestep $t = 0$ **while** \mathbf{w} not converged **do**1: Get gradients w.r.t. stochastic objective at timestep t :

$$\mathbf{g}_t = \nabla_{\mathbf{w}} f_t(\mathbf{w}_{t-1})$$

2: Update biased first moment estimate:

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t$$

3: Update biased second raw moment estimate H :

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2$$

4: Compute bias-corrected first & second moment estimate:

$$\hat{\mathbf{m}}_t = \mathbf{m}_t / (1 - \beta_1^t), \hat{\mathbf{v}}_t = \mathbf{v}_t / (1 - \beta_2^t)$$

5: Update parameters

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \alpha \hat{\mathbf{m}}_t / (\sqrt{\hat{\mathbf{v}}_t} + \epsilon)$$

end while**Return** \mathbf{w}

3

term for training the neural network is chosen as l_2 -norm for its good performance in controlling the weights from ‘memorizing’ the data, which can be expressed as

$$\lambda R(\mathbf{w}) = \lambda \|\mathbf{w}\|_2 \quad (3.10)$$

Apart from the regularization terms added on the loss function, in the neural network, *dropout* can be applied on the layer to randomly deactivate some neurons, by a specified probability. In each update, some of the neurons become deactivated so that the situation in which only a part of the same neurons contribute to the output can be prevented. In this work, *dropout* can also be applied to the first fully connected layer to prevent overfitting. Figure 3.5 illustrates how *dropout* works in the fully connected layers.

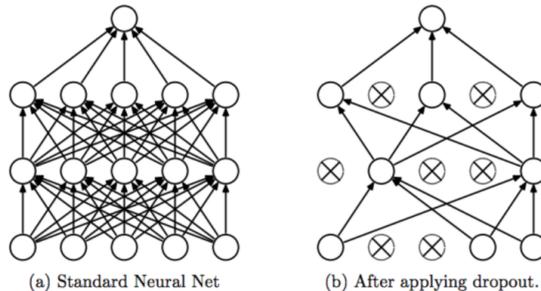


Figure 3.5: *Dropout* Mechanism During One Weight Update, Figure by [4]

Normalization: In machine learning, normalization is the process that transfers the numeric range of the data into a new range, without distorting differences in the ranges of values or losing information. Commonly used normalization approaches are *Z Normal-*

ization, *Min-Max Normalization*, *Unit Vector Normalization* etc.. In practice, these normalization approaches usually modified for batches or layers, e.g. *batch normalization* [45], *layer normalization* [46], etc.. Applying normalization prior to the training phase is crucial to obtain good results and significantly speed up the calculations towards convergence [47]. In this work, *batch normalization* is used in some of the neural networks. In the 1D scenario, *batch normalization* can be expressed as

$$y = \frac{x - E[x]}{\sqrt{\text{Var}[x + \epsilon]}} * \gamma + \beta \quad (3.11)$$

where γ , β are trainable parameters that adjust the standard deviation and bias of the distribution of hidden layers where *batch normalization* is applied. Empirically, *batch normalization* is usually chosen for ANNs, CNNs, while *layer normalization* is usually applied to RNNs.

3.3 Part 1: ROI Detection

Inspired by the ROI detection in the wall position detection module in the DSP approach section 2.2.2, ROI detection can also be done in a neural network by training such a kernel to aggregate the ensemble information over a large amount of the data including the artery-lumen structure. With the neural network, it is expected that every time an envelope data is fed in, the output should be an position representing the lumen center, and hence the window can be defined with a fixed window length and a lumen center coordinate. This localization problem is similar to many object classification / detection tasks in which the neural network is trying to regress the bounding box coordinates and size of targeted objects [48]. Usually such neural network applies deep convolution neural networks (DCNN) as a feature extractor, followed by fully connected layers to regress the locations of the target object, for example, in [48] a 5-layer CNN was deployed to extract the features, and in the extension works [49] and [50] large-scale pre-trained neural network(e.g. 16 layers VGG-16, etc.) are applied as the feature extractor.

3.3.1 Problem Formulation

Considering the complexity of the localization problem for images, it can be expected that a more simple structure should be sufficient in our task where only one ROI structure is to detected in 1D data sequence. The problem of detecting the lumen center can be formulated with the following heuristics:

Large-sized 1D Kernel: Based on the hyperechoic property of the lumen (low amplitude or noise levels), the ROI is expected to have a peak-low level-peak structure, roughly symmetric on two sides. Such simple feature should be able to be well detected by a few large-size convolution kernels, with approximate the size of larger than the artery-lumen structure.

Few Convolution Layers: As the convolution kernels are sliding over the envelope of one A-mode data frame, with well-trained kernels, the output after the convolution should give a high response at the lumen center position. Since it is in fact a correlation between a kernel and the signal, there is no necessity to have many layers which will extract many

and deep features from the envelope data. On the other hand, the feature extraction of a simple structure over 1D data sequence should be sufficiently tackled by a shallow neural network without too much parameters, saving computation resources for further processing on the tracking or segmentation task.

Averaging & Post-processing: Instead of applying fully connected layer to regress the location of lumen center from the extracted feature of convolution layers, averaging pooling and an *argmax* are applied to reduce the training and computation needed for a highly non-linear localization problem.

These heuristics break down the detection problem into a convolution neural network to train the kernels and deterministic post-processing to extract the lumen center (center of the ROI). The design of these ROI detection network is explained in the following section.

3

3.3.2 ROI Detection Network Design

Based on the heuristics, the ROI detection network should consist following parts: convolution layers, averaging pooling and an *argmax* operation that return the argument of the maximum value. The core of the ROI detection are the settings for the convolution layers: convolution kernels and response vector, they are explained as below:

Convolution Kernels: To detect the artery-lumen structure and produce an easily interpreted result for post-processing modules, the convolution kernel is expected to penalize locations which are not lumen center, this will result in a kernel that has a similar shape to the artery-lumen structure, with larger values on both sides, and low/negative values in the middle of the kernel, so as to have a high response at the center of lumen, with great penalty on the amplitudes that occur in the middle.

Apart from the kernel size, kernels should be few and multiple to tackle the variation of the artery-lumen structure of different test subjects. Biologically, the diameter of the artery, the width of the media, and the impedance mismatch are unique in different individuals, and such variation can be averaged in multiple kernels to have a better generalization than just using one kernel.

Hence, in ROI detection, the convolution part consists 2 layers, with the first layer: 3 kernels of size $1 * 501$ (include the entire structure of artery and lumen), and the second layer of 1 kernel of size 1 (synthesis over 3 outputs of 3 convolution kernels).

Training Labels: Response Vector (Convolution Layer Output): As the target output is representing the response of convolution layers, one of the straight way is to take the lumen center and its neighboring positions to have higher response, and the response decreases as the distance to the lumen center increases. At the same time, this output vector should be constrained to have very low or no response as the position close to artery walls and other positions. It resembles the probability of the occurrence of lumen center across all depth, while different in the specific value settings.

As the label of the training process, the response vector of an frame of envelope data can be calculated as the following:

The lumen center positions are not directly available from the DSP approach, however, a good estimation can be the midpoint between the estimated wall positions. Hence, the lumen center can be calculated as

$$Lumen_c = (P_{Ante.} + P_{Post.})/2 \quad (3.12)$$

With the lumen center available, the response vector can be constructed by the distance from the lumen center to the current depth position. Apart from the artery wall positions and lumen center, the midpoints of the lumen center and both wall positions are defined as *cut-off points*, the response becomes zero with current depth position lying further than the *cut-off points*. The presence of the *cut-off points* ensures the output of convolution layers to have high response only when the kernels are at the vicinity of lumen center, while it also keeps a relatively smooth profile similar to the probability density function (PDF) to avoid imbalance distribution in the latent space in the output. The *cut-off points* are defined below

$$\begin{aligned}
 P_{cut-off(a)} &= (Lumen_c + P_{Ante.r})/2 \\
 P_{cut-off(p)} &= (Lumen_c + P_{Post.})/2
 \end{aligned}
 \tag{3.13}$$

With the *cut-off points* defined, the maximum distance before set to 0 is the distance between *cut-off points* and the lumen center. By setting up the labels for corresponding ultrasound envelope for every sample, the convolution layers of the ROI detection should be able to approximate the kernels for outputting large values at around the lumen center location. Denote the Euclidean distance of the lumen center and current point d , current position at i -th depth as p_i , we have

$$d_i = (p_i - Lumen_c)^2, P_{cut-off(a)} < i < P_{cut-off(b)}
 \tag{3.14}$$

Normalize the distance to scale 0-1 and flip, the response vector R_i can be expressed as

$$R_i = \begin{cases} 1 - \frac{d_i}{(Lumen_c - P_{cut-off})^2} & P_{cut-off(a)} < i < P_{cut-off(b)} \\ 0 & \text{Other} \end{cases}
 \tag{3.15}$$

The images below are to illustrate how the response vectors are generated, the width of the non-zero indices on the depth dimension controls the penalty of the deviation from the correct lumen center position. The hyperparameters and settings of the ROI detection

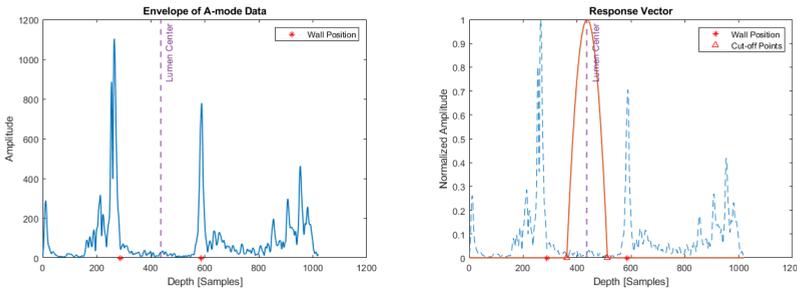


Figure 3.6: Response Vector(Labels for ROI Detection). The left figure shows the original envelope of A-mode data, the right figure shows the response vector calculated by the artery wall positions

network are listed in table 3.2.

The neural network structure of ROI detection can be illustrated as figure 3.7.

Setting	Symbol	Type or Value
Network Structure	-	1D Convolution, Feed forward
Activation Function	g	LeakyReLU
Loss Function	L	MSE
Optimizer	-	Adam
Learning rate	α	$10^{-7} - 10^{-6}$
Regularization	R	<i>dropout</i> , l2-norm
Dropout probability	$P_{Dropout}$	0.05
Weight decay	λ	1^{-4}
Training epochs	N	>10
Number of inputs	x	1020
Number of outputs	y	1

Table 3.2: Hyperparameters and Settings of the ROI Detection Network

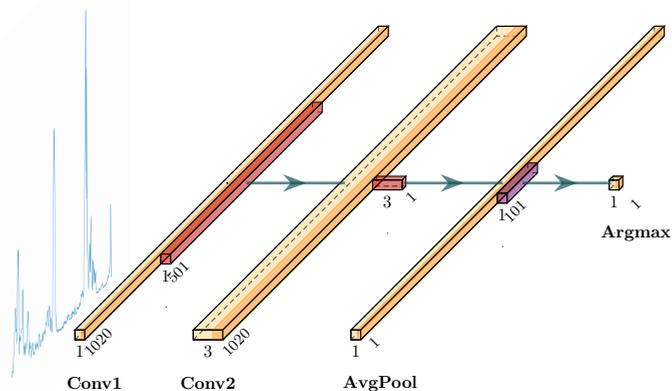


Figure 3.7: ROI Detection Neural Network Structure

3.4 Part 2: Time Domain Segmentation

In the segmentation part, a neural network is expected to segment a long sequence of ultrasound signal recording by the marking event, the cardiac cycle marker (CC marker), which is the same as the moment of ECG R-peak. Consequently, each segment is a heart cycle (heart beat) that starts from the detected CC marker, ends when a new heart cycle is detected. These segmented sections contain full information happened inside the artery (wall movements, distension, etc.). In the whole ML pipeline, the ECG marker can be used to determine the relative movement of the artery wall, the diameter with respect to a reference point, it is also a key feature in estimating the pulse arrival time (PAT). In the DSP pipeline, the PAT estimation and the parameters like artery wall motion are computed over two modalities, ultrasound and ECG, while here it is assumed that the characteristics of the movement revealed by ultrasound image where a CC marker is detected, should be able to be captured by the neural networks; hence the estimation of PAT or artery parameters can be completed only using one modality, ultrasound. As CC markers give the moment when an ECG R-peak arrives, such neural networks should be trained by matching the CC markers on the ultrasound data.

Observing the ultrasound image of one measurement, with a temporal sampling frequency of 500 Hz, one heart cycle that is approximately 0.6 - 1.0 s on average, will result in 300 - 500 time samples in the ultrasound data. The illustration of the detected CC marker and ultrasound image is shown in figure 3.8. It can be observed that in every heart cycle, an upstroke of the artery distension takes place in the middle of one heart beat, following the relatively slow motion of the artery contracting and shrinking in diameter. These characteristics represent the systolic and diastolic phase of one heart cycle, and important biological features can be extracted in the arrival time and waveform of them (e.g., blood pressure, CVD risk etc. [51]). Also, the anterior and posterior walls of the artery moves in the opposite direction (towards or back to the transducer), this results in some roughly symmetric patterns along the lumen center on depth dimension of the both walls. Since ROI detection has made it possible to detect the lumen center, the identification of CC markers can be carried out on the cropped data that mostly contain the artery. On

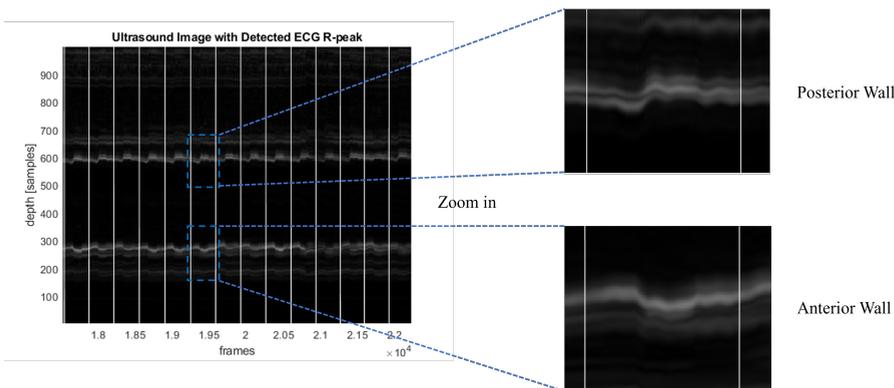


Figure 3.8: Example of Detected CC Marker on the Ultrasound Image (White Vertical Line implies Detected CC Markers)

the basis of these observations of the CC Marker and ultrasound data, the heuristics and requirements for such a segmentation neural network are discussed in the next section.

3.4.1 Problem Formulation

Unlike many image segmentation tasks [23], where pixel-wise labels are assigned for images, the segmentation based on CC markers can be reduced to the detection of certain features in images, and further segment the heart cycles according to these detected moments, since only one dimension segmentation is needed. In order to solve this problem in the image perspective, the data format fed into the neural network should be images, containing a window of time samples at a given time point. Following this philosophy, sliding window of a certain length is set as the input of the neural network. Consequently, two different strategies are proposed to train such a neural network.

The first is to treat every heartbeat data as one input image, and by feeding the neural network with numerous heartbeat data, the distribution of intensity, temporal features, and statistics should be ‘learned’ by the neural network, such that a representation in latent space is given in the output of the neural network. The training process is similar to an ‘averaging’ process that integrates all heart beat data in the training dataset. To satisfy these requirements, an autoencoder that is trying to reconstruct the image itself, with 2D convolution layers extracting geometric features, is needed. In the end, it is expected that with the input of the entire measurement of ultrasound data, this network outputs the distance to the ‘learned’, averaged heartbeat in the latent space representation, such that the local minimum in latent space distance is regarded as the occurrence of one CC marker. However, due to the variability of the heart rate of different people, different situations, the length of one heartbeat does not have a fixed value.

The second is to input the ultrasound image with a fixed length of frames, with labels specifying the probability (or distances to) a CC marker occurs.

To formulate this problem, following heuristics are come up with as either requirements or properties of the desired segmentation neural network:

Difference Processing over Depth and Time: From the example of the recorded CC markers in ultrasound data 3.8, it is obvious that depth information occurs only on a few samples, while each sample on the time axis would be informative in inferring the probability of the occurrence of the CC marker. As the data are arranged in images, it is important that depth and time be treated discriminatively. Since the information on depth dimension is more sparse, compression, convolution, down-sampling or integration may be useful to synthesis the original image into a time sequence. Also, along the time dimension, less processing should be applied, which reduces the length of one window, to keep the temporal information as intact as possible. This will require the neural network to either equip nonsquare kernels in the convolution layers or process in depth and time dimension sequentially.

Sliding Mechanism: As every heart cycle differs in length slightly, it is very difficult to determine a fixed window length that divides the long sequence into short intervals of the same length and, meanwhile, have the CC marker occur in every heart cycle. Also, dividing the measured ultrasound image into subimages without overlapping will result in much fewer images for training, and the prediction of the probability (distance) of an occurrence of an CC marker in the current subimage will not be coherent and the accuracy

will be greatly decreased. Therefore, a sliding window is constructed over the measured ultrasound image with a step size much smaller than the window length (a large part of the image overlapping with neighboring images). In this way, the number of images available in a measurement is

$$N_{samples} = \lfloor (L_0 - L_{win}) / l_{step} \rfloor + 1 \tag{3.16}$$

where $N_{samples}$ is the number of samples available, L_0 is the length of the entire ultrasound image in time, L_{win} is the window length, l_{step} is the step size. The division is floored to the nearest integer before adding one.

Shift Sensitive: Observation of ultrasound images reveals that the CC marker occurs prior to the upstroke in the artery diameter (approximately 80 - 110 time samples), when the artery is in the systole phase. As above, the sliding window is applied to the ultrasound recording, the regional feature of the image is expected to shift in the consecutive windows. To correctly infer the probability or distance of the occurrence of the CC marker, the neural network should be shift sensitive to certain spatial features, unlike the translation-invariant conventional CNNs where only the combination of such features is interested in classification or regression. An illustration of the difference between shift-sensitive and translation-invariant is given in figure 3.9.

3

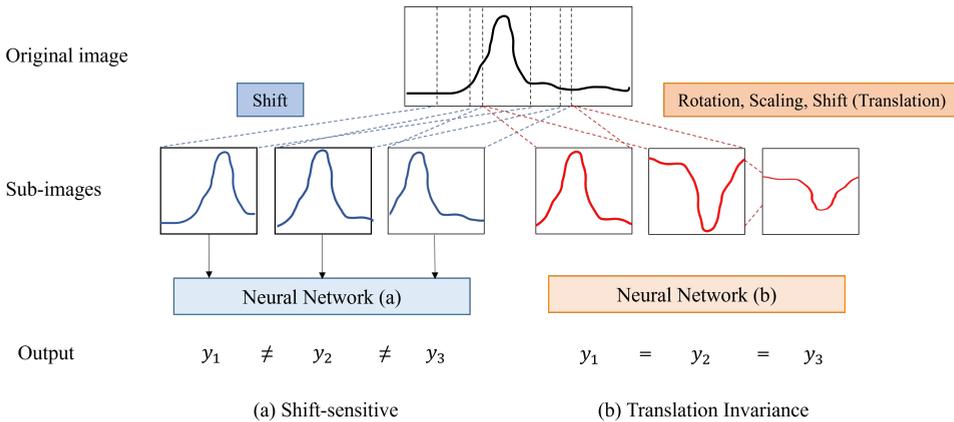


Figure 3.9: Example of Shift-sensitive Network and Translation-invariant Network

Proper Labeling: Unlike *dense prediction* where each pixel in an image is labeled with a certain class or value in computer vision [52], in the available data here, only the CC markers (ECG R-peak arrival time) are provided; hence, the segmentation of different phases (diastole, systole, etc.) of the ultrasound image in time is not very likely. Since only a flag event is defined for segmentation, instead of an area or continuous for a certain duration, the distribution of the classes will be extremely unbalanced if the CC markers and non CC-marked moments are treated as two parallel classes. Such a binary classification problem turns out to fail as the CC markers are regarded as noise and outliers by the neural network due to the unbalance in the dataset. To handle this problem, a multi-class classi-

fication or a regression with a well-designed, evenly distributed mapping from dataset to labels will be more likely to succeed.

3.4.2 Segmentation Network Design

Following the heuristics discussed in the previous section, a 2D convolution neural network is designed to segment the long ultrasound image into individual heart cycles, based on CC markers. The input size of the subimages is set to 500 *300 (depth *time), which can include the full range of movements of both arteries and almost in time dimension, and contains almost one heart cycle to include as many temporal features as possible, while keeping only one CC marker in the current frame at most. Unlike the small-sized, square convolution kernels (3 *3, 5 *5, etc.) applied in convention computer vision tasks, non-square, large-sized kernels are used to better integrate the information on the depth axis and compress it into a sequence similar to a time series. In the choice of specific size for the depth and the time dimension, it is decided that the size in depth dimension should cover at least half of the diameter of the lumen (central dark region) plus the width of the artery, and without padding, a few convolution operations in the depth dimension should be able to compress the ultrasound image into a sequence with only a few variables for every time sample. For the time dimension, the kernel size is expected to include the regional movement of the artery (to be able to identify whether the artery wall is going inwards or outwards), while after the convolution layers, the resulting sequence is similar in the length of the original time samples. In this way, the features in the images are extracted by the convolution layers into a time series efficiently without sequential processing of data in time and depth dimension. To make the neural network shift-sensitive, *max pooling* is not applied after each convolution layers to preserve the location property of the extracted features, hence the compression (downsampling) of the image data into the feature map is only achieved by the large kernels without padding. In this way, the feature map after each convolution layer is expected to decrease in size, and the width and height of the feature maps can be expressed as the following equations.

$$H_{out} = \lfloor \frac{H_{in} + 2 \times padding[0] - dilation[0] \times (kernel_size[0] - 1) - 1}{stride[0]} + 1 \rfloor \quad (3.17)$$

$$W_{out} = \lfloor \frac{W_{in} + 2 \times padding[1] - dilation[1] \times (kernel_size[1] - 1) - 1}{stride[1]} + 1 \rfloor \quad (3.18)$$

where input would be a batch of (multi-channel) image samples (matrices), denoted as $(N, C_{in}, H_{in}, W_{in})$, and output would be $(N, C_{out}, H_{out}, W_{out})$. H is the height of the image, W is the width, they can be both considered the two dimensions of a matrix.

To properly design the label for the CC markers, it is decided that a distance metric is a good representation related to the probability of the occurrence of the CC marker. Starting from the first frame of one window, the label is set to be the distance to the nearest ECG marker. The labeling scheme can be shown in figure 3.10. To have more continuous and a large range, the label is chosen to be $l1$ distance, without normalization to 0-1 range because of the max distance is not constant in different positions on time dimension. Apart from the regression of $l1$ distance, a classification scheme of quantized distance of

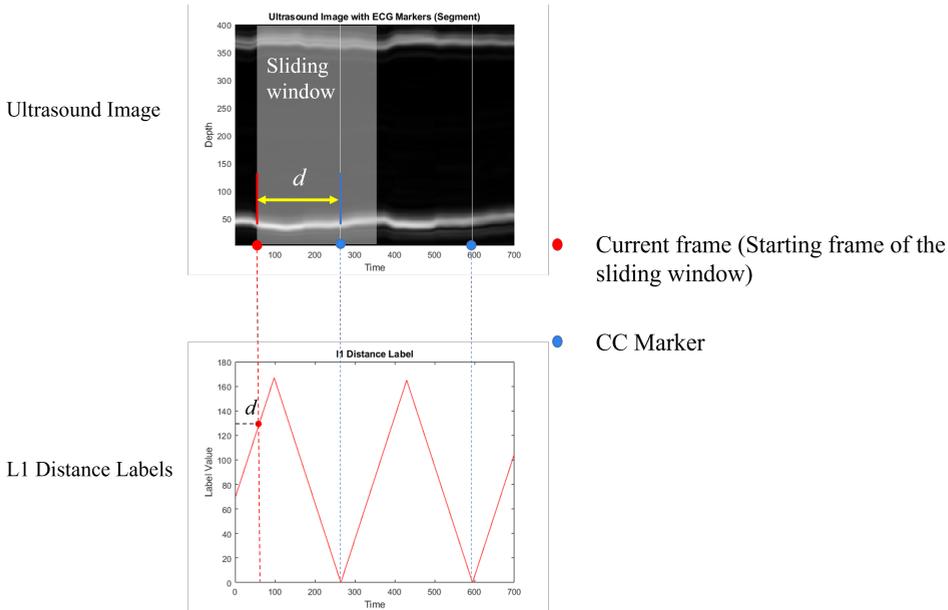


Figure 3.10: An Illustration of the l_1 distance Labeling

the starting frame and the succeeding ECG marker is also designed. It turns out that the regression scheme has a better accuracy since the quantization is downsampling the time samples and quantized distance in classification only relates ECG marker and current frame in one direction.

After the convolution layers, fully-connected layers are applied to readout the output of the feature maps like most of the CNNs. Depending on the label, the number of output neurons on the last layer can be altered (1 output neuron for regression, number of classes for classification). The complete structure of this neural network and the evolution of the size of the feature map is illustrated in the figure 3.11. This neural network consists of 5 convolution layers and 3 fully-connected layers. The hyperparameters and training settings are listed in Table 3.3.

3.4.3 Post-processing for Time Domain Segmentation

The segmentation neural network is a one-to-one relation for input and output: for every subimage located on a time sample, an inferred distance is outputted. Inputting a long ultrasound data sequence will result in a sequence that infers the distance to the nearest CC marker from the current frame. To decide where the CC markers are, it is important to include the neighboring distance information of the current frame, as there is no guarantee that an ECG marker always results in an '0' in the output of the neural network. Hence, a local minimum detector is applied to the output sequence to find the ECG markers. In order to obtain the desired CC markers, the regressed sequence is firstly flipped down to convert the local minimum into local maximum; then a peak-finding algorithm specifying minimal peak distance is applied to find the exact locations of CC markers.

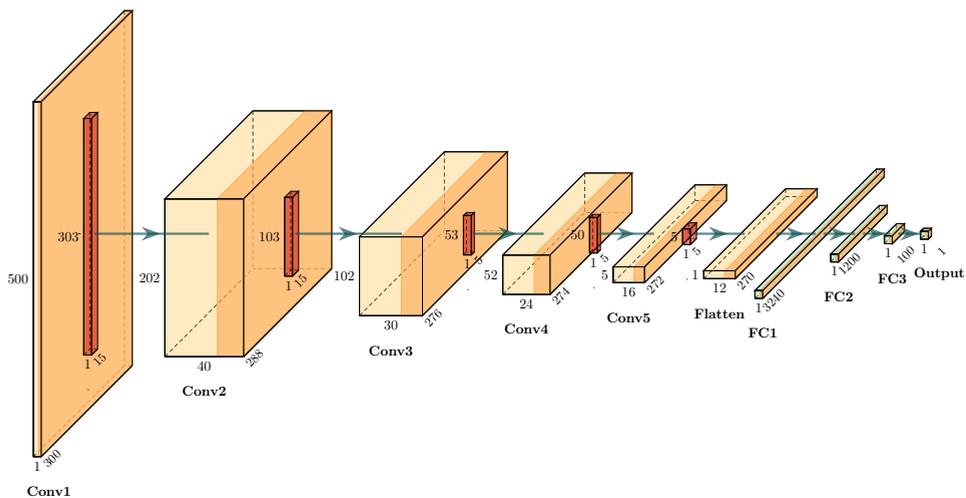


Figure 3.11: Convolution Neural Network for Heartbeat Segmentation

Post-processing is implemented in *Matlab* and the peak-finding is performed by using close-source function *findpeaks*. The result of the regressed sequence outputted by the neural network and the result after post-processing are given in the chapter 4.

Setting	Symbol	Type or Value
Network Structure	-	2D Convolution, Feed forward
Activation Function	g	ReLU
Loss Function	L	MSE
Initialization	w	Xavier Uniform
Optimizer	-	Adam
Learning rate	α	$10^{-7} - 10^{-4}$
Regularization	R	l2-norm
Weight decay	λ	10^{-4}
Training epochs	N	>20
Number of inputs	x	500
Number of outputs	y	1

Table 3.3: Hyperparameters and Settings of the Segmentation Network

3.5 Part 3: Diameter Tracking

In the tracking part, the neural network is designed as a regression task over the diameters and the wall positions. From the previous part figure 2.5, the artery diameter(distension) tracking is done via frame-to-frame phase shift estimation and wall position detection in every heart cycle, segmented by the ECG R-peak. Eventually, the diameter (distension) is inferred for every time instance; the variation in time reveals the pattern of movement of the artery within a heart cycle (illustrated in figure 3.12).

Given the current availability of the data(raw RF ultrasound data, analytic signal (after Hilbert Transform) and envelope of the analytic signal), the artery diameter or artery wall positions should be extracted by one or combination of these signals. Since the tracking task only outputs a target parameter after it processes the current time frame, the input for the neural network should be organized in time domain (single sample or window of samples) with complete signal (in depth dimension) at every time sample. Based on the fact that the envelope of signal can provide a very obvious peak-low intensity-peak structure to indicate the locations of artery walls and lumen, the detection of artery walls is hence taking the envelope data (selected region by the ROI detection network. The phase of the signal only implies subsample movement between consecutive frames, it does not have the position information of the artery wall, and hence the use of the phase information should not be prior to the envelope (magnitude of the analytic signal). To find the method that is capable of regressing the wall position, a few experiments are carried out to see which kind of data can yield the correct and desired result. It turns out that with similar neural network structure and complexity, the raw data input fails to converge in the learning and cannot have a reasonable output. However, the envelope (acquired by Hilbert Transform) can converge fast and output better waveforms as the loss decreases. As a result, the envelope of the data (magnitude of the analytic signal) is used as the main input of the tracking neural network.

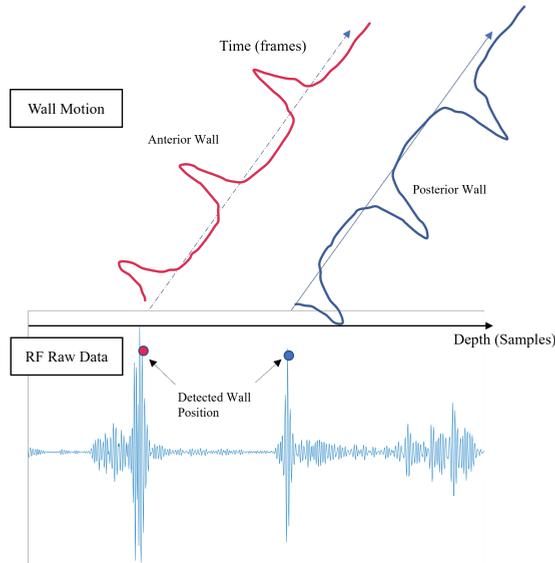


Figure 3.12: Illustration of the Motion of Artery Wall Movement

3.5.1 Problem Formulation

In order to detect the position of the artery wall in a very long time sequence, the tracking problem can be seen as either a time series or a one-to-one detection problem. The different perspectives result in different approaches: neural network for time series (Recurrent Neural Network, RNN) or a time-invariant problem (CNN or ANN structure). Similar to the DSP approach where the time relevance of neighboring samples is the phase shift, it is expected that the neural network can somehow also establish a short time correlation between samples. While the RNN has the recurrent connection that can feed the output of the network (hidden states in RNN) back to the input, the current output will relate to the past samples as far as the back propagation allows (backpropagation through time, BPTT). Such a mechanism enables backpropagation to compute the influence of past input on current output, making current output correlates with past input with long enough intervals. For such reasons, RNNs are very useful when handling the task with contextual data, e.g., speech recognition [53], translation, etc. The time-invariant problem, however, treats all the samples as individual and perform detection and regression over each single sample, while the correlation in time is not explicitly implemented, it is expected that with a well trained model, the gradual and slight difference lied in the progression of the ultrasound signal will result in similarly slight and gradual changes in the output of the neural network. On the other hand, this time-invariant ANN (CNN) structure does not need to trace back the gradient in time, which is more efficient in computation. In theory, both structures can be suitable for this tracking task; the choice is made and explained in the following section 3.5.2.

Based on the heuristics discussed above, the neural network designed for tracking should have the following features:

Low Complexity: As mentioned in the DSP approach figure 2.5, the backbone of the tracking is the complex cross-correlation between the submatrices of the analytic signal. Such a computation is costly in terms of computation sources and time. As in the neural network, without correlation operation, it is expected that the neural network should regress over each time sample (or window) without very deep and complicated neural network structure. On the other hand, since the ROI for the artery can be detected in the previous section, the selection on the ROI can further reduce the input size for the tracking task, and also remove the part containing noise, interference, etc..

Smooth Output Waveform: The smoothness of the output waveform for diameters, distortions is significant for the further pulse arrival time (PAT) estimate, which employs derivatives to detect peaks or maximum. Also, the physic movement of the artery is analytic, continuous as time goes, requiring the smoothness in the wall motions, diameters without oscillations in short time intervals. Since in the DSP approach the motion of the artery walls is detected via phase shift, the smoothness is guaranteed inside one heart cycle. Given that the neural network inputs the samples without correlation or any differential operations between the samples and its discrete nature, to preserve the smoothness in the output, the labels should not be altered with different scales or downsampled, and operations should be deployed that average the latent representation for each time sample or the output of the neural network.

3.5.2 Tracking Network Design

Based on the heuristics mentioned above section 3.5.1, the tracking neural network should be compact, efficient and have averaging mechanisms either in the layers of the network or the output of the network. In order to find out the better model, a few experiments with different parameter settings are carried out to investigate which neural network structure performs better (RNN or ANN (CNN)), where the averaging should take place (inside the network or after the output). As the result shows, the CNN structure that employs a few 1D convolution layers, with individual sample input and averaging after the output has a better performance over other settings. It is observed that the CNN structure with only one sample in time as input can converge and is relatively more accurate compared to other schemes. The output of the RNN structure network has obvious discontinuities every time the back propagation is truncated, and the windowed input over time, or averaging inside the network does not yield a better result than one time sample input. This results in the tracking network being designed as a 1D convolution neural network, with a few convolution layers followed by fully connected layers as read-out layers, which means that such a tracking network works as a position detector for the artery walls without explicitly correlating the current sample with the past ones.

The neural network structure for tracking the anterior wall is illustrated as figure 3.13. Similar to the ROI detection section 3.3, the tracking neural network takes the envelope of the data processed by the ROI detection every time sample, passes through a few 1D convolution layers and fully connected layers to output the result. The hyperparameters and the settings to train are listed in the table 3.4.

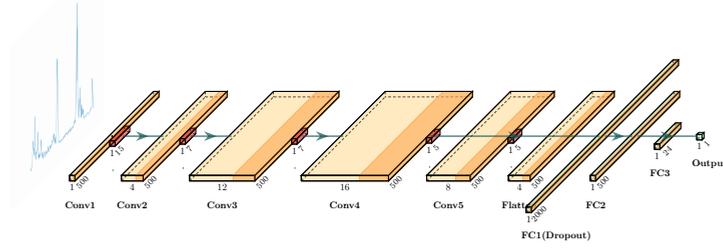


Figure 3.13: Artery Wall Diameter Tracking Neural Network Structure

3.5.3 Post-processing

Based on the design of the tracking neural network, the diameter of the artery is directly regressed by positional information of the artery walls, not the phase shift between consecutive frames. Due to the discrete nature of neural networks, this position-detection based approach does not have a mechanism that can guarantee the smoothness in the output sequence. Since in the later stage the biological features (PAT) need to be extracted via the derivatives of the diameter waveform, a smoothing process should be applied after the diameter sequence is produced by the neural network. Here a Savitzky-Golay filter is applied to smooth the data. Based on least-squares smoothing of signals, the Savitzky-Golay filter applies convolution coefficients on data sub-sets consecutively. Proposed in [54], this filter can increase the precision of the data without distorting signal tendency, and is widely used in many numerical tasks. By altering the order and frame length, this filter can give different smoothing effect of the processed data. Generally, the Savitzky-Golay filter can be written as

$$Y_j = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i y_{i+1}, \quad \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2} \quad (3.19)$$

where y_j is the observed value in the dataset, m is the order of the polynomial and i specifies which data point in the frame. The exact effect of different choices for frame length and order on the data is beyond the scope of this thesis, while the details of the derivation and application are discussed in [55]. The results of the diameter output sequence from the neural network and the corresponding smoothed data are given in chapter 4.

Setting	Symbol	Type or Value
Network Structure	-	1D Convolution, Feed forward
Activation Function	g	ReLU
Loss Function	L	HuberLoss
Initialization	w	Xavier Uniform
Optimizer	-	Adam
Learning rate	α	$10^{-8} - 10^{-4}$
Regularization	R	<i>dropout</i> , l2-norm
Dropout probability	$P_{Dropout}$	0.08
Normalization	-	<i>Batchnorm1d</i>
Weight decay	λ	$5 * 10^{-5}$
Training epochs	N	>50
Number of inputs	x	500
Number of outputs	y	1

Table 3.4: Hyperparameters and Settings of the Tracking Network

3.6 Cardiac Parameter Estimation: Pulse Arrival Time

As the other major part to estimate the cardiac parameter, after acquiring the CC markers (stands for the delayed ECG R-peak moments) and artery diameter (motion), the pulse arrival time (PAT) can be estimated by the time difference between the ECG R-peak collected at the heart and the reference point in the movement of the periphery artery (e.g., carotid artery, femoral artery, etc.) [5]. At the periphery artery, there are a few different reference points that can be use to compute PAT, e.g. foot, peak or the slope on the diameter of the artery [56], shown in figure 3.14. In post-processing to estimate the PAT, the maximum

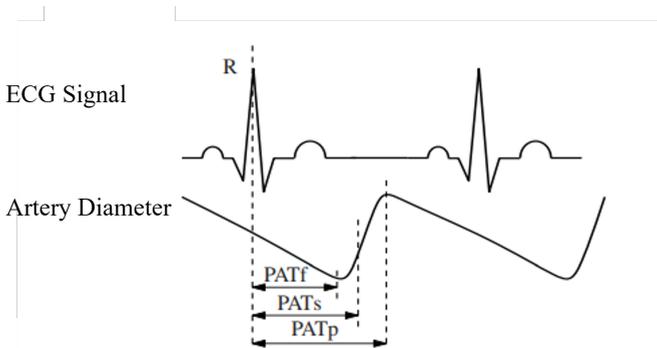


Figure 3.14: An Illustration Reference Points on the Artery Diameter for PAT Computation, Partly Adapted from [5]

slope in the artery diameter is chosen as the reference point to compute the PAT. Due to the discrete nature of the output diameter, the foot and the peak will be relatively hard to identify, while the max slope can be easily extracted in the first derivative by finding

the first maximum after the CC markers. The computation of the PAT can be illustrated in the figure 3.15. When the PAT is available, the pulse wave velocity (PWV) can then be

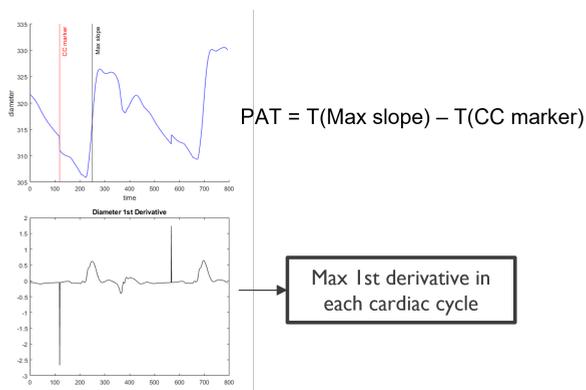


Figure 3.15: An Illustration of PAT Computation

computed to aid in the assessment of the health situation of the subjects, after acquiring the biological distance from the artery to the heart.

4

Results & Evaluation

In this chapter, the results from the machine learning (ML) pipeline in each part (ROI detection, segmentation, and tracking) are presented, including intermediate results, and subsequently measured in different metrics for evaluation. To prove the effectiveness of the ML approach pipeline, these results will be given both in visualization (with original ultrasound images) and tables containing the numbers of errors, in qualitative and quantitative measures.

4.0.1 Validation Scheme

To assess how the results of the machine learning model generalize to an independent dataset, validation schemes that applied different splits of train and test dataset are often used in machine learning. In this work, a non-exhaustive cross-validation scheme is used to test the generalizability of the ML pipeline 4.1. Out of the 7 available subjects, each including different interventions and sections (pace breathing, hand gripping, and resting), the validation is carried out in different subjects to evaluate the generalization of the model when given different personal characteristics in the arteries. Due to the large size of the entire dataset, the validation scheme does not test on all sessions of a subject and does not train on all the remaining datasets. If this limited training and validation scheme has a good generalization performance, it can be safely deduced that a model trained on larger size training dataset also has a good generalization ability.

Validation Scheme:							Training Data
Split 1	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Test Data
Split 2	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	
Split 3	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	
Split 4	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	
Split 5	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	
Split 6	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	
Split 7	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	

Figure 4.1: Validation Scheme

Data	MSE(Response vector)	MAE(Lumen center)
Split 1	0.01039	6.0514
Split 2	0.00821	10.1607
Split 3	0.01196	5.9619
Split 4	0.00714	4.5126
Split 5	0.01028	6.8277
Split 6	0.00862	5.3201

Table 4.1: Error in ROI Detection (response vector and lumen center)

4

4.1 Intermediate Results & Analysis

The obtained results of the ML pipeline of different parts (ROI Detection, Segmentation and Tracking, including intermediate results) are given in each subsection, followed by the analysis.

4.1.1 ROI Detection

In the ROI detection part section 3.3, a neural network consists only of convolution layers and a deterministic postprocessing to identify the lumen center given the envelope data of the raw ultrasound data. The training scheme applies a response vector as the label to let the neural network learn the peak-low intensity-peak structure through large-size kernels. The original ultrasound image, compared to the image of the response vectors (labels), and the output of the convolution layers are shown in figure 4.2.

It can be observed that the response vectors represent the region where the lumen center will occur with high change. From the output of the convolution layers, the large kernels successfully absorbed the features of the artery wall-lumen structure. The output image of response vectors shares the same moving pattern as in the label response vectors. As the lumen center in both images, the movement of the lumen center can reveal the low-frequency motion of the artery, for example, respiratory. To quantitatively assess the performance of ROI detection, the MSE is computed for the output/label response vector to directly evaluate the performance of the neural network, for the extracted lumen center, since the ROI detection only regresses a location index indicating lumen center position, the mean absolute error (MAE) is computed to assess the overall accuracy of ROI detection. The table 4.1 shows the result of the error of MSE (response vector) and MAE (mean absolute error) of the lumen centers.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - x_i| \quad (4.1)$$

where y_i is the inferred lumen center, x_i is the label lumen center. The results are given in Table 4.1. According to the table above, it is clear that ROI detection can find the lumen center with a tolerable error (compared to the typical artery diameter 300-450 [samples]) that ensures that the ROI (artery wall-lumen-artery wall) structure is still present for further time domain segmentation or diameter tracking.

Data	MAE[Sample]
Split 1	10.7258
Split 2	9.1979
Split 3	19.7692
Split 4	32.2438
Split 5	8.5000
Split 6	27.4827

Table 4.2: Error in Time Segmentation: The Detection of CC Markers

4.1.2 Time Domain Segmentation

For every inputted window of the ultrasound image, the segmentation neural network regresses a value to infer the distance from the starting frame of this window to the nearest cardiac cycle (CC) markers. Since there is no normalization for both the distance label and the output of the regression neural network, the moment of occurrence of the CC marker should be determined by looking for the local minimum of the output sequence. For illustration, the result after the neural network is demonstrated in figure 4.3.

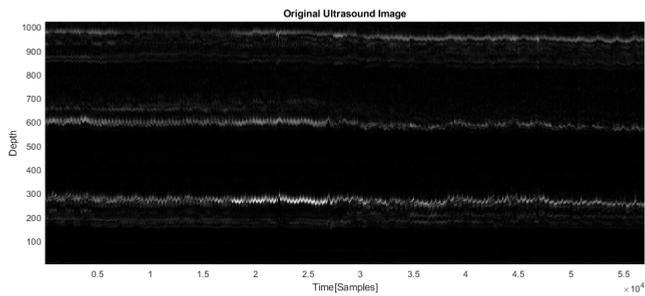
Since the output of the segmentation neural network represents how close (likely) a possible CC marker is, the local minimum in a short time interval can reveal the CC marker, as introduced in section 3.4.3. By fixing a minimum local minimum distance (number of samples of approximately 0.6-0.8 heart cycle on average), the CC markers can be derived by filtering out unwanted local minimum. To investigate the performance, the MAEs of their positions are computed, given in Table 4.2 The result shows that the post-processing (local minimum detection) can effectively find the CC markers, however, the misalignment between detected CC markers and label CC markers can cause large absolute errors. The MAE is expected to be greatly reduced after filtering out the outliers. Compared to the typical duration of a heart beat (300-500 samples), these MAEs are still safe to estimate CC markers.

4.1.3 Diameter Tracking

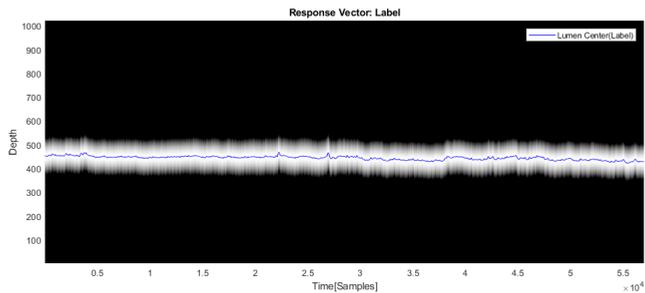
The tracking neural network tracks the evolution of the artery diameter in ultrasound data and returns an inferred sequence of the artery diameter. The direct result from the neural network is given in the figure 4.4. As explained in section 3.5.3, a Savitzky-Golay filter is applied to smooth the output sequence in order to reserve important information in the first derivative. For each split in the validation scheme, the errors of Diameter Tracking are given in the table 4.3. As it turns out that the regression on the diameter of the artery does not have very good accuracy in tracking the actual artery diameters, the important feature for further processing, the upstroke of diameter (max slope in the first derivative) can still be preserved.

Data	Huber Loss	MSE
Split 1	7.0987	82.45
Split 2	6.5784	80.24
Split 3	7.3044	92.75
Split 4	8.6217	204.57
Split 5	3.3608	17.87
Split 6	9.1812	114.72

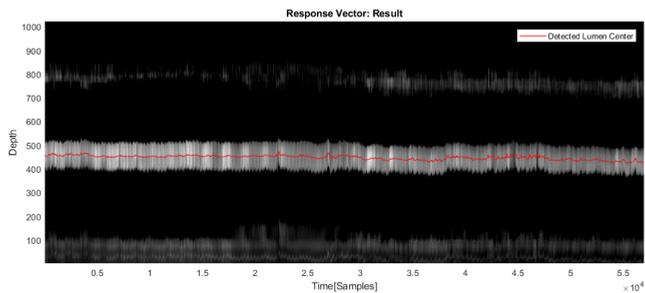
Table 4.3: Error in Diameter Tracking



(a) Original Ultrasound Image



(b) Label Response Vectors & Lumen Center



(c) Output Response Vectors & Lumen Center

Figure 4.2: Comparison of the Original Ultrasound Image, Response Vectors and Lumen Centers (Label, output). The Images Display the Intensities in Gray Scale

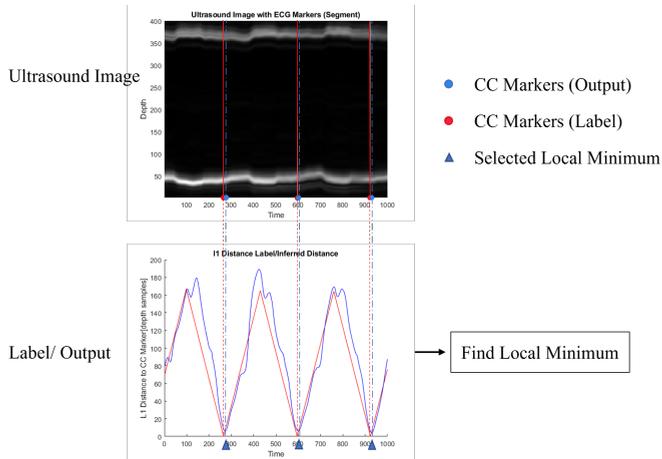


Figure 4.3: Time Domain Segmentation: Result (excerpt). In the Label/Output image, the label is plotted in red line, the regressed distance to a CC marker is plotted in blue

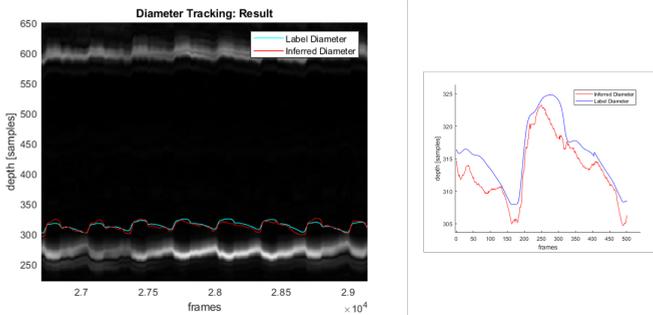


Figure 4.4: Diameter Tracking: Result (excerpt)

4.2 Numerical Results & Analysis

In this section, the errors in estimating the reference point from the first derivative and the CC markers are shown numerically, and the estimation of the pulse arrival time (PAT) from the original data (label) and from the ML processing pipeline in intra-subject and inter-subject perspective.

4.2.1 Intra-subject Results

Since the pulse arrival time (PAT) calculation has to involve two estimated values (CC markers, tracked diameter), the estimated CC marker moments and max slope moments detected from the first derivative of diameter (introduced in section 3.6) will have their own error distribution. Here, a histogram on the error distribution of the mentioned detected moments is given in figure 4.5. The histogram shows that for both CC markers

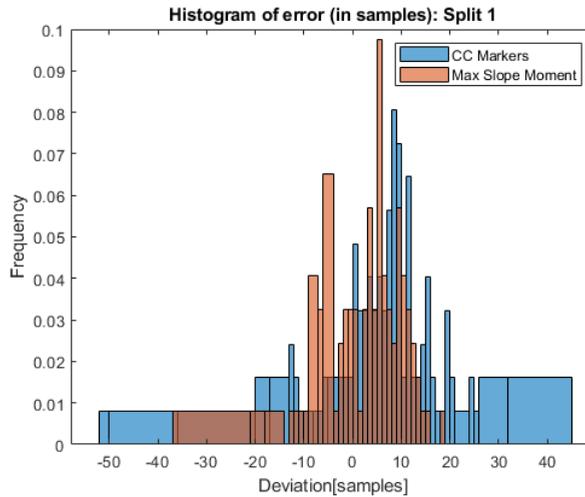


Figure 4.5: Histogram of Error from CC Markers, Regressed Diameter, Split 1

and maximum slope moments extracted from the diameter and a mean error (MAE) can be computed. The deviation of the mean error can directly demonstrate how good the estimation is.

The PATs are also estimated in every heart beat, and the reference PATs computed from the labels (derived from the DSP pipeline) are used as ground truth to evaluate the performance of the ML-based processing pipeline (figure 4.6). It can be observed that although the estimate PAT is very noisy and oscillating for every heart cycle, the average of these estimates is very close to the average PAT of the ground truth. Therefore, it can be concluded that the ML-based pipeline can better estimate the PAT given a long enough measurement (consists of multiple heart beats), while it cannot effectively estimate instantaneous PAT for each heart cycle.

To discover whether the estimated CC markers can follow the heart rate variation, the duration of each heart cycle is computed by subtracting neighboring CC markers from the

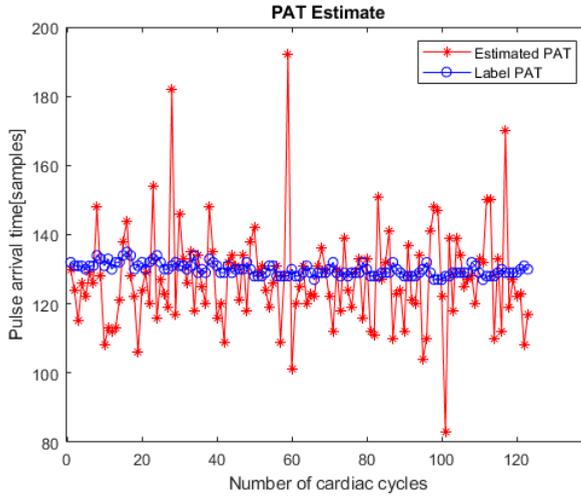


Figure 4.6: Estimated, Ground Truth PAT, Split 1 (Mean Label PAT: 129.9106, Mean Estimated PAT: 127.0579)

Data/Metric [samples]	Split 1	Split 2	Split 3	Split 4	Split 5	Split 6
Mean Label PAT	129.9106	143.5053	114.2586	120.8154	121.3401	122.3334
Mean Estimated PAT	127.0579	145.8438	107.0345	127.4250	120.3673	136.4865
Mean Absolute Error	11.3659	17.1158	14.8966	34.2914	6.5102	26.1883

Table 4.4: Estimated & Label Mean PAT

estimated one and the ground truth. Figure 4.7 shows the result of the possible correlation of the variation of heart rate. In this subject, it can be observed that the time domain segmentation in the ML processing pipeline can follow the trend of heart variation, with some noise and outliers.

With only the results of Split 1 are shown here, the rest are given in the Appendix.

4.2.2 Inter-Subject Results

To evaluate the generalizability and effectiveness of the estimation of the mean PAT of the ML processing pipeline, the mean PAT of the measurements are calculated for all subjects on the detected heartbeat (CC markers). With 6 splits of testing, their results are shown in table 4.4 and figure 4.8. The correlation between the estimated PAT and ground truth PAT can be evaluated by the correlation coefficient ρ as

$$\rho(x, y) = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{x_i - \mu_x}{\sigma_x} \right) \left(\frac{y_i - \mu_y}{\sigma_y} \right) \quad (4.2)$$

where x, y are the variables (mean estimated, ground truth PAT), μ is the mean and σ is the stand deviation of the variable. The non-diagonal element in the result correlation coefficient (matrix) represents the correlation between the two variables; for the 6 test subjects,

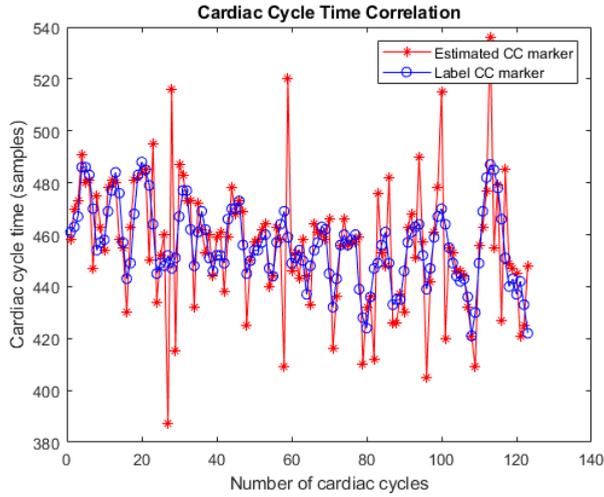


Figure 4.7: Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 1

the correlation coefficient is $\rho(x, y)_{(2,1)} = 0.8250$, indicating that there is a strong correlation between the mean estimated and the label PAT, and hence proves the effectiveness of the mean PAT estimated by the proposed ML processing pipeline.

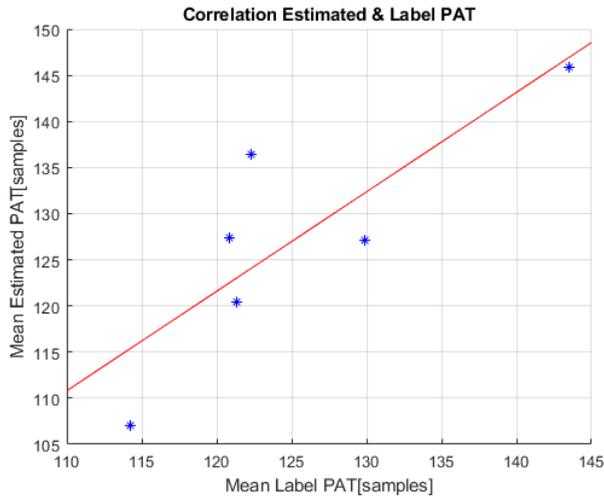


Figure 4.8: Correlation between Estimated & Ground Truth PAT

4.3 Concluding Remarks

From the results in this chapter, it can be observed that the results from the neural networks (subparts of the ML processing pipeline) generally function well with good accuracy

in their objectives (identify ROI region, locate CC markers and max-slope moments). The performance of the neural networks themselves (loss, convergence) is not evaluated directly, and the final neural networks on the ML processing pipeline may not be optimal. On the other hand, the combination of these neural networks (subparts) also may not reach the optimal point for a minimal loss in general.

Since the label PATs are calculated from the labels of the DSP pipeline from the recorded ECG signal and the computed diameters, the fidelity (correctness) in the label PAT depends on the (fixed) ECG data and estimated diameter. Since the ECG data itself is a measurement directly to the heart activity, and due to the complex cross correlation mechanism in the DSP approach, the smoothness of output diameter is well preserved and has high quality waveform, it can be safely concluded that the label PATs in different heart cycles has very good accuracy of the actual PAT.

The estimation of PAT is the direct outcome from the previous results from the neural networks in the ML pipeline. As errors accumulate in the processing of the ML pipeline (it can be observed in the Appendix), several error sources will negatively affect the result of the PAT estimation. As a result, PAT estimation is not effective for instantaneous heart beats, but it is good enough for an average PAT over a few heart cycles. The possibility of improving accuracy by optimizing the current pipeline is discussed in the chapter 5.

5

Conclusion & Discussion

5

In this chapter, the entire proposed machine learning processing pipeline is reviewed for its special characteristics and its disadvantages and limitations. The research questions are answered based on the design and the results of the ML pipeline. On the perspective of further improvements, recommendations are proposed as suggestions for further research in this topic or extension of this work in a more general picture. Finally, we draw the conclusion of the work in this thesis and summarize its significance. And future perspective is given to possibly extend this work into practice applications.

5.1 Discussion

In this section, from a more general and extensive perspective, the significance and limitations of the research in this work will be discussed, and consequently the recommendations on possible improvements are given. At last, the future perspective that could turn the current work into practice is proposed.

5.1.1 Research Questions

With the detailed explanation of proposed machine learning processing pipeline and the results, the proposed research questions can be answered

1. How machine learning algorithms can obtain the same artery motion parameters through proper training, without heavy computation and processing?

By constraining the size of neural networks and utilizing the heuristics, the ML pipeline can produce the most important cardiac parameters: artery diameter and CC marker without involving operations on complex numbers, correlation, etc. And meanwhile, the ML pipeline can utilize the advantage of parallel processing in the GPUs, potentially increasing computation efficiency

2. Can we re-intepret the existing DSP pipeline in the perspective of machine learning? What is the approach to assemble/decompose the subparts in the DSP pipeline in Machine Learning?

Yes. The existing DSP pipeline is re-intepreted into an ML processing pipeline with 3

subparts of neural networks and one part of postprocessing. The decomposition of the functionality facilitates the design and training of the ML pipeline.

3. Is ultrasound data enough for estimation of pulse arrival time (PAT) and without ECG signal?

With a good correlation in the mean PAT per subject (mean estimate and label PAT), using ultrasound alone is enough for the estimation of pulse arrival time (PAT) without ECG signal.

5.1.2 Limitation

From a more general point of view, the limitations of the current design of ML processing can be listed as follows.

- **Cascading Structure:** The pulse arrival time (PAT) is estimated by a sequence of neural networks and post-processing. As it is not derived directly and uniquely from the neural network, it is difficult to quantitatively analyze the error originated from the sub-parts, and it is not easy to carry out an overall optimization over the full process. Since the optimization/pruning of the neural networks is left for further exploration, it remains unknown how much can the models in the pipeline be improved.
- **Sequential Processing:** Unlike many popular image/video processing pipelines, the proposed ML pipeline works in a sequential way, instead of parallel processing, it is very hard for the current ML processing pipeline to estimate PAT in real time. The read-write operation, the memory use of ultrasound data is not emphasized in the design, causing redundant memory use, and computation in the intermediate variables.
- **Limited information from raw ultrasound data:** In the full processing pipeline, only the magnitude of the ultrasound signal (envelope) is fed into the neural networks, while the phase of the signal is discarded. Without the subsample movement information in the phase, the result of the diameter tracking is not guaranteed with smoothness, leading to the errors/difficulties in the later post-processing stage. The possibilities of interpreting phase information in machine learning perspective without deterministic processing are worth exploring, which certainly requires a proper processing for heterogeneous data.

The recommendation for extensive improvements of the work will focus on these limitations and other practical considerations.

5.1.3 Recommendation

Here, a few suggestions on how to further improve the performance of the current ML-based pipeline and possible research directions for related projects are listed as follows.

- **Phase-aware Neural Network:** Unlike the DSP-based approach, machine learning tries to model the problem only according to the statistics of the data, whereas the physical information and relation between variables are not fully exploited. With

the powerful ability of approximating any functions, it is certain that the diameter tracking can be improved by feeding neural networks with proper phase-encoded inputs. The question of how to design an encoding scheme and implement it in a neural network is definitely a worthy topic to explore.

- **Pruning & Quantization of the Neural Network:** The process of pruning & quantization is a trade-off between accuracy and computation efficiency. It is widely used in the hardware-related neural network designing where the chip/embedded system should be able to run the algorithms within its power consumption and memory. By restricting bits of data (data types, e.g. *float64*, *float32*, *int16*, etc.), less computation resource is needed for a single operation. The purpose of pruning is to find inactive connections between layers and delete them. The neural network is more likely to integrate into the chip with these two techniques.
- **Different Architectures, Generative Models:** As the desired parameters (pulse arrival time, max slope moments, etc.) and the intermediate result (artery diameter, CC markers) are related to each other, a generative model that is able to 'learn' the connection (joint distribution) between these parameters could be intriguing. In this way, it is expected that the pulse arrival time or other biomedical-related features can be directly acquired from the neural network, without post-processing.

5.2 Conclusion

In this thesis, a machine learning (ML)-based processing pipeline is proposed as an alternative to the existing digital signal processing (DSP) pipeline to estimate pulse arrival time (PAT) from the ultrasound data. Following the heuristics from the DSP pipeline or the observation of the data itself, the design of neural networks in the ML pipeline has a different structure than the convention neural networks in the sense of kernels, processing orders, etc.. The entire ML processing pipeline consists of 4 parts, ROI Detection, Time Domain Segmentation, Diameter tracking and post-processing. Through training, the cardiac cycle markers (CC Markers), artery diameters can be inferred by the model to further estimate the PAT. By choosing the max slope moments in the diameter waveform as the reference point, the PAT can be successfully estimated as a average quantity over multiple heart beats in one measurement. Further, the correlation of the estimated mean PAT and label mean PAT of different test subjects is studied and the conclusion is proved by a correlation coefficient of 0.8250, which implies a strong correlation between the two PATs. As a data-driven approach, the ML-based approach has been successful in solving a biomedical problem without involving explicit physical-related transformation/interpretation on the ultrasound data. The difference in the processing also leads to the discrete nature output of the ML-based approach, against to the continuous one in DSP-based pipeline. Nonetheless, without fully preserving the features in the ultrasound data, the most important moment of upstroke of artery and CC markers can still be detected for the PAT estimation. It can be expected that with more data and further optimization, the data-driven approach can greatly improve the performance in accuracy of regression and estimation.

5.3 Future Perspective

Given the functionality of the ML processing pipeline, the possibilities of the proposed ML pipeline application are discussed.

Hardware Implementation: The integration of neural networks on the integrated circuits/chips for portable devices and embedded systems has been in trends recently, with proper mapping and quantization, the circuit integrated neural networks turn out to be very powerful and effective at their specialized fields. Following this philosophy, a possible application of the proposed ML pipeline is to be integrated into an embedded system that is part of medical devices. The valuable information extracted by the ML-based pipeline is expected to help in the assessment of the heart of the test subjects, combined with other biological characteristics.

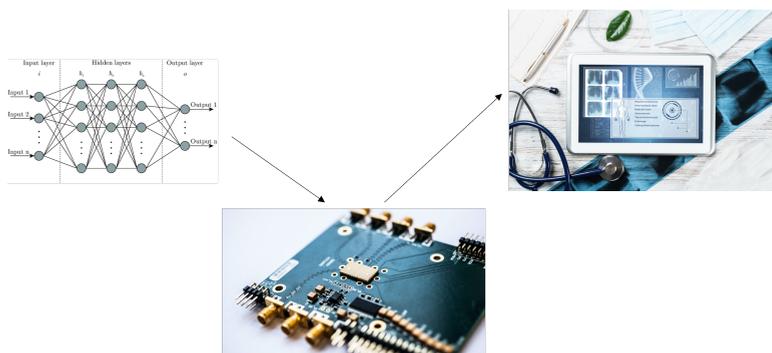


Figure 5.1: On-chip Neural Network Applications

Integration with Other Modalities The booming industry of communications and computations has greatly pushed forward the developments of the *Cloud*, Internet of Things (IoT), etc., which is making it possible for multiple sensors (modalities) to connect each other and compute meaningful results. As a trending technology, multi-modal sensors and sensing technologies have gained increased popularity on various application scenarios. For the proposed ML processing pipeline, it can be further improved and integrated into a larger pipeline that takes multiple kinds of sensor data for a more powerful and robust model.

Bibliography

URLs in this thesis have been archived on Archive.org. Their link target in digital editions refers to this timestamped version.

References

- [1] Thais Coutinho. Arterial stiffness and its clinical implications in women. *Canadian Journal of Cardiology*, 30, 07 2014.
- [2] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.
- [3] Fabian Beutel. Advancing pulse wave velocity methods - from novel ultrasonic concepts to ambulatory monitoring, 2021.
- [4] Deepak Battini. Implementing drop out regularization in neural networks, 2018.
- [5] Yong-Seok Park, Sung-Hoon Kim, Yoon Se Lee, Seung-Ho Choi, Seung-Woo Ku, and Gyu-Sam Hwang. Real-time monitoring of blood pressure using digitalized pulse arrival time calculation technology for prompt detection of sudden hypertensive episodes during laryngeal microsurgery: Retrospective observational study. *J Med Internet Res*, 22(5):e13156, May 2020.
- [6] Elizabeth Wilkins, L Wilson, Kremlin Wickramasinghe, Prachi Bhatnagar, Jose Leal, Ramon Luengo-Fernandez, R Burns, Mike Rayner, and Nick Townsend. *European Cardiovascular Disease Statistics 2017*. European Heart Network, Belgium, February 2017.
- [7] NHS. Cardiovascular disease, 2022.
- [8] Flávio D. Fuchs and Paul K. Whelton. High blood pressure and cardiovascular disease. *Hypertension*, 75(2):285–292, 2020.
- [9] Stephane Laurent, Pierre Boutouyrie, Pedro Guimarães Cunha, Patrick Lacolley, and Peter M. Nilsson. Concept of extremes in vascular aging. *Hypertension*, 74(2):218–228, 2019.
- [10] Pierre Boutouyrie and Rosa-Maria Bruno. The Clinical Significance and Application of Vascular Stiffness Measurements. *American Journal of Hypertension*, 32(1):4–11, 10 2018.

- [11] P. M. Nabeel, V. Raj Kiran, Jayaraj Joseph, V. V. Abhidev, and Mohanasankar Sivaprakasam. Local pulse wave velocity: Theory, methods, advancements, and clinical applications. *IEEE Reviews in Biomedical Engineering*, 13:74–112, 2020.
- [12] Sofie Huybrechts, Daniel Devos, Sebastian Vermeersch, Dries Mahieu, Eric Achten, Tine Backer, Patrick Segers, and Luc Bortel. Carotid to femoral pulse wave velocity: A comparison of real travelled aortic path lengths determined by mri and superficial measurements. *Journal of hypertension*, 29:1577–82, 06 2011.
- [13] Michel E. Safar, Roland Asmar, Athanase Benetos, Jacques Blacher, Pierre Boutouyrie, Patrick Lacolley, Stéphane Laurent, Gérard London, Bruno Pannier, Athanase Protogerou, Véronique Regnault, and null null. Interaction between hypertension and arterial stiffness. *Hypertension*, 72(4):796–805, 2018.
- [14] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.
- [15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [17] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014.
- [18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.
- [19] Juan Shan, S. Kaisar Alam, Brian S. Garra, Yingtao Zhang, and Tahira Ahmed. Computer-aided diagnosis for breast ultrasound using computerized bi-rads features and machine learning methods. *Ultrasound in medicine & biology*, 42 4:980–8, 2016.
- [20] Yuya Hiramatsu, Chisako Muramatsu, Hironobu Kobayashi, Takeshi Hara, and Hiroshi Fujita. Automated detection of masses on whole breast volume ultrasound scanner: false positive reduction using deep convolutional neural network. In Samuel G. Armato III and Nicholas A. Petrick, editors, *Medical Imaging 2017: Computer-Aided Diagnosis*, volume 10134, page 101342S. International Society for Optics and Photonics, SPIE, 2017.
- [21] Phillip M. Cheng and Harshawn S. Malhi. Transfer learning with convolutional neural networks for classification of abdominal ultrasound images. *J. Digit. Imaging*, 30(2):234–243, 2017.
- [22] Natalia Antropova, Benjamin Q. Huynh, and Maryellen L. Giger. A deep feature fusion methodology for breast cancer diagnosis demonstrated on three imaging modality datasets. *Medical Physics*, 44(10):5162–5171, 2017.

- [23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2014.
- [24] Nima Torbati, Ahmad Ayatollahi, and Ali Kermani. An efficient neural network based method for medical image segmentation. *Computers in biology and medicine*, 44:76–87, 01 2014.
- [25] Rosa-María Menchón-Lara and José-Luis Sancho-Gómez. Fully automatic segmentation of ultrasound common carotid artery images based on machine learning. *Neurocomputing*, 151(Part 1):161–167, 2015.
- [26] Fang Chen, Dan Wu, and Hongen Liao. Registration of ct and ultrasound images of the spine with neural network and orientation code mutual information. In Guoyan Zheng, Hongen Liao, Pierre Jannin, Philippe Cattin, and Su-Lin Lee, editors, *Medical Imaging and Augmented Reality*, pages 292–301, Cham, 2016. Springer International Publishing.
- [27] Y. Gao, M. A. Maraci, and J. A. Noble. Describing ultrasound video content using deep convolutional neural networks. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 787–790, 2016.
- [28] Hao Chen, Qi Dou, Dong Ni, Jie-Zhi Cheng, Jing Qin, Shengli Li, and Pheng-Ann Heng. Automatic fetal ultrasound standard plane detection using knowledge transferred recurrent neural networks. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 507–514, Cham, 2015. Springer International Publishing.
- [29] Laura Brattain, Brian Telfer, Manish Dhyani, Joseph Grajo, and Anthony Samir. Machine learning for medical ultrasound: status, methods, and future opportunities. *Abdominal Radiology*, 43, 04 2018.
- [30] Peter J. Brands, Arnold P.G. Hoeks, Jean Willigers, Christine Willekes, and Robert S. Reneman. An integrated system for the non-invasive assessment of vessel wall and hemodynamic properties of large arteries by means of ultrasound. *European Journal of Ultrasound*, 9(3):257–266, 1999.
- [31] LLC. WebMD. Picture of the carotid artery, 2014.
- [32] Chapter 1 - time-frequency and instantaneous frequency concepts0. In Boualem Boashash, editor, *Time-Frequency Signal Analysis and Processing (Second Edition)*, pages 31–63. Academic Press, Oxford, second edition edition, 2016.
- [33] Peter J. Brands, Arnold P.G. Hoeks, Léon A.F. Ledoux, and Robert S. Reneman. A radio frequency domain complex cross-correlation model to estimate blood flow velocity and tissue motion by means of ultrasound. *Ultrasound in Medicine Biology*, 23(6):911–920, 1997.

- [34] Jaroslaw Krejza, Michal Arkuszewski, Scott E. Kasner, John Weigele, Andrzej Ustymowicz, Robert W. Hurst, Brett L. Cucchiara, and Steven R. Messe. Carotid artery diameter in men and women and the relation to body and neck size. *Stroke*, 37(4):1103–1105, 2006.
- [35] Qinghua Huang, Fan Zhang, and Xuelong Li. Machine learning in ultrasound computer-aided diagnostic systems: A survey. *BioMed Research International*, 2018:5137904, Mar 2018.
- [36] Tong Wu, Laith R. Sultan, Jiawei Tian, Theodore W. Cary, and Chandra M. Sehgal. Machine learning for diagnostic ultrasound of triple-negative breast cancer. *Breast Cancer Research and Treatment*, 173(2):365–373, Jan 2019.
- [37] George V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.
- [38] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. *Object Recognition with Gradient-Based Learning*, pages 319–345. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.
- [39] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021.
- [40] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. <http://www.deeplearningbook.org>.
- [41] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.
- [42] Geoffrey Hinton. Neural network for machine learning, 2018.
- [43] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.
- [44] Peter. Bühlmann. *Statistics for High-Dimensional Data Methods, Theory and Applications*. Springer Series in Statistics. Springer Berlin Heidelberg, Berlin, Heidelberg, 1st ed. 2011. edition, 2011.
- [45] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.
- [46] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.
- [47] J. Sola and J. Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3):1464–1468, 1997.
- [48] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013.

- [49] Ross Girshick. Fast r-cnn, 2015.
- [50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.
- [51] Alexander C. Flint, Carol Conell, Xiushui Ren, Nader M. Banki, Sheila L. Chan, Vivek A. Rao, Ronald B. Melles, and Deepak L. Bhatt. Effect of systolic and diastolic blood pressure on cardiovascular outcomes. *New England Journal of Medicine*, 381(3):243–251, 2019. PMID: 31314968.
- [52] Tom Sercu and Vaibhava Goel. Dense prediction on sequences with time-dilated convolutions for speech recognition, 2016.
- [53] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.
- [54] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.
- [55] Ronald W. Schafer. What is a savitzky-golay filter? [lecture notes]. *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.
- [56] Federico Cattivelli and Harinath Garudadri. Noninvasive cuffless estimation of blood pressure from pulse arrival time and heart rate with adaptive calibration. pages 114–119, 06 2009.

Appendix

In the appendix, the intermediate results (from section 4.2.1) are given from the rest test splits (2, 3, 4, 5, 6). The histograms of the errors from the max slope moments and CC markers are

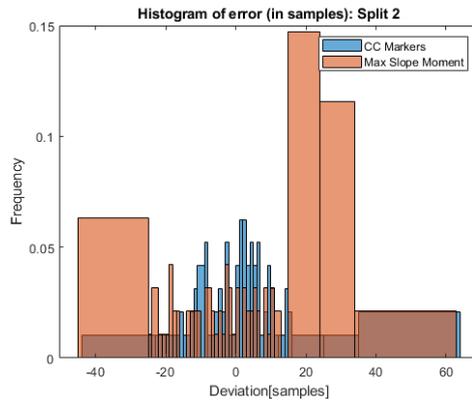


Figure 2: Histogram of Error from CC Markers, Regressed Diameter, Split 2

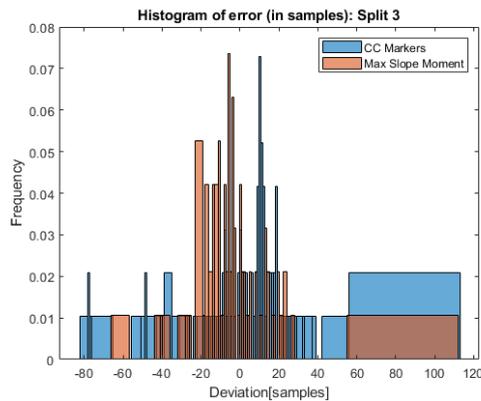


Figure 3: Histogram of Error from CC Markers, Regressed Diameter, Split 3

The visualization of estimated, ground-truth PAT from every heart beat in the ultrasound recording is given below.

The variation of the heart rates on the rest test subjects are displayed as follows

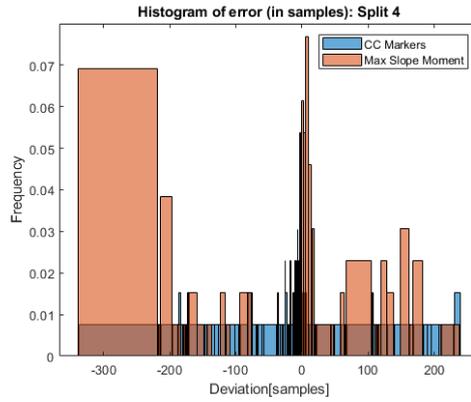


Figure 4: Histogram of Error from CC Markers, Regressed Diameter, Split 4

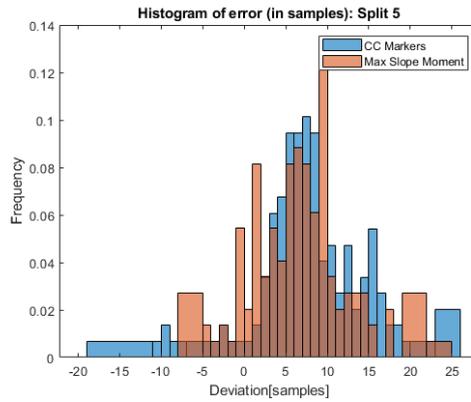


Figure 5: Histogram of Error from CC Markers, Regressed Diameter, Split 5

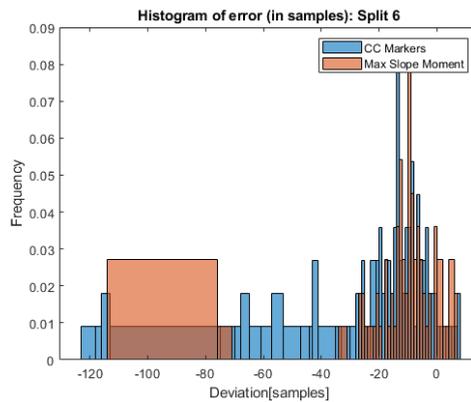


Figure 6: Histogram of Error from CC Markers, Regressed Diameter, Split 6

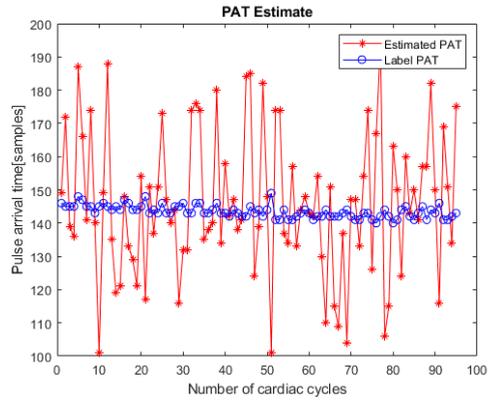


Figure 7: Estimated, Ground Truth PAT, Split 2

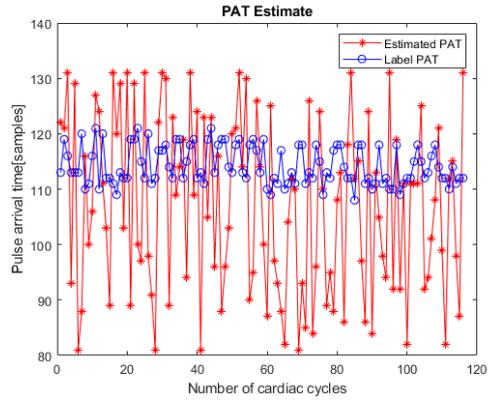


Figure 8: Estimated, Ground Truth PAT, Split 3

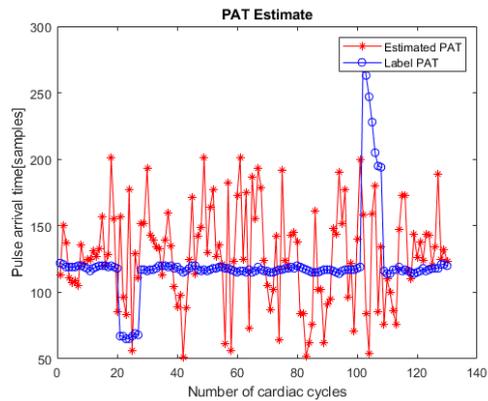


Figure 9: Estimated, Ground Truth PAT, Split 4

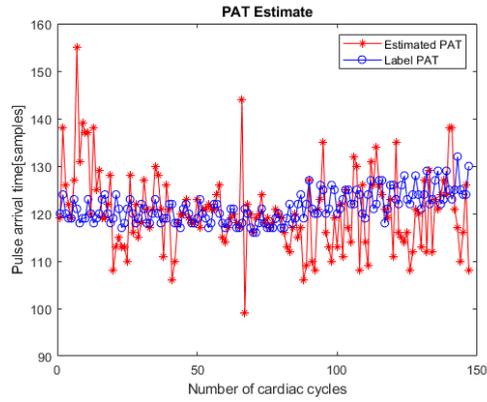


Figure 10: Estimated, Ground Truth PAT, Split 5

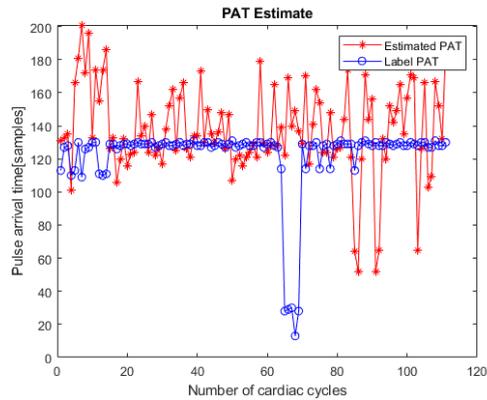


Figure 11: Estimated, Ground Truth PAT, Split 6

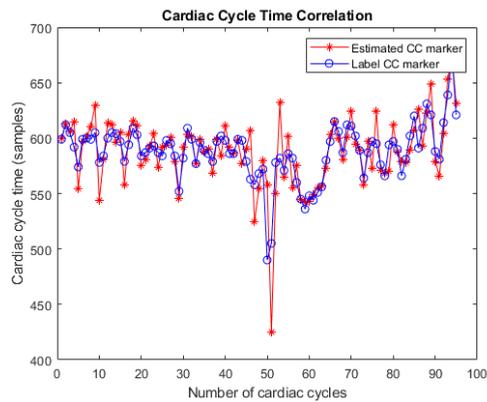


Figure 12: Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 2

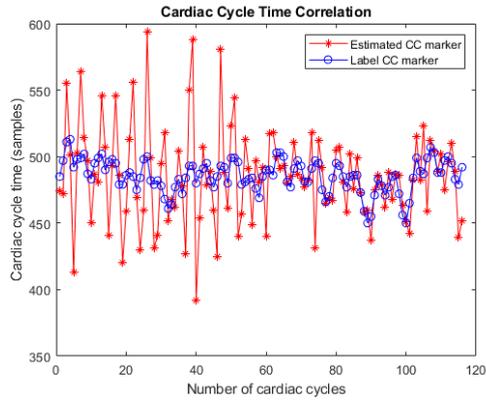


Figure 13: Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 3

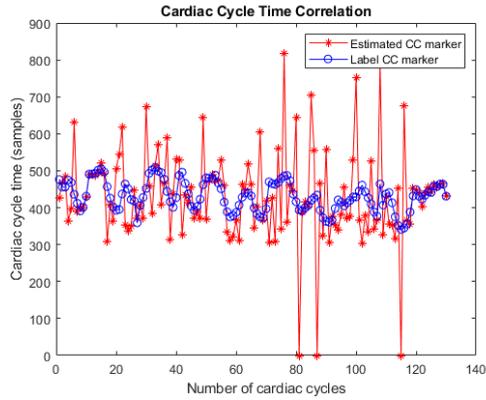


Figure 14: Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 4

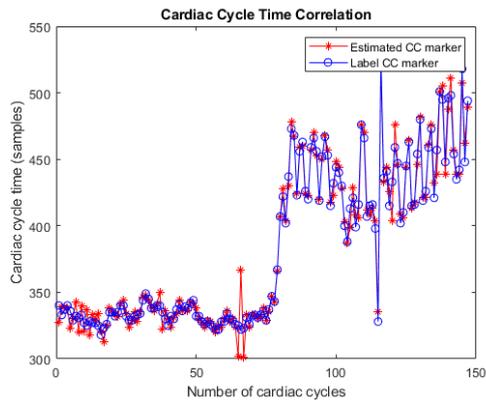


Figure 15: Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 5

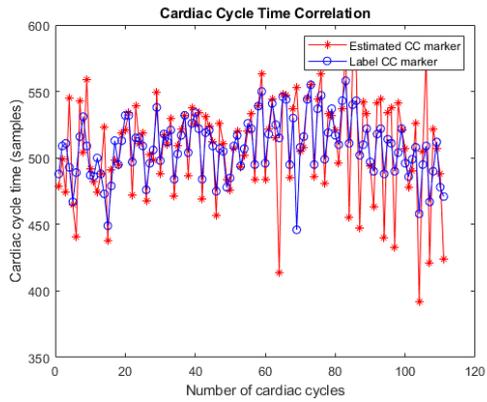


Figure 16: Duration of Heart Cycle from Detected & Ground Truth CC Markers, Split 6