University of Padova

Department of Information Engineering

Master Thesis in ICT for Internet and Multimedia Engineering

Estimation of continuous arterial blood pressure from photoplethysmography using convolutional neural networks

Supervisor
Prof. Michele Rossi
University of Padova

Master Candidate
Carlo Facchin

Student ID
1234374

Academic Year
2021-2022

Graduation date
28-11-2022

Abstract

Arterial blood pressure (ABP) monitoring is a fundamental method in preventing and detecting different cardiovascular health diseases such as hypertension, which represent one of the leading causes of death in the world. Currently, the most commonly adopted noninvasive blood pressure measurement system is the sphygmomanometer, while continuous monitoring of this parameter still often requires an invasive process.
For this reason, recent research has been highly conducted on the measurement of the ABP using signals that can be measured with a non-invasive process, such as the photoplethysmogram (PPG) and electrocardiogram (ECG).

After briefly introducing some of the state-of-the-art methodologies involved in this measurement, this work proposes a Deep Convolutional Neural Network Architecture (DCNN) that uses only a photoplethysmogram signal (PPG) as input to estimate continuous arterial blood pressure (ABP) signal non-invasively. This study also aims to measure and analyze systolic blood pressure (SBP), diastolic blood pressure (DBP), and mean arterial pressure (MAP) from the estimated ABP waveform and covers the data pre-processing techniques used.
The method was carried out on a subset of 942 patients of two publicly available datasets: MIMIC-II and MIMIC-III, resulting in a Mean Absolute Error (MAE) of 42.96 mmHg for SBP, 27.82 for DBP, 9.56 for the MAP value and a Root Mean Squared Error (RMSE) of 49.49 mmHg for SBP, 34.84 for DBP and 12.27 for the MAP value.
The results show that the proposed architecture is able to model the dependency between PPG and ABP, making it a good process to estimate continuous blood pressure using only a non-invasive photoplethysmography signal.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AAMI	Advancement of Medical Instrumentation
ABPM	Ambulatory Blood Pressure Monitoring
ABP	Arterial Blood Pressure
ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
BCE	Binary Cross Entropy
BP	Blood Pressure
BW	Baseline Wander
CNN	Convolutional Neural Network
CVD	Cardiovascular Disease
DBP	Diastolic Blood Pressure
DCNN	Deep Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DP	Diastolic Pressure
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
FCN	Fully Convolutional Network
FFNN	Feed Forward Neural Network
GD	Gradient Descent

HBPM	Home Blood Pressure Monitoring
IBI	Inter-beat Interval
IBP	Invasive Blood Pressure
ICU	Intensive Care Unit
LED	Light-Emitting-Diode
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAE	Mean Absolute Error
MAP	Mean Arterial Pressure
MIMIC	Multi-parameter Intelligent Monitoring in Intensive Care
ML	Machine Learning
MSE	Mean Square Error
MSE	Mean Squared Error
NADAM	Nesterov Accelerated Adaptive Moment Estimation
NAG	Nesterov Accelerated Gradient
NIBP	Non Invasive Blood Pressure
NN	Neural Network
OGRU	Optimized Gated Recurrent Unit
PAT	Pulse Arrival Time
PPG	Photoplethysmography
PP	Pulse Pressure
PTT	Pulse Transit Time
PWB	Pulse Wave Begin
PWD	Pulse Wave Duration

PWE	Pulse Wave End
PWV	Pulse Wave Velocity
RMSE	Root Mean Square Error
ReLU	Rectified Linear Unit
SBP	Systolic Blood Pressure
SGD	Stochastic Gradient Descent
SP	Systolic Pressure
SV	Stroke Volume
WHT	White Coat Hypertension

1

Introduction

In recent years studies have highlighted the clinical importance of continuous blood pressure monitoring [1]. Fluctuation in BP has, in fact, in the case of hypertension, a direct correlation with a variety of cardiovascular diseases (CVDs) [2].
For a long time these diseases have remained the leading causes of death globally [3] and according to the World Heart Federation, about the half of ischemic strokes are caused by hypertension [4]. Furthermore, it also increases the risk of hemorrhagic stroke, heart failure, heart attack, and chronic kidney disease.

Constant BP measuring and monitoring is thus fundamental for the general population, but particularly important for people already suffering from hypertension or related cardiovascular diseases, as such people are highly vulnerable to elevated blood pressure [5].
Despite Blood pressure (BP) measurement is the most commonly performed medical office test [6], there are not proper standard to easily monitory blood pressure. In particular, current clinical practice for the diagnosis of hypertension includes blood pressure monitoring with two different non-invasive methods: Ambulatory Monitoring (ABPM) and Home Monitoring (HBPM) [7]. Cuff-based devices are the standard for non-invasive measurements and are commonly recommended by physicians. These devices offer the highest measurement accuracy, however, they require a relatively strict measuring protocol in order to ensure the measured values are correct. The measuring procedure can be tedious and requires dedicated time and effort, also, physical activity typically does not allow for simultaneous measuring of BP

with a cuff [8]. On the other hand, is possible to have also an invasive continuous BP measurement technique, which is also known as direct BP measurement or invasive ABP. A catheter is inserted into an artery to perform real-time blood pressure monitoring. It can complete BP estimation in every cardiac cycle and monitor BP changes more precisely. Therefore, it is recognized internationally as the gold standard of BP monitoring methods [9]. But this approach is used only for critical patients and it has the risk of a range of complications, including infection [10].
Furthermore, there are cases in which hypertension does not manifest clearly to the patient: it is the case of white coat hypertension (WHT). It is a condition where a patient's blood pressure is higher when taken in a medical setting, while it is normal during daily activities. It is believed that the phenomenon is due to anxiety experienced during a clinic visit [11].

In recent years, since biological parameters such as blood volume, flow and pulse wave are easily collectable in patients peripheries with non-invasive techniques and and hold a non-linear correlation with blood pressure, research has driven into the use of Machine Learning (ML) to solve this problem. In particular through a Neural Network framework, is possible to extract the optimal features required to monitor SBP and DBP and developing algorithm able to produce values that meet system validation requirements defined by the Association for the Advancement of Medical Instrumentation (AAMI) [12].

In the next part of this introductory chapter will be discussed the related works in this field of research.
In the second chapter will be a brief overview of the biomedical background, will be discussed the concepts of arterial blood pressure and plethysmography and their relevant proprieties.
In the third chapter are introduced the basic deep learning tools and the complex deep learning framework used in this type of signal regression.
The fourth chapter describes the methodology used, explaining the data acquisition pre-processing and augmentation, as well as the deep learning architecture used and the experimental setup.
The last two chapters describe the experimental results, and the conclusion of this work, as well as final considerations and possible future developments.

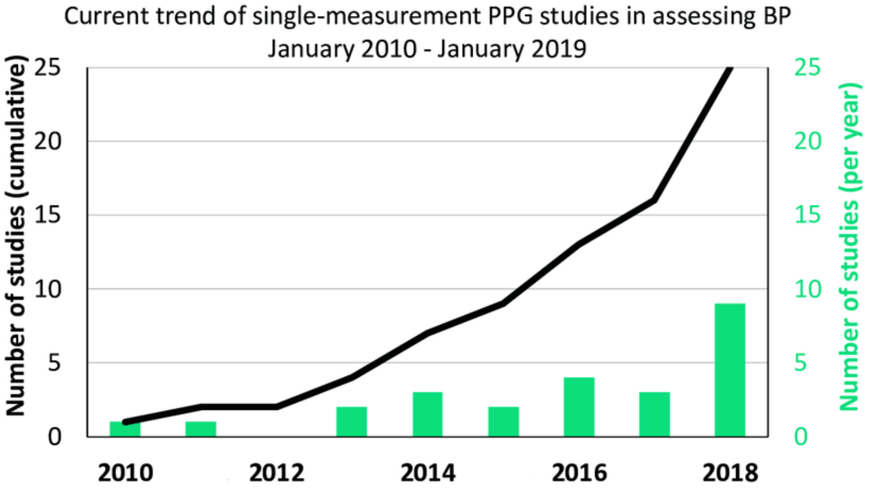

Figure 1.1: Number of PubMed publications that utilized single-measurement PPG to estimate BP (January 2010 - January 2019).

1.0.1 RELATED WORKS

With the advent of digital sensors, signal-processing and artificial intelligence models help gather important human vital signs using wearable sensors [13] [14].
In particular photoplethysmography (PPG) reveled being a good signal, demonstrating a clear correlation with the ABP signal, as well as having a versatile and low-cost technology [15] [16].

As represented in Figure 1.1, the number of studies relative to single-measurement PPG for BP estimation has increased exponentially over the last decade.

In the beginning research were focused towards more indirect approaches using the combination of two signals PPG and ECG: [17] [18] for example, showed a strong negative correlation between signals such as BP and pulse transit time (PTT), but also pulse wave velocity (PWV) and pulse arrival time (PAT) [19].

In 2013 [20] showed that artificial neural networks (ANN) can perform better than standard linear regression. The study used a feed forward ANN with two hidden layers, with 35 neurons on the first layer and 20 neurons on the second one.
The signals analyzed were taken from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) database [21], with a quite small sample of size 15000 pulsations ($\sim$ 4 hours

of recording), resulting in 3.80±3.46 mmHg for systolic and 2.21±2.09 mmHg for diastolic pressure, meeting the American National Standards of the Association for the Advancement of Medical Instrumentation (AAMI), where the maximal accepted errors is 5±8 mmHg.
[22] used a complex recurrent network on 22 features extracted from the PPG and ECG signals, root-mean-square error (RMSE) is used as metric and achieved good performances: RMSE of 3.63 on SBP and 1.48 on DBP.
It is important to note though, that models performances are very dependent on the dataset choice and on sample selection. In particular, having a data subset with a low variation of values, for example in BP, could influence the performances of the network, resulting in a smaller error if compared with a model utilizing a dataset with a wider range of BP values.
Reviewing the difficulties of using a combination of signals, and as PPG is the most convenient signal to be used in wearable devices, contrary to the ECG data acquisition that results very complex and expensive, some researchers are trying to measure BP using only the PPG signal.
Already in 2003 [23] explored the possibility to measure BP using only PPG signals, with the requirement of calibration, involving three sessions: rest, step-climbing exercise, and recovery from exercise. This publication was one of the first study based only on PPG signals, proving possible to compute BP values from photoplethysmography.

In recent years with the the advancements in deep neural techniques new approaches were evaluated, one of the very first studies based on complex deep learning frameworks is [24], that used the PPG alongside its first and second derivative as inputs into a novel spectro-temporal deep neural network with residual connections. This study analyzed the MIMIC III database, resulting in over 700 h of signals after preprocessing, and achieved mean absolute errors of 9.43 mmHg for SBP and 6.88 mmHg for DBP. Despite the fact that this ResNet-GRU-based model resulted computationally expensive and with a slow convergence, it pointed that personalization of models is important and substantially improves the results, as well as showing the potential of deep convolutional neural network (DCNN) for this type of task.
Similarly, another research work [25], modelled an optimized gated recurrent unit (OGRU) neural network, a variant of the Long short-term memory (LSTM) neural network. Their method still does not satisfy the Association for the Advancement of Medical Instrumentation (AAMI) standard error range.
Another relevant result was obtained recently in [26], that has managed to model a lighter and faster deep convolutional U-Net, using PPG only, without computing the signal derivatives.

The model was evaluated on a subset of 100 subjects from MIMIC and MIMIC-III dataset and resulted in a mean absolute error of 3.68 ± 4.42 mmHg for SBP, 1.97 ± 2.92 mmHg for DBP, satisfying the AAMI requirements.

2

Biomedical background

The biomedical background of this work is related to the cardiovascular system, more specifically to the arterial blood pressure (ABP), his monitoring and the implications that hypertension have on the human health. Along with the ABP and its signal, is relevant the understanding of another body functional measurement: the plethysmography.
Will therefore follow a brief introduction of these concepts.

2.1 CARDIOVASCULAR SYSTEM

The cardiovascular system is the system responsible for delivering blood to different parts of the body. It consists of the following organs and tissues [27]:

- The **heart** is the key organ of the cardiovascular system, as it enables the continuous blood flow.
  It is composed of a particular muscle cells tissue, called myocardium, and a specialized conduction system. The combination of these two elements produce a physiological pump for the blood inside the body.
  The human heart is composed of four chambers, two atria and two ventricles: while the atria receive the blood, the ventricles pump it by mean of a strong myocardium contraction

- A closed system of **blood vessels**: these vessels include:

  - **Arteries**: vessels that carry blood away from the heart.

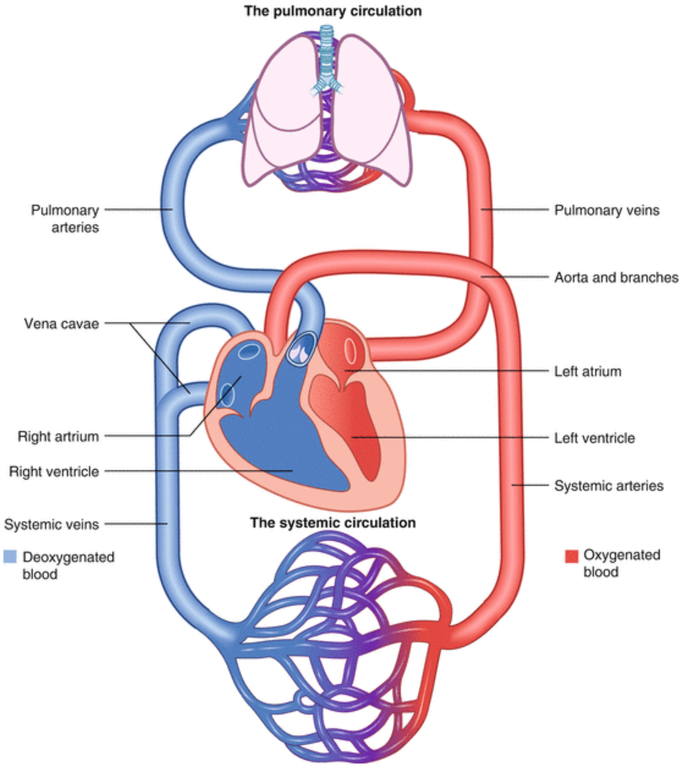

Figure 2.1: Cardiovascular system.

- **Veins**: vessels that bring blood back to the heart.
- **Capillaries**: tiny vessels that branch off from arteries to deliver blood to all body tissues.

There are two blood circulatory systems in the body. The first is the **systemic circulatory system**. This is the main blood circulatory system that transports blood to the organs, tissues, and cells throughout the body.
The second is the **pulmonary circulatory system**. This circulatory system moves blood between the heart and lungs. It is where oxygen enters the blood and carbon dioxide leaves the blood. Fig.(2.1)

2.2 ARTERIAL BLOOD PRESSURE (ABP)

In general, systemic arterial pressure or simply blood pressure (BP), refers to the pressure measured within large arteries in the systemic circulation, and among the respiratory rate, heart rate, oxygen saturation, and body temperature, reflects the status of the circulatory system.

Arterial pressure levels are determined by many factors. These include age, gender, body weight, cardiac output, peripheral vascular resistance, volume and viscosity of circulating blood, as well as agents like mental stress and hormonal levels.
It can also be influenced by many behaviors like eating, drinking, physical conditioning and drug use [28][29][30].

The arterial blood pressure derives from the pumping action of the left ventricle of the heart; therefore, the level of arterial pressure at any point in the arterial vascular compartment reflects functioning of the left ventricle. During each contraction of the left ventricle (termed *systole*), the highest systemic pressure generated within the arteries is termed the **systolic pressure (SBP)**.
When the left ventricle stops contracting, the heart valve controlling outflow from the left ventricle into the aorta closes and the left ventricle relaxes and refills (between beats). This phase of the heart is termed diastole. During diastole the arterial pressure drops as the arterial blood rapidly flows out of the arterial compartment into the capillaries. The lowest arterial pressure during this rest phase of the left ventricle is termed the **diastolic pressure (DBP)** [31].

2.2.1 ABP WAVEFORM

The arterial waveform is produced through the dynamic interactions between the volume of blood ejected by the heart during each beat, also called stroke volume (SV), the speed with which this volume is ejected by the heart, the ability of the vascular tree to distend and accommodate this ejected volume and finally the rate at which the ejected volume of blood is able to flow away from the central arterial compartment into the peripheral tissues.
The waveform also depends on where it is recorded, narrower arteries (usual in the periphery of the body) are less compliant, therefore, the pressure here has different shapes, in particular the anacrotic limb (the ascending part of the wave) is steeper and the SBP is generally higher.

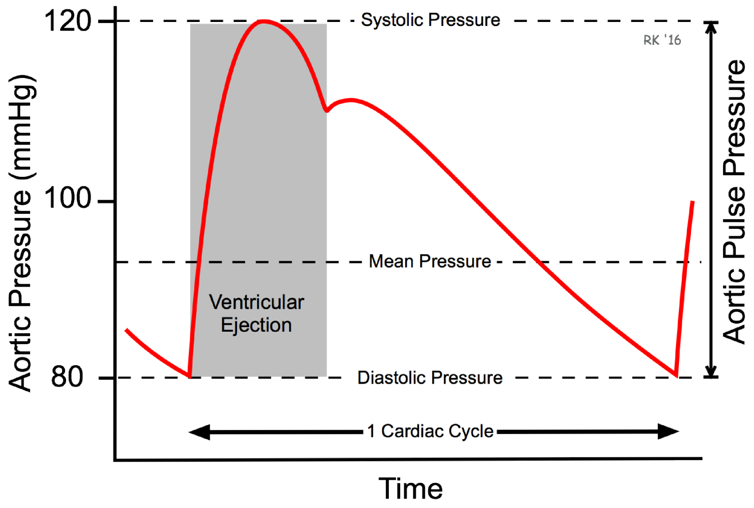

Figure 2.2: Aortic pressure curve during 1 cardiac cycle time-period

When evaluating an arterial waveform it is important to recognize two very broad components of this wave form: a steady component which is reflected by the **mean arterial pressure (MAP)**; and a pulsatile component or pulse pressure (PP) which oscillates around this mean value. Fig.(2.2) [32]. The higher and the lower peaks of the oscillation around the waveform mean value are the systolic blood pressure (SBP) and diastolic blood pressure (DBP) pressure, which are the maximum and minimum pressures, respectively (equation 2.1).

Usually, the normal range, at rest, is within 100-130 millimeters of mercury (mmHg) for the systolic one and 60-80 mmHg for the diastolic one.

$$PP = SBP - DBP \tag{2.1}$$

While MAP can only be measured directly by invasive monitoring, it can be estimated from the systolic and diastolic pressures. Equation 2.2 [33]

$$MAP = DBP + 1/3(SBP - DBP) \tag{2.2}$$

Mean arterial pressure is a useful concept because it can be used to calculate overall blood flow, and thus delivery of nutrients to the various organs. Infact, it is a good indicator of perfu-

sion pressure (ΔP). The ideal blood pressure irrorate all of the various organ systems without causing damage. Any organ not adequately perfused will suffer ischemic damage and will be unable to perform adequately [34] [35].
For this reason another way to find the MAP is to use the Systemic Vascular Resistance (R) and the total flow through the vasculature (Q), which is represented mathematically by the formula 2.3:

$$R = \Delta P/Q \tag{2.3}$$

2.2.2 HYPERTENSION

SP and DP values are used to understand if the pressure status is within specific healthy ranges. In fact, a too low SP means that the peripheral body regions are not perfused enough with nutrients and oxygen, while a too high SP is a risk for the vessels and organs integrity.

Hypertension, also called high blood pressure, is a long term pathology where the arterial blood pressure is persistently elevated. Hypertension is called a ”silent killer” due to the fact that most people with hypertension are unaware of the problem, because it may have no warning signs or symptoms. For this reason, it is essential that blood pressure is measured regularly. Long-term high blood pressure is a major risk factor for stroke, coronary artery disease, heart failure, atrial fibrillation, peripheral arterial disease, vision loss, chronic kidney disease, and dementia [36] [37] [38].

The prevalence of hypertension varies across regions and country income groups. The WHO African Region has the highest prevalence of hypertension (27%) while the WHO Region of the Americas has the lowest prevalence of hypertension (18%) [39].
The number of adults with hypertension increased from 594 million in 1975 to 1.13 billion in 2015, with the increase seen largely in low- and middle-income countries. This increase is due mainly to a rise in hypertension risk factors in those populations [39].

Figure 2.3 represented a simple and most commune used classification of hypertension by SBP and DBP values.

The American College of Cardiology and the American Heart Association divide blood pressure into four general categories:

- **Normal blood pressure**. Blood pressure is between 90-119 mmHg for the SBP and

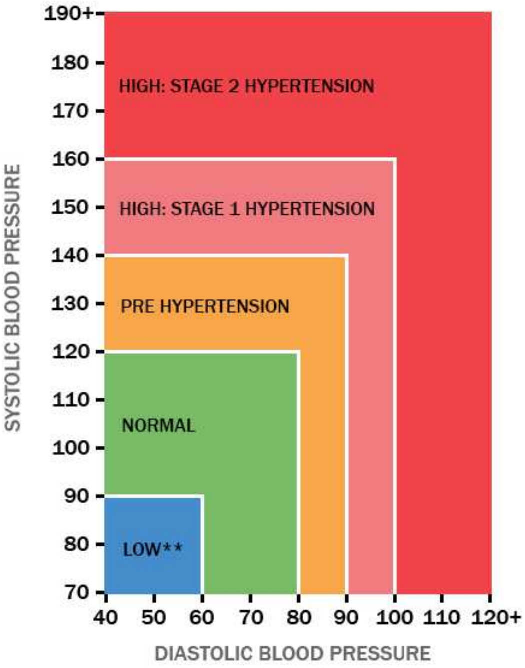

Figure 2.3: Classification of BP values into categories according to the American College of Cardiology and the American Heart Association.

60-79 mmHg for the DBP. Values below this threshold are considered hypotension.

- **Elevated blood pressure or pre-hypertension**. The top number ranges from 120 to 139 mmHg and the bottom number is between 80 and 89 mmHg.
- **Stage 1 hypertension**. The top number ranges from 140 to 159 mmHg or the bottom number is between 90 and 99 mmHg. These values covers the most cases and are typically caused by genetic or lifestyle factors.
- **Stage 2 hypertension**. The top number is 160 mmHg or higher or the bottom number is 100 mmHg or higher. It is often related to a specific disorder.
- Blood pressure higher than 180/120 mmHg is considered a **hypertensive emergency** and requires urgent medical care.

2.2.3 ABP MONITORING METHODS

There here exist several methods for (BP) measurement, usually divided in two categories: invasive blood pressure (IBP) monitoring and non invasive blood pressure (NIBP) monitoring.

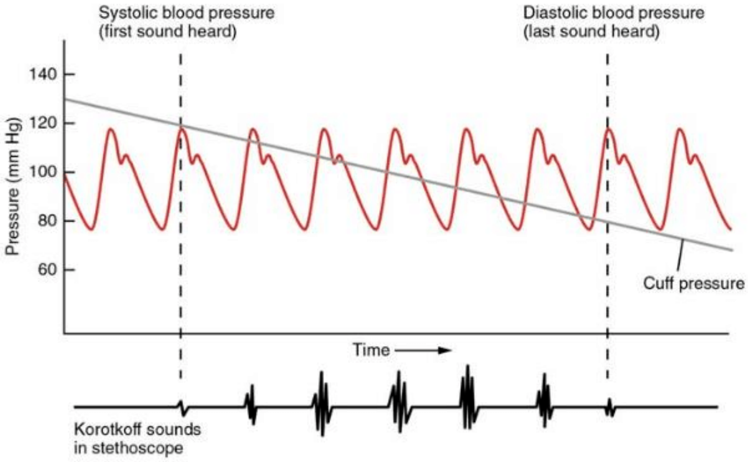

Figure 2.4: Cuff pressure during deflation on blood pressure during auscultatory method.

IBP is commonly used in the intensive care unit (ICU) and avoided in hypertensivepatients. This technique involves the use of a cannula inserted in a suitable artery (radial, femoral or brachial).
The main advantages are the continuous monitoring (beat-to-beat) of the patient's blood pressure, which may undergo rapid changes, the displaying of the corresponding waveform, the accurate measurement at very low pressure values (e.g shocked patients) and in people not suitable for NIBP (e.g obese patients).

An invasive set up is not appropriate in routine clinical practice; non-invasive methods are of common use in ambulatory measurements.
The most used technique is represented by the auscultatory method, using a mercury sphygmomanometer and a stethoscope. The cuff pressure, positioned around the upper portion of the arm, is brought up to a level higher than the SP, up to the total compression of the artery, determined by the palpatory method in the distal side of the hand. Once the cuff is gradually deflated, with the stethoscope, it is possible to hear the appearance of the Korotkoff sounds, used to determine SP and its disappearance to determine DP (Figure 2.4)[40] [7].

A minimum systolic value can be roughly estimated by palpation, most often used only in emergency situations since it could result imprecise. A third non-invasive technique is the oscillometric method and involves the observation of oscillations in the sphygmomanometer cuff pressure which are caused by the pulse pressure (PP) oscillations. It employs either deformable membranes that are measured using differential capacitance, or differential piezoresistance, and they include a microprocessor to automatically interpret the sensor results. In order to maintain accuracy the pressure sensor should be calibrated periodically [41].

2.3 PHOTOPLETHISMOGRAPY

The term **Photoplethysmography (PPG)**, introduced in the 1930s by Hertzman and colleagues, refers to a non -invasive technique for measuring the volume of blood flowing within the vessels [42]. It overcomes some of the limits of the classical methods for detecting cardiovascular diseases, such as ECG where the device need to acquire the signal from at least two points of the body. The PPG devices in fact require only a sensor that is usually integrated into the device case, resulting in an easier and more comfortable set up for low specificity heart monitoring [43].

The device used for measuring a PPG signal use a sensor to detect the pulse of the oxygenated blood to the periphery, that causes the relaxation of arteries and arterioles in the subcutaneous tissue.
This change in volume is detected by illuminating the skin with the light from a light-emitting diode (LED) and then measuring the amount of light either transmitted or reflected to a photodiode or photodetector.

The skin illumination by the LED has to be calibrated in order to reduce the ionization of skin cell an organic tissues. Furthermore, wavelength choice is also important and depends on the application and must be considered both susceptibility to motion artifacts and sensitivity in poor skin perfusion. Still, because of its relative freedom from artifacts, green light is often used in ambulatory monitoring applications.
Wearable PPG sensors have two operating modes (Figure 2.5):

- **Transmission mode**: light source and photodetector are placed in diametrically opposite sides, facing each other. The photodetector catches the light not absorbed by the tissues. Only a few quite thin body regions are suitable for this technique, however, this technique allows to isolate in a better way the sensor from the environmental light, so as to collect higher quality signals [44].

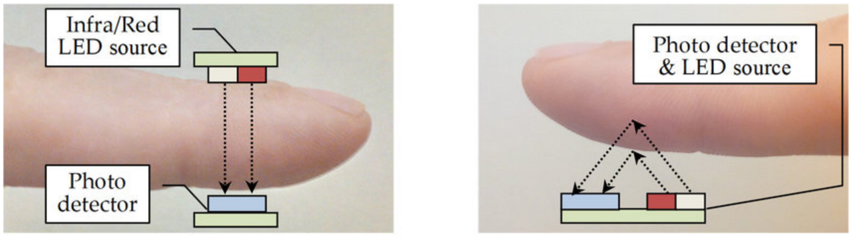

Figure 2.5: Basic principle of PPG sensors working in: (a) transmission, (b) reflection modes.

- **Reflection mode**: light source and photodetector are placed on the same side. Not requiring special conditions, it can be applied to thicker sites of the body, for instance: wrist, forehead and chest. On the other hand its high sensitivity to motion artifacts and to the presence of ambient light, limits the accuracy of the physiological measure [44].

2.3.1 PPG WAVEFORM

The PPG waveform is composed by a pulsatile (AC) physiological waveform attributed to cardiac synchronous changes in the blood volume with each heart beat, and is superimposed on a slowly varying (DC) baseline with various lower frequency components attributed to respiration, sympathetic nervous system activity and thermoregulation (Image 2.7) [43].
The analysis of the AC component is very important for heart activity monitoring and has a charateristic periodic shape composed by two phases: the *anacrotic phase* that represents the rising edge of the pulse so the contraction of the heart during the systolic phase, and the *catacrotic phase* that is identified with the diastolic phase.

The anacrotic phase is the characteristic that changes the most from subject to subject; it is in fact affected by subjective vascular conditions like arteries stiffness. Usually it is composed of three phases: an initial first pre-dicrotic dip, a successive dicrotic notch, and a final dip at the end [45] (Figure 2.7). The dicrotic notch is due to a reflexed wave, caused from arterial elasticity. An important aspect is the fact that the anacrotic phase is the characteristic lost when the monitored patient suffers from vascular diseases that increase the vascular resistance. For this reason, when this occurs, the dicrotic notch can be invisible if the signal is acquired in periphery.

The pulse wave begin (PWB), indicates the beginning of the systolic phase, that finish at the pulse wave systolic peak. The pulse wave (PWE) indicates the end of the diastolic phase. The

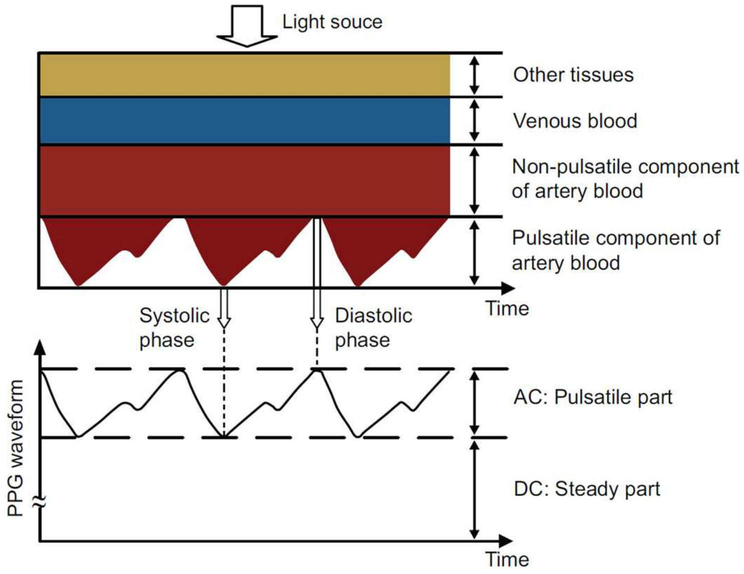

Figure 2.6: PPG waveform propagation from the photopletysmogram sensor.

time between these two phases is called pulse wave duration (PWD). The time elapsing from a pulse wave systolic peak to the consecutive is usually expressed in milliseconds and is called inter-beat interval (IBI) (Figure 2.7).

One of the most popular usage by PPG waveform is the heart rate for sport and daily home monitoring, since can be placed in multiple comfortable body regions without being invasive [46]. Furthermore, due to the recent COVID-19 pandemic crysis the research activity and the commercialization of PPG based devices increased [47], proving the importance given to this type of not invasive and accessible monitoring methods.
Some examples of wearable photoplethysmography devices (Figure 2.8).

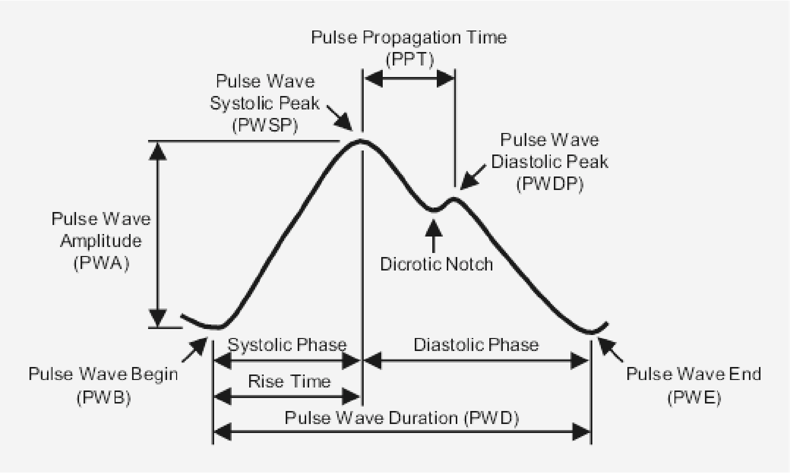

Figure 2.7: Representation of PPG waveform descriptors.

Figure 2.8: Some different types of wearable photoplethysmography devices.

3

Deep Learning Methods

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that gives calculators the ability to learn to perform a specific task without being explicitly programmed to do so, that consist into the development of computational models of learning.
Deep Learning (DL) in particular employs models that are inspired by human brain structure, in order to achieve computational capabilities such as data sensing and data understanding, as well as decision making based on previous and present outcomes, common capability of the networks of biological systems [48].

While classical Machine Learning algorithm are model based, i.e. human comprehensible, Deep Learning discovers intricate structure in large data sets in order to automatically learn and capture hierarchical features.

3.1 MACHINE LEARNING REGRESSION TECHNIQUES

The Regression [49] is a statistical process used to estimate relationships between variables. In particular, regression determines the functional relation between the independent variables (x vectors) and the dependent target variables (y vector). In machine learning regression, the target values (y) are numerical and continuous (in R), and the goal of training is to learn a function f, such that $f(x) \to y$. It is considered that the independent variable is exact while the dependent variable is affected by error (e.g. measurement inaccuracy).

The simplest method for regression is linear regression. This model establishes a linear predictor function between the dependent variables (Y) and the independent ones (X). There are many ways to find the best regression line that fits the data distribution, and they are based on the minimization of the distance. The linear regression equation is 3.1:

$$y_i = \alpha + \beta x_i + \varepsilon_i \tag{3.1}$$

where α and β are the parameter of the curve. The objective of the regression is to find the best $\hat{\alpha}$ and $\hat{\beta}$ for the fitting of the data that minimize the error term Q, i.e. the sum of the squared residuals ε_i (equation3.2).

$$Q(\hat{\alpha}, \hat{\beta}) = \sum_{i=1}^{n} \hat{\varepsilon}^2 \tag{3.2}$$

Still there are application where linear regression is not enough for modelling the data, in those cases other types data representations in the space are considered like polynomial regression.
Furthermore, other two concepts are needed to model regression algorithm:

1. Bias: error due to inaccurate assumption or simplification made by the model. It is a measure of how much does the average model over all training sets from the true model.
2. Variance: metric for evaluating the dependence of the model from the input data. It is a measure of how much models estimated from different training sets differ from each other.

When a model has high bias and low variance means that there are few parameters for the modelling, in this case we can say that the model is underfitting. A low bias and a high variance instead, indicates overfitting and it happens when the model is too accurate just for the training data. For this reason, is important to make a variance and bias trade-off that keep the model accurate in learning from the training data but still maintaining good generalization capabilities (figure 3.1).

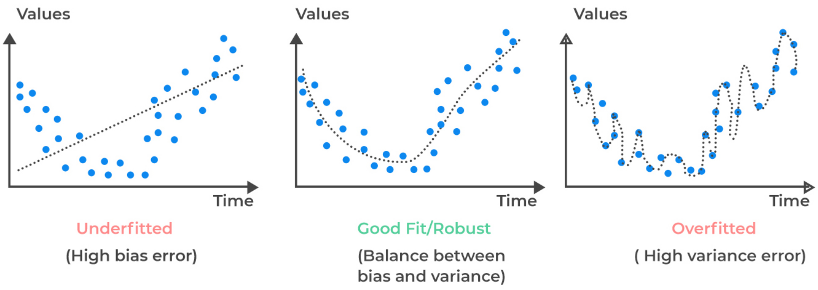

Figure 3.1: Representation of underfitting, good fit and overfitting models on the same data.

3.2 Deep Learning Models Structure

Deep Learning models are composed by hierarchically simple computational units, called nodes or neurons, connected by links or edges. In DL implementations, the value at each connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. This function is called activation function and emulates the human biological process of neuron activation and firing in the brain, resulting in a more expressive network, capable to compute highly complicated tasks, like for instance pattern recognition, classification/clustering, prediction.

The weights are modified by a learning algorithm according to a loss function, which models the observed errors. Adjusting these weights the network can learn and improve the accuracy of its predictions. Learning is complete when examining additional observations do not usefully reduce the error rate.

A Neural network is a Deep Learning model defined by several elements. Will therefore follow a quick outline of his structure and components (Figure 3.2):

- Neurons: the neurons are the basic information processing units of a NN, they receive an input, pass it through an activation function, and produce an output using an output function. The activation function limits the amplitude of the output of the node and adds some non linearity.
- Architecture: defines the network structure that is the number of artificial neurons in the network and their inter-connectivity.
- Learning algorithm: is the procedure used to perform the learning process, it modifies the weights to compensate for each error found during learning. One of the most popu-

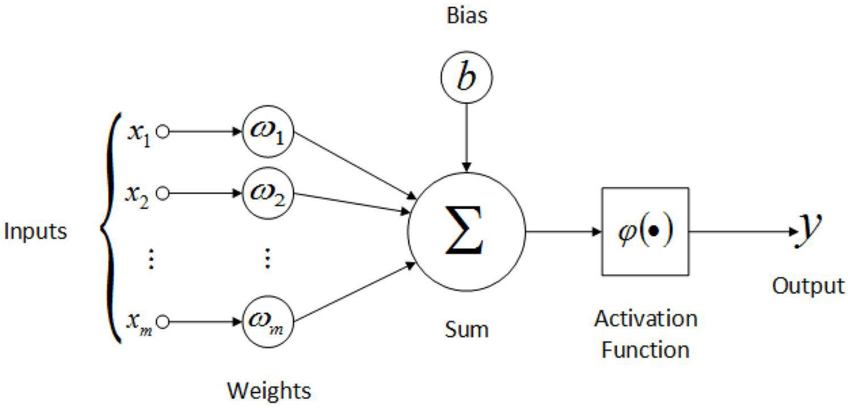

Figure 3.2: Single neuron structure.

lar learning algorithms methods is based on gradient descent and it is called backpropagation or generalized delta rule. For each training instance the backpropagation algorithm first makes a prediction (forward pass), measures the error, then goes through each layer in reverse to measure the error contribution from each connection (reverse pass), and finally tweaks the connection weights to reduce the error (Gradient Descent step). How much the weights are modified in a single step is computed by different optimizers algorithms according to a parameter specified as learning rate.

3.2.1 Fully Connected Layer

The first modeled layer is the fully-connected layer, where the totality of the nodes in previous layer are connected to each neuron in the target layer. It represents a function from $\mathbb{R}$ to $\mathbb{R}^n$.

In this architecture the nodes are disposed in an input layer receiving data in input, one or more hidden layers, and an output layer producing the final output of the network. Since the layers are fully connected, when they are stacked one after another they create a deeper model, called Feed Forward Neural Network (FFNN), capable to capture higher level features.
The layers close to the input layer are usually called the lower layers, and the ones close to the outputs are usually called the upper layers. With one hidden layer we can represent any continuous function of the input data, and with two hidden layers even discontinuous functions can be represented.

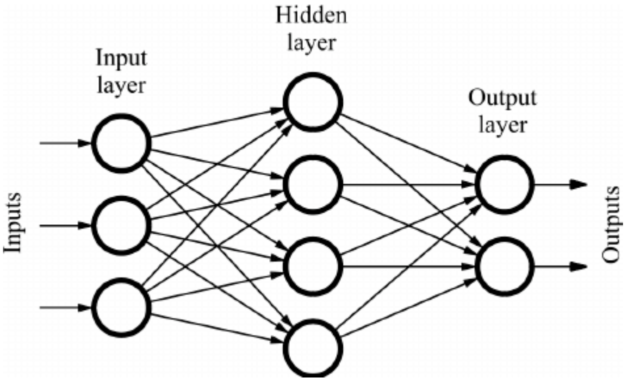

Figure 3.3: Feed Forward Neural Network structure.

Feed Forward Neural Network models are computationally demanding due to the number of connections, and have complexity O(mn) for each layer: for this reason they are usually used as last layers to encode high level features, or as final outputs.

3.2.2 CONVOLUTIONAL LAYER

Convolutional neural network (CNN) were inspired by the functionality of visual cortex and are mainly known for their applications with two-dimensional data like for instance image recognition. Nevertheless CNNs are not restricted to visual perception, they are also successful at many other tasks, such as voice recognition, natural language processing and in general time series analysis.

In the visual cortex neurons usually have a small local receptive field, reacting only to visual stimuli located in a limited field of view. Other neurons instead have larger receptive fields, and they react to more complex patterns that are combinations of the lower-level patterns. These observations led to the idea that the higher-level neurons are based on the outputs of neighboring lower-level neurons.

The core building block of a CNN is the convolutional layer, a layer based on convolution, a linear operation that involves the multiplication of a set of two dimensional array of weights, called filter or kernel, with the array of input data.
This model allows the NN to focus on small low-level features in the first hidden layer, then collects them into larger higher-level features in the next hidden layer, and so on.

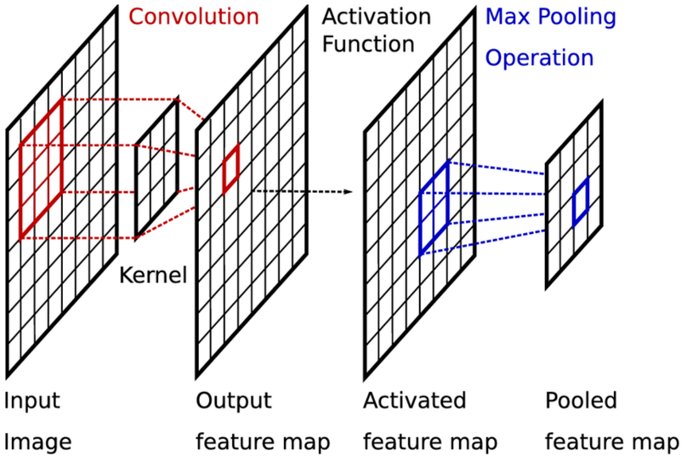

Figure 3.4: Basic CNN with a single layer convolutional block and pooling.

During the forward phase each convolutional kernel is convolved on its input producing a two-dimensional activation map of that kernel. This map is called feature map.
Convolutional layers usually have more than one filter, returning in outputs one feature map per filter.
Since a neuron receptive field extends across all the previous layers' feature maps, the network can learn through filters that activate when it detects some specific type of feature at some spatial position in the input, characteristic that can be also defined as translation invariance.
Furthermore, in convolutional layers all neurons within a given feature map share the same weights and bias term, reducing drastically the number of parameters in the model, that combined with the lower number of connections required in convolutional layers, generally make them more efficient with respect to fully-connected ones.

Another important concept in CNN models is pooling, which is a type of non-linear down-sampling, with the function to create a lower resolution version of the input signal, that still contains the large or important structural elements. It also reduces the number of parameters and memory usage.
The pooling layer has a receptive field but with no weights associated to it, it just aggregates the inputs using an aggregation function. There are several non-linear functions to implement pooling, among which max pooling is the most common. This last one partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum.

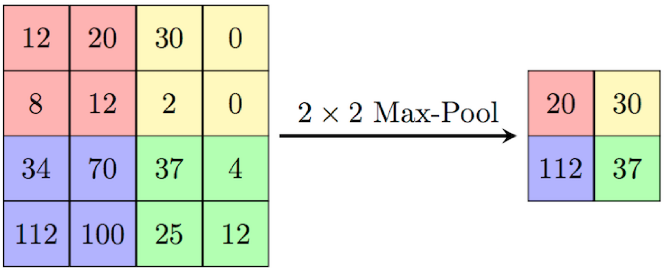

Figure 3.5: Example of max pooling function with stride=2 and kernel 2x2.

3.2.3 ACTIVATION FUNCTIONS

Activation functions are used to break down the linear relation between input/output and enhance the expressiveness of the model, helping the network to learn complex patterns in the data. Various activation functions have been proposed, the following are the most popular ones:

- Linear activation function: called also as ”no activation,” or ”identity function”. The output is proportional to the input, and is used as activation layer of neural networks to extract features for the final classification block or to interpolate the input (regression) (equation 3.3).

$$y = \sum_{i=1}^{D} w_i x_i = w^T x \tag{3.3}$$

- Sigmoid activation function: the function output is returned in the range [0, 1]. The sigmoid has a characteristic shape of an ”S” and is often used to predict probabilities. Since it is affected by the gradient vanishing problem with gradient near 0 or 1, this function is not suited as inter-layer activation because it could break the training phase (equation 3.4).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.4}$$

- Tanh activation function: called also Hyperbolic tangent function. The output is mapped in range [-1,1]. The tanh function has and ”S” strong shape and has the advantage that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero. This function is usually employed in solutions targeting two-class classification problems (equation 3.5).

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.5}$$

- Rectified Linear Unit function (ReLU): the function output is returned in the range [0, ∞), if the input is smaller than zero the function return as output zero while if the input equal or greater than zero (x >= 0) the function will be linear returning as output x. It is preferred to the Sigmoid and Tanh functions, because it greatly accelerates the convergence during the training (equation 3.6).

$$ReLU(x) = max(0, x) \tag{3.6}$$

- Leaky Rectified Linear Unit function (Leaky ReLU): it is based on the ReLU, but it has a small slope for negative values instead of a flat slope. In this way the function output is returned in the range (-∞, ∞). Having a small negative slope is one attempt to fix the problem that in some cases affect the ReLU, when his gradient tend to converge to zero.

3.2.4 Loss Functions

The loss function is used to compute the difference distance between the output of a DL model and the expected output. It's a method to evaluate how your network models the data.
The purpose in the majority of the cases is to minimize or maximize loss by updating the network weights with an optimization algorithm. The loss function has to be chosen according to the problem that we want out model to perform: for a regression task is usually used Mean Absolute Error (MAE) or Mean Squared Error (MSE) while for binary class classification task is used the Binary Cross Entropy (BCE) (equations , ,).

$$MAE = \frac{1}{N} \sum_i |y_i - \hat{y}_i| \tag{3.7}$$

$$MSE = \frac{1}{N} \sum_i (y_i - \hat{y}_i)^2 \tag{3.8}$$

$$BCE = -\frac{1}{N} \sum_i y_i \cdot log(P(y_i)) + (1 - y_i) \cdot log(1 - P(y_i)) \tag{3.9}$$

3.2.5 OPTIMIZERS

An optimization algorithm is a procedure computed iteratively, used to minimize a derivable convex function obtained as output of a deep learning model. The principle behind optimization is the fact that the gradient is point towards the direction of the largest increase of the function in the region close to w and this implies that moving in the opposite direction equals to moving towards a minimum point.

Gradient Descent (GD) is one of the most used optimization algorithms to optimize deep learning models. The aim of gradient Descent is to minimize the target function $J(\Theta)$, with the model parameter $\Theta \in R^d$. The parameters are updated in the opposite direction of the gradient of the objective function $\nabla_{\Theta} J(\Theta)$ with respect to the parameters.
The parameter η is the learning rate and is used to control the step-size in each update.
Since the classic GD computes the gradients for the whole dataset to perform single update, it can be very slow when applied to large datasets. For this reason in this case is more optimal to apply the Stochastic Gradient Descent (SGD) optimization algorithm, that computes an update for a subset of the training samples instead.
It is faster compared to GD and allows to learn with new examples on-the-fly, but it is also less stable [50].

A relevant problem with all the gradient descent methods discussed above, is the long time it takes for the algorithm to update significantly when the gradient is small, for example around a saddle point. To solve this problem is used the idea of momentum incorporated to gradient descent.
Momentum is a parameter used to accelerate descent in the relevant direction by adding a fraction γ of the update vector of the past steps to the current update. In this way if the updates points on the same direction for multiple iterations, they will be accumulated and the updating step size is increased by the momentum multiplier [51] [50].

Nesterov accelerated gradient (NAG) is a method that implement concept of the momentum (momentum-based gradient descent) but improving his capabilities to predict and ”slow down” when the gradient is about to increase again after a minima. Standard momentum computes first the current gradient and then takes a big jump in the direction of the previous accumulated gradient. Differently, NAG firstly move with a big update in the direction of the

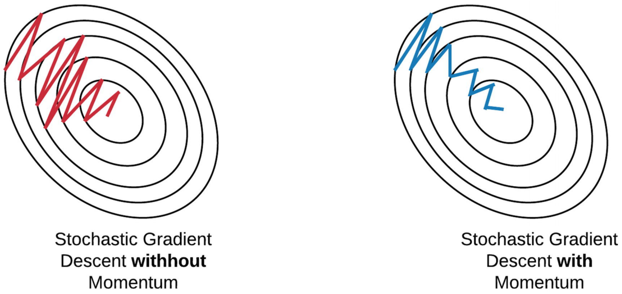

Figure 3.6: Representation of the difference between the use of momentum in gradient descent.

previous accumulated gradient, then estimates the next position of the parameters and makes a correction in the gradient direction, avoiding to moving too fast [52].

Adaptive Moment Estimation (ADAM) is a second order optimizer that computes adaptive learning rates for each parameter, keeping an exponential decaying average variable, obtained from the estimates of the mean and variance of the past squared gradients [52] [53]. ADAM become more popular due to the faster convergence speed respect to Gradient Descent, even if on longer training GD has shown better generalization. Nesterov Accelerated Adaptive Moment Estimation (NADAM) combines ADAM and NAG, with the idea to compute an accurate step in the gradient, assisting the ADAM optimizer with the generalization capabilities of Gradient Descent [53].

3.3 U-Net

The U-NET is an end-to-end fully convolutional network (FCN) firstly developed by Olaf Ronneberger in [54] in order to work on biomedical images segmentation. The architecture contains two paths. First path is the contraction path, also called as the encoder, which is used to capture the context in the image. The encoder is just a standard sequence of convolutional layers and max pooling layers. The second path is the symmetric expanding path, also called as the decoder, which is used to enable precise localization using transposed convolutions.
The name comes from the ”U” shape structure, where the input pass multiple down-sizing and expanding stage. In figure 3.7 is represented the original U-Net model structure.

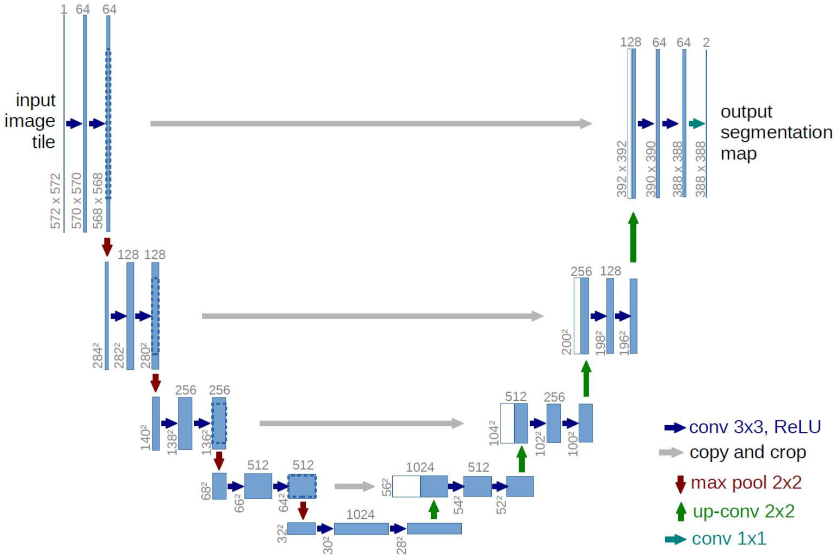

Figure 3.7: Original U-Net model structure. It present 4 down/up sampling stages.

4

Methodology

4.1 DATASET

4.1.1 MIMIC CLINIC DATABASE

In recent years there has been a concerted move towards the adoption of digital health record systems in hospitals. In the US, for example, the number of non-federal acute care hospitals with basic digital systems increased from 9.4 to 75.5% over the 7 year period between 2008 and 2014 [55].

To evaluate how PPG and ABP are related was used the Multi-Parameter Intelligent Monitoring in Intensive Care (MIMIC-III) waveform database, because of his size and relevance in past works, and also because the MIMIC collection of signals are representative of the full range of pathophysiologies that result in sudden blood pressure changes [21].
MIMIC is a large, freely-available database comprising deidentified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center (Boston, Massachusetts, United States) between 2001 and 2012. The database includes information such as demographics, vital sign measurements made at the bedside ($\sim$ 1 data point per hour), laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality (including post-hospital discharge) [56].
MIMIC-III waveform is a subset of the MIMIC database which contains 67,830 records of

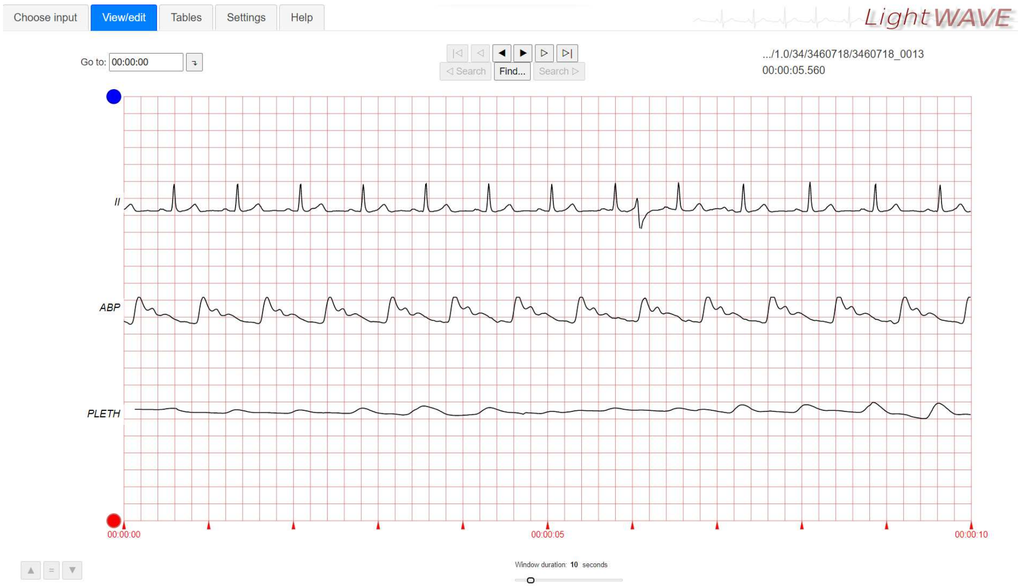

Figure 4.1: Part of record 3460718 visualized using LightWAVE. ECG, ABP and PPG waveform are present.

multiple physiologic signals (“waveform”) and time series of vital signs (“numerics”) collected from approximately thirty thousand bedside patient, in adult and neonatal intensive care units (ICUs).

Waveform almost always include one or more types of ECG signals, and often include continuous arterial blood pressure (ABP) waveform, fingertip photoplethysmogram (PPG) signals, and respiration, with additional waveform (up to 8 simultaneously) as available. Numerics typically include heart and respiration rates, SpO2, and systolic, mean, and diastolic blood pressure, together with others as available. Recording lengths also vary; most are a few days in duration, but some are shorter and others are several weeks long [57].

The recordings vary in length from 1 to 80 hours depending on patients. The data obtained from the bedside monitors are divided into files each containing 10 minutes of recorded signals, which can then be assembled without gaps to form a continuous recording. The data were written in ten-minute segments in order to limit possible loss of data from power interruptions. The ECG, PPG and ABP signals are sampled at 125 Hz with 8, 10 or 12-bit precision. It is common for the physiologic signals to be interrupted or changed occasionally during recordings of such long duration. When using a viewer such as LightWAVE (Figures 4.1, 4.2), all signals available at any time during a record are listed, although in most cases only a subset is visible at any given time.

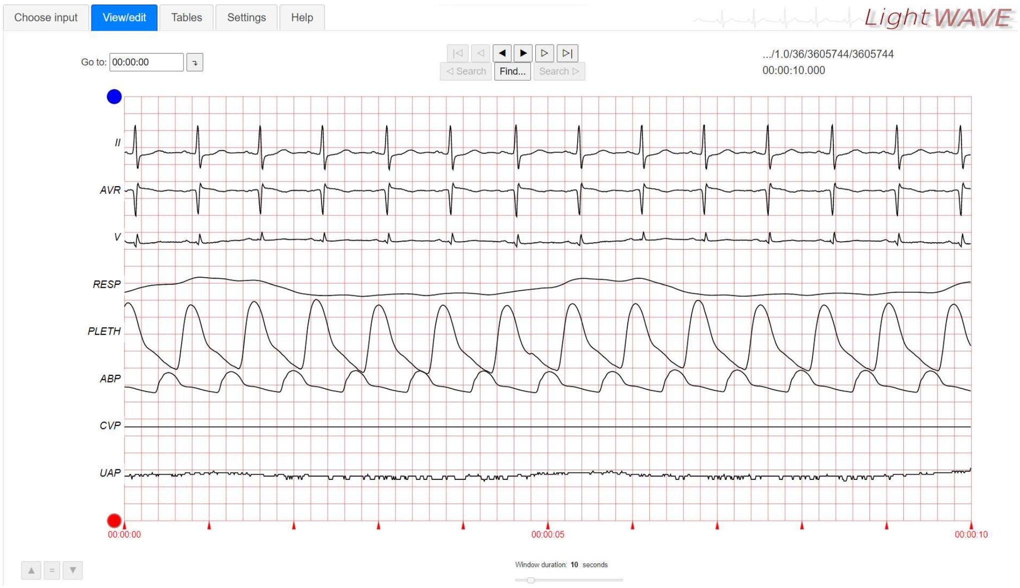

Figure 4.2: Part of record 3605744 visualized using LightWAVE. All 8 possible waveform are present in this record, including respiration rates, different ECG and BP signals.

4.2 DATA PRE-PROCESSING AND AUGMENTATION

Since the total uncompressed size of the MIMIC-III waveform database is large (6.7 TB), initially the data extraction of this work planned to code an automatic algorithm based on the WFDB python library, with the possibility to discard every record without the requested signals.

However during this implementation we discovered some technical limitation related to the use of this database: waveform or numerics missing, a signal could not be available throughout an entire record, gaps and patient identification, but above all, inter-waveform alignment problems and not uniform bit-rate.

For this reason, in this work has been decided to use the Cuff-Less Blood Pressure Estimation Dataset [58] from the UCI Machine Learning Repository [59]. The UCI Dataset is a filtered version of the MIMIC-III Waveform database, only the instances presenting simultaneously ABP, PPG, and ECG signals where filtered, resulting in 12000 records of 942 patients.

The selected records of the UCI Dataset were uniformly divided into four parts, each part containing 3000 instances, and the data is available in MATLAB file format (.mat).

After the data loading and removing the ECG waveform from the records, were performed

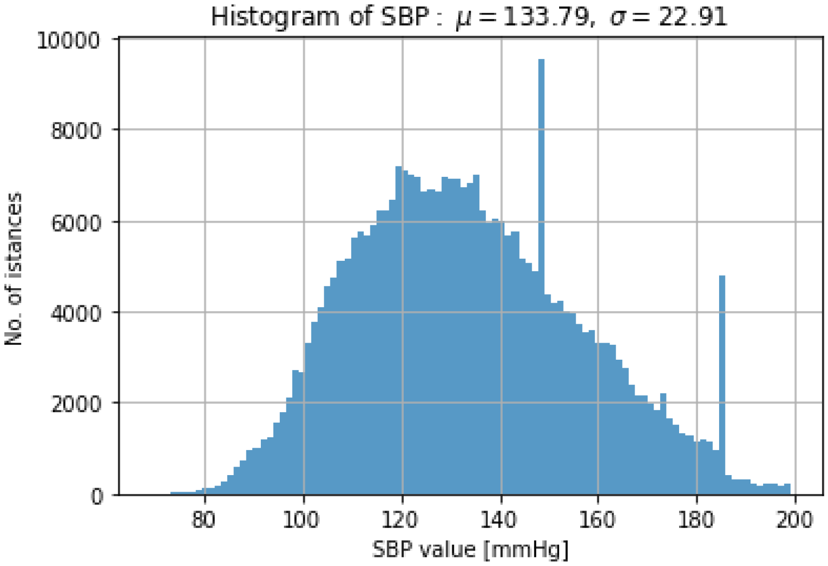

Figure 4.3: Systolic blood pressure distribution.

manual analysis on the waveform and computed the statistics of the totality of instances, in order to retrieve information about the SDP and DBP distributions (Figures 4.3, 4.4).

Data pre-processing was one of the most important and time-consuming part of this study since multiple signal processing techniques were analyzed and tried. The Figure 4.5 demonstrates the data pre-processing pipeline.

The first step in the pre-processing is to filter the PPG signal. For this stage we primarily tried three possible methods. The first filtering technique included a single Equiripple FIR filter with cutoff frequencies of 0.5 Hz to 8 Hz [60]. This range of frequency were chosen because values below 0.5 Hz were identified as baseline wandering, while values above 8 Hz were considered high-frequency noise.
The second method consisted in a low pass 4th order Butterworth filter followed by a baseline draft or wandering correction.
Baseline wander (BW) is a low-frequency artefact present in electrocardiogram (ECG) and also PPG signal recordings of a subject [61]. The removal of this artifact is an important step in processing because BW makes interpretation of ECG recordings difficult. The main cause of the BW are the movement and respiration of the patient or the improper use of the electrodes (electrode-skin impedance) [62].

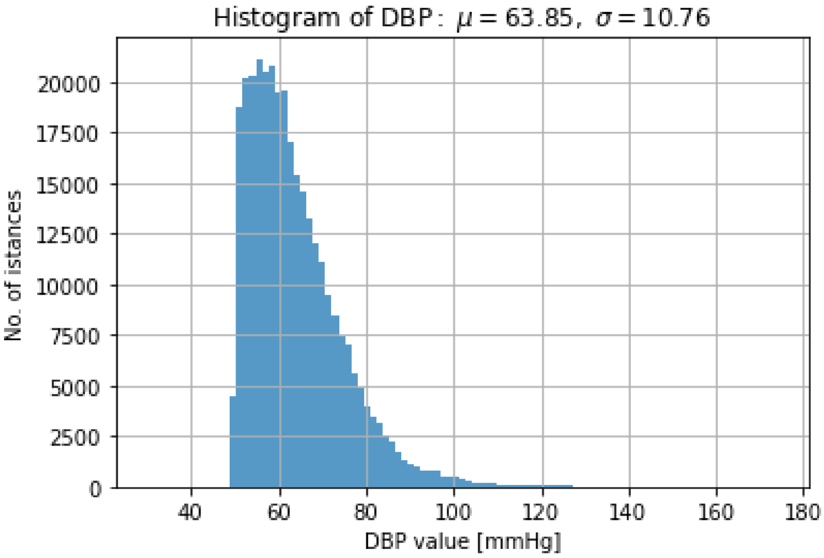

Figure 4.4: Diastolic blood pressure distribution.

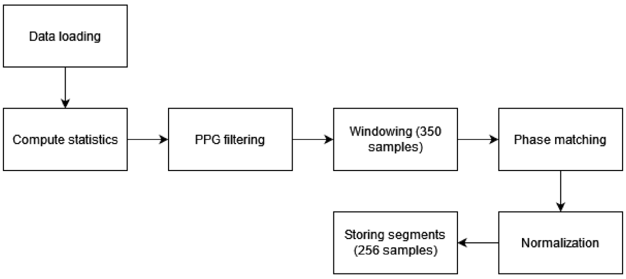

Figure 4.5: Pre-processing pipeline.

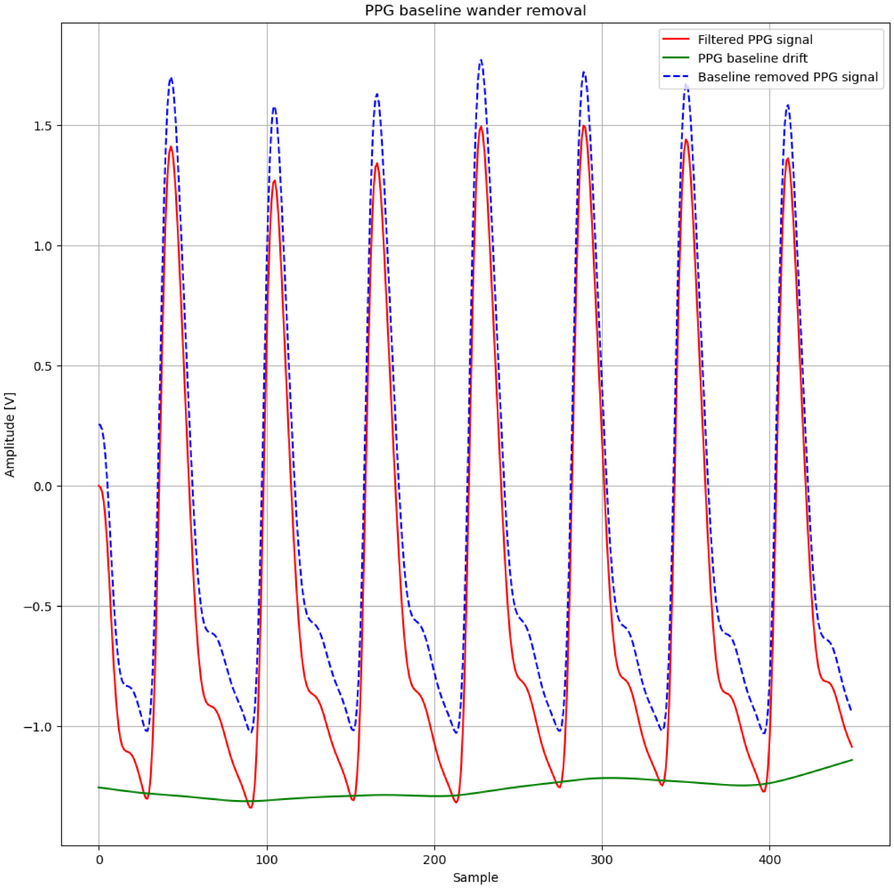

Figure 4.6: PPG signal baseline wander removal.

In order to correct this artifact, we used an asymmetrically reweighted penalized least squares smoothing algorithm [63] [64]. The baseline drift removal effect is represented in figure 4.6.

The third method use a discrete wavelet transform decomposition (DWT) filter. The wavelet transform is performed to 10 decomposition levels, with Daubechies 8 (db8) as the mother wavelet [65]. Then, the low (0-0.25 Hz) and high frequency (250-500 Hz) components are eliminated by setting the decomposition coefficients to zero, followed by soft Rigrsure thresholding. Finally, the signal is retrieved by reconstructing the decomposition. This technique allows to remove both high-frequency noise and baseline drift [65].
Between the three possible techniques the first performed better. The filtering obtained with the Equiripple FIR band-pass filter in fact, performed in the same way as the discrete wavelet decomposition (DWT) filter, but with less computational impact instead. Furthermore, the

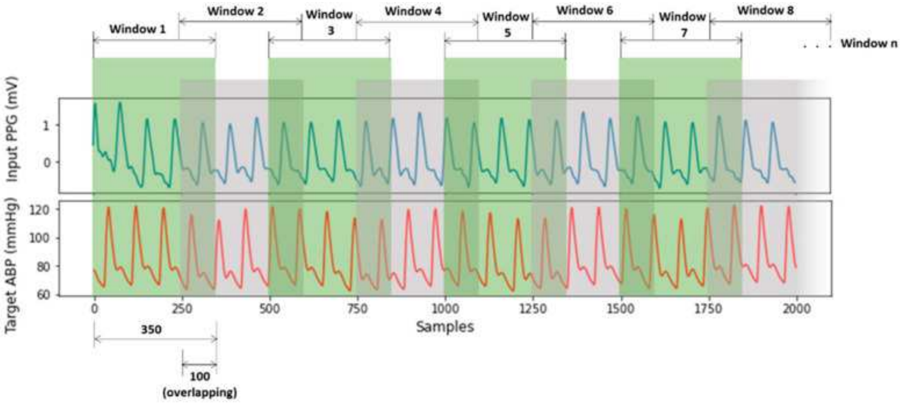

Figure 4.7: PPG and ABP signal slicing.

baseline drift removal algorithm used in the second method resulted too impactful on the signal dynamic where simply filtering the lower frequencies could obtain it with more success.

The second phase of the processing pipeline was windowing both the PPG and the ABP signals. The PPG and ABP signals were divided into windows of 350 samples sequentially, with overlapping of 100 samples (Figure 4.7) [26].

After windowing the signals, for each segment was performed a bad signal detection. This algorithm check if both the PPG and the ABP segments presents anomalies such as flat lines and/or flat peaks, and if the ABP segment has dysfunctional systolic or diastolic pressure values, or if they present anomalies with the frequency of the peaks (Figure 4.8). The segments which presented these type of anomalies were thus discarded.
Flat lines are usually indication of a device disconnection during the parameter measurement, or an incorrect memory saving of the signal while flat peaks reveal the saturation of the signal. In order to detect these problems is checked if are present 6 consecutive samples (2% of the total segment length) with unchanged values.
In order to detect ABP segments with missing peaks has been used a find-peaks function that given an one dimensional array, finds all local maxima by simple comparison of neighboring values. With the list of peaks values, the algorithm simply checked if the time between a peak and the next was greater than 5 seconds [24]. In that case the anomaly was detected.

Then the algorithm check the systolic and diastolic range in the ABP segments. In [58] are considered only signals with a range of (60 mmHg ≤ DBP ≤ 130 mmHg) and (80 mmHg ≤

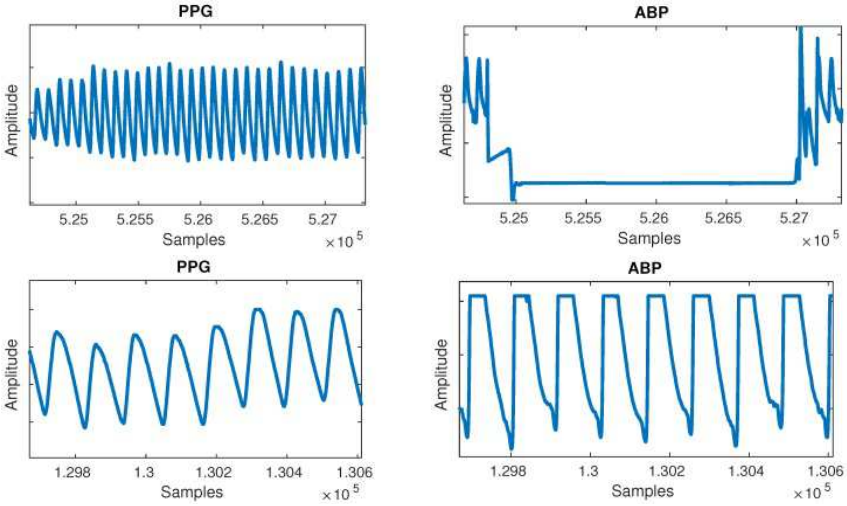

Figure 4.8: Representation of flat lines and saturated peaks in ABP waveforms.

	Min	Max	Mean	Std
SBP	68.40	199.99	133.79	22.91
DBP	50.00	174.49	63.85	10.76
MAP	58.49	184.99	88.84	13.73

Table 4.1: Distribution of the dataset ABP

SBP ≤ 180 mmHg). However in this work, similarly to [66] we wanted to set our algorithm on a broader range of pressure segments since a real-world application scenario might exhibit extremely small and high BP values. Therefore, we considered signals with DBP as low as 50 mmHg and as hight as 140, and SBP as low as 70 and high as 190 mmHg.
The distribution of the dataset ABP are presented also in Table 4.1. It can be observed that the systolic pressure has a greater value of standard deviation; as stated in [58] this large range is more likely to cause difficulties when computing a SBP prediction.

The fourth phase in the preprocessing pipeline is the phase-matching between the PPG segment and the corresponding target ABP segment.
In order to compute the location difference between the two signal windows, cross-correlation was performed. The sample position indicated by the maximum value of the cross-correlation indicates the time lag between the two segments. Keeping the position PPG window fixed the ABP signal was then translated by the computed time difference. In figure 4.11 is represented the effect of the time-shift processing on the PPG and ABP windows.

After phase-matching the segments, both the PPG and ABP signals were further windowed into 256 sample segments. This is due the fact that the tried architectures is 256 as input size. In this way the total time duration of each window is 2.048 s.

Finally the PPG and ABP segments were normalized using the equation 4.1 where x_i refers to i-th signal window, and x_{max} and x_{min} are the respectively maximum and minimum values of all the signals after the filtering and bad signal detection process. The normalized PPG and ABP are then stored in order to prepare the train validation and test dataset.
Since the training setup in use has very limited computational limits compared to the machines used in the recent studies, with only a gtx1060 GPU and 6 GB of RAM, after the pre-processing pipeline the segments were saved in .h5 format. This prevents some of the computational problems, reducing the processor memory usage that will otherwise overflow interrupting the process.

$$X_{normalized(i)} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

The SBP and DBP distribution after the processing pipeline is represented in the figures 4.9 and 4.10.

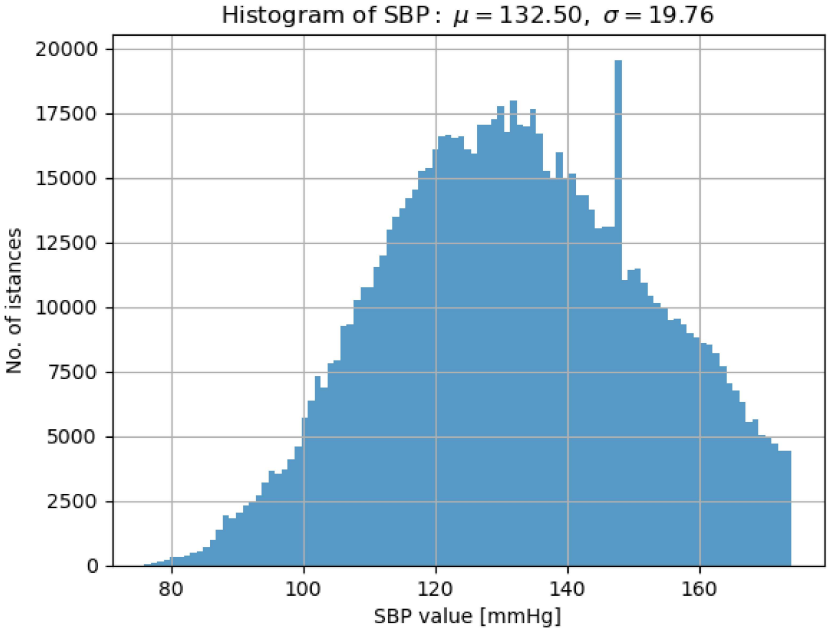

Figure 4.9: SBP distribution after processing.

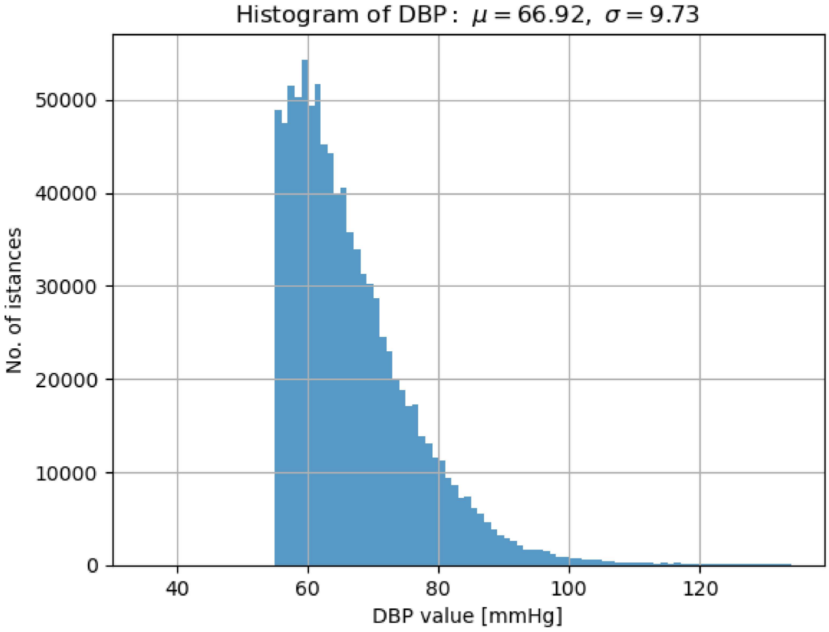

Figure 4.10: DBP distribution after processing.

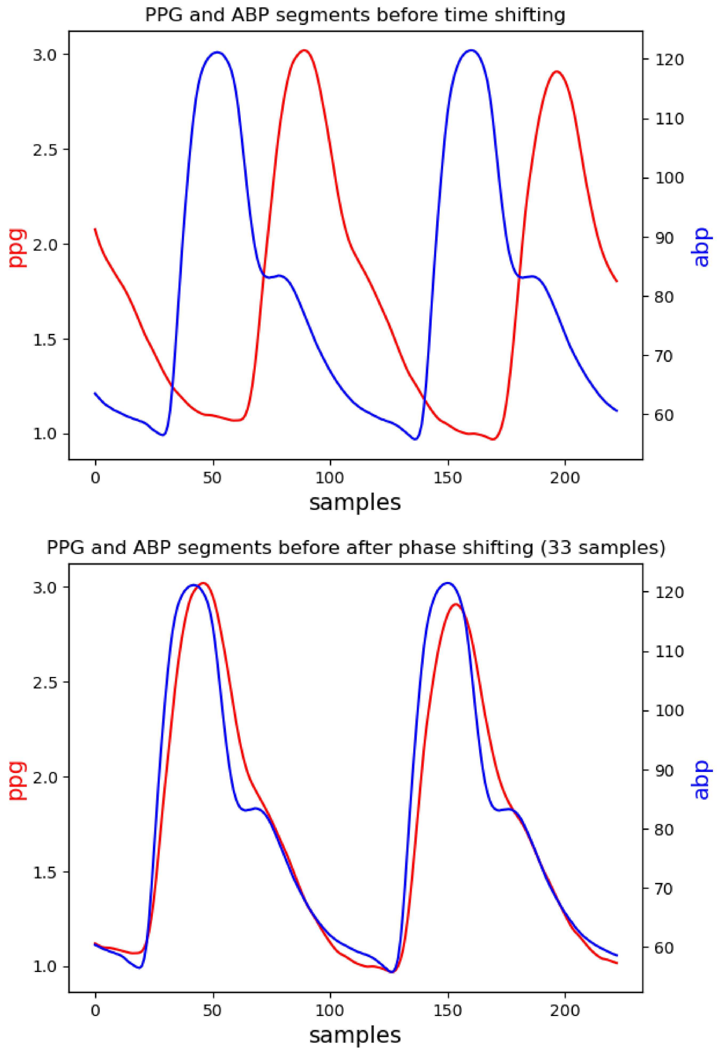

Figure 4.11: Before and after effect of phase-matching algorithm.

4.3 DEEP LEARNING MODELS

4.3.1 BASEUNET

The models have been implemented using the Anaconda framework. The main dependencies of the work are: Python 3.9, cudatoolkit 11.3.1, cudnn 8.2.1, tensorflow 2.9.1 and pytorch 1.12.0.
The setup includes a single machine with a gtx1060 GPU with 6 GB of GDDR5 RAM.
Though the original U-Net is designed to perform semantic segmentation on images, for our purpose, we tried to follow the recent trends of models for this type of problem, using this architecture to reconstruct 1-dimensional signals, which is primarily a regression task. The scope of the modeling phase was to find the best performing U-Net architecture which can be used as an autoencoder for optimum feature extraction.

The first configuration implemented is a UNet with some modifications respect to the one used in [67].
The coded model is represented in 4.1. The structure present:

- The input of the network is a PPG signal with a length of 256 samples of 1 dimensional (1D) convolution layer.
- On each layers are applied two 3 × 1 convolution layers with a Leaky Rectified Linear Unit activation function, Max pooling and batch normalization.
  The Leaky ReLU activation function was used to avoid the dying ReLU problem.
- The number of feature channels after each block doubles in the contracting path so that the architecture can learn the complicated structures effectively. As the original model, the structure present a total of 4 down and up sampling stages. The respective features for contracting phase are 64 in input, 128, 256, 512, 1024.
- The bottleneck present two 3 × 1 convolution layers followed by a 2 × 1 up-sampling layer. This path can create a feature space able to generates relevant features.
- A dropout block of 50% has been applied at the last layers of the contracting path and on bottleneck path in order to reduce the overfitting of the model.
- The expansion path is symmetric to the contracting one, each expansion block passes the input to one 2 × 1 and two 3 × 1 convolution layers followed by a Leaky ReLU activation

function after each convolution layer and a 2 × 1 up-sampling layer. The number of feature channels used in the expansion path is halved to maintain symmetry.

- Finally, at the last expansion path, two extra 3 × 1 convolution layers are used to map each 64 feature vector equal to the input dimension. Using this network, the PPG signal window of 256 samples is mapped into the ABP signal window of 256 samples.

Has been used 70% of the four .mat dataset total data for training our model, 15% for validation, and the remaining 15% for testing. Previous shuffling at fixed random seed has been performed on the dataset.

```python
def DUnet(input_size = (256,1)):

    inputs = Input((input_size))

    conv1 = Conv1D(64, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(inputs)
    conv1 = Conv1D(64, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv1)
    pool1 = MaxPooling1D(pool_size=(2))(conv1)
    conv2 = Conv1D(128, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(pool1)
    conv2 = Conv1D(128, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv2)
    pool2 = MaxPooling1D(pool_size=(2))(conv2)
    conv3 = Conv1D(256, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(pool2)
    conv3 = Conv1D(256, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv3)
    pool3 = MaxPooling1D(pool_size=(2))(conv3)
    conv4 = Conv1D(512, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(pool3)
    conv4 = Conv1D(512, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv4)
    drop4 = Dropout(0.5)(conv4)
    pool4 = MaxPooling1D(pool_size=(2))(drop4)

    conv5 = Conv1D(1024, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(pool4)
    conv5 = Conv1D(1024, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv5)
```

```
    drop5 = Dropout(0.5)(conv5)

    up6 = Conv1D(512, 2, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(UpSampling1D(size = (2))(drop5))
    merge6 = concatenate([drop4,up6], axis = 2)
    conv6 = Conv1D(512, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(merge6)
    conv6 = Conv1D(512, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv6)

    up7 = Conv1D(256, 2, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(UpSampling1D(size = (2))(conv6))
    merge7 = concatenate([conv3,up7], axis = 2)
    conv7 = Conv1D(256, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(merge7)
    conv7 = Conv1D(256, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv7)

    up8 = Conv1D(128, 2, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(UpSampling1D(size = (2))(conv7))
    merge8 = concatenate([conv2,up8], axis = 2)
    conv8 = Conv1D(128, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(merge8)
    conv8 = Conv1D(128, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv8)

    up9 = Conv1D(64, 2, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(UpSampling1D(size = (2))(conv8))
    merge9 = concatenate([conv1,up9], axis = 2)
    conv9 = Conv1D(64, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(merge9)
    conv9 = Conv1D(64, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv9)
    conv9 = Conv1D(2, 3, activation = 'leaky_relu', padding = 'same',
    kernel_initializer = 'he_normal')(conv9)
    conv10 = conv10 = Conv1D(1, 1)(conv9)

    model = Model(inputs=[inputs], outputs=[conv10])

    model.summary()

```

```
    return model
```

Listing 4.1: UNet architecture

4.3.2 DUNET

Another configuration implemented is a different U-Net variation. It also present 4 down/up sampling stages, the number of feature utilised are reduced to 16, 32, 64, 128, 256, in order to decrease the computational complexity of the model and the time needed to the training phase. For both the contraction and the expanding phase has been chosen two 3 × 1 convolution layers, followed by batch normalization. ReLu activation has been chosen 4.2.

```
def UNet(length=256, n_channel=1):

    x = 16

    inputs = Input((length, n_channel))
    conv1 = Conv1D(x,3, activation='relu', padding='same')(inputs)
    conv1 = BatchNormalization()(conv1)
    conv1 = Conv1D(x,3, activation='relu', padding='same')(conv1)
    conv1 = BatchNormalization()(conv1)
    pool1 = MaxPooling1D(pool_size=2)(conv1)

    conv2 = Conv1D(x*2,3, activation='relu', padding='same')(pool1)
    conv2 = BatchNormalization()(conv2)
    conv2 = Conv1D(x*2,3, activation='relu', padding='same')(conv2)
    conv2 = BatchNormalization()(conv2)
    pool2 = MaxPooling1D(pool_size=2)(conv2)

    conv3 = Conv1D(x*4,3, activation='relu', padding='same')(pool2)
    conv3 = BatchNormalization()(conv3)
    conv3 = Conv1D(x*4,3, activation='relu', padding='same')(conv3)
    conv3 = BatchNormalization()(conv3)
    pool3 = MaxPooling1D(pool_size=2)(conv3)

    conv4 = Conv1D(x*8,3, activation='relu', padding='same')(pool3)
    conv4 = BatchNormalization()(conv4)
    conv4 = Conv1D(x*8,3, activation='relu', padding='same')(conv4)
    conv4 = BatchNormalization()(conv4)
```

```
    pool4 = MaxPooling1D(pool_size=2)(conv4)

    conv5 = Conv1D(x*16, 3, activation='relu', padding='same')(pool4)
    conv5 = BatchNormalization()(conv5)
    conv5 = Conv1D(x*16, 3, activation='relu', padding='same')(conv5)
    conv5 = BatchNormalization()(conv5)

    up6 = concatenate([UpSampling1D(size=2)(conv5), conv4], axis=2)
    conv6 = Conv1D(x*8, 3, activation='relu', padding='same')(up6)
    conv6 = BatchNormalization()(conv6)
    conv6 = Conv1D(x*8, 3, activation='relu', padding='same')(conv6)
    conv6 = BatchNormalization()(conv6)

    up7 = concatenate([UpSampling1D(size=2)(conv6), conv3], axis=2)
    conv7 = Conv1D(x*4, 3, activation='relu', padding='same')(up7)
    conv7 = BatchNormalization()(conv7)
    conv7 = Conv1D(x*4, 3, activation='relu', padding='same')(conv7)
    conv7 = BatchNormalization()(conv7)

    up8 = concatenate([UpSampling1D(size=2)(conv7), conv2], axis=2)
    conv8 = Conv1D(x*2, 3, activation='relu', padding='same')(up8)
    conv8 = BatchNormalization()(conv8)
    conv8 = Conv1D(x*2, 3, activation='relu', padding='same')(conv8)
    conv8 = BatchNormalization()(conv8)

    up9 = concatenate([UpSampling1D(size=2)(conv8), conv1], axis=2)
    conv9 = Conv1D(x, 3, activation='relu', padding='same')(up9)
    conv9 = BatchNormalization()(conv9)
    conv9 = Conv1D(x, 3, activation='relu', padding='same')(conv9)
    conv9 = BatchNormalization()(conv9)

    conv10 = Conv1D(1, 1)(conv9)

    model = Model(inputs=[inputs], outputs=[conv10])

    return model
```

Listing 4.2: DUNet architecture

4.3.3 LITEUNET

Due to the problems had with computational complexity required on the previous one, the third model implemented is a lite version of the last architecture 4.3.

It present two reduced down/up sampling stages. The number of feature channels after each block has been increased to still maintain the feature learning capabilities of the model. They are respectively 32, 64, 128. The convolutional setting configuration remains the same of the first implemented UNet model.

```
def UNetLite(length = 256, n_channel=1):

    inputs = Input((length, n_channel))
    conv1 = Conv1D(32,3, activation='relu', padding='same')(inputs)
    conv1 = BatchNormalization()(conv1)
    conv1 = Conv1D(32,3, activation='relu', padding='same')(conv1)
    conv1 = BatchNormalization()(conv1)
    pool1 = MaxPooling1D(pool_size=2)(conv1)

    conv2 = Conv1D(64,3, activation='relu', padding='same')(pool1)
    conv2 = BatchNormalization()(conv2)
    conv2 = Conv1D(64,3, activation='relu', padding='same')(conv2)
    conv2 = BatchNormalization()(conv2)
    pool2 = MaxPooling1D(pool_size=2)(conv2)

    conv3 = Conv1D(128,3, activation='relu', padding='same')(pool2)
    conv3 = BatchNormalization()(conv3)
    conv3 = Conv1D(128,3, activation='relu', padding='same')(conv3)
    conv3 = BatchNormalization()(conv3)
    pool3 = MaxPooling1D(pool_size=2)(conv3)



    up8 = concatenate([UpSampling1D(size=2)(conv3), conv2], axis=2)
    conv8 = Conv1D(64, 3, activation='relu', padding='same')(up8)
    conv8 = BatchNormalization()(conv8)
    conv8 = Conv1D(64, 3, activation='relu', padding='same')(conv8)
    conv8 = BatchNormalization()(conv8)

    up9 = concatenate([UpSampling1D(size=2)(conv8), conv1], axis=2)
```

```
    conv9 = Conv1D(32, 3, activation='relu', padding='same')(up9)
    conv9 = BatchNormalization()(conv9)
    conv9 = Conv1D(32, 3, activation='relu', padding='same')(conv9)
    conv9 = BatchNormalization()(conv9)

    conv10 = Conv1D(1, 1)(conv9)

    model = Model(inputs=[inputs], outputs=[conv10])

    return model
```

Listing 4.3: LiteUNet architecture

5

Hyperparameters and experimental results

We report the metrics obtained from the training and validation set with the different models. Multiples configuration of parameters has been tried on the different networks architectures. The best configuration tried on the first network featured the use of Adam with 0.0001 learning rate as optimizer, 40 epochs, batch size = 8, Mean Square Error (MSE) as loss (Figure 5.1). We kept low learning rate due to the small batch size, to avoid unstable fluctuations. For each epoch we decided to save the model only if the performance in test-set is better than the previous, in order to avoid over-fit. Already with 40 max iterations the full process lasted about an day, while the training loss had still to fully converge, the test MSE settled on around 200 pretty quickly. For this reason, selecting a larger iteration number would lead only to a more overfitting network.

On the second network has been tried a configuration with Adam with 0.0008 learning rate as optimizer, 45 epochs, batch size = 8, Mean Square Error (MSE) as loss (Figure 5.2). Unfortunately this training process did not produced better result, computing for almost 2 days but failing to converge after few epochs.

In order to reduce the computational complexity and to have a model with faster training phase has been tried the lite U-Net configuration using both a larger batch size and learning rate and slighter increase number of iterations: Adam with 0.001 learning rate as optimizer, 50 epochs, batch size = 32, Mean Square Error (MSE) as loss (Figure 5.3).

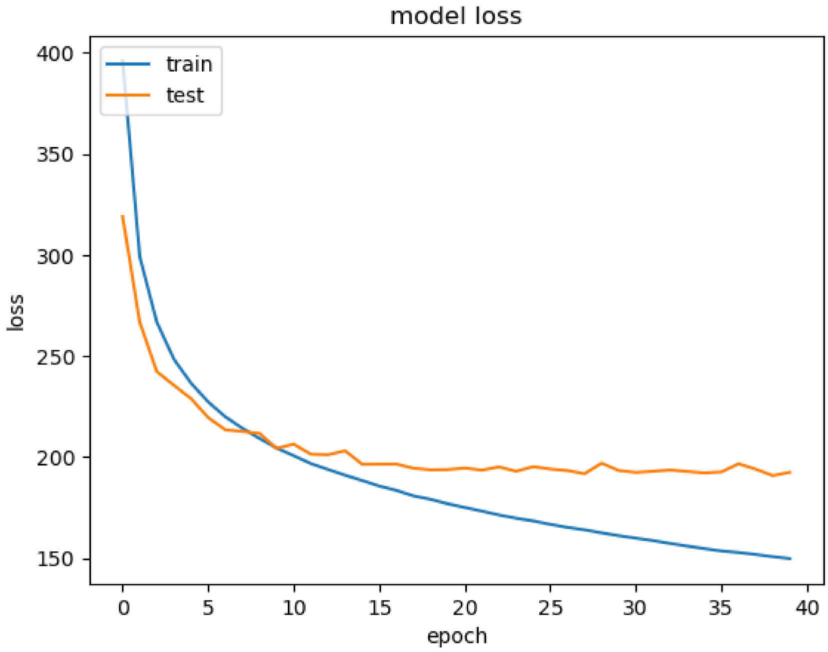

Figure 5.1: Train and validation (not test as image legend suggest) loss on the fist network.

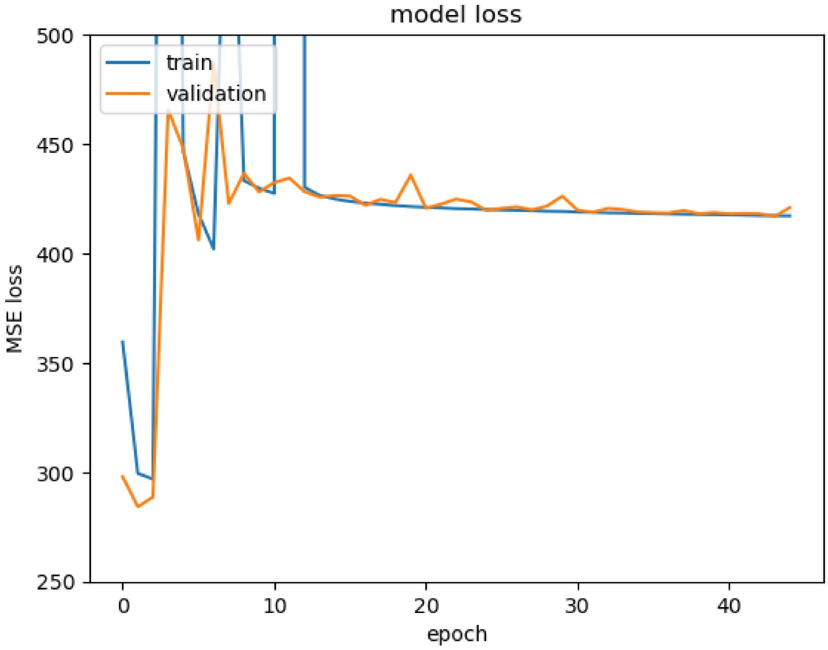

Figure 5.2: Train and validation (not test as image legend suggest) loss on the second network.

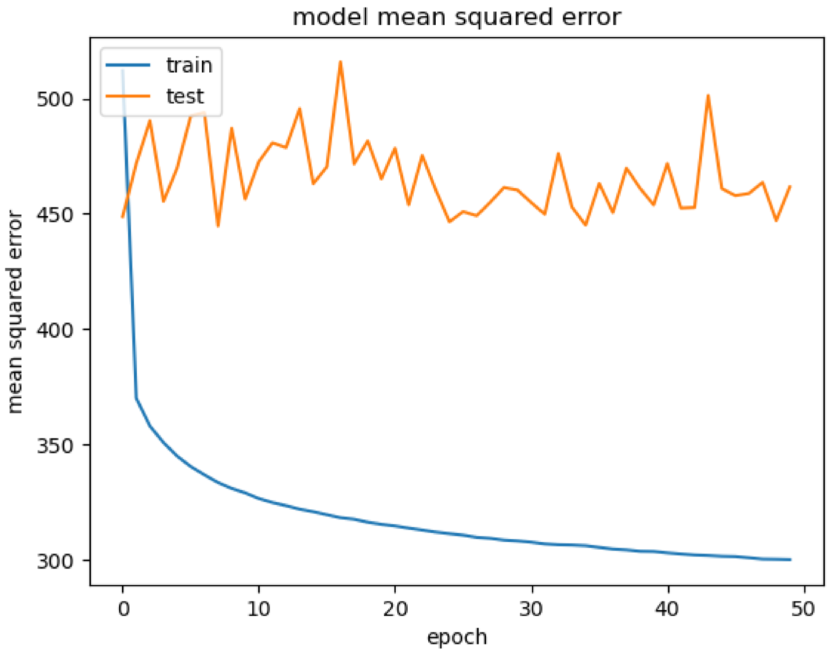

Figure 5.3: Train and validation (not test as image legend suggest) MSE loss on the third lite network.

Then has been then tried a configuration using the first model, that so far performed better result, with an input of 1024x1 segments instead of 256 long ones. This could bring the model to learn better the signal dynamics and the correlation between the waveform, leading to better results.
The chosen parameters are Adam with 0.0004 learning rate as optimizer, 50 max epochs, batch size = 32, Mean Square Error (MSE) as loss (Figure 5.4).

Has finally been applied the trained models on the test dataset, in order to evaluate Mean Squared Error (MSE), Mean Absolute Error (MAE) as well as represent the prediction waveform that the model has in output compared to the real target signal.
As described in table 5.1, the best performing model is the latter described.

The predicted signals for the models are represented in figures 5.5, 5.6 and 5.7.
The best model BaseUNet1024 is able to successfully predict the dynamic of the waveform and detect the pressure notch in the release phase of the pressure, but unfortunately has still difficulty to predict the max peaks of the signal.

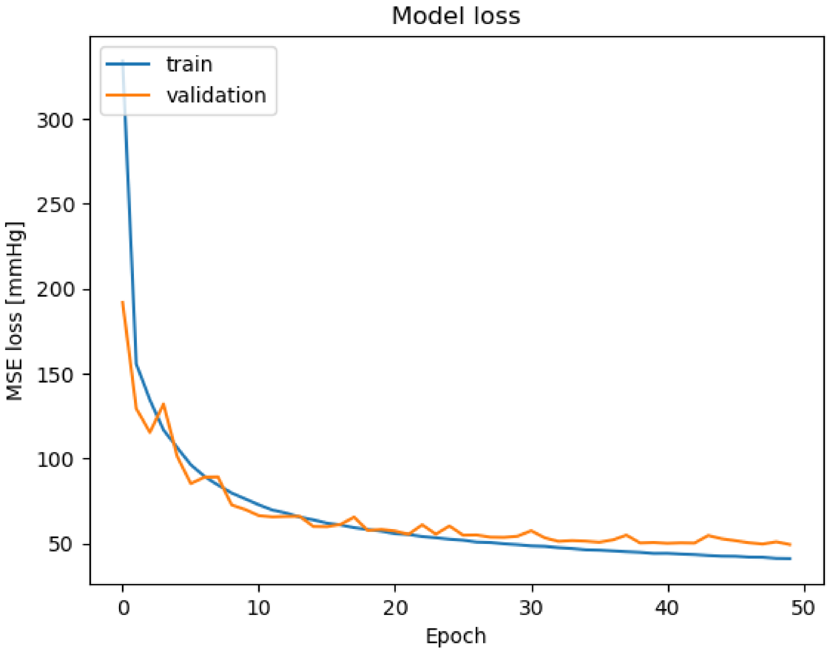

Figure 5.4: First model with 1024x1 input.

From the predicted signal segments has been then computed the SBP, DBP and MAP values, the model overall resulted in a Mean Absolute Error (MAE) of 42.96 [mmHg] for SBP, 27.82 [mmHg] for DBP, 9.56 [mmHg] for the MAP value and a Root Mean Squared Error (RMSE) of 49.49 [mmHg] for SBP, 34.84 [mmHg] DBP and 12.27 [mmHg] MAP.

Test Loss Metrics			
Model	Epoch	MSE	MAE
BaseUNet	40	440.95	16.74
DUNet	45	437.82	16.78
LiteUNet	50	452.12	16.51
BaseUNet (1024 samples input)	**50**	**239.65**	**11.79**

Table 5.1: Test Loss Metrics

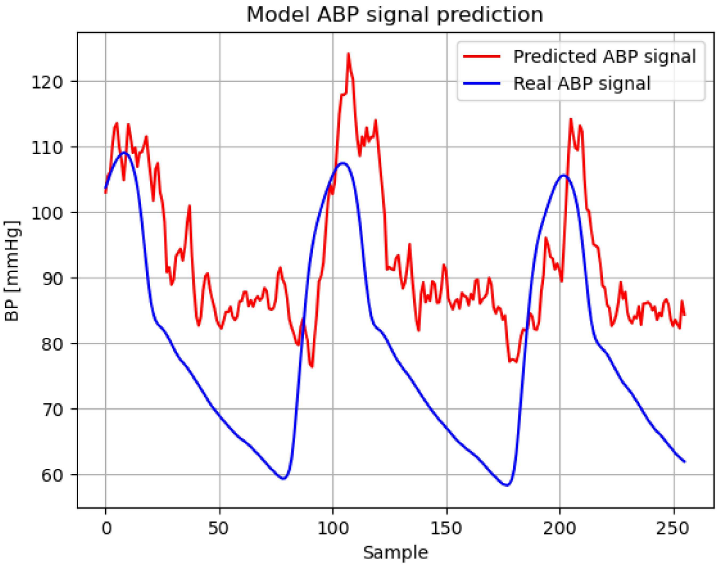

Figure 5.5: First tried U-Net architecture

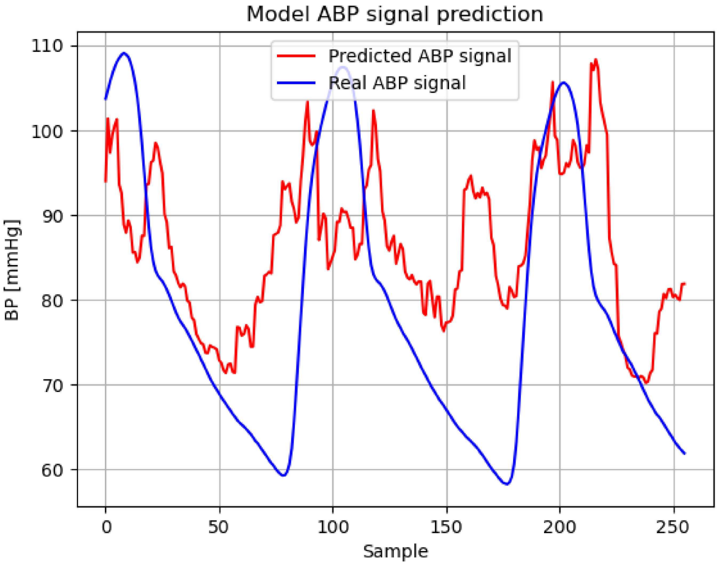

Figure 5.6: First tried U-Net architecture

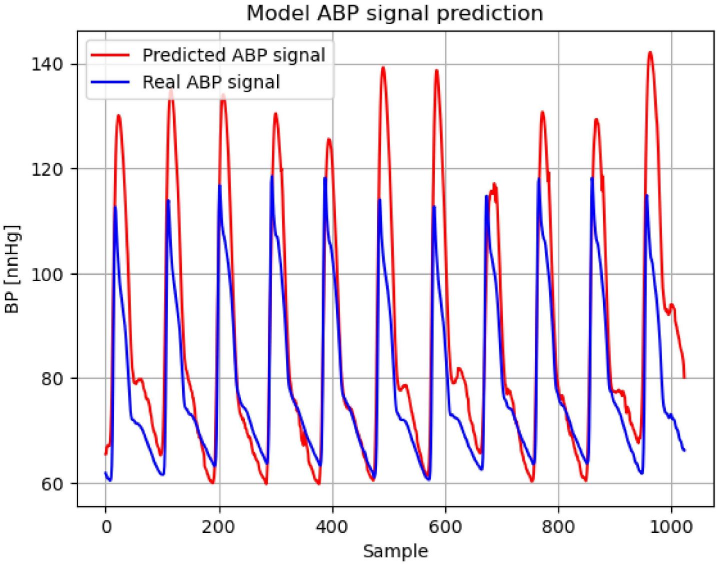

Figure 5.7: First tried U-Net architecture

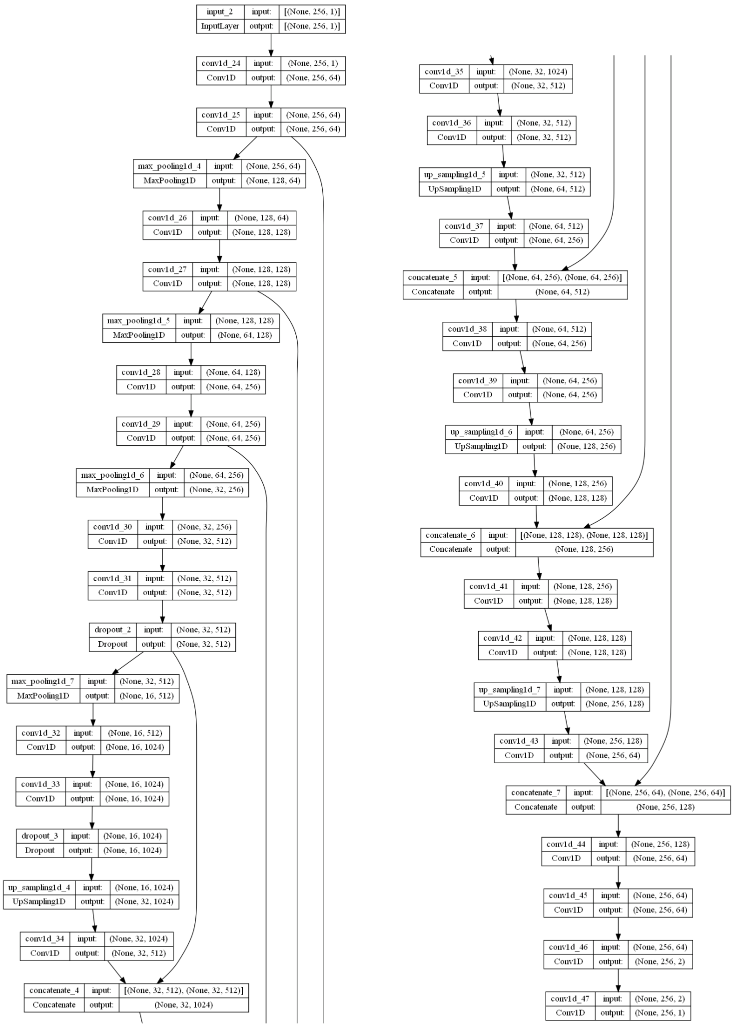

Figure 5.8: BaseUNet architecture

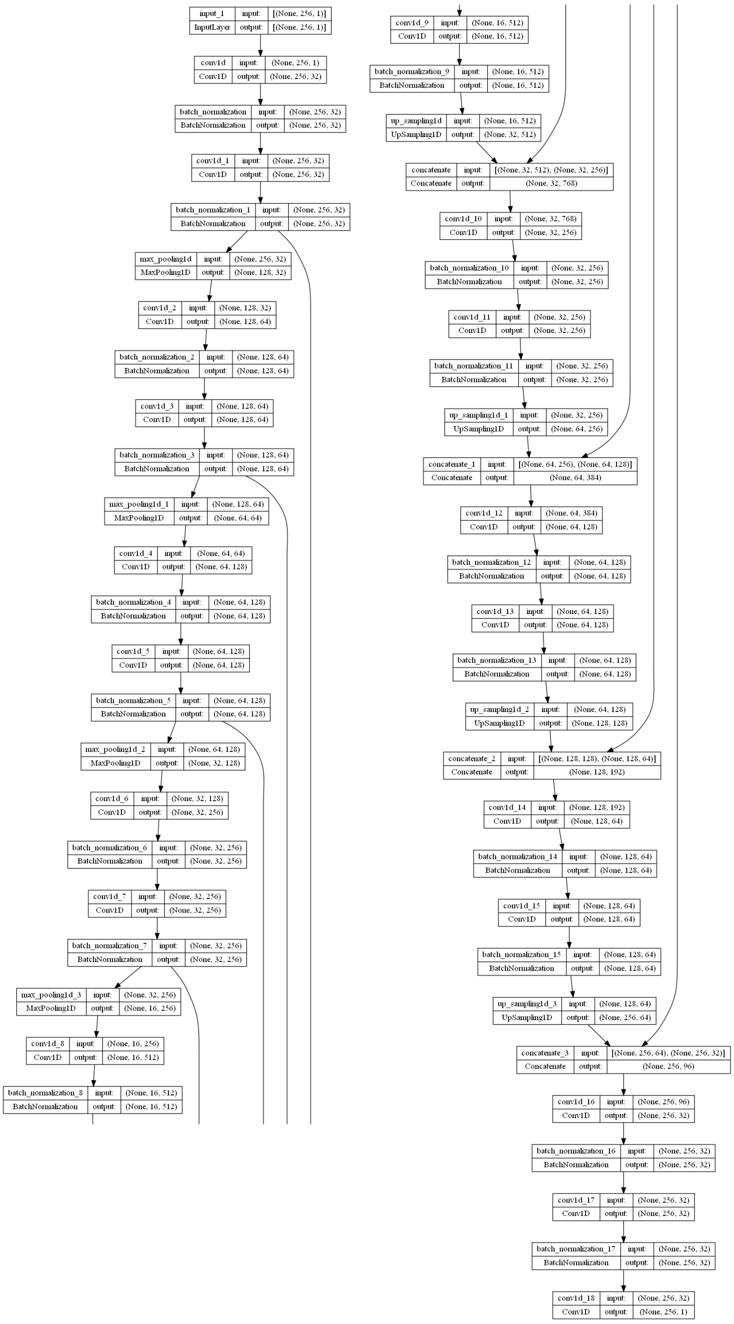

Figure 5.9: DUNet architecture

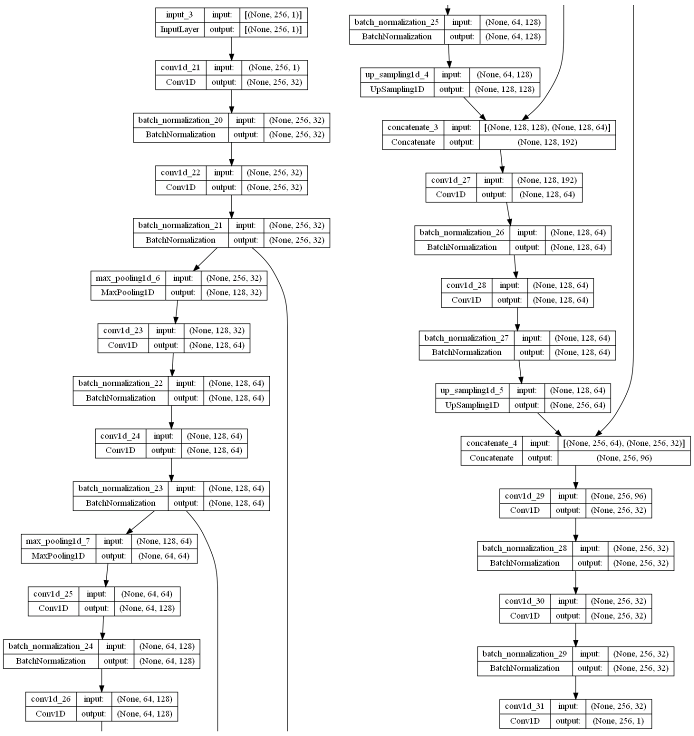

Figure 5.10: LiteUNet architecture

6

Conclusion

In this work we studied the use of Deep Learning models, in particular Deep Convolutional U-Net architectures, for the prediction of the arterial blood pressure (ABP) directly from the photoplethysmography (PPG) signal. We then implemented different U-Net models that took 256 samples PPG segments, computing the ABP prediction.

We briefly introduced the ABP measuring methods and his problems, as well as providing context to the related works and to state of the art methods used in this type of problem.
Then has been briefly described the medical background of this work, more specifically to the arterial blood pressure, his monitoring and the implications that hypertension have on the human health, as well as describing the photoplethysmography measurement and his waveform.
Later we introduced the basic concepts of Machine Learning and Deep Learning, in particular describing the main components used on Deep Convolutional Neural Networks.

We then explained the pipeline used for the dataset acquisition and the data processing, starting from the raw set of signals of the MIMIC-III database, data pre-processing and data augmentation. We developed several U-Net based models, trying to improve the performance to better results, still keeping an eye to the complexity of each architecture, given the quite low computational potential of our experimental setup.
For this reason has been tried also lighter versions of some models, presenting a least number of variables.

The best performing model is a variant of the first implemented, BaseUNet, that received in input 1024 samples segments instead of 256 samples ones, resulting in a prediction of Mean Absolute Error (MAE) of 42.96 [mmHg] for SBP, 27.82 for DBP and 9.56 for the MAP value.

Since this type of Deep Learning data-based models are very dependent on the quantity and quality of the data in input is quite possible to imagine an improved version of the model presented.
Future developments can obtain better results not only by increasing the size of the dataset but also considering more complex configurations and processing pipelines, possible by more performing experimental setups.
In future studies will be crucial also to find a balance between model complexity and performances, in order to consider an architecture that could be embedded in wearable portable devices. This would bring an important advancement in the field, enabling an easy method to perform continuous healthcare monitoring of arterial blood pressure, to prevent the onset of irreversible damages, such us cardiovascular diseases and hypertension.

References

[1] G. Parati, G. S. Stergiou, E. Dolan, and G. Bilo, "Blood pressure variability: clinical relevance and application," *The Journal of Clinical Hypertension*, vol. 20, no. 7, pp. 1133–1137, 2018.

[2] M.-C. Irigoyen, K. De Angelis, F. Dos Santos, D. R. Dartora, B. Rodrigues, and F. M. Consolim-Colombo, "Hypertension, blood pressure variability, and target organ lesion," *Current hypertension reports*, vol. 18, no. 4, pp. 1–13, 2016.

[3] J. S. Rana, S. S. Khan, D. M. Lloyd-Jones, and S. Sidney, "Changes in mortality in top 10 causes of death from 2011 to 2018," *Journal of general internal medicine*, vol. 36, no. 8, pp. 2517–2518, 2021.

[4] W. H. Organization *et al.*, *Global atlas on cardiovascular disease prevention and control: published by the World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization*. World Health Organization. Regional Office for Europe, 2011.

[5] D. Levy, M. G. Larson, R. S. Vasan, W. B. Kannel, and K. K. Ho, "The progression from hypertension to congestive heart failure," *Jama*, vol. 275, no. 20, pp. 1557–1562, 1996.

[6] J. Handler, "The importance of accurate blood pressure measurement," *The Permanente Journal*, vol. 13, no. 3, p. 51, 2009.

[7] A. S. Meidert and B. Saugel, "Techniques for non-invasive monitoring of arterial blood pressure," *Frontiers in medicine*, vol. 4, p. 231, 2018.

[8] K. Kario, "Management of hypertension in the digital era: small wearable monitoring devices for remote blood pressure monitoring," *Hypertension*, vol. 76, no. 3, pp. 640–650, 2020.

[9] K. B. Siaron, M. X. Cortes, S. E. Stutzman, A. Venkatachalam, K. M. Ahmed, and D. M. Olson, "Blood pressure measurements are site dependent in a cohort of patients with neurological illness," *Scientific reports*, vol. 10, no. 1, pp. 1–7, 2020.

[10] A. Bur, H. Herkner, M. Vlcek, C. Woisetschläger, U. Derhaschnig, G. Delle Karth, A. N. Laggner, and M. M. Hirschl, "Factors influencing the accuracy of oscillometric blood pressure measurement in critically ill patients," *Critical care medicine*, vol. 31, no. 3, pp. 793–799, 2003.

[11] P. Verdecchia, J. A. Staessen, W. White, Y. Imai, and E. O'brien, "Properly defining white coat hypertension," *European heart journal*, vol. 23, no. 2, pp. 106–9, 2002.

[12] S. Rastegar, H. GholamHosseini, and A. Lowe, "Non-invasive continuous blood pressure monitoring systems: current and proposed technology issues and challenges," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 1, pp. 11–28, 2020.

[13] M. E. Chowdhury, K. Alzoubi, A. Khandakar, R. Khallifa, R. Abouhasera, S. Koubaa, R. Ahmed, and A. Hasan, "Wearable real-time heart attack detection and warning system to reduce road accidents," *Sensors*, vol. 19, no. 12, p. 2780, 2019.

[14] M. E. Chowdhury, A. Khandakar, K. Alzoubi, S. Mansoor, A. M. Tahir, M. B. I. Reaz, and N. Al-Emadi, "Real-time smart-digital stethoscope system for heart diseases monitoring," *Sensors*, vol. 19, no. 12, p. 2781, 2019.

[15] M. Hosanee, G. Chan, K. Welykholowa, R. Cooper, P. A. Kyriacou, D. Zheng, J. Allen, D. Abbott, C. Menon, N. H. Lovell *et al.*, "Cuffless single-site photoplethysmography for blood pressure monitoring," *Journal of clinical medicine*, vol. 9, no. 3, p. 723, 2020.

[16] M. Elgendi, "On the analysis of fingertip photoplethysmogram signals," *Current cardiology reviews*, vol. 8, no. 1, pp. 14–25, 2012.

[17] M. Y.-M. Wong, C. C.-Y. Poon, and Y.-T. Zhang, "An evaluation of the cuffless blood pressure estimation based on pulse transit time technique: a half year study on normotensive subjects," *Cardiovascular Engineering*, vol. 9, no. 1, pp. 32–38, 2009.

[18] R. Shriram, A. Wakankar, N. Daimiwal, and D. Ramdasi, "Continuous cuffless blood pressure monitoring based on ptt," in *2010 International Conference on Bioinformatics and Biomedical Technology*. IEEE, 2010, pp. 51–55.

[19] C. P. Chua and C. Heneghan, "Continuous blood pressure monitoring using ecg and finger photoplethysmogram," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2006, pp. 5117–5120.

[20] Y. Kurylyak, F. Lamonaca, and D. Grimaldi, "A neural network-based method for continuous blood pressure estimation from a ppg signal," in *2013 IEEE International instrumentation and measurement technology conference (I2MTC)*. IEEE, 2013, pp. 280–283.

[21] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[22] Ü. Şentürk, İ. Yücedağ, and K. Polat, "Repetitive neural network (rnn) based blood pressure estimation using ppg and ecg signals," in *2018 2Nd international symposium on multidisciplinary studies and innovative technologies (ISMSIT)*. Ieee, 2018, pp. 1–4.

[23] X. Teng and Y. Zhang, "Continuous and noninvasive estimation of arterial blood pressure using a photoplethysmographic approach," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*, vol. 4. IEEE, 2003, pp. 3153–3156.

[24] G. Slapničar, N. Mlakar, and M. Luštrek, "Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network," *Sensors*, vol. 19, no. 15, p. 3420, 2019.

[25] S. Shimazaki, H. Kawanaka, H. Ishikawa, K. Inoue, and K. Oguri, "Cuffless blood pressure estimation from only the waveform of photoplethysmography using cnn," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 5042–5045.

[26] T. Athaya and S. Choi, "An estimation method of continuous non-invasive arterial blood pressure waveform using photoplethysmography: A u-net architecture-based approach," *Sensors*, vol. 21, no. 5, p. 1867, 2021.

[27] P. I. Aaronson, J. P. Ward, and M. J. Connolly, *The cardiovascular system at a glance*. John Wiley & Sons, 2020.

[28] J. L. Lapum, M. Verkuyl, W. Garcia *et al.*, *Vital Sign Measurement Across the Lifespan–1st Canadian Edition*. OpenStax Anatomy and Physiology, 2017.

[29] P. Brennan, G. Greenberg, W. Miall, and S. Thompson, "Seasonal variation in arterial blood pressure." *Br Med J (Clin Res Ed)*, vol. 285, no. 6346, pp. 919–923, 1982.

[30] N. Hjortskov, D. Rissén, A. K. Blangsted, N. Fallentin, U. Lundberg, and K. Søgaard, "The effect of mental stress on heart rate variability and blood pressure during computer work," *European journal of applied physiology*, vol. 92, no. 1, pp. 84–89, 2004.

[31] S. J. Kopp, J. T. Barron, and J. P. Tow, "Cardiovascular actions of lead and relationship to hypertension: a review." *Environmental Health Perspectives*, vol. 78, pp. 91–99, 1988.

[32] R. E. Klabunde. (2016) Cardiovascular physiology concepts. [Online]. Available: https://www.cvphysiology.com/Blood20Pressure/BP002

[33] O. Gauer, "Lehrbuch der physiologie des menschen," *Urban und Schwarzenberg München*, 1960.

[34] H. K. Walker, W. D. Hall, and J. W. Hurst, "Clinical methods: the history, physical, and laboratory examinations," 1990.

[35] E. A. Wehrwein and M. J. Joyner, "Regulation of blood pressure by the arterial baroreflex and autonomic nervous system," *Handbook of clinical neurology*, vol. 117, pp. 89–102, 2013.

[36] D. T. Lackland and M. A. Weber, "Global burden of cardiovascular disease and stroke: hypertension at the core," *Canadian Journal of Cardiology*, vol. 31, no. 5, pp. 569–571, 2015.

[37] S. Mendis, P. Puska, B. Norrving, W. H. Organization *et al.*, *Global atlas on cardiovascular disease prevention and control*. World Health Organization, 2011.

[38] I. Hernandorena, E. Duron, J.-S. Vidal, and O. Hanon, "Treatment options and considerations for hypertensive patients to prevent dementia," *Expert opinion on pharmacotherapy*, vol. 18, no. 10, pp. 989–1000, 2017.

[39] WorldHealthOrganization. (2022) Cardiovascular physiology concepts. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/hypertension

[40] P. Bonnafoux, "Auscultatory and oscillometric methods of ambulatory blood pressure monitoring, advantages and limits: a technical point of view." *Blood pressure monitoring*, vol. 1, no. 3, pp. 181–185, 1996.

[41] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, S. Rajan, and I. Batkin, "Oscillometric blood pressure estimation: past, present, and future," *IEEE reviews in biomedical engineering*, vol. 8, pp. 44–63, 2015.

[42] Y. Sun and N. Thakor, "Photoplethysmography revisited: from contact to noncontact, from point to imaging," *IEEE transactions on biomedical engineering*, vol. 63, no. 3, pp. 463–477, 2015.

[43] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiological measurement*, vol. 28, no. 3, p. R1, 2007.

[44] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, "Wearable photoplethysmographic sensors—past and present," *Electronics*, vol. 3, no. 2, pp. 282–302, 2014.

[45] E. Mejia-Mejia, J. Allen, K. Budidha, C. El-Hajj, P. A. Kyriacou, and P. H. Charlton, "Photoplethysmography signal processing and synthesis," in *Photoplethysmography*. Elsevier, 2022, pp. 69–146.

[46] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur, and H. Nazeran, "A review on wearable photoplethysmography sensors and their potential future applications in health care," *International journal of biosensors & bioelectronics*, vol. 4, no. 4, p. 195, 2018.

[47] A. Chauhan, K. Farmah, A. Goel, and A. Gandotra, "A novel patient monitoring system using photoplethysmography and iot in the age of covid-19," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2021, pp. 427–437.

[48] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[49] G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012.

[50] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[51] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[52] T. Dozat, “Incorporating nesterov momentum into adam,” 2016.

[53] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[54] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[55] D. Charles, M. Gabriel, and M. F. Furukawa, “Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2012,” *ONC data brief*, vol. 9, pp. 1–9, 2013.

[56] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, “Mimic-iii, a freely accessible critical care database,” *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[57] M. G. V. M. C. G. D. Moody, Benjamin and I. Silva., “Mimic-iii waveform database (version 1.0),” *PhysioNet*, 2020.

[58] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, “Cuffless blood pressure estimation algorithms for continuous health-care monitoring,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 4, pp. 859–869, 2016.

[59] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[60] T. Athaya and S. Choi, “Evaluation of different machine learning models for photoplethysmogram signal artifact detection,” in *2020 International conference on information and communication technology convergence (ICTC)*. IEEE, 2020, pp. 1206–1208.

[61] M. Blanco-Velasco, B. Weng, and K. E. Barner, “Ecg signal denoising and baseline wander correction based on the empirical mode decomposition,” *Computers in biology and medicine*, vol. 38, no. 1, pp. 1–13, 2008.

[62] T. Ji, Z. Lu, Q. Wu, and Z. Ji, “Baseline normalisation of ecg signals using empirical mode decomposition and mathematical morphology,” *Electronics letters*, vol. 44, no. 2, p. 1, 2008.

[63] S.-J. Baek, A. Park, Y.-J. Ahn, and J. Choo, "Baseline correction using asymmetrically reweighted penalized least squares smoothing," *Analyst*, vol. 140, no. 1, pp. 250–257, 2015.

[64] P. Gupta, K. K. Sharma, and S. D. Joshi, "Baseline wander removal of electrocardiogram signals using multivariate empirical mode decomposition," *Healthcare technology letters*, vol. 2, no. 6, pp. 164–166, 2015.

[65] B. N. Singh and A. K. Tiwari, "Optimal selection of wavelet basis function applied to ecg signal denoising," *Digital signal processing*, vol. 16, no. 3, pp. 275–287, 2006.

[66] N. Ibtehaz and M. S. Rahman, "Ppg2abp: Translating photoplethysmogram (ppg) signals to arterial blood pressure (abp) waveforms using fully convolutional neural networks," *arXiv preprint arXiv:2005.01669*, 2020.

[67] S. Mahmud, N. Ibtehaz, A. Khandakar, A. M. Tahir, T. Rahman, K. R. Islam, M. S. Hossain, M. S. Rahman, F. Musharavati, M. A. Ayari *et al.*, "A shallow u-net architecture for reliably predicting blood pressure (bp) from photoplethysmogram (ppg) and electrocardiogram (ecg) signals," *Sensors*, vol. 22, no. 3, p. 919, 2022.