University of Padova

Department of Information Engineering

Master Thesis in Telecommunication Engineering

A Deep Learning-Based ECG Delineator:

Evaluation and Comparison on Standard Databases

Supervisor

Michele Rossi

University of Padova

Co-supervisor

Matteo Gadaleta

University of Padova

Master Candidate

Umberto Cappellazzo

Academic Year 2018/2019

Abstract

The ECG signal, which represents the electrical activity of the heart, is an essential tool to monitor the status of the heart and evaluate its condition. Several algorithms have been proposed for the automatic detection of the ECG characteristic waves, namely P wave, T wave and QRS complex, with particular focus on the localization of the R peaks. In fact, the R peak conveys very important and diagnostic valuable information, such as the heart rate, the heart rate variability, and it is the main feature used for arrhythmia detection. This Thesis aims to leverage the standard Convolutional Neural Network (CNN) to propose a new Deep Learning-based ECG delineator for the individuation of the P, R and T peaks.

The processing pipeline of my work will be investigated, made up of a pre-processing stage, a learning framework and a post-processing phase. My algorithm will be compared with other state-of-the-art techniques on standard databases. In order to assess the resilience to noise of the algorithms, more and more noisy scenarios will be taken into consideration.

Contents

Listing of figures

Listing of tables

1 Introduction

ECG signals are reflective of electric activities of the heart muscle. They are related to a variety of intertwined and complex chemical, electrical, and mechanical processes present in the heart. They convey a great deal of valuable diagnostic information, not only describing functioning of the heart, but also other systems such as circulation or nervous systems [1]. A characteristic feature of ECG signal comes with a cyclic occurrence of its components, consisting of the P-QRS-T complex. The ECG signal in this region is reflective of the performance of the conduction system of the heart that pertains to a single heart evolution, involving contraction of atria and ventricles. This part of the ECG signal forms a region of interest (ROI) as it contains the most essential diagnostic information. During ECG signal processing and analysis, an important task is to detect the onset, peak and offset locations of the P wave, QRS complex and T wave. In particular, an accurate R peak detection is essential in signal processing equipment for heart rate measurement and it is the main feature used for arrhythmia detection.

ECG signal are one of the best-known biomedical signals. Given their nature, they bring forward a number of challenges during their registration, processing and analysis. Characteristic properties of biomedical signals include their non-stationarity, noise susceptibility, and variability among individuals. In particular, a very undesirable aspect of these signals is their high susceptibility to various types of noise. The main sources of noise include:

- Changes of the body-electrode impedance (baseline wandering);

- Changes of a mutual position of the heart and electrodes caused by respiratory movements of a patient;
- Contraction of skeletal muscles;
- Interference of electromagnetic nature;
- Interferences caused by high power devices.

The ECG signal is one of the best-recognized biomedical signals. Its high diagnostic capabilities have been demonstrated. In the last decades there has been a significant growth of interest in the development of efficient methods of processing and analysis of ECG signals with the intent to create useful diagnostic information. The first proposed algorithms were derivative-based and filtering-based approaches [4], [5]. Later on, several different other approaches have been investigated, based on Hilbert trasnform [6], Hidden Markov Models [8], mathematical morphology [11], Phasor transform [12]. Anyways, the most adopted approach is that based on the Wavelet Transform [14], [15]. The effectiveness of the Wavelet Transform relies on its intrinsic property of describing the ECG signal at different scales.

In this thesis, a new Deep Learning-based delineator has been implemented and discussed. The processing pipeline of my algorithm is divided into three parts. The first one is the so-called *pre-processing* stage. Through it, the raw ECG signal is processed. In particular, the signal is filtered with a specific band-pass filter in order to remove the baseline wandering, as well as scaled.

The second step is the core of the work: the *learning framework*. It is constituted by a Convolutional Neural Network (CNN) with 6 convolutional layers for the P and T peaks' delineation and with 4 convolutional layers for the R one, respectively. The CNN proposed in my work is different from the standard one since it does not perform classification or regression, hence only the feature learning block is present. In fact, the output of my CNN has the same size of the input. This entails that the network does not depend on the input's length and so I can give in input to the CNN signals with different sizes. The third and last step is the *post-processing* stage. Here, a sigmoid operation is applied to the output of the CNN and, then, a peak detector is used so as to find out the locations of the peaks.

My algorithm will be compared with other classical approaches in terms of the F_1 score and the statistics of the error (i.e., the difference between the real locations of the peaks and the ones obtained by one specific algorithm) on standard databases. Moreover, the algorithms will be tested in more and more noisy scenarios so that we will be able to evaluate their robustness against noise.

The thesis is organized as follows:

1. Chapter 2 gives a brief description of the electrical conduction system of the heart, as well as of the goals of the ECG delineation;

2. Chapter 3 is devoted to giving an outline of the most common algorithms for ECG delineation;

3. Chapter 4 describes in details each step of my proposed processing pipeline: pre-processing stage, learning framework and post-processing step;

4. Chapter 5 describes the algorithms for the comparison, the metrics used for the comparison, the simulation scenarios; the final results are showed by means of tables, boxplots and figures;

5. Finally, Chapter 6 draws the conclusions and proposes some guidelines for future works based on my work.

2 Preliminaries: the Electrical Conduction System of the Heart and ECG Interpretation/Delineation

2.1 Introduction

Electrocardiography is the process of producing an **electrocardiogram** (ECG or EKG), a recording of the electrical activity of the heart by means of electrodes placed on the skin. The ECGs are usually printed on a grid where the horizontal axis represents time and the vertical axis represents voltage. The electrodes detect the changes in electrical potential difference (voltage) during depolarization and repolarization [1] of the myocardial fibers during each cardiac cycle. The sources of the electrical potentials are contractile cardiac muscle cells (*cardiomyocytes*).

The ECG is used to investigate some types of abnormal heart function including arrhythmias and inadequate coronary artery blood flow (e.g., myocardial ischemia and myocardial infarction), as well as hearth morphology (such as the orientation of the heart in the chest cavity

[1] In biology, *depolarization* is a change within a cell, during which the cell undergoes a shift in electric charge distribution, resulting in less negative charge inside the cell. On the contrary, the *repolarization* refers to the change in membrane potential that returns it to a negative value just after the depolarization phase.

and the hypertrophy). Furthermore, it is useful for gauging the performance of pacemakers.

2.2 The Electrical Conduction System of the Heart

The **electrical conduction system of the heart** transmits signals usually generated by the sinoatrial node to cause contraction of the heart muscle. The pacemaking signal generated in the sinoatrial node travels through the right atrium to the atrioventricular node, along the bundle of His and through bundle branches to bring about contraction of the heart muscle. This signal firstly stimulates the contraction of the right and left atrium, and then the right and left ventricles. This process permits blood to be pumped all over.
Cardiac muscle consists of two main cell types: *cardiomyocytes*, which generate electrical potentials during contraction and make up the atria and the ventricles, and *cardiac pacemaker cells*, specialized in the generation and conduction of the electrical impulses and distributed throughout the heart. These specialized electrical cells depolarize spontaneously. At rest, cardiomyocytes are polarized with an electrical membrane potential of around -90 mV. Excitation by an external stimulus can trigger a rapid reversal of the electrical potential of working myocardial cells (*depolarization*). The depolarization is usually due to a sudden increase in permeability of the membrane to sodium. The downward swing of the action potential, or *repolarization* phase, is mainly due to the movement of potassium ions out of the cell. After depolarization, the muscle returns to its original electrical state. During the repolarization, the cardiac muscle is incapable of being stimulated (e.g., it is refractory), which protects it against premature activation.

The conduction system of the heart is depicted in Fig. 2.1. The *sinoatrial node* (S-A) has the highest rate of spontaneous depolarization and acts as the primary pacemaker. Its activity is regulated by the autonomic nervous system (ANS). At normal condition, the S-A node generates impulses that stimulate the atria to contract. This node is located in the superior wall of the right atrium, close to the opening of the superior vena cava. Other elements of the conduction system include the *atrioventricular node* (A-V), located between the atria and the ventricles, in the lower atrial septum adjacent to the annulus of the mitral valve, and the *bundle of His*. The bundle of His divides into a right and left branch at the level of membranous part of the interventricular septum. The left branch is further branched into an anterior and posterior bundle. The *Purkinje fibers* are the final component of the conduction system, which are intertwined with muscle fibers and papillary muscles. Their task is to conduct the wavefronts directly to the two ventricles so that they contract simultaneously. The Purkinje

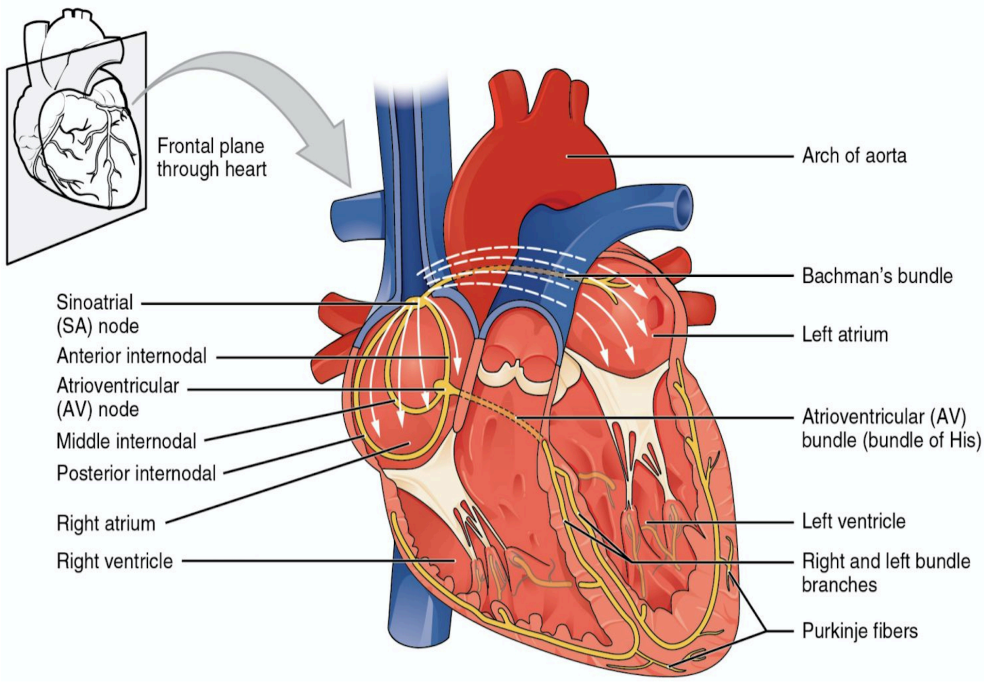

Figure 2.1: The electrical conduction system of the heart, from [36].

fibers have intrinsic automaticity (ventricular escape rhythm) generating approximately 30 bpm (beats per minute). The cells of the A-V node also depolarize spontaneously but at a higher rate (about $40 - 50$ bpm).

Using the terminology associated with electrical devices, the conduction system of the heart can be described as a pacemaker (S-A node), a resistor that simultaneously acts like a fuse (the A-V node) and two insulated electrical wires (branches of the bundle of His). The term "resistor" for the property of the A-V node is appropriate since it slows down the depolarization (conduction velocity through the A-V node is slower than in other parts of the conducting system, 0.05 m/s vs. 4 m/s, respectively). This delay enables the transfer of blood from the atria to the ventricles and is responsible for ensuring that the sequence of ventricular contraction follows atrial contraction.

The comparison between the A-V node and a "fuse" is appropriate because the A-V node possesses Wenckebach's point, which is thought to maintain the ratio of maximum conduction of the supraventricular impulses to ventricles at $1:1$.

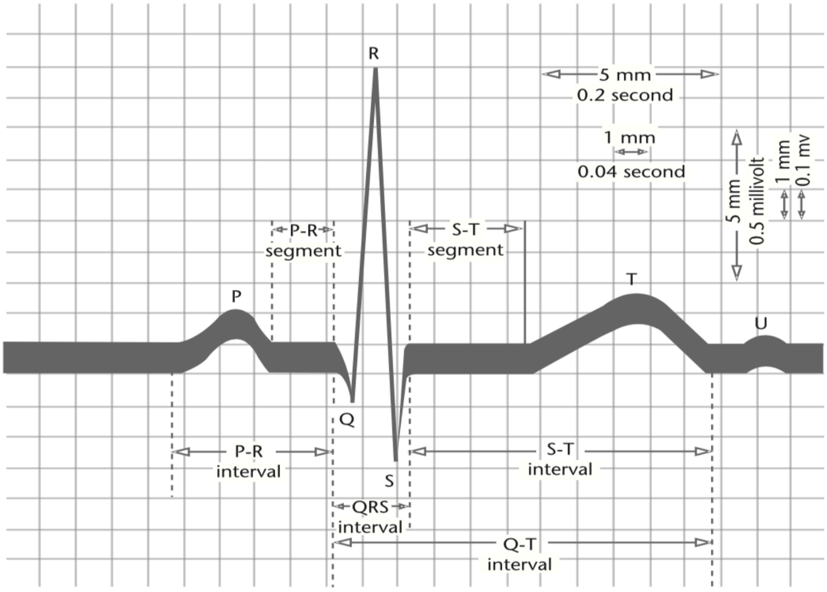

Figure 2.2: ECG of a heart in normal sinus rhythm, from [2].

2.3 ECG Delineation

A normal rhythm of the heart produces an ECG waveform that is akin to the one showed in Fig. 2.2. The goal of ECG delineation is that of evaluating the onset and ending points of all the P-waves, T-waves and QRS complexes in a specific ECG trace, as well as the position of the peaks of the above-mentioned waves. A normal ECG waveform shows some typical morphologies:

- The *isoelectric line*: a horizontal line when there is no electrical activity on the ECG;
- *Segments*: the time duration of the isoelectric line between waves;
- *Intervals*: the time duration between the beginning of a wave and the beginning of a subsequent wave (e.g., PR interval) or the duration between two consecutive equal peaks (e.g., RR interval).

2.3.1 Specification of Waves, Intervals and Segments of an ECG Waveform

In this section, the description of the waves and the intervals between them on an ECG tracing will be given. It is possible to visualize them in Fig. 2.3.

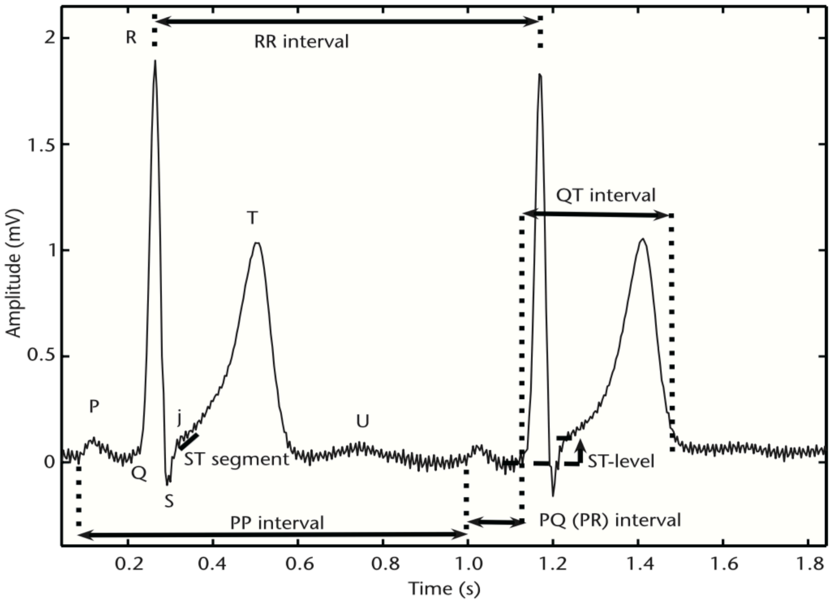

Figure 2.3: Standard fiducial points in the ECG (P, Q, R, S, T and U) together with clinical features, from [2].

- **P wave** is the first deflection of the ECG and it represents the depolarization of the atria. Atrial depolarization spreads from the SA node towards the AV node, and from the right atrium to the left atrium. The normal duration of the P wave is no longer than 120 ms. An abnormality of this wave can be due to atria enlargement, hypertrophy or ectopic atrial pacemaker;

- **PR interval** is measured from the beginning of the P wave to the beginning of the QRS complex. This interval reflects the time the electrical impulse takes to travel from the SA node through the AV node.The standard span of time is $120 - 200$ ms. A PR interval shorter than 120 ms suggests that the electrical impulse is bypassing the AV node, as in Wolf-Parkinson-White syndrome. A PR interval consistently longer than 200 ms diagnoses first degree atrioventricular block;

- **QRS complex** is the largest group of waves on the ECG and represents the depolarization of the right and left ventricles via the bundle of His and Purkinje fibers. The ventricles have a large muscle mass compared to the atria, hence the QRS complex usually has a much larger amplitude than the P wave. The normal duration of this wave is $80 - 100$ ms. If the QRS complex is longer than 120 ms, it suggests disruption of the heart's conduction system or ventricular rhythms such as ventricular tachycardia. An unusually tall QRS complex may represent left ventricular hypertrophy, while a very low-amplitude QRS complex may represent a pericardial effusion or infiltrative myocardial disease;

- **ST segment** connects the QRS complex and the T wave. It represents the period when the ventricles are depolarized. It is usually isoelectric, but may be depressed or elevated with myocardial infarction (MI) or ischemia;

- **T wave** represents the repolarization of the ventricles. It is generally upright in all leads except aVR and V1. Inverted T waves can be a sign of myocardial ischemia, left ventricular hypertrophy, high intracranial pressure, or metabolic abnormalities. Peaked T waves can be a sign of hyperkalemia or very early myocardial infarction;

- **QT interval** is measured from the onset of the QRS complex to the end of the T wave. The length of the QT interval is directly affected by the heart rate (the slower the heart rate, the longer the interval). To eliminate this influence, a corrected QT (QTc) should be worked out using either Bazett's formula ($QTc = \frac{QT}{\sqrt{RR}}$) or Fridericia's formula ($QTc = \frac{QT}{RR^{0.33}}$), where QT is the length of the QT interval in seconds and RR is the cardiac cycle in seconds. The disadvantage is that these formulas overestimate the QT interval when the heart rate is fast and underestimate it when the heart rate is slow. Normally the duration of the QT interval is less than 440 ms. A prolonged QTc interval is a risk factor for ventricular tachyarrhythmias and sudden death. Long QT can arise as a genetic syndrome, or as a side effect of certain medications. An unusually short QTc can be seen in severe hypercalcemia;

- **U wave** is hypothesized to be caused by the repolarization of the interventricular septum. It normally has a low amplitude, and even more often is completely absent.

In Table 2.1 it is possibile to see some typical values of the intervals and amplitudes of the waves under specific characteristics.

Table 2.1: Typical Lead II ECG Features and Their Normal Values in Sinus Rhythm at a Heart Rate of 60 bpm for a Healthy Male Adult.

$Feature$	$Normal\ Value$	$Normal\ Limit$
P width	110 ms	±20 ms
PQ/PR interval	160 ms	±40 ms
QRS width	100 ms	±20 ms
QTc interval	400 ms	±40 ms
P amplitude	0.15 mV	±0.05 mV
QRS height	1.5 mV	±0.5 mV
ST level	0 mV	±0.1 mV
T amplitude	0.3 mV	±0.2 mV

Note: There is some variation between lead configurations. Heart rate, respiration patterns, drugs, gender, diseases, and ANS activity also change the values. QTc = αQT, where $\alpha = (RR)^{-\frac{1}{2}}$. About 95% of (normal healthy adult) people have a QTc between 360 ms and 440 ms. Female durations tend to be approximately 1% to 5% shorter except for the QT/QTc, which tends to be approximately 3% to 6% longer than for males. Intervals tend to elongate with age, at a rate of approximately 10% per decade for healthy adults.

3

State of the Art: a Survey of the Most Common Algorithms for ECG Delineation

3.1 Introduction

This chapter is devoted to giving an outline of the most-adopted techniques for the ECG-delineation problem. As the time went by, new approaches have been proposed in order to outperform the existing algorithms. It worths noticing that, albeit some methods have been contrived quite a long time ago, their performances are very competitive and wrangle with the most recent one. The most-used methods rely on the wavelet transform (sometimes in combination with other techniques) due to its intrinsic property of describing the signal at different scales. Truly, I will tarry mainly on some works related to it. Nevertheless, other disparate techniques will be addressed so as to grant a thorough insight and show how much this task has been dissected. Note that some of the following algorithms aim at only individuating the QRS complex, which is the most important wave of an ECG signal and provides much information about the current state of the heart.

3.2 Derivative-Based Algorithms

The derivative-based algorithms are the oldest and most intuitive approaches. They are often used in real-time analysis or for large datasets since they do not require extensive computations. These methods also have the advantage of not necessitating manual segmentation of data, training of the algorithms, or patient-specific modifications that are usually required for other detection methods. Among these algorithms, one of the most famous is surely that proposed by Pan and Tomkins in the 1985 [4]. It is a real-time QRS detector, thus it provides an incomplete delineation. This algorithms is composed of three processing steps:

1. *Linear digital filtering*, which includes a bandpass filter (used for attenuating noise), a derivative (which provides information about the slope of the QRS), and a moving window integrator (it produces a signal that includes information about both the slope and the width of the QRS complex);

2. *Non-linear transformation*, in particular signal amplitude squaring, which intensifies the slope of the frequency response curve of the derivative and helps restrict false positives caused by T waves with higher than usual spectral energies;

3. *Decision rule algorithm*, performed by adaptive thresholds and T-wave discrimination techniques.

Once the processing procedure is completed, the algorithm is divided into three processes: learning phase 1, learning phase 2, and detection. Learning phase 1 requires about $2s$ to initialize detection thresholds based upon signal and noise peaks detected during the learning process. Learning phase 2 requires two heartbeats to initialize RR-interval average and RR-interval limit values. The subsequent detection phase does the recognition process and produces a pulse for each QRS complex. The thresholds and other parameters of the algorithm are adjusted periodically to adapt to changing characteristics of the signal.

The crucial part of this algorithm is the choice and adaptation of the thresholds. They use two sets of thresholds: one set thresholds the filtered ECG, and the other thresholds the signal produced by moving window integration. The usage of two sets of thresholds improves the reliability of the detection. Moreover, thanks to the initial bandpass filtering, the signal-to-noise ratio (SNR) is increased, thereby letting the use of lower thresholds. The algorithm uses a dual-threshold technique to find missed beats. There are two separate threshold levels in each of the two sets of thresholds. One level is half of the other. The thresholds continuously adapt to the characteristics of the signal since they are based upon the most-recent signal and noise peaks that are detected in the ongoing processed signals.

This algorithm has been evaluated on the MIT/BIH arrhythmia database [38] (it was the best database available in the middle of 80s), consisting of 48 half-hour recordings for a total of 24 h of ECG data. The total detection error rate achieved by the algorithm is 0.675 %, the

sensitivity S_e is equal to 99.75 and the positive predictivity is equal to 99.54. This algorithm fails in specific records that display unusually tall, peaked P waves, or non-QRS waves with highly unusual morphologies.
As a final remark, in general the derivative-based methods achieve very good results in terms of accuracy; yet, they show quite high time errors.

3.2.1 Low-Complexity Algorithm for Real-Time Embedded Systems

More recently, Bote et al. [13] have presented a new modular and *low-complexity* algorithm that provides a complete ECG delineation. Involving a reduced number of operations per second and having a small memory footprint, the algorithm is intended for real-time delineation on resource-constrained embedded systems. The modular design allows the algorithm to automatically adjust the delineation quality in runtime to a wide range of modes and sampling rates, from a ultralow-power mode when no arrhythmia is detected, to a complete high-accuracy delineation mode.

The standard approaches based on Wavelet Transform (e.g., [14], [15]) involve high computational cost due to its complexity. If the delineation is executed off-line or without any restriction of time, they can be good approximations to medical annotations. However, the increasing use of miniaturized embedded wearable devices for real-time health monitoring and diagnosis, is driving the need for more efficient algorithms. Currently, the wearables market is steadily growing up, developing new devices that are able to monitor heart rate (HR) 24/7, using GPS for running or monitoring sleep patterns and quality. Fitness trackers or smartwatches can be used to record and/or analyze real time signals without anything but a sensor mounted on the device or a chest strap. The proposed algorithm for real-time ECG delineation helps to enable the next-generation of wearable monitoring systems, that can rely on the delineation results to perform online on-board arrhythmia detection.

The algorithm in [13] proposes a new delineation algorithm that is not only less computationally complex and more energy efficient than state-of-the-art techniques, but also much more flexible and modular, being able to always achieve an optimal trade-off between energy consumptions and delineation accuracy, depending on the available energy budget and the required performance, to carry out a precise arrhythmia detection.

The simple yet accurate delineation algorithm is based on the first and second derivative of a low-pass FIR filtered ECG signal whose cut-off frequency is 14 Hz. Noise reduction, using another low-pass FIR filter with frequency of 40 Hz, and baseline wander removal, applying different morphological filters, are also considered.

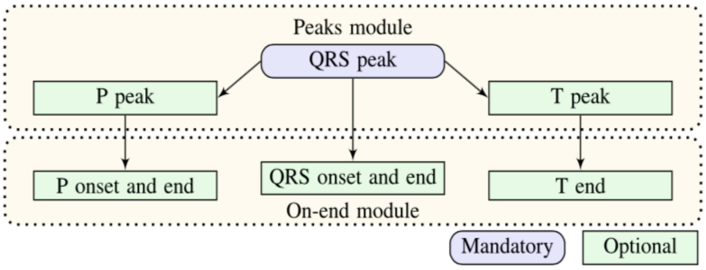

Figure 3.1: Conceptual flowchart of the ECG delineation, from [13].

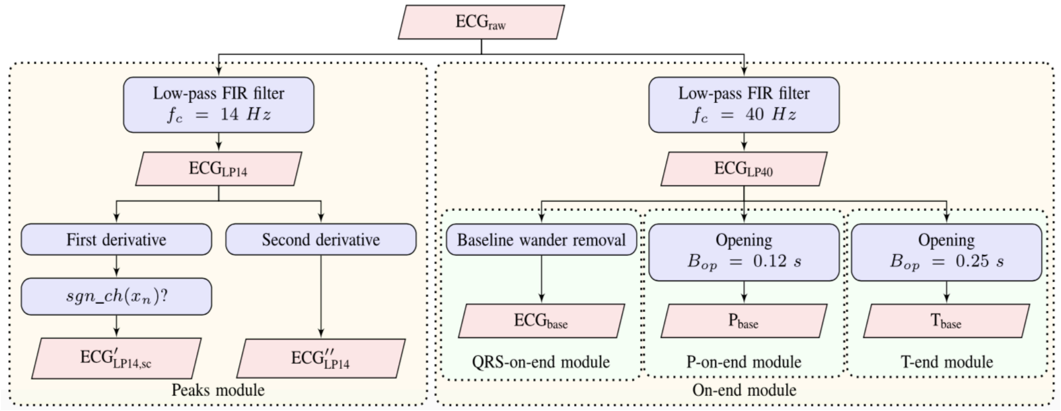

Figure 3.2: Flowchart of the proposed modular ECG signal processing, from [13].

The conceptual flowchart of the ECG delineation can be viewed on Fig. 3.1 and schematized as follows:

1. The delineation algorithm begins with a QRS peak detection;

2. Once a QRS peak has been detected, its boundaries (onset and end) can be found, if needed, before and after that QRS peak (this defines the QRS complex);

3. Also P and T peaks may be searched before and after that QRS complex, with their onset and end.

In Fig. 3.2 it is possible to see the *Peaks* and *On-end* modules. The purpose of the *Peaks module* is to detect the QRS peank and, also, the P and T peaks if needed. In order to do this, a low-pass FIR filter with a cut-off frequency of 14 Hz is applied to the ECG_{raw} signal. With this aggressive filter, any type of high-frequency signal is eliminated, while P, QRS and T wave peak positions are preserved. To detect the peak of the QRS complex, the first and

Table 3.1: Delineation performance of the algorithm in [13].

Mode	*High-accuracy*	*Standard*	*Low-power*	*Ultra-low-power*
Features	High frequency High-order FIR filters	High frequency Low-order FIR filters	Low frequency Continuous sampling	Low frequency Discontinuous sampling
Detectable waves	QRS, P, T peaks, onsets, ends	QRS, P, T peaks, onsets, ends	QRS peak	QRS peak
Sampling rate (Hz)	250	250	50	50
Filter order	40	10	10	10
Computational cost	Medium-high	Medium	Low	Ultra-low
Accuracy	Ultra-high	High	Medium-high	Medium
QRS peak delineation				
Se - PPV (%)	$100 - 99.51$	$99.50 - 99.78$	$97.27 - 98.27$	$95.38 - 98.78$
$m \pm \sigma(ms)$	-6.0 ± 4.6	-5.5 ± 6.6	20.47 ± 24.46	22.15 ± 26.45

second derivatives are computed. The details of the method can be consulted by the reader in [13].

In parallel to the *Peaks module*, the *On-end module* can be enabled for detecting onsets and ends of the previously detected ECG waves. A less aggressive low-pass FIR filter is applied to ECG_{raw} with a cut-off frequency of 40 Hz.

The algorithm has been tested on the QT database [37]. The main advantage of the proposed technique with respect to previous works is its ability to operate in different modes, that can be selected dynamically at run time, depending on which ECG fiducial points have to be detected and how accurate the detection must be, and therefore drastically reducing the CPU usage. There are four possible modes:

- *High-accuracy*;
- *Standard*;
- *Low-power*;
- *Ultra-low-power*.

The results obtained by the algorithm on a specific mode and its specifications can be found on Table 3.1.

3.3 Hilbert Transform-Based Algorithms

In [6], a new approach to QRS detection using the properties of the Hilbert transform is presented. In a nutshell, the algorithm exploits the first differential of the ECG signal and its Hilbert transformed data to locate the R peaks in the ECG waveform. The combinational use of both signals has a number of advantages, such as the minimization of unwanted effects of large peaked T and P waves and the excellent performance of the algorithm under significant noise contamination. Now, some definitions and properties of the Hilbert transform at the base of the algorithm are presented.

Given a real time function $x(t)$, its Hilbert transform is defined as:

$$\hat{x}(t) = H[x(t)] = \frac{1}{\pi}\int_{-\infty}^{+\infty} x(\tau)\frac{1}{t-\tau}d\tau \tag{3.1}$$

It is known that $\hat{x}(t)$ is also a time dependent function and a linear function of $x(t)$. It holds that:

$$\hat{x}(t) = \frac{1}{\pi t} * x(t) \tag{3.2}$$

Furthermore, the Fourier transform of the Hilbert transform of $x(t)$ may be re-expressed as:

$$F\{(\hat{x})\} = -j\,\mathrm{sgn}\,f \cdot F\{x(t)\} \tag{3.3}$$

Therefore, the Hilbert transform of the original function $x(t)$ represents its harmonic conjugate.

A very useful property of the Hilbert transform is that it is an odd function. This is equivalent to say that it will cross zero on the x-axis every time that there is an inflexion point in the original waveform. Similarly, a crossing of the zero between consecutive positive and negative inflexion points in the original waveform will be represented as a peak in its Hilbert transformed conjugate.This interesting property is used by [6] to develop an elegant and much easier way to find the peak of the QRS complex in the ECG waveform corresponding to a zero crossing in its first differential waveform d/dt (ECG). The proposed approach is depicted in Fig. 3.3. It is worthwhile to underline that the implementation of the last block (i.e., the threshold peak detector) is quite simple because the P and T waves are minimized in relation to the relative peak corresponding to the peak of the QRS complex in the Hilbert sequence. The threshold must be adaptive in order to guarantee accurate detection of the R peaks.

The detector has been tested using entire records from the MIT-BIH Arrhythmia database

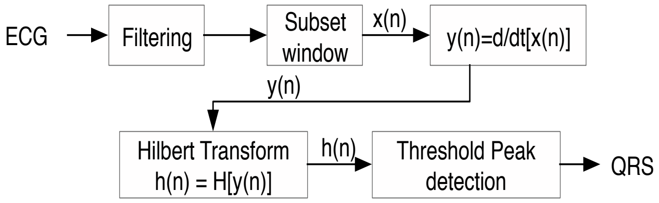

Figure 3.3: Block diagram of the QRS detector, based on [6].

[38]. The performances achieved by the detector are very good. In particular, it works well as the signals are corrupted by pronounced muscular and baseline artifacts (MIT-BIH Noise Stress Test Database). A QRS detection error rate of 0.36%, a sensitivity (Se) of 99.81% and a positive prediction (P^{+}) of 99.83% were achieved against the MIT-BIH database.

3.4 Phasor Transform-Based Algorithms

The algorithm proposed by Rodrigo et al. in [12] is based on the Phasor Transform (PT) and it provides a overall delineation of the ECG signal (P and T waves, QRS complex), operating in single lead recordings. The most important feature of this algorithm is that it is a fast algorithm rather than the Wavelet Transform-based ones, as well as very accurate. The most important steps of this method follow, referring the reader to [12] for a more detailed description.

The *Phasor Transform* (PT) is a tool able to represent a sinusoidal function in the complex domain. The result is a complex number, called the phasor, which preserves the signal information regarding root mean square and phase values. For a generic discrete sinusoid such as $x[n] = A\cos(\omega n + \varphi) = \Re\{Ae^{j(\omega n+\varphi)}$, being A the amplitude and φ the phase of the sinusoid, its PT would provide a rotating phasor in the complex plane with magnitude A, rotation speed ω and initial phase φ, i.e., $PT\{x[n]\} = Ae^{j\varphi} = A\cos(\varphi) + jA\sin(\varphi)$.

To enhance the ECG waves, PT is used to convert each instantaneous ECG sample into a phasor. A constant value R_v is considered as the real part, whereas the original value of the ECG is used as the imaginary component of the phasor. Thus, if $x[n]$ denotes an ECG recording of N samples, being n the discrete time, the phasor $y[n]$ can be defined for each sample as $y[n] = R_v + jx[n]$, for $n = 1, \ldots, N$. Let $M[n]$ and $\varphi[n]$ be the magnitude and the phase of the phasor, respectively. In this way, by considering the instantaneous phase variation in consecutive samples of the phasor transformed ECG, the slight variations provoked by P and T waves in the original recording are maximized, regardless of their eventually low

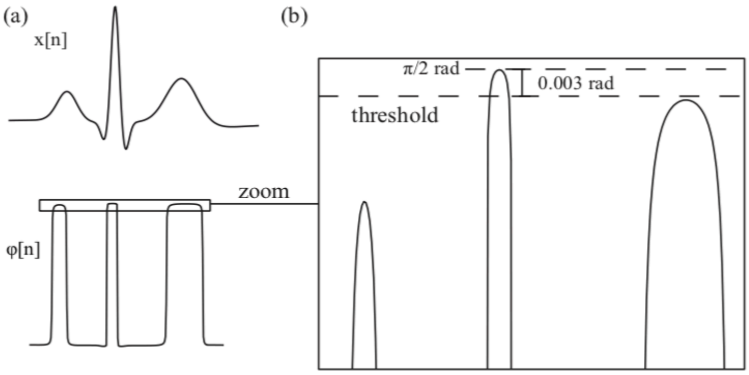

Figure 3.4: (a) Representation of a normal beat from a typical ECG (top) together with its phase variation obtained from the phasor transform (bottom). (b) Detailed representation of the phasor transformed signal $\varphi[n]$ to illustrate the process for R peak identification. Image taken from [12].

amplitude. The value of R_v determines the degree with which ECG waves are enhanced into the phasorial signal and this value is calculated experimentally for each wave by using the QT database as learning set.

QRS complexes are detected by applying directly the PT to the absolute value of the preprocessed ECG, with $R_v = 0.001$. As can be seen in the lower part of Fig. 3.4 (a), P and T waves are notably enlarged by the PT operation. By establishing a threshold of 0.003 rad below the maximum phase variation of $\pi/2$ rad (see Fig. 3.4 (b)), the QRS complexes can be detected as the segments that exceed the threshold.

Once the R peak is detected, it is used as a reference for the identification of Q and S waves. For both waves, a window of 35 ms is considered, and the PT is newly applied to the absolute value of the ECG, subtracting the median of the segment. R_v is set to 0.005. The local minimum of φ is searched inside the window: if any point presents a phase higher than 75% of the maximum variation of $\pi/2$ rad, that is the Q or S peak. The delineation of T and P waves proceeds in a similar way with some additional gimmicks.

The method has been tested on the QT database [37]. In Table 3.2 the results obtained from the proposed method are showed. The performances are very similar to the most significant single-lead delineators, keeping in mind that this algorithm presents a low computational cost and a mathematical simplicity. Note that the standard deviation (s in the table) for P wave is quite high.

Table 3.2: Delineation performance of the algorithm in [12].

$Feature$	Pon	$Ppeak$	$Pend$	$QRSon$	$QRSend$	$Tpeak$	$Tend$
Se (%)	98.65	98.65	98.65	99.85	99.85	99.20	99.20
$m \pm s(ms)$	2.6 ± 14.5	32 ± 25.7	0.7 ± 14.7	-0.2 ± 7.2	2.5 ± 8.9	5.3 ± 12.9	5.8 ± 22.7

3.5 Mathematical Morphology-Based Algorithms

A mathematical morphology-based QRS detector has been introduced for the first time in 1993 from [10]. The work proposed by [11] describes an algorithm that carry out a complete ECG-delineation, thus it is more interesting to be investigated.

The algorithm in [11] is a multiscale morphological derivative (MMD) transform-based technique which operates on a single lead (in particular the ECG lead II). As a nonlinear filtering, it has been proven that morphological dilation and erosion satisfy the causality and the additive semigroup property required by multiscale analysis for signal of any dimension with local maxima and minima as singular points (peak, onset and offset of P, T and QRS waves for ECG signal). By applying a morphological derivative transform defined at different scales, noise sensitivity inherent in single scale operation can be reduced in MMD method. Moreover, the problem of position deviation existed in wavelet transform-based techniques can be avoided due to the nonlinearity of morphological transform.

In [11], the signal to be processed is limited to continuous function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ with only finite oscillations on a closed interval which is differentiable everywhere except at some singular points. A singular point in the one-dimensional signal is defined as a point whose derivatives on the right and the left exist with different signs. The derivative on the right can be represented by morphological sup-derivative M_f^+, which is defined as:

$$M_f^+(x) = \lim_{s \to 0} \frac{(f \oplus g_s)(x) - f(x)}{s} \quad . \tag{3.4}$$

Analogously, the derivative on the left is established as:

$$M_f^-(x) = \lim_{s \to 0} \frac{f(x) - (f \ominus g_s)(x)}{s} \quad . \tag{3.5}$$

Denoting the functions, $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_s : G_s \subset \mathbb{R}^n \rightarrow \mathbb{R}$ $(s > 0)$, the two fundamental operations of multiscale morphology are:

$$Dilation : (f \oplus g_s)(x) = \sup_{t \in (G_s \cap D_x)} \{f(x-t) + g_s(t)\} \tag{3.6}$$

$$Erosion : (f \ominus g_s)(x) = \inf_{t \in (G_s \cap D_x)} \{f(x+t) - g_s(t)\}, \tag{3.7}$$

where D_x is the translation of D, $D_x = x + t : t \in D, \sup(f)$ and $\inf(f)$ refer to the supremum (least upper bound) and infimum (greatest lower bound) of f, s is scale, and g_s is the scaled structuring function.

The authors propose a multiscale morphological derivative difference M_f^d, defined as:

$$M_f^d(x) = M_f^+(x) - M_f^-(x) = \lim_{s \to 0} \frac{(f \oplus g_s)(x) - f(x)}{s} - \lim_{s \to 0} \frac{f(x) - (f \ominus g_s)(x)}{s} = \lim_{s \to 0} \frac{(f \oplus g_s)(x) + (f \ominus g_s)(x) - 2f(x)}{s} \quad . \tag{3.8}$$

By choosing a flat structuring function, where $g_s(x) = 0, x \in G$, where $G = \{x : ||x|| \leq s\}$, the above multiscale morphological derivative transform described by Equation (6) is simplified to the following process: choose a moving window with a length of $(2s + 1)$ samples and find the maximum and minimum values in the window, as well as the value of the signal at the central point $f(x)$. Then, the MMD transform at the central point can be specified as:

$$M_f^{d_s}(x) = \frac{\max_{t \in [x-s, x+s]}\{f(t)\} + \min_{t \in [x-s, x+s]}\{f(t)\} - 2f(x)}{s} \quad . \tag{3.9}$$

At a positive peak in the ECG signal, its left derivative is positive and its right derivative is negative, therefore, positive peaks in the ECG signal correspond to the local minima in $M_f^{d_s}$. At the onset or offset of a positive peak, there is an abrupt increase in its derivative value from left to right. So, the onsets and offsets correspond to the local minima in $M_f^{d_s}$. As applied to ECG lead II signal, the R peak, Q wave and S wave correspond to the local minima of the $M_f^{d_s}$, while the onsets and offsets of the P and T waves correspond to the local maxima of the $M_f^{d_s}$. Hence, the characteristic QRS complex, P and T waves, can be detected using the just-described MMD detector by detecting the local extrema in the MMD transformed signal, setting specific thresholds as well. In addition to this, in order to reduce the noise and

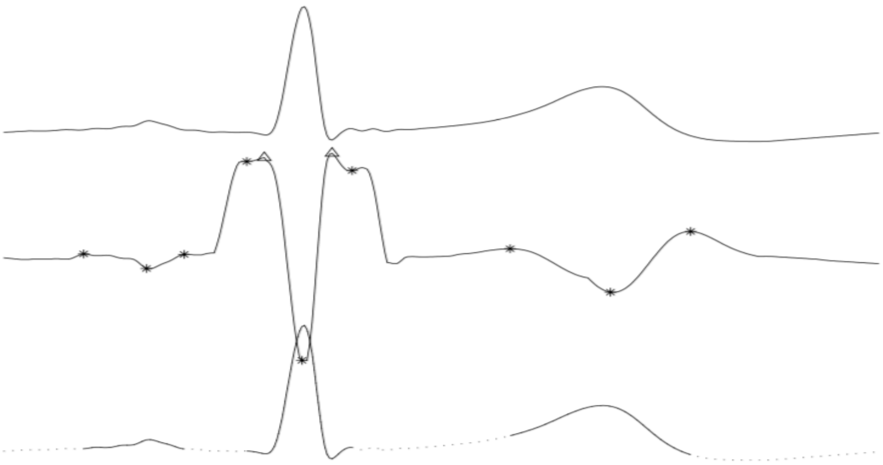

Figure 3.5: Result of characteristic wave detection for single ECG beat. From top to bottom: original ECG signal; the MMD transformed signal at scale 20 with the detected characteristic points marked; detected characteristic waves in solid line. Image taken from [11].

Table 3.3: Delineation performance of the algorithm in [11].

$Parameter$	Pon	$Poff$	$QRSon$	$QRSoff$	Ton	$Toff$
Se $(\%)$	97.2	94.8	1004	100	99.8	99.6
$m \pm s(ms)$	9.0 ± 9.4	12.8 ± 13.2	3.5 ± 6.1	2.4 ± 10.3	7.9 ± 15.8	8.3 ± 12.4

correct the baseline, as first step, the ECG is preprocessed by morphological filtering.

The algorithm has been tested using the first ECG leads from the MIT-BIH arrhythmia database and the QT database. In Fig. 3.5 it is possible to see the original signal and the MMD transformed signal at scale 20 with the detected characteristic points marked. The results obtained by the MMD detector are showed in Table 3.3. The performance on QRS and T waves are extremely good, whereas the algorithm struggles with the P wave (in general P wave is the most demanding one since it has a low peak and the noise can be even stronger than the P wave).

3.6 Wavelet-Based Algorithms

The *Wavelet Transform*-based algorithms are the most adopted ones for the ECG delineation [14], [15], [16], [16]. The use of WT is due to some reasons, among others:

- The drawbacks of filtering-based approach: frequency variations in the characteristic waves often adversely affect its performance. The frequency distribution of QRS complexes generally overlaps with that of the noise, resulting in both false positive and false negative detections. Plus, this approach has a low efficiency when dealing with odd morphologies;

- WT is attractive as a mathematical tool because of two main properties: first, when the WT is applied, the ECG can be described at different scales of temporal and frequency resolution, thus high frequency waves (such as the QRS complex) can be distinguished from low frequency waves (such as P and T waves); second, the WT can be easily implemented with a cascade of Finite-Impulse Response (FIR) filters.

- The noise and artifacts affecting the ECG signal also appear at different frequency bands, thus having different contribution at the various scales.

The algorithm in [14] is probably the most famous and benchmark WT-based approach. It is worthwhile to describe how it works because almost all other approaches proceed in a similar way, with some obvious slight variations (e.g., the values of the thresholds or the prototype wavelet).

The wavelet transform is a decomposition of the signal as a combination of a set of basis functions, obtained by means of dilation (a) and translation (b) of a single prototype wavelet $\psi(t)$. Thus, the WT of a signal $x(t)$ is defined as

$$W_a x(b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-b}{a}\right)dt, \quad a > 0. \tag{3.10}$$

The greater the scale factor a is, the wider is the basis function and consequently, the corresponding coefficient gives information about lower frequency components of the signal, and viceversa. In this way, the temporal resolution is higher at high frequencies than at low frequencies.

If the prototype wavelet $\psi(t)$ is the derivative of a smoothing function θ, it can be shown that the wavelet transform of a signal $x(t)$ at scale a is

$$W_a x(b) = -a\left(\frac{d}{db}\right) \int_{-\infty}^{+\infty} x(t)\theta_a(t-b)dt, \tag{3.11}$$

where $\theta_a(t) = (1/\sqrt{a})\theta(t/a)$ is the scaled version of the smoothing function. The wavelet transform at scale a is proportional to the derivative of the filtered version of the signal with a smoothing impulse response at scale a. Therefore, the zero-crossings of the WT correspond to the local maxima or minima of the smoothed signal at different scales, and the maximum

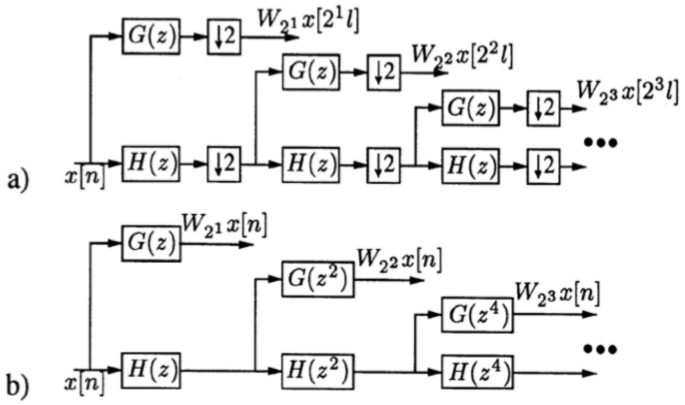

Figure 3.6: Two filter-bank implementation of DWT. (a) Mallat's algorithm. (b) Implementation without decimation (*algorithme à trous*). Image taken from [14].

absolute values of the wavelet transform are associated with maximum slopes in the filtered signal.

The scale factor a and/or the translation parameter b can be discretized. The usual choice is to follow a dyadic grid on the time-scale plane: $a = 2^k$ and $b = 2^k l$. The transform is then called *dyadic wavelet transform*, with basis functions:

$$\psi_{k,l}(t) = 2^{-k/2}\psi(2^{-k}t - l); \quad k, l \in \mathbb{Z}^+. \tag{3.12}$$

For discrete-time signals, the dyadic discrete wavelet transform (DWT) is equivalent, according to Mallat's algorithm, to an octave filter bank [39], and can be implemented as a cascade of identical cells [low-pass and high-pass finite impulse response (FIR) filters], as illustrated in Fig. 3.6(a).

The downsamplers after each filter in Fig. 3.6 remove the redundancy of the signal representation. As side effects, they make the signal representation time-variant, and reduce the temporal resolution of the wavelet coefficients for increasing scales. To keep the time-invariance and the temporal resolution ad different scales, the same sampling rate in all scales is used, what is achieved by removing the decimation stages and interpolating the filter impulse responses of the previous scale. This algorithm, called *algorithme à trous* [40], is shown in Fig. 3.6(b).

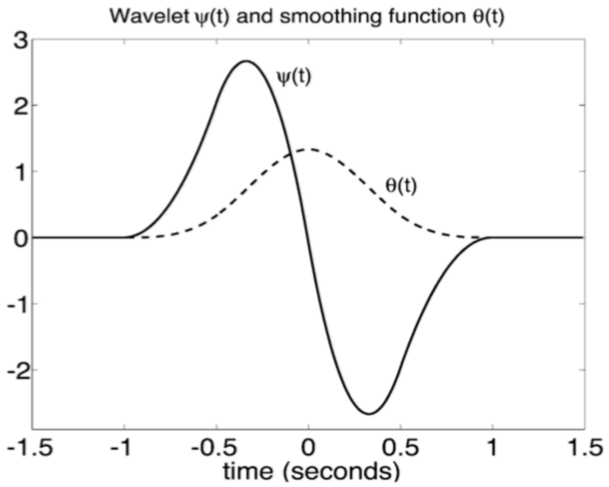

Figure 3.7: Prototype wavelet $\psi(t)$ and smoothing function $\theta(t)$, from [14].

Using this algorithm, the equivalent frequency response for the k-th scale is

$$Q_k(e^{j\omega}) = \begin{cases} G(e^{j\omega}), & k = 1 \\ G(e^{j2^{k-1}\omega}) \prod_{l=0}^{k-2} H(e^{j2^l\omega}), & k \geq 2 \end{cases} \tag{3.13}$$

In [14], a quadratic spline is used as prototype wavelet $\psi(t)$. The quadratic spline Fourier transform is

$$\Psi(\Omega) = j\Omega\left(\frac{\sin(\frac{\Omega}{4})}{\frac{\Omega}{4}}\right)^4 \tag{3.14}$$

The wavelet can be easily identified as the derivative of the convolution of four rectangular pulses, i.e., the derivative of a low-pass function. Fig. 3.7 represents the wavelet and smoothing function used in [14].

A brief description of the algorithm follows, the complete steps and details can be found in the related paper. The algorithm applies directly over the digitized ECG signal without any prefiltering. According to the spectrum of the ECG signal waves, most of the energy of the ECG signal lies within the scale 2^1 to 2^5. Fig. 3.8 shows several simulated waves similar to those in the ECG, together with the first five scales of their DWT. As exemplified by (a), monophasic waves produce a positive maximum-negative minimum pair along the scales, with a zero crossing between them. Each sharp change in the signal is associated to a line of maxima or minima across the scales. In wave (b), small Q and S wave peaks have zero crossings associated in the WT, mainly at scales 2^1 and 2^2. P or T-like waves (c) have their major

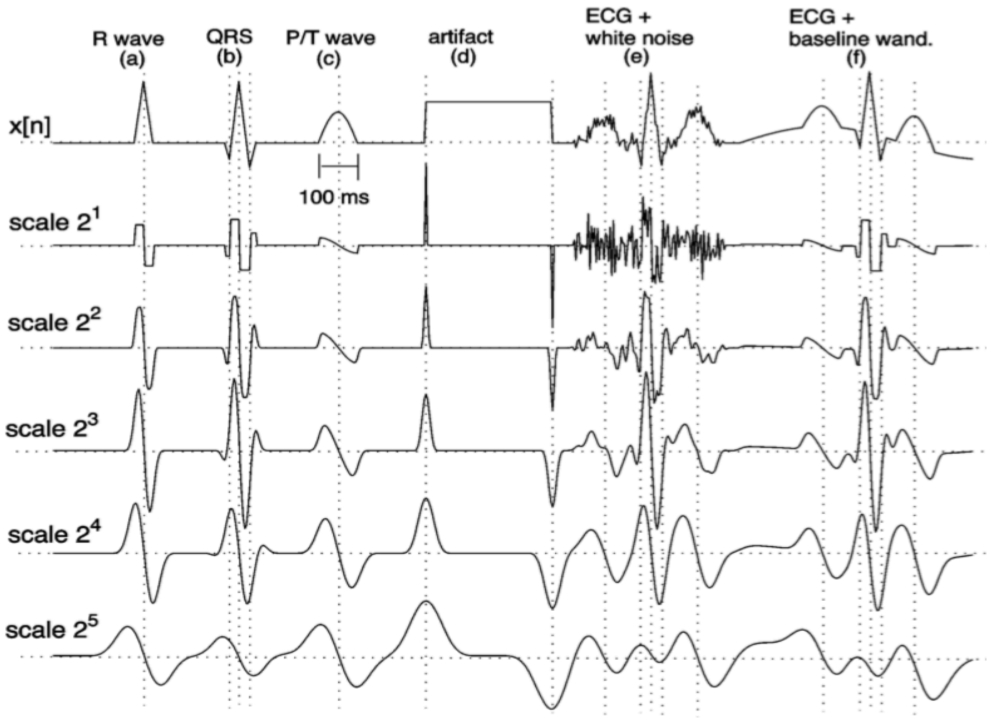

Figure 3.8: WT at the first five scales of ECG-like simulated waves, from [14].

component at scales 2^4 to 2^5, whereas artifacts like (d) produce isolated maximum or minimum lines which can be easily discarded. If the signal is contaminated with high-frequency noise (e), the most affected scales are 2^1 and 2^2, being higher scales essentially immune to this sort of noise. Baseline wander (f) affects only at scales higher than 2^4.

Using the information of local maxima, minima and zero crossings at different scales, the algorithm identifies the significant points in the following four steps: 1) detection of QRS complexes; 2) detection and identification of the QRS individual waves (Q, R, S, R'), and determination of the QRS complex boundaries; 3) T wave detection and delineation; and 4) P wave detection and delineation.

1. *QRS Detection*: QRS complexes are detected using an algorithm based on the multiscale approach. This algorithm searches across the scales for "maximum modulus lines" exceeding some thresholds at scales from2^1 to 2^4. After rejecting all isolated and redundant maximum lines, the zero crossing of the WT at scale 2^1 between a positive maximum-negative minimum pair is marked as a QRS.

2. *QRS Delineation (Onset, End and Individual Waves)*: the algorithm departs from the position given by the detector, which must be flanked by a pair of maximum moduli

with opposite signs at scale 2^2, namely at n_{pre} and n_{post}. The delineator looks before n_{pre} and after n_{post} for significant maxima of $|W_{2^2}x[n]|$ accounting for other adjacent slopes within the QRS complex. To consider a local maximum modulus as significant, it must exceed the threshold. The zero crossings between these significant slopes at scale 2^1 are assigned to wave peaks, and labeled depending on the sign and the sequence of the maximum moduli. The onset (end) of the QRS is before (after) the first (last) significant slope of the QRS, which is associated with a maximum of $|W_{2^k}x[n]|$. So, firstly the samples of the first and last peaks associated with the QRS in $W_{2^k}x[n]$, say n_{first} and n_{last}, are identified. Then, candidates to onset and end are determined by applying two criteria: i) searching for the sample where $|W_{2^2}x[n]|$ is below a threshold relative to the amplitude of the maximum modulus; ii) searching for a local minimum of $|W_{2^2}x[n]|$ before n_{first} or after n_{last}. Finally the QRS onset and end are selected as the candidates that supply the nearest sample to the QRS.

3. *T Wave Detection and Delineation*: a search window for each beat is defined, relative to the QRS position and depending on a recursively computed RR interval. Within this window, we look for a local maxima of $|W_{2^4}x[n]|$. If at least two of them exceed the threshold ϵ_T, a T wave is considered to be present. In this case, the local maxima of WT with amplitude greater than γ_T are considered as significant slopes of the wave, and the zero crossings between them as the wave peaks. Depending of the number and polarity of the found maxima, we assign one out of six possible T wave morphologies: positive (+), negative (-), biphasic (+/- or -/+), only upwards, and only downwards. If the T wave is not found in scale 2^4, the above process is repeated over $|W_{2^5}x[n]|$. Attending to the loss of time resolution in the growing scales, the peak(s) of the T wave correspond to the zero crossing(s) at scale 2^3, if they exist, or at the scale 2^k in which T wave was found. To identify the wave limits, the same criteria as for QRS onset and end is used.

4. *P Wave Detection and Delineation*: The P wave algorithm is similar to the T wave algorithm, using an appropriate RR-dependent search window and adequate thresholds. For P wave only four different morphologies are admitted: positive (+), negative (-), and biphasic (+/-, -/+).

As a last thing, I want to show Table 3.4 that summarizes the performances of four WT-based algorithms. The algorithms have been tested on the QT database.

3.7 Other approaches

In addition to the already described approaches, in the literature it is possible to find other algorithms for the ECG delineation. Among them, I want to remember those based on the Low Pass Differentiation (LPD) [9], on the Hidden Markov Model (HMM) [8] and

Table 3.4: Delineation performance of the WT-based algorithms.

$Method$	$Parameters$	Pon	$Ppeak$	$Poff$	$QRSon$	$Rpeak$	$QRSoff$	$Tpeak$	$Toff$
Martínez et al. [14]	$Se(\%)$	98.87	98.87	98.75	99.97	99.92	99.97	99.77	99.77
	$P^+_{min}(\%)$	91.03	91.03	91.03	-	-	-	97.79	97.79
	$m \pm s(ms)$	2.0 ± 14.8	3.6 ± 13.2	1.9 ± 12.8	4.6 ± 7.7	-	0.8 ± 8.7	0.2 ± 13.9	-1.6 ± 18.1
Cesari et al. [15]	$Se(\%)$	-	-	-	100.00	-	100.00	99.50	98.73
	$P^+(\%)$	-	-	-	-	-	-	-	-
	$m \pm s(ms)$	-	-	-	2.8 ± 7.7	-	2.7 ± 9.7	-2.6 ± 12.2	-2.7 ± 20.7
Di Marco et al. [16]	$Se(\%)$	98.15	98.15	98.15	100.00	-	100.00	99.72	99.77
	$P^+_{min}(\%)$	91.00	91.00	91.00	-	-	-	97.76	97.76
	$m \pm s(ms)$	-4.5 ± 13.4	-4.7 ± 9.7	-2.5 ± 13.0	-5.1 ± 7.2	-	0.9 ± 8.7	-0.3 ± 12.8	1.3 ± 18.6
Rincón et al. [17]	$Se(\%)$	99.87	99.87	99.91	99.97	-	99.97	99.97	99.97
	$P^+_{min}(\%)$	91.98	92.46	91.70	98.61	-	98.72	98.91	98.50
	$m \pm s(ms)$	48.6 ± 11.2	10.1 ± 8.9	0.9 ± 10.1	3.4 ± 7.0	-	3.5 ± 8.3	3.7 ± 13.0	-2.4 ± 16.9
$2\sigma_{cse}$ Tolerance		10.2	-	12.7	6.5	-	11.6	-	30.6

on a combination of HMM and Wavelet transform [7]. In particular, the HMM-based approaches provide classification of the typology of the beats, too.

4
Processing Pipeline

4.1 Introduction

The processing pipeline of my proposed work is constituted by three stages: 1) Pre-Processing 2) Learning Framework (Deep Learning Network) and 3) Post-Processing. The processing pipeline can be visualized in Fig. 4.1. Note that the steps with an asterisk are applied only to the records of the QT database.

The Pre-Processing phase consists of all those elaboration steps that are applied to the ECG signals of the databases and that at last bring about the final dataset, whose elements

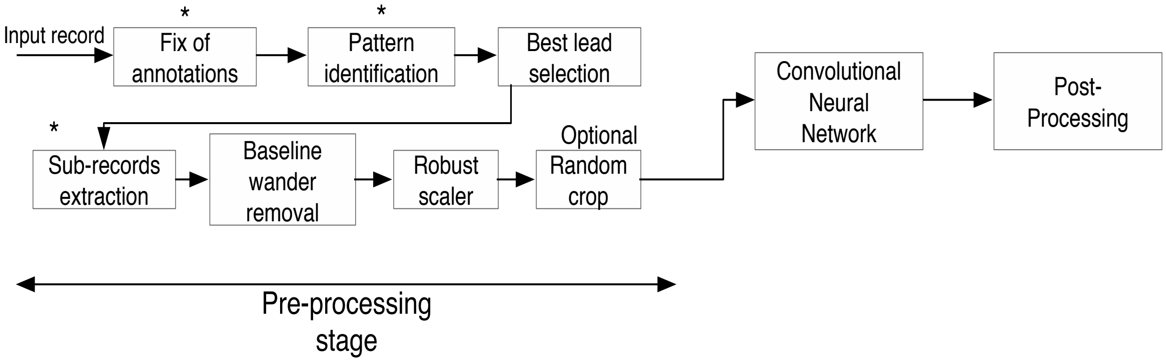

Figure 4.1: Processing pipeline of my proposed work. The steps with an asterisk are employed on the qtdb only.

Table 4.1: Distribution of the 105 records in the QT database.

MIT-BIH Arrhyt.	MIT-BIH ST DB	MIT-BIH Sup. Vent.	MIT-BIH Long Term	ESC STT	MIT-BIH NSR DB	Sudden Death
15	6	13	4	33	10	24

will be the input to the Deep Learning network. The elaborations have been applied to the QT database [37], the MIT-BIH Supraventricular Arrhythmia (abbreviated in SVDB) database [18] and MIT-BIH Arrhythmia (abbreviated in MITDB) database [38]. The QT database has required more steps since it provides annotations for all the waves (onset, peak and offset). Besides, quite a lot of records in the QT database have required direct tweaks of the annotations.

Afterwards, the description of the Deep Learning framework used for the training phase will be characterized.

Eventually, during the post-processing stage, from the output of the DL framework, the locations of the peaks will be individuated.

4.2 Pre-Processing Stage

The pre-processing phase is the most elaborated one and is made up of four building blocks, plus three more blocks applied only to the QT database. Before delving into the details of these steps, a description of the three databases is provided.

The *QT* database (QTDB) [37] includes ECGs which were chosen to represent a wide variety of QRS and ST-T morphologies, in order to challenge QT detection algorithms with real-world variability. The records were chosen primarily from among existing ECG databases, including the MIT-BIH Arrhythmia Database, the European Society of Cardiology ST-T Database, and several other ECG databases collected at Boston's Beth Israel Deaconess Medical Center. Table 4.1 shows the sources of the data.

The QT database contains a total of 105 fifteen-minute excerpts of two channel ECGs, selected to avoid significant baseline wander or other artifacts. Within each record, between 30 and 100 representative beats were manually annotated by cardiologists, who identified the beginning, peak and end of the P-wave, the beginning and end of the QRS-complex, the peak and end of the T-wave, and (if present) the peak and end of the U-wave. In order to permit the study of beat-to-beat variations such as alternans, 30 consecutive beats of the dominant morphology were annotated in each case if possible.

All records were sampled at 250 Hz. Each record includes a signal file (with extension .dat), a header file describing the format of the signal file (extension .hea) and several annotation files (among them, the .art contains the original annotations from the source database, .man contains the annotations of the R peak made automatically, .q1c and .q2c contain the annotations added by two experts (1 and 2, respectively) of the waveform boundaries, as well as the peaks of P, T and U waves).
The regular annotation symbols are: '(', ')', 't', 'p' and 'u'. Note that an opening round bracket can point out the onset of any of the four waves in an ECG signal. This feature is important for the second step of the pre-processing.

The *MIT-BIH Arrhyhmia* database (MITDB) [38] contains 48 half-hour exceprts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Twenty-three recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60 %) and outpatients (about 40 %) at Boston's Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample.
The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mv range. Two or more cardiologists independently annotated each record; disagreements were solved to obtain the computer-readable reference annotations for each beat (approximately 110000 annotations in all) included with the database.

The *MIT-BIH Supraventricular Arrhythmia* database (SVDB) [18] includes 78 half-hour ECG recordings chosen to supplement the examples of supraventricular arrhythmias in the MITDB database. The sampling frequency of the database is 128 Hz.

Since some records of the QTDB have been selected among the ones in the MITDB (15 records) and the ones in the SVDB (13 records), these 29 records have been discarded in order not to have duplicates. In addition to this, three records from QTDB ("$sel233$", "$sel44$" and "$sele0104$") have been discarded because they present issues for the developed algorithm. Therefore, if we count all the records from the three databases, the total size of the overall starting database is equal to 201.

Obviously, in order for all records to have the same sampling frequency, the records from MITDB and SVDB have been resampled to the 250 Hz frequency (i.e., the one of the QTDB).

As a last remark, the term *record* consists of a signal, an header file and one or more annotations files. From now on, however, I will use the terms *record* and (*physical*) *signal* interchangeably for indicating the same object.

4.2.1 Fix of Annotations & Pattern Identification

These two steps are applied **only** to the QT database. In [37] it is possible to see the waveform pattern of each record; namely, the events that have been annotated. There are 9 different patterns, such as: (p)(N)(t) and (p)(N)t)u). The non-uniformity of the records' pattern complicates the algorithm and it is necessary to get the pattern of each record with a dedicated function. Even worse, several records don't comply with the established pattern, thereby bringing about the algorithm to fail. In order to mend these glitches, all the inappropriate annotations have been manually discarded. Note that this issue does not exist for the MITDB and SVDB since only the R-peak is annotated, so the only possible problem is the wrong position of a peak and obviously this cannot be fixed (this was in charge of the cardiologists).

4.2.2 Best Lead Selection

Each record has two channels or leads. As the algorithm proposed in this dissertation works on a single channel, it is imperative to choose the best channel; namely, the one providing the highest Signal to Noise Ratio (SNR). In order to accomplish this, for each R-peak of the record, its SNR is computed. The power of the peak is calculated as the sum of the squares of the amplitudes of the samples inside a window of length 12 centered in the position of the R-peak. Remember that each sample corresponds to 4 ms. As for the power of the noise, the same window is shifted at the left of the R peak for a certain number of samples. In fact, between the P wave and the Q location, there is the P-R segment that usually is an horizontal line without electrical activity. Hence, this segment is constituted only by "noise". The window with the lowest value is selected as the noise power. Given this, the mean of the SNRs for the two channel is computed. Finally, the channel with the highest average SNR is kept.

4.2.3 Sub-Records Extraction

This step affects only the QTDB. In fact, there is still another issue with this database. For several records, the annotations of the beats are not consecutive. This means that, for instance, a certain number of beats are annotated, then there is an interruption, and, after a certain number of beats, other beats are annotated. Thus, it is crucial to extract these series of consecutive beats from the annotation file and from the record, and discard the "holes". If not, when these signals will be inputted to the learning framework, these peaks will be

detected, providing false positives. In order to do this, I have to check the R-R distance between consecutive annotated labels: if this distance is greater than a fixed one (k_R in the python function), then the two R peaks are not consecutive; otherwise, they are adjacent. The value of k_R depends on the specific record. I set it by default equal to 350 samples (1.4 s). Nevertheless, for some records, k_R has been adjusted ad hoc.
Eventually, among the extracted sub-records, only the ones with a length greater than 750 samples are stored, so that on average at least three R peaks are present. Thus, it is possible that, from one specific record, some sub-records are extracted. During the learning phase, it is important to keep all the sub-records belonging to the same record/patient into the test or train set.

4.2.4 Baseline Wander Removal & Scaling & Random Crop (Optional)

ECG records acquired on an ambulatory way or in stress tests are heavily contaminated with noise. One of the main sources of noise in these acquisition conditions is the baseline wander (BLW). It is a low-frequency noise ($0.05 - 3$ Hz in stress tests). BLW is mainly due to movement during breathing, patient movements, poor contact between electrode cables and ECG recording equipment, inadequate skin preparation where the electrode is placed and dirty electrodes. In order to remove the baseline, the *Neurokit,* a Python module, has been used [41]. In particular, the *ecg_preprocess* function implements a FIR band-pass filter, with order 0.3 and cutoff frequencies $[3, 45]$ Hz. Plus, a scaling operation is performed.

The last step of the pre-processing phase is an optional random crop. The random crop operation consists in extracting randomly a portion of the signal with a desired length ($crop\_span$). In this way, I can create a dataset of records with the same size or one whose records have their original size. Indeed, during the training phase, I created a dataset with a $crop\_span$ equal to 750 samples; whereas, for the testing phase, I considered the full-length records.

4.3 Learning Framework

Hitherto the pre-processing steps have been proposed. The resulting signals will be input to a Convolutional Neural Network (CNN for brevity), which is in the charge of the learning/training and testing phase. Two different CNNs for the individuation of the R peaks and the P-T peaks have been devised, respectively. They have been implemented in Pytorch and they will be investigated and showed in the next sections.

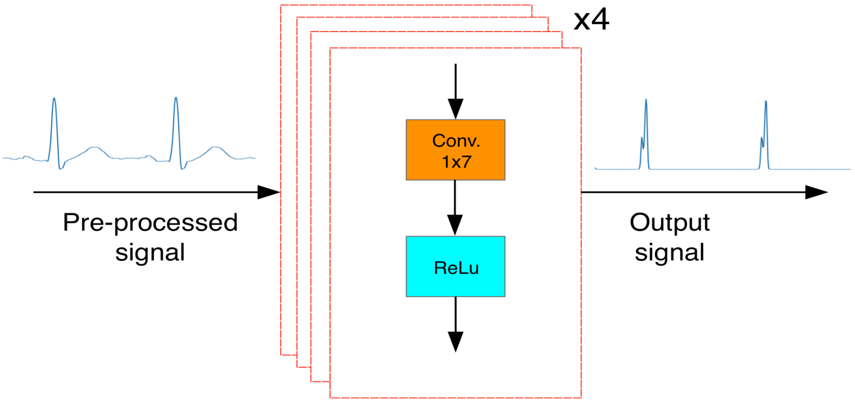

Figure 4.2: Structure of the CNN4 for the R-peaks' delineation.

Table 4.2: Parameters' tuning for each layer of the CNN4.

Layer #	Kernel Size	in_channels	out_channels	Stride	Dilation	Input Size	Output Size	Receptive Field
1	7	1	8	1	1	L	L	7
2	7	8	16	1	1	L	L	13
3	7	16	32	1	1	L	L	19
4	7	32	1	1	1	L	L	25

4.3.1 CNN for the R-peak Identification

For the location of the R peaks, a 4-layers CNN has been implemented (henceforward dubbed CNN4). It is depicted in Fig. 4.2. The details of each stacked layer can be consulted in Table 4.2 (note that all the parameters are expressed in samples, and one sample corresponds to 4 ms).

The pre-processed signal, x_{in}, which is input to the network, has dimensions equal to 1xL. Throughout the whole network, the dimensions of the input and output signals are the same (i.e., $L_{in}^{i} = L_{out}^{j} = L, \forall\, i, j \in \{1, 2, 3, 4\}$, where i, j represents the i-th and j-th layer, respectively). This is accomplished by zero-padding both sides of the input signal with pad zeros, where pad can be computed with the following formula:

$$pad = \left\lfloor \frac{(L_{out} - 1)stride - L_{in} + 1 + dilation(kernel - 1)}{2} \right\rfloor. \qquad (4.1)$$

The number of channels increases layer by layer, till at the output of the network it becomes again equal to 1. Except for the last layer, after each convolution operation, a ReLu

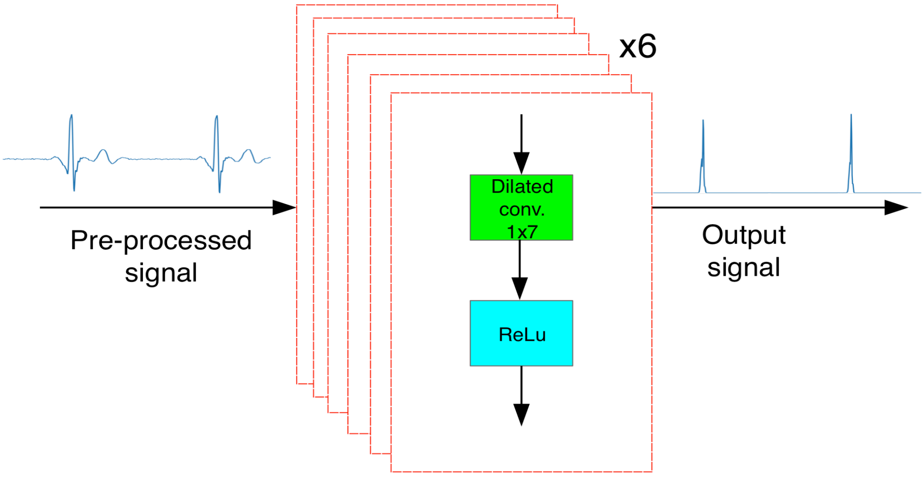

Figure 4.3: Structure of the CNN6 for the P/T-peaks' delineation.

activation function is applied. Dropout has not been included since it did not bring about any upturn (L2 regularization penalty is already included in the Adam optimizer). For the R peak, the dilation factors have been all set to 1 (i.e., a standard convolution operation). Indeed, since the R peak is very prominent with respect to the P-T peaks, there is no use expanding the receptive field. The simulations have confirmed this argument. As for the choice of the kernel size, the best tradeoff between complexity and performance was given by 7. Furthermore, I have added *batch normalization* and *residuals*; however, they have not brought any betterment, hence I dropped them.
The output of CNN4, x_{out}, is a signal that should show a spike in the proximity of each R peak.
One final fundamental remark: the length of the input signal L can be of any size. This decision makes all the system dynamic and free of length's constraints (in contrast with the typical decision of giving in input to a CNN signals/images with fixed dimensions). In particular, for the training phase, L has been set to 750 samples, thereby involving a random crop of the regarded signal; whereas, for the testing phase, each signal has been considered with its original length.

4.3.2 CNN for the PT-peaks Identification

For both P and T peaks, a 6-layers CNN has been designed (dubbed CNN6). Its structure is schematized in Fig. 4.3, whereas the specifics of each layer can be consulted in Table 4.3.

Table 4.3: Parameters' tuning for each layer of the CNN6.

Layer #	Kernel Size	in_channels	out_channels	Stride	Dilation	Input Size	Output Size	Receptive Field
1	7	1	8	1	1	L	L	7
2	7	8	16	1	1	L	L	13
3	7	16	32	1	2	L	L	25
4	7	32	64	1	4	L	L	49
5	7	64	128	1	8	L	L	97
6	7	128	1	1	16	L	L	193

The network is made up of 6 layers. It is deeper than CNN4 because the identification of the P and T peaks' position is by far more challenging than the R ones. This is due to the fact that the P and T peaks have a lower intensity and, consequently, now the presence of the noise becomes an important hassle, which leads to a deterioration of the global performances. For this reason, two additional layers have been added and the receptive field has been step-by-step extended by means of dilation convolutions (also known as atrous convolutions). In fact, systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage.
Apart from these two modifications, the reasonings made for the CNN4 still apply.

4.4 Post-Processing Stage

The post-processing step consists in identifying the peaks, given the output signal. The shape of the output signal is similar to the input one, with the difference that the desired peaks' intensity approaches zero, whereas the other peaks have a negative magnitude which is quite bigger. This behaviour can be better understood in Fig. 4.4, which refers to the P peak, but obviously it holds for the R and T peaks, too. In order to emphasize the desired peaks, and since whose absolute value is smaller than the undesirable peaks, a *sigmoid* operation is applied to the output signal. By doing this, the resulting signal will have a relevant spike only in the exact location (bottom signal in the figure). From now on, x_{out} will refer to the signal after the sigmoid operation. The effectiveness of the sigmoid can be seen in Fig. 4.5, where its shape is shown. Mathematically, the sigmoid function is defined as:
$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$.

At this point, a peak detector is needed. I have chosen the one offered by the signal processing packet of the Scipy library: *find_peaks*. The parameters of this method are the same for the P, R and T peaks detection, but one: the *prominence*. The prominence of a peak measures how much a peak stands out from the surrounding baseline of the signal and is defined

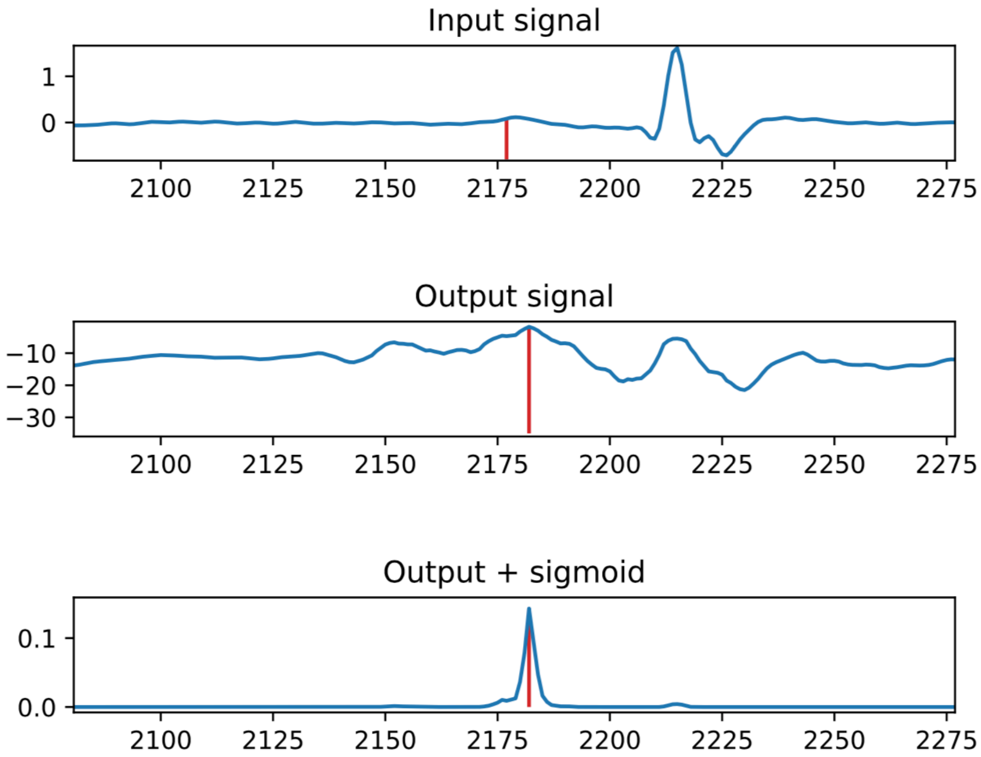

Figure 4.4: From top to bottom: input signal to the CNN; output signal; resulting signal after performing sigmoid function.

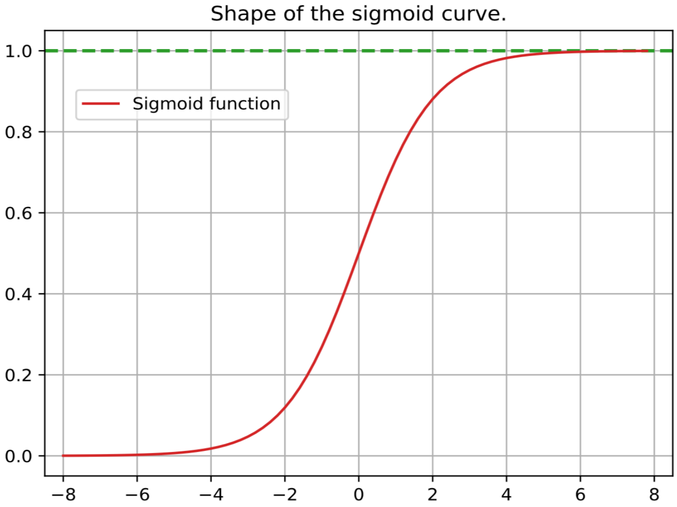

Figure 4.5: Shape of the sigmoid function.

as the vertical distance between the peak and its lowest contour line. This parameter has been computed as the k-th percentile of x_{out}: $prominence = percentile(x_{out}, k)$. The values of k for the P, R and T peaks are 97, 92 and 90, respectively. The gauging of the parameter k was merely deterministic.

Then, since some small peaks can be present and they do not represent a correct peak, only the ones with a magnitude above a certain threshold are kept, the others are wiped out. The criterion for keeping or rejecting a peak follows. Let p and p_{max} be the magnitude of a peak and the magnitude of the peak with the highest magnitude, respectively; then, a peak is kept if $p > \epsilon \cdot p_{max}$, where ϵ is equal to 0.3 for the P and T peaks, otherwise it is equal to 0.15 for the R peak.

At this point, a list with the positions of the candidates for being correct peaks is available. The last thing to do is to compare the candidates with the real peaks given by the annotations.

5 Simulations and Performance Analysis

5.1 Introduction

In this final chapter, the results obtained with my algorithm will be exhibited and compared with those of other algorithms. It will be shown how the addition of white gaussian noise to the ECG signal will affect the performances. Plus, it will be shown the accuracy of the algorithms by changing the length of the tolerance window.

5.2 Algorithms for the Comparison

It has been quite difficult to find out online some consistent and well-documented algorithms for the comparison purpose. A brief description of them, along with the corresponding references, follows.

- **XQRS** algorithm. It is a QRS detector algorithm provided by the Python WFDB package, thus it does not perform P-T peaks detection. [42]. It belongs to the class of the thresholding "algorithms". First of all, it loads the signal and configuration parameters. It performs a bandpass filtering and a moving wave integration (mwi) with a ricker wavelet. If specified, a learning is conducted in order to initiliaze running parameters of noise and QRS amplitudes, the QRS detection threshold, and recent RR intervals. Then, the algorithm iterates through the local maxima of the mwi signal. For each local maxima: 1) It checks whether it is a QRS complex. To be classified as a QRS, it must come after the refractory period, cross the QRS detection threshold, and not be classified as a T-wave if it comes close enough to the previous QRS. If successfully classified, it updates the running detection threshold and heart rate parameters. 2) If not a QRS, the algorithm classifies it as a noise peak and updates the running parameters. 3) Before continuing to the next local maxima, if no QRS was detected within 1.66 times the recent RR interval, it performs backsearch QRS detection: it checks the previous peaks using a lower QRS detection threshold.

- **Wavelet-based** algorithm and **Pan-Tomkins** algorithm. I have already discussed them in Chapter 3. The ECGKit is a toolbox that collects Matlab tools for reading, processing and presenting ECG signals [43]. Among the diversified applications, it implements R-detection using a Wavelet approach based on [14] and an approach based on the famous Pan-Tomkins algorithm [4]. The Wavelet algorithm performs P and T delineation, too.

- Three algorithms provided by the **NeuroKit** module [41]. It provides different methods of segmentation, among them I have chosen the three ones with acceptable performances and references. The first algorithm is documented in [44]. This algorithm is based on [4] and [45]. The second algorithm is based on [46] and [47]: the former presents the main threshold-based algorithm, whereas the latter introduces some features to increase the performances. The third and last algorithm is presented in [48]. It is based on the comparison between absolute values of summed differentiated electrocardiograms of one of more ECG leads and adaptive threshold. The threshold combines three parameters: an adaptive slew-rate value, a second value which rises when high-frequency noise occurs, and a third one intended to avoid missing of low amplitudes beats. All the three algorithms provide P, R and T segmentation.

In order of presentation, hereafter the algorithms will be dubbed as: XQRS, Wavelet, PanTom, Hamilton, Engzee and Christov. My own algorithm will be dubbed CNN4 for the R peak and CNN6 for the P-T peaks.

5.3 Metrics for the Comparison

The performances of the algorithms are compared by means of appropriate metrics. Each algorithm outputs a vector that contains the position of the estimated peaks ($estimated\_events$). This vector is compared with the one containing the positions of the peaks manually-annotated by the cardiologists ($true\_events$). In particular, I have calculated:

- *Precision/Predictivity* (P^+)= TP/(TP + FP)

- *Recall/Sensitivity* (Se) = TP/(TP+ FN)

- *Detection Error* (Fd) = (FN + FP)/ TB
- F_1 *score* (F1) = 2/(1/P^+ + 1/Se)

where TP denotes a *true positive* (when an automatic annotation matches a manual annotation), FP represents a *false positive* (when for an automatic annotation there is not a correspondent manual annotation), FN stands for *false negative* (when for a manual annotation there is not a correspondent automatic annotation), TB denotes the true beats. For each TP, I have calculated the time difference (i.e., error) between automatic and manual annotations. In order to compare the error distribution of the different algorithms, I have computed its *mean*, *median*, *standard deviation* (std) and *interquartile range* (IQR, it is the difference between the 75-th and 25-th percentile of the data). These metrics have been computed on the absolute value of the errors since outliers with opposite signs can be averaged out. Whereas, for the computation of the boxplots (see next sections), I have considered the errors with their sign.
A final remark for a TP. An estimated peak is considered as a TP if the absolute value of its error is below a specific threshold, which has been set to 12 samples (96 ms). This means that the length of the tolerance window (win_len) is 22 samples.

5.4 Tuning of the Hyperparameters for the Training of CNN4 & CNN6

In this section, the choice of the tuning of the hyperparameters for CNN4 and CNN6 will be exposed. If no network is mentioned during the description of a specific hyperparameter, then that choice applies to both networks. First of all, in order to derive a more accurate and truthfulness estimate of the model performance, I have used a 5-fold cross validation. In this way, the total dataset is split into 5 folds and, in turn, each fold will be used as test set. The number of *epochs* that has led to the best results of the training phase for CNN4 and CNN6 is 75 and 60, respectively. The choice of the *batch size* has been 16 for CNN4 and 8 for CNN6. This choice is due to the fact that the number of total signals for the R peak is 231; whereas, for the P and T peaks, the number of signals is 142 and 155, respectively. As for the *optimizer*, I have opted for the Adam algorithm. The *learning rate* is equal to 0.001. So as to prevent overfitting, the L2 penalty of the optimizer has been set to 0.0001. This is the only regularization technique I have used (dropout for the networks have brought no improvement). It worths noticing that I have also tried to implement a learning rate scheduler in order to adjust the learning rate based on the number of epochs. Nevertheless, neither does it led to better results. As a last thing, a binary cross-entropy loss function has been utilized. Plus, the weights for the positive examples have been set to 10.

5.5 Simulations' Results

The simulations have been run in two different scenarios. A noiseless scenario and a nosy one. In fact, the former consists in the original signals without the addition of AWG (additive white gaussian) noise; the former comprises the signals with the addition of AWG noise with ascending power. The noise has been created from Gaussian distributions with 0 mean and standard deviation equal to: 0.01, 0.03, 0.07, 0.1. Note that this is a noise created by myself to see how the algorithms react to a more and more noisy "environment". The physical noise due to the movements of the patient and the baseline wander is still present in both cases.

The results obtained for the R peak and for the P-T peaks will be recapitulated in a table for inspecting in details all the metrics. Furthermore, by means of boxplots, the statistical description of the time error will be displayed. Then, the plots of the F_1 score in function of the window's length will be shown. Finally, having selected a specific win_len, I will investigate the trend of the algorithms as the noise power increases.

5.5.1 Noiseless Scenario

In Table 5.1, the metrics for the R peak are summarized. We can observe that the XQRS algorithm is the best one, both in terms of precision and recall. My own algorithm is slightly behind. The PanTom algorithm performs quite well with respect to precision, yet it is more prone to committing FNs. The Wavelet alg performs well for both metrics. The remaining three algs have poorer performances than the others, and they behave in a very similar fashion (this trend is maintained for all the following simulations).

With regard to the statistics of the time error, my own alg performs by a hair better than all the others in terms of median and mean; howbeit, it shows worse results for the std and IQR than the XQRS, PanTom and Wavelet algs. Still, it behaves better than Hamilton, Engzee and Christov. Remember that the metrics in the table have been computed for the absolute value of the time errors. It is possible to visualize the statistical description of the time error by means of a boxplot, such as in Fig. 5.1. The boxes represent the inter-quartile range (IQR), the green lines show the median values, and the whiskers depict the 5th and the 95th percentile (the outliers are not showed). From the boxplot we can infer that CNN4, XQRS, Wavelet and PanTom have a very similar IQR. The median value for Hamilton (Engzee and

Table 5.1: Performance results for the R-peak segmentation, noiseless scenario, win_len = 22.

R peak detector	**CNN4**	**XQRS**	**PanTom**	**Wavelet**	**Hamilton**	**Christov**	**Engzee**
TB	296158	296158	296158	296158	296128	296128	296128
TP	279406	281215	241648	287386	246366	233545	243176
FP	18218	13589	12953	24495	47022	43712	48452
FN	16752	14943	61070	15332	49762	62583	52952
Fd	12.01	9.63	24.99	13.45	32.53	35.75	34.27
P^+ (%)	93.77	95.39	94.91	92.15	84.06	84.29	83.39
Se (%)	94.25	94.95	79.83	94.94	83.27	78.97	82.11
Mean err (samples)	3.18	3.41	3.59	3.35	5.21	5.36	5.30
Median err (samples)	2.09	3.23	3.47	3.25	3.67	3.69	3.68
Std err (samples)	3.01	1.39	1.48	1.27	4.42	4.72	4.56
IQR err (samples)	3.46	1.34	1.54	1.32	6.89	6.46	6.58

Christov's performances are very similar to the Hamilton's, hence they are not reported) is very close to 0, but it is clear that, due to the high IQR and 5-95th percentile, several large time errors average out. My own algorithm shows a slightly bigger 5-95 percentile rather than XQRS, Wavelet and PanTom.

Another interesting analysis is to see how the change of the win_len bears on the performance of the algorithms. For this purpose, the F_1 score is computed since it takes into account both precision and recall. Consider Fig. 5.2. Theoretically, we expect that, as the win_len increases, the F_1 score should increase as well and tend to 1 (i.e., 100 %). This behaviour is respected in practice. The slight downside of CNN4 with respect to XQRS is completely nullified when the win_len is pushed to very high values. Also the Hamilton alg benefits from a larger and larger window. The PanTom alg is the only one that does not increase too much its performance. However, except for PanTom, all the algs show a comparable trend and values. The vertical grey dashed line corresponds to the reference window length.

Now we move on to the P peak. It has been quite difficult to find out online some good P-T peaks detectors. In fact, now the total algorithms are 5. In Table 5.2, it is possible to see the results. First of all, we can see that the number of total beats is by far smaller than the R peak ones. CNN6 outstrips all the other algs. The recall and sensitivity are a little worse than the ones obtained for the R peak (in particular the recall), yet this is normal because the detection of the P peak is more challenging. The Wavelet alg shows quite good results, even though the recall is critical (77.39). Hamilton, Christov and Engzee perform good in terms

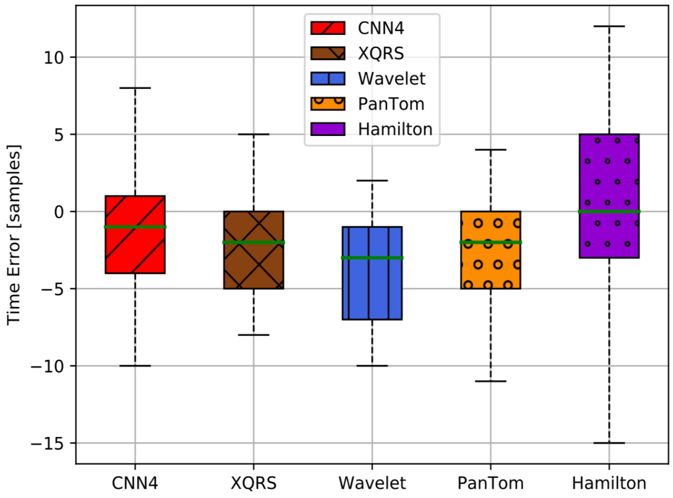

Figure 5.1: Statistical description of the time error for R peak, noiseless scenario.

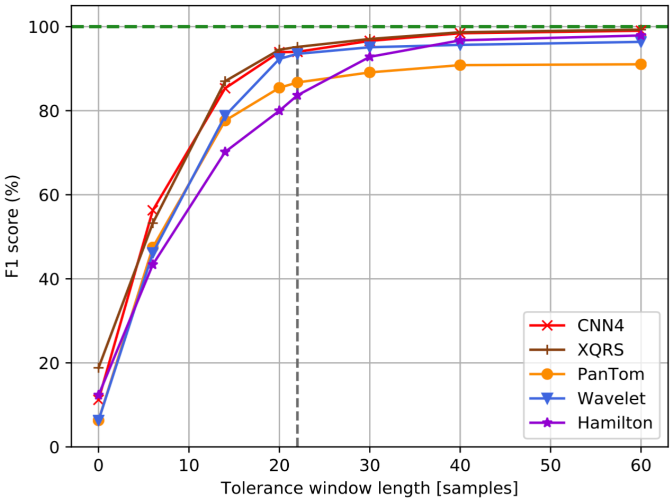

Figure 5.2: F_1 score vs win_len, R peak, noiseless scenario.

Table 5.2: Performance results for the P-peak segmentation, noiseless scenario, win_len = 22.

P peak detector	**CNN6**	**Wavelet**	**Hamilton**	**Christov**	**Engzee**
TB	2711	2741	2711	2707	2711
TP	2418	2098	1805	1809	1814
FP	141	323	225	222	234
FN	243	612	815	801	806
Fd	14.25	31.15	38.36	37.79	38.36
P^+ (%)	92.97	86.66	88.92	89.07	88.57
Se (%)	89.39	77.39	68.89	69.31	69.24
Mean err (samples)	3.75	3.92	3.86	3.71	3.92
Median err (samples)	2.27	3.15	3.63	3.38	3.61
Std err (samples)	4.00	2.91	1.65	1.84	1.78
IQR err (samples)	2.57	2.79	1.90	2.10	2.11

of precision, while the recall is problematic (it falls below 70). It is clear that the recall is the most challenging metrics and it entails that the algorithms are prone to missing the peaks. With regards to the error statistics, all the algorithms behave in a similar way. The std of the error is a bit worse for CNN6 and Wavelet.

From the boxplot of Fig. 5.3 we can see that the statistics of the error among the algs are quite similar. Note the median of the CNN6 approaching zero and the considerably high upper whisker of Wavelet (i.e., it tends to posticipate the position of the peaks).

Eventually, in Fig. 5.4, it is showed the trend of the F_1 score vs the win_len. It is evident how the increment of the F_1 score from win_len = 22 to 60 is small, thereby denoting a good accuracy for all the algs in locating the P peaks.

It is important to note one thing. Since for the P and T peaks the number of total beats is scant if compared with that of the R peak, the existence of problematic signals can heavily affect the performances. Indeed, for the P peaks, I have noticed two signals with a strange pattern of the P peaks. Consequently, my alg is not able to identify the P peaks, causing a raise of the FNs (actually, the recall value is worse than the precision's). In Fig. 5.5, these two signals are showed. They both display a very imperceptible P peak. It is clear that my CNN has issues in identifying it and around 60 out of the 243 FNs are due to these signals, thus more than the 20 %. These strange patterns can be caused by the presence of a highly noisy environment. Therefore some signals with pathological patterns bring about a slump in the overall performance.

We conclude with the analysis of the T peak. From Table. 5.3 we can see that it is the

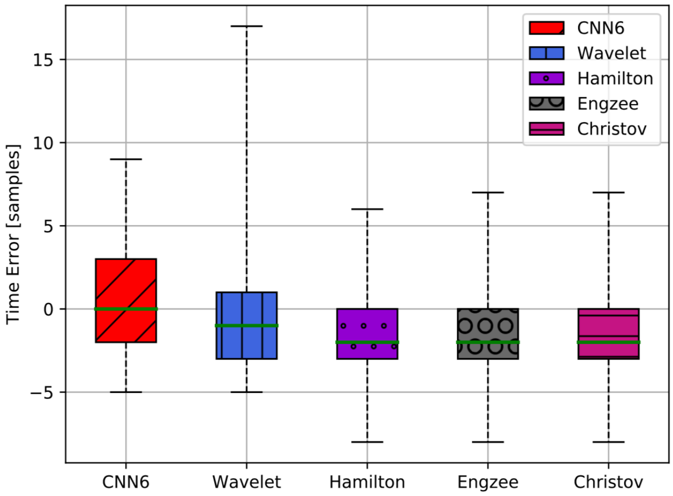

Figure 5.3: Statistical description of the time error for P peak, noiseless scenario.

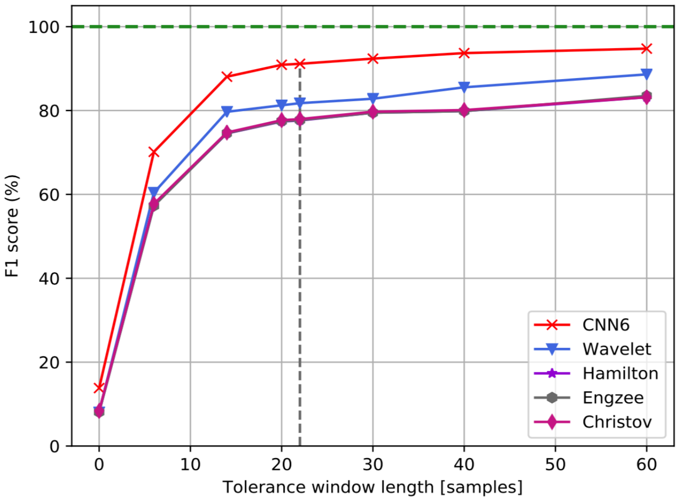

Figure 5.4: F_1 score vs win_len, P peak, noiseless scenario.

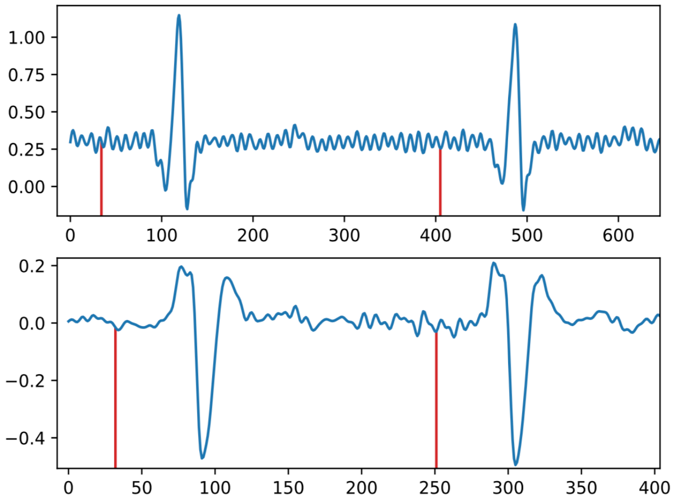

Figure 5.5: Signals with imperceptible P peaks. At the top, signal *sele0116_0* is depicted. At the bottom, signal *sel40_0*.

most challenging one. CNN6 loses about 10 % with respect to the P peak. The Wavelet alg loses around 3/4 % to the P delineation, yet it is still below my algorithm in terms of F_1 score (81.38 vs 77.71). The remaining algs struggle a lot (F_1 score abates under 50 %). The trouble of the P identification lies in the difficulty of locating exactly its position. From Table 5.6 it is evident how the mean and median of the error is rather high. In fact, the position of the T peak is not univocal because it does not have a simple morphology like the P peak. In Fig. 5.7 we can see some examples. At the top it is possible to visualize two very clear and quite high T peaks. Instead, in the middle and at the bottom, the peaks are very low and can be confused with other adjacent spikes. Remember that some signals sometimes exhibit also the U peak, and it could be confused with the T peak. Consequently, since the exact location of the T peak is hard to find out, the algorithm is more prone to giving in output a position which is outside the tolerance window length of 12 samples.

So, since there is this variability in the morphology of the T peak, we should expect that, as the win_len is dilated more and more, the F_1 score should increase as well. The trend is sketched in Fig. 5.8. In contrast to the P peak, now the increase of the F_1 score is very significant. All the algorithms increase their performance of more than 10 points; in particular, my algorithm goes from around 80 % to more than 96 %.

Table 5.3: Performance results for the T-peak segmentation, noiseless scenario, win_len = 22.

T peak detector	**CNN6**	**Wavelet**	**Hamilton**	**Christov**	**Engzee**
TB	2921	2951	2921	2917	2921
TP	2358	2133	1232	1221	1233
FP	518	436	917	880	908
FN	563	788	1579	1586	1578
Fd	37.01	41.48	85.45	84.54	85.11
P^+ (%)	82.04	83.03	57.33	58.12	57.59
Se (%)	80.73	73.02	43.83	43.50	43.86
Mean err (samples)	5.74	5.45	6.95	6.88	7.1
Median err (samples)	3.14	5.26	6.82	6.79	6.98
Std err (samples)	5.45	2.69	1.80	1.70	1.76
IQR err (samples)	6.04	3.25	2.20	2.08	2.13

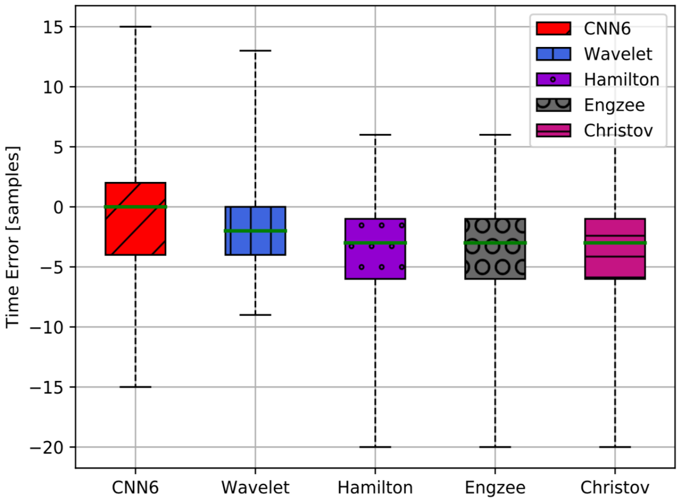

Figure 5.6: Statistical description of the time error for T peak, noiseless scenario.

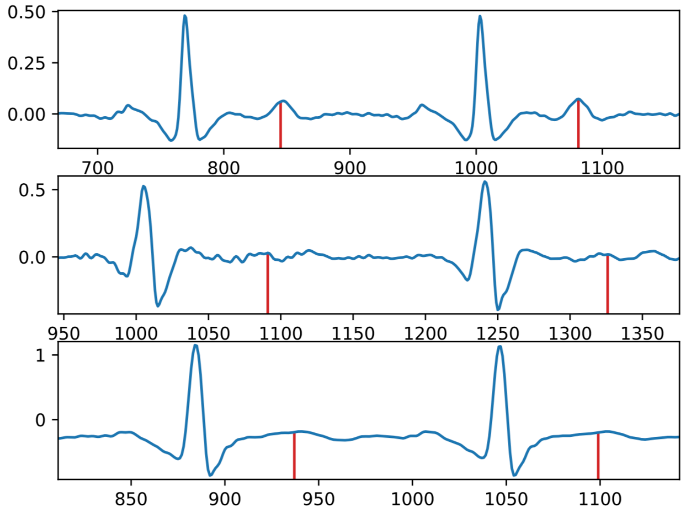

Figure 5.7: Morphology of the T peak. At the top: regular and well-visible T peaks. In the middle and at the bottom: two signals with hard-to-locate T peaks.

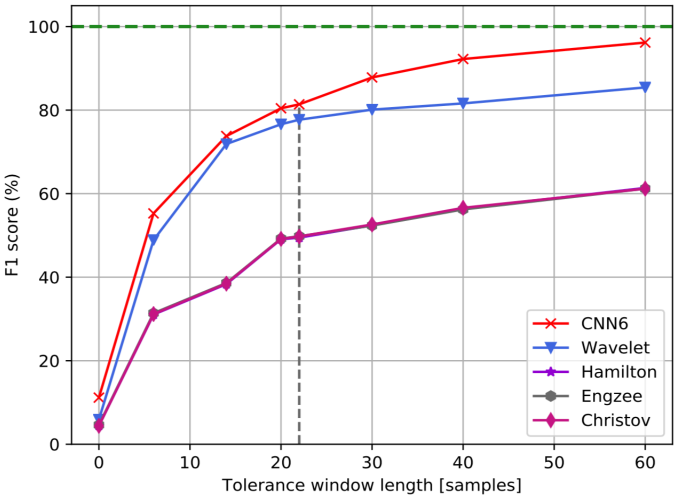

Figure 5.8: F_1 score vs win_len, T peak, noiseless scenario.

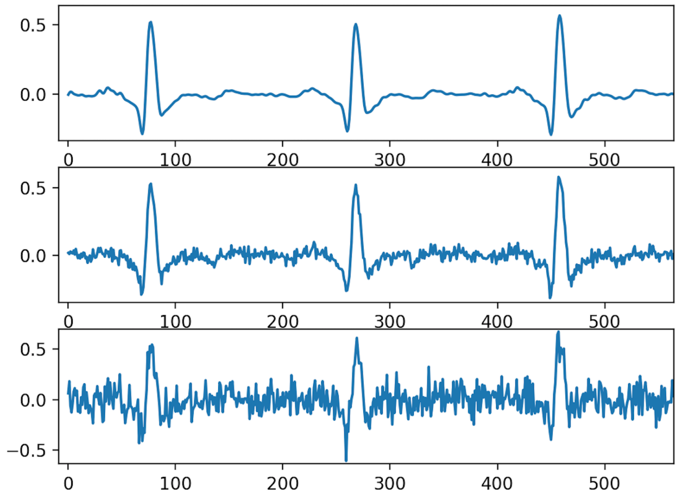

Figure 5.9: Impact of the noise in the input signal. From top to bottom: noiseless input signal; input signal with AWGN, std = 0.03; input signal with AWGN, std = 0.1.

5.6 Noisy Scenario

I wanted to test the robustness and resilience of the algorithms against the noise. I have added to each signal an additive white gaussian noise (AWGN). I have tested the algorithms in four different situations based on the standard deviation of the noise. The four increasing std values I have considered are: 0.01, 0.03, 0.07 and 0.1. It is possible to understand how the addition of a AWGN with different standard deviations changes the input signal in Fig. 5.9.

Regarding my algorithm, I have implemented two different methods. The first one consists in training a different network for each level of added noise and then test the network on the signals with the same std (i.e., I train a network where all signals have a noise with std = 0.01, another network where all signals have a noise with std = 0.03, and the like). The second method consists in training one single network where the signals can be noisy or not, and then I test all the signals with the same std, for each std. The std of the added noise is modelled like a discrete random variable X. The alphabet of X is $\mathcal{X} = \{0, 0.01, 0.03, 0.07, 0.1\}$ and its pdf is $p_X(0) = 0.35, p_X(0.01) = 0.2, p_X(0.03) = 0.2, p_X(0.07) = 0.15, p_x(0.1) = 0.1$. The std equal to 0 means that no noise is summed up at all. I have created this pdf based on the fact that in practice the signals have only the standard noise due to the baseline wander etc, hence the signals without any additional noise are the most likely, whereas the probability of a more and more noisy signal is less and less probable. Thus, during the training phase, for each signal in a batch, I generate a std value, using the afore-mentioned random variable,

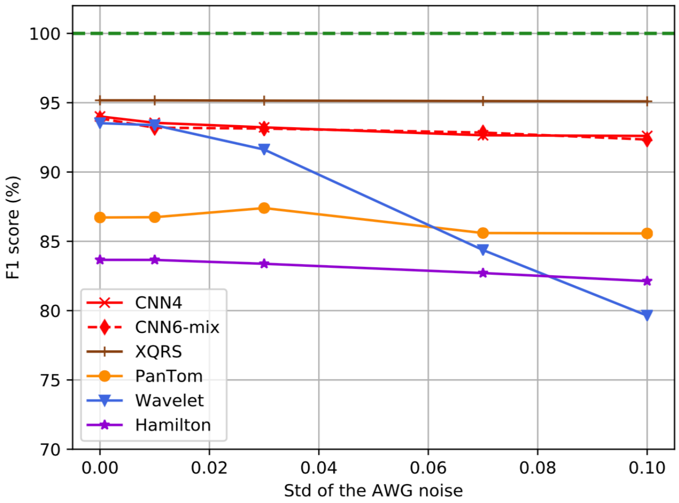

Figure 5.10: F_1 score vs std of the noise, R peak.

for the addition of the noise signal.
The second method is the most realistic one since in practice the signals are affected by random noises and not by noises with the same std. Plus, the second method is computationally better since I train only one network for all the cases rather than train each network for each level of noise. The first method is dubbed CNN4/6, the second one CNN4/6-mix.

The results for the noisy scenario for the R, P and T peaks are depicted in Fig. 5.10, 5.11 and 5.12, respectively. For the R peak, the XQRS alg continues to be the best one. The addition of the noise for all the std values does not make any difference. I can say that it is the most robust alg against the noise. Also my own alg performs very well even when a noise with std = 0.1 is added. The difference between CNN4 and CNN4-mix is indiscernible. PanTom and Hamilton loses slightly more than 1 % in the worst scenario. The Wavelet alg is the only one that suffers from the insertion of noise, already for std = 0.03. We have also to consider that, as it is evident from Fig. 5.9, albeit the presence of high noise, the R peak is quite perceivable. This is why almost all the algs keep on performing very well.

The reasoning does not hold for the P and T peaks. In fact, now, all the algorithms, some more than others, have a clear drop in performances. This trend is more significant for the P peak. The Wavelet alg still shows a very prominent fall. In regard to my algorithm, now the CNN4-mix is behind CNN4 for 1-3 %. This is quite obvious since CNN4 is trained with signals with the same std. I can surmise that my algorithm is very robust against noisy signals, even as the noise's std is high. Into the bargain, CNN4/6-mix, although trained with noisy

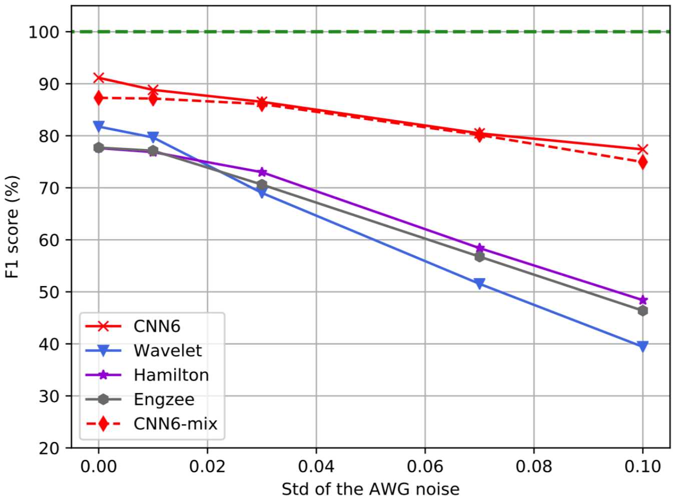

Figure 5.11: F_1 score vs std of the noise, P peak.

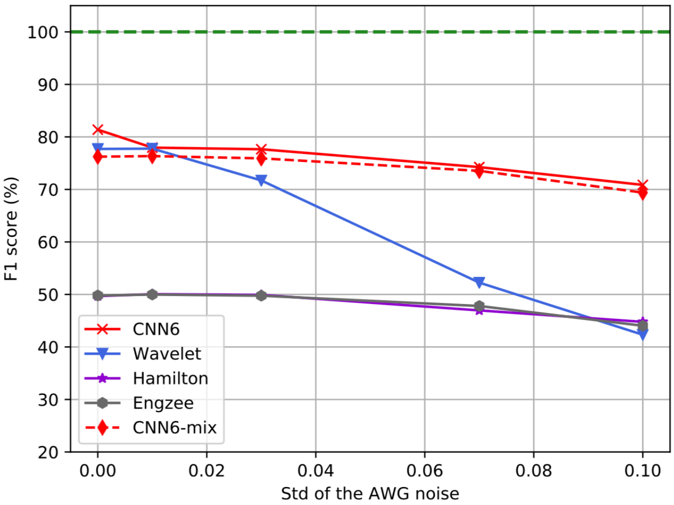

Figure 5.12: F_1 score vs std of the noise, T peak.

and noiseless signals, performs almost equal to the ones trained with a specific level of noise, thereby showing its high adaptation to a very diversified scenario.

6 Conclusions and Future Work

In this thesis I have proposed a brand-new Deep Learning-based ECG delineator. I have not found anything similar on the Net, thus I consider my work a novelty. In the field of ECG processing, CNNs have been used mainly for classification tasks, yet, in my work, I have showed that they can be used for the ECG delineation, too. I have implemented a thorough processing pipeline, starting off with the pre-processing stage, passing through the core of my thesis, the learning framework, and eventually finishing up with the post-processing leg.

The pre-processing stage is crucial for obtaining optimal results; still, it's been the most demanding and tedious part due to the issues related to the QT database.

One of the most interesting aspect of my thesis is that I have got very good results using very efficient Convolutional Neural Networks. An efficient CNN entails a small computational load and memory footprint, thereby permitting my algorithm to work in real-time scenarios. However, I have tried to add recent features to my CNNs, such as inverted residual and bottleneck, but they led no improvement. This is due to the fact that an ECG signal is not complex like a 3D image could be, hence a relatively non-deep CNN is able to capture entirely its features. A key feature for pulling out the P and T delineation with good results has been the use of atrous convolution, that has allowed the network to increase the receptive field without increasing the total number of parameters. Plus, my CNNs has the advantage of not depending on the input's length, thus not having that constraint of the standard CNNs.

Although the total number of signals was quite scant for a DL approach, the results obtained by my implementation are better than or very similar to the other state-of-the-art algorithms I have tested. For the P and T peaks, my algorithm outstrips the others. Furthermore, my algorithm performs very well also in noisy scenarios, being the best algorithm. This means that my algorithm is robust against the presence of severe noisy conditions.

6.1 Future Work

As future work, it will be interesting to add at least the individuation of the onset and offset of the QRS complex (i.e., Q and S peaks). Surely, this should be done in coordination with some deterministic approaches. Regarding the offset and onset of the P and T peaks, I esteem that is very difficult for the CNNs to learn their position, or at least they have to be supported with some kind of derivative/wavelet-based technique.

Another future development could be that of enriching the post-processing step in order to decrease even more the number of TPs. This stage does not rely on the CNNs, but it is a parametric phase.

In conclusion, it would be very interesting to see if the addition of more examples for the training phase would increase the global performances. This depends on whether new databases will be made public.

References

[1] A. Gacek, W. Pedrycz, "ECG Signal Processing, Classification and Interpretation: A comprehensive Framework of Computational Intelligence", 1st Edition, Springer Publishing Company, Incorporated, 2011.

[2] G. D. Clifford, F. Azuaje, P. McSharry, "Advanced Methods and Tools for ECG Data Analysis", Artech House, Inc. , Norwood, MA, USA, 2006.

[3] N. M. Arzeno, Z. Deng, and C. Poon, "Analysis of First-Derivative Based QRS Detection Algorithms", *IEEE Transactions on Biomedical Engineering*, Vol. 55, No. 2, February 2008.

[4] J. Pan, and W. J. Tompkins, "A Real-Time QRS Detection Algorithm", *IEEE Transactions on Biomedical Engineering*, Vol. BME-32, No. 3, March 1985.

[5] A. S. M. Koeleman, H. H. Ros, and T. J. Van Den Akker, "Beat-to-beat in- terval measurement in the electrocardiogram," Med. Biol. Eng. Comput., vol. 23, pp. 213–219, 1985.

[6] D. S. Benitez, P. A. Gaydecki, A. Zaidi, and A. P. Fitzpatrick, "A new QRS detection algorithm based on the Hilbert transform", *Computers in Cardiology* 2000, Vol. 27, No. Cat. 00CH37163, pp. 379-3824,20004.

[7] R. V. Andreão, and J. Boudy, "Combining Wavelet Transform and Hidden Markov Models for ECG segmentation", *EURASIP Journal on Applied Signal Processing*, 2007, Vol. 2007, No. 1, pp. 95-102.

[8] R. V. Andreão, B. Dorizzi, and J. Boudy, "ECG Signal Analysis Through Hidden Markov Models", *IEEE Transactions on Biomedical Engineering*,Vol. 53, No. 8, August 2006.

[9] P. Laguna, R. Jané, and P. Caminal, "Automatic Detection of Wave Boundaries in Multilead ECG Signals: Validation with the CSE Database", *Comput. Biomed. Res. 27*, 1 February 1994, 45-60. DOI=http://dx.doi.org/10.1006/cbmr.1994.1006

[10] P. E. Trahanias, "An Approach to QRS Complex Detection Using Mathematical Morphology", *IEEE Transactions on Biomedical Engineering*, Vol. 40, No. 2, February 1993.

[11] Y. Sun, K. L. Chan, and S. M. Krishnan, "Characteristic wave detection in ECG signal using morphological transform", *BMC Cardiovascular Disorders*, 2005.

[12] A. M. Rodrigo, R. Alcaraz, and J. J. Rieta, "Automatic Electrocardiogram Delineator Based on the Phasor Transform of Single Lead Recordings", *Computing in Cardiology*, 2010.

[13] J. M. Bote, J. Recas, F. Rincó, D. Atienza, and R. Hermida, "A Modular Low-Complexity ECG Delineation Algorithm for Real-Time Embedded Systems", *IEEE Journal of Biomedical and Health Informatics*, Vol. 22, No. 2, March 2018.

[14] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, "A Wavelet-Based ECG Delineator: Evaluation on Standard Databases", *IEEE Transactions on Biomedical Engineering*, Vol. 51, No. 4, April 2004.

[15] M. Cesari, J. Mehlsen, A. -B. Mehlsen, and H. B. D. Sorensen, "A New Wavelet-Based ECG Delineator for the Evaluation of the Ventricular Innervation", *IEEE J. Transl. Eng. Health Med.* , Vol. 5, Jul. 2017, Art. no. 2000215.

[16] L. Y D. Marco, and L. Chiari, "A Wavelet-Based ECG Delineation Algorithm for 32-bit Integer Online Processing", *BioMedical Engineering OnLine*, 2011.

[17] F. Rincón, J. Recas, N. Khaled, and D. Atienza, "Development and Evaluation of Multilead Wavelet-Based ECG Delineation Algorithms for Embedded Wireless Sensor Nodes", *IEEE Transactions on Information Technology in Biomedicine*, Vo. 15, No. 6, November 2011.

[18] S. D. Greenwald, "Improved Detection and Classification of Arrhythmias in Noise-Corrupted Electrocardiograms using Contextual Information", Thesis (Ph. D.), Harvard University, MIT Division of Health and Technology, Program in Medical Engineering and Medical Physics, 1990.

[19] B. Köhler, C. H. Hennig, and R. Orglmeister, "The Principles of Software QRS Detection", *IEEE Engineering in Medicine and Biology*, 2002.

[20] L. Chun-Lin, “A Tutorial of the Wavelet Transform”, NTUEE, Taiwan, 2010.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder architecture for image segmentation”, *IEEE Trans. Pattern Anal. Mach. Intell.* , Vol. 39, No. 12, pp. 2481-2495, Dec. 2017.

[22] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation”, arXiv:1706.05587, 2017.

[23] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”, arXiv:1606.00915, 2017.

[24] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs”, arXiv:1412.7062v4, 2016.

[25] F. Yu, and V. Koltun, “Multi-Scale Context Aggregation by Dilated Convolutions”, arXiv:1511.07122v3, 2016.

[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour Detection and Hierarchical Image Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vo. 33, Issue 5, May 2011.

[27] S. Ioffe, and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariance Shift”, arXiv:1502.03167v3, 2015.

[28] K. Simonyan, and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, arXiv:1409.1556v6, 2015.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”, arXiv:1512.03385v1, 2015.

[30] A. Howard, M. Zhu, B. Chen, et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv:1704.04861v1, 2017.

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, arXiv:1801.04381v4, 2019.

[32] S. H. Jambukia, V. K. Dabhi, and H. B. Prajapati, "Classification of ECG signals using Machine Learning Techniques: A Survey", *ICACEA*, IMS Engineering College, Ghaziabad, India.

[33] W. Ahmed, and S. Khalid, "ECG Signal Processing for Recognition of Cardiovascular Diseases: A Survey", *INTECH*, 2016.

[34] I. Beraza, and I. Romero, "Comparative study of algorithms for ECG segmentation", *Biomedical Signal Processing and Control*, Vo. 34, pp. 166-173, 2017.

[35] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning", *Proceedings of the* 30*th International Conference on Machine Learning*, Atlanta, Georgia, Usa, 2013.

[36] https://www.thoughtco.com/heart-nodes-anatomy-373242

[37] P. Laguna, R. G. Mark, A. Goldberger, and G. B. Moody, "A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG", *IEEE Computers in Cardiology*, pp. 673-676, 1997.

[38] Massachusetts Institute of Technology. MIT-BIH ECG database. Available: http://ecg.mit.edu/.

[39] S. Mallat, "Multifrequency channel decompositions of images and wavelet models", *IEEE Trans. Acoust. Signal Processing*, Vol. 37, pp. 2091–2110, Dec. 1989.

[40] A. Cohen, and J. Kovačević, "Wavelets: The mathematical background", *Proc. IEEE*, Vol. 84, pp. 514–522, April 1996.

[41] https://neurokit.readthedocs.io/en/latest/

[42] https://wfdb.readthedocs.io/en/latest/processing.html

[43] https://ecg-kit.readthedocs.io/en/master/index.html

[44] P. Hamilton, "Open Source ECG Analysis", 2012.

[45] P. S. Hamilton, W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database", IEEE Trans. Biomed Eng. , 1986, pp. 1157-65.

[46] W. A. H. Engelse, C. Zeelenberg, “A single scan algorithm for QRS-detection and feature extraction”, Computers in cardiology, 6(1979), 37-42.

[47] A. Lourenço, H. Silva, P. Leite, R. Lourenço, A. L. Fred, “Real Time Electrocardiogram Segmentation for Finger based ECG Biometrics”, in Biosignals (pp. 49-54), February 2012.

[48] I. I. Christov, “Real time electrocardiogram QRS detection using combined adaptive threshold”, Biomed. Eng. Online, 3(1), 28, 2004.