QRS Complex Detection in Paced and Spontaneous Ultra-High-Frequency ECG

Zuzana Koscova[1], Adam Ivora[1], Petr Nejedly[1], Josef Halamek[1], Pavel Jurak[1], Magdalena Matejkova[2], Pavel Leinveber[2], Lucie Znojilova[3], Karol Curila[3], Plesinger Filip[1]

[1]Institute of Scientific Instruments of the Czech Academy of Sciences, Brno, Czechia
[2]ICRC at St. Anne´s University Hospital, Brno, Czechia
[3] Cardiocenter of University Hospital Kralovske Vinohrady, Prague, Czechia

Abstract

Background: *Analysis of ultra-high-frequency ECG (UHF-ECG, sampled at 5,000 Hz) informs about dyssynchrony of ventricles activation. This information can be evaluated in real-time, allowing optimization of a pacing location during pacemaker implantation. However, the current method for real-time QRS detection in UHF-ECG requires suppressed pacemaker stimuli.*

Aim: *We present a deep learning method for real-time QRS complex detection in UHF-ECG.*

Method: *A 3-second window from V1, V3, and V6 lead of UHF-ECG signal is standardized and processed with the UNet network. The output is an array of QRS probabilities, further transformed into resultant QRS annotation using QRS probability and distance criterion.*

Results: *The model had been trained on 2,250 ECG recordings from the FNUSA-ICRC hospital (Brno, Czechia) and tested on 300 recordings from the FNKV hospital (Prague, Czechia). We received an overall F1 score of 97.11 % on the test set.*

Conclusion: *Presented approach improves UHF-ECG analysis performance and, consequently, could reduce measurement time during implant procedures.*

1. Introduction

Ultra-high-frequency ECG (UHF-ECG, passband up to 1,000 Hz) has been shown to describe electrical ventricular dyssynchrony of the heart [1, 2]. Also, a significant association of UHF-ECG analysis results with the effect of cardiac resynchronization therapy (CRT) [3] was shown. However, since the location of the pacing lead affects the impact of the therapy, we developed a software called VDI vision (version 1.0) in the past; the VDI vision analyses the UHF-ECG signals in real-time.

The real-time UHF-ECG analysis (refined with each incoming beat, usually 100-200 heartbeats are required) can describe the effect of pacing on ventricles activation even during pacemaker implantation [4].

This paper will focus on improving the entry point into the UHF-ECG analysis - the QRS detection.

Although the QRS detection in ECG signals is a very well mapped area, UHF-ECG (Fig. 1) is specific due to high sampling frequency. Therefore, when QRS complexes are paced, pacing artifacts (Fig. 1 - triangles) have a much higher amplitude than in regular ECG. This behavior might confuse QRS detection unless the pacing stimuli are suppressed or removed. Consequently, the current solution for real-time QRS detection in the VDI vision is computationally intensive due to removing these pacing artifacts [5]. Also, its F1 performance at 90% could be further improved since non-detected QRS prolong measurement time while false QRS detections result in excessive computations during QRS morphology clustering [6]. Therefore, to improve real-time UHF-ECG analysis performance, it is beneficial to reduce the preprocessing pipeline required for QRS detection. Nevertheless, it is also important to improve the QRS detector's performance, which would shorten measurement time, finally leading to quicker optimization of pacing and, therefore, shorter surgery.

Here, we present a deep learning method for QRS complex detection in paced and spontaneous UHF-ECG.

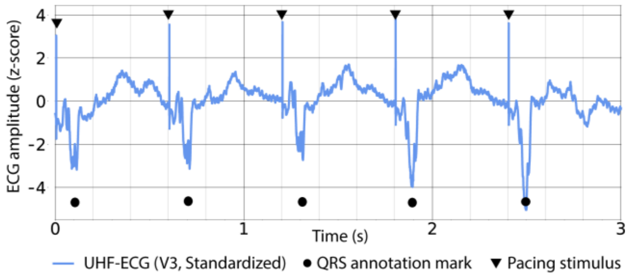


Figure 1. Standardized UHF-ECG signal with bipolar septum stimulation from V3 lead with pacing stimuli (triangles). Circles point to QRS complexes.

2. Data

In this study, we used two independent datasets; both recorded at 5,000 Hz. The dataset used for training and validation contained 12-lead UHF-ECG signals from the FNUSA hospital (Brno, Czechia) from 78 healthy and 942 CRT subjects (before and after implantation). 2,250 ECG records acquired from 780 subjects were used as a training set, and 768 ECG records acquired from 240 subjects were used as a validation set. The dataset contained records with spontaneous and stimulated QRSs (mostly different settings of biventricular pacing). The length of the recordings was 2-15 minutes. QRS annotation marks for this dataset were produced automatically with our previous UHF-ECG solution, incapable of real-time processing (UHF-Solver software).

The second dataset (FNKV hospital, Prague, Czechia) was used for independent testing. The test set contained 300 records acquired from 47 subjects to pacemaker implantation (before, during, or after the procedure). These records contained spontaneous QRSs (57 records from 11 subjects) and paced QRSs (247 records from 47 subjects). FNKV subjects were mostly treated by His bundle or Parahisian stimulation. Multiple recordings per patient were acquired for different positions of the pacing electrode. QRS annotation marks for the test set were manually prepared in SignalPlant software [7].

Output vectors for each recording were based on QRS annotation marks. Samples up to proximity 0.05 ms (250 samples) to the left and right side from QRS annotations were marked as QRS class (value 1); hence all the other samples belonged to a background class (value 0). Although 12-lead ECGs were recorded, we used only precordial leads V1, V3, and V6 to make the model as lightweight as possible.

3. Method

The core of this work is a deep-learning model to detect QRS. We selected a convolutional neural network with UNet architecture [8, 9] for this task (Fig. 2). Because the intended use is for real-time QRS detection, we decided on using a 3-second input window.

The neural network processes ECG signal from V1, V3, and V6 precordial chest leads, meaning the input array of size 3x15,000 samples (3 leads x 3 seconds by 5,000 samples per second).

The output of the network is an array of size 2x15,000 (QRS and non-QRS class x 3 seconds by 5,000 samples per second) samples representing the background and QRS complex probability of each examined sample.

3.1. Preprocessing and data augmentation

The described method is designed to work in real-time with minimal signal preprocessing. Therefore, each lead was transformed to a z-score only. We augmented the dataset by simulating higher heart rates. For every record, a distant 6-second window was used to prepare an artificial, 3-second signal with twice as fast heart rate: the signal was downsampled from 5,000 to 2,500 Hz using Butterworth antialiasing filter (the fifth-order with a cut-off frequency of 1,200 Hz, and decimation with factor 2).

3.2. The neural network architecture

The difference between the proposed architecture and the original UNet architecture is in different hyperparameters in convolution and MaxPooling layers and, most importantly, in using 1D convolution instead of 2D convolution.

The UNet architecture of the convolutional neural network (Fig. 2) consists of two symmetric sections: The contraction (Fig. 2 - left) and the expansion section (Fig. 2 - right).
The contractions section is made of 4 contraction blocks. Each of these blocks contains 2 convolution layers (kernel size = 9, stride = 1, padding = 4), batch normalization and ReLU activation function followed by MaxPooling layer (kernel size = 2, stride = 2). After each block, the signal length is downsampled with a factor of two and after first three blocks the number of feature maps doubles.

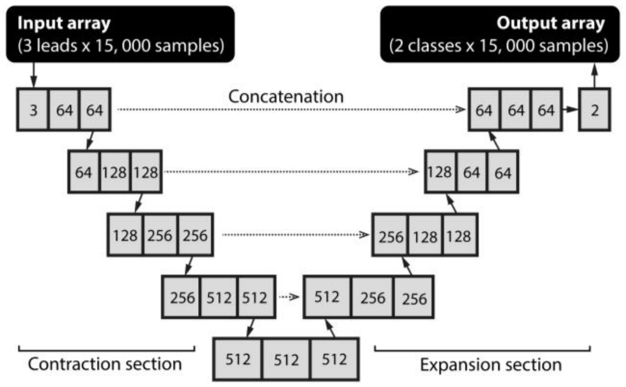


Figure 2. The architecture of UNet CNN. Rectangles with digits represent a number of feature maps in each block. Input leads refer to V1, V3, and V6; output classes refer to sample probability of being QRS or non-QRS.

Similar to the contraction section, the expansion section contains four expansion blocks. Each of these blocks includes a transpose convolution layer (kernel size = 9, stride = 2) which upsamples feature maps length with the factor of two. Transpose convolution layer is followed by two convolution layers (kernel size = 9, stride = 1, padding = 4) with batch normalization and ReLU activation function. Output from each block is concatenated with output from a corresponding block in the contraction section, and after the first three expansion blocks feature maps number is cut in half.
The final layer of the network is the convolution layer (kernel_size = 1, stride = 1) with a softmax activation function that outputs the class probabilities. We used Python [10] and PyTorch [11] for developing the network.

3.3. Model Training

The model has been trained for 30 epochs, using Adam optimization with a learning rate of 0.0001. Weighted cross-entropy loss function was used due to imbalanced output classes. We used a graphic-processing unit (GPU) with "Compute Unified Device Architecture" (CUDA) for training (GeForce RTX 2080 Ti).

3.4. Postprocessing

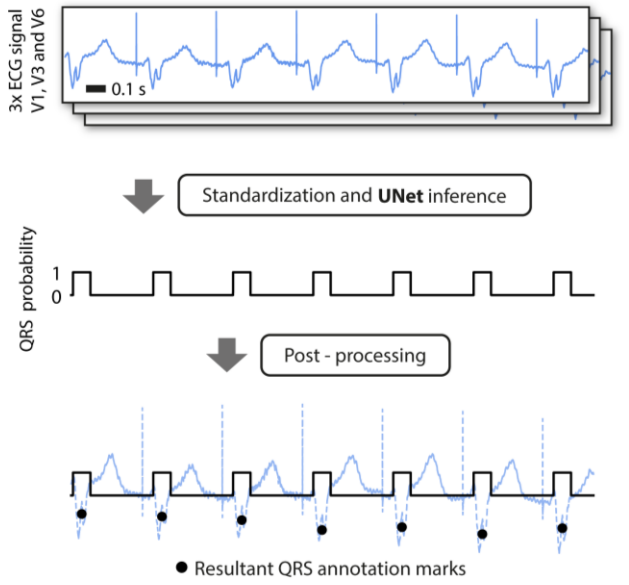
The network output is an array (2x15,000), representing the class probability of each sample. The first class is considered a background (i.e., not QRS), and the second a QRS complex (Fig.3-middle). The resultant QRS annotations are based on QRS probability and distance criterion. Segments of samples with QRS probabilities higher than threshold 0.955 (empirically determined on validation dataset) and longer than 50 samples (i.e., 0.01 seconds) were classified as QRS. The final positions of QRS were calculated as a midpoint of detected segments.

4. Results

The model performance on train, validation and test set is shown in Table 1. The test set results are presented separately for records with paced and non-paced QRS.

Table 1. Model results for used datasets. N_{QRS} refers to the total number of QRS in records. Se represents sensitivity, PPV is a positive predictive value, and F1 is F1-score.

Dataset	N_{QRS}	Se [%]	PPV [%]	F1 [%]
Train	7,904	98.99	97.42	98.19
Validation	2,745	98.36	96.95	97.65
Test (total)	1,480	96.49	97.74	97.11
Test (spont.)	214	97.20	95.41	96.30
Test (paced)	1,266	96.37	98.15	97.25


3x ECG signal V1, V3 and V6
0.1 s
Standardization and UNet inference
QRS probability
1
0
Post - processing
● Resultant QRS annotation marks
Figure 3. Top - original ECG signal (V3 lead). The signal (ECG leads V1, V3, and V6) is standardized and pushed into the UNet model, resulting in a QRS probability vector (middle). Post-processed output probability results in QRS annotation marks (bottom).

Furthermore, the proposed model performance was compared to the method already used in VDI vision (up to version 1.0) and to the Pan-Tompkins algorithm [12] for QRS detection (Tab. 2).

Table 2. Comparison of performance for specific QRS detection models on the test set (300 records from 47 patients). Results are reported for spontaneous (Spont.) and paced records and for the entire test set (Total). VDI refers to the detection method used in software VDI vision version 1.0 or earlier.

Model	Record type	Se [%]	PPV [%]	F1 [%]
UNet	Spont.	97.20	95.41	96.30
	Paced	96.37	98.15	97.25
	Total	96.49	97.74	97.12
VDI	Spont.	92.52	94.29	93.40
	Paced	86.97	93.07	89.91
	Total	87.77	93.25	90.43
PanTompkins	Spont.	81.31	75.98	78.56
	Paced	29.51	28.55	29.02
	Total	37.1	35.7	36.39

We measured inference time for a single 3-second ECG block using the target framework (C# and .NET with ONNX runtime ver. 1.7.1). The average computing time (100 trials) was 757 ± 24 and 56 ± 1 milliseconds for CPU (Intel Xeon Silver 4116 at 2.1 GHz) and GPU (CUDA acceleration, GTX 1080 Ti), respectively.

5. Discussion

The proposed deep learning model achieved higher performance (F1 score of 96%) on the unseen test data than the QRS detector from VDI vision (F1 score of 90%). An important benefit of the presented solution is significantly increased sensitivity from 87 to 96% (Tab. 2 - Paced of UNet and VDI), suggesting that the average time for UHF-ECG analysis can be shortened. We also compared the Pan Tompkins detector implemented in "py-ecg-detectors" Python package version 1.0.2. The Pan Tompkins achieved an F1-score on the test set of 29.02% and 78.56% for stimulated and spontaneous activity, respectively. However, it should be noted that the Pan Tompkins method was not designed for UHF-ECG signals with stimulated QRS, which refers to a nearly 50% difference in F1-score between paced and spontaneous signals (Table 2).

Table 2 shows that the performance of the presented model is higher for spontaneous QRS complexes than for stimulated ones. This difference reflects the distribution of spontaneous and paced records in the training set. We also investigated the most common reason for false negatives (FN). The most non-captured QRS are ventricular fusion beats, with random stimuli in their descending edge. However, for further UHF-ECG analysis, these FN detections have no negative impact since only the major morphology group of QRS is analyzed; if captured, fusion beats would be excluded from the analysis.

We showed that real-time model inference is possible but should be supported by GPU with CUDA acceleration.

6. Conclusion

We described a deep-learning model to detect QRS complexes in ultra-high-frequency ECG with an intentional focus on pacing. We used data from two independent centers (FNUSA-ICRC, Brno, Czechia for training and validation; FNKV, Prague, Czechia for testing) to build a unique multicentric dataset covering a variety of diseases connected to treatment by pacemakers. Thanks to minimal signal preprocessing and high detection performance, the presented model is likely to be implemented in the next generation of real-time VDI vision software. The most beneficial effect for patients is shorter measurement time, which is essential during implant procedures.

Acknowledgments

The research was supported by a research grant no. FW01010305 and FW03010434 by the Czech Technological Agency and by the Czech Academy of Sciences (project RVO:68081731).

References

[1] P. Jurak, J. Halamek, J. Meluzin et al., "Ventricular dyssynchrony assessment using ultra-high frequency ECG technique," in Journal of Interventional Cardiac Electrophysiology, vol. 49, pp. 245-254, 2017

[2] P. Jurak, K. Curilla, P. Leinveber et al., "Novel ultra-high-frequency electrocardiogram tool for the description of the ventricular depolarization pattern before and during cardiac resynchronization," in Journal of cardiovascular electrophysiology, pp. 300-307, 2020

[3] F. Plesinger, P. Jurak, J. Halamek, P. Nejedly et al., "Ventricular electrical delay measured from body surface ECGs is associated with cardiac resynchronization therapy response in left bundle branch block patients from the MADIT-CRT Trial (Multicenter Automatic Defibrillator Implantation-Cardiac Resynchronization Therapy)," in Circulation: Arrhythmia and Electrophysiology, 2018

[4] K. Curila, P. Jurak, J. Halamek et ac., "Ventricular activation pattern assessment during right ventricular pacing: Ultra-High-Fequency ECG study," in Journal of Cardiovascular Electrophysiology, Mar. 2021

[5] P. Andrla, F. Plesinger, J. Halamek et ac., "A method for removing pacing artifacts from Ultra-High-Frequency Electrocardiograms," in CinC, 2018

[6] F. Plesinger, J. Jurco, J. Halamek, P. Leinveber, T. Reichlova, P. Jurak, "Multichannel QRS morphology clustering data preprocessing for Ultra-High-Frequency ECG analysis ," in Cardiotechnix, pp. 11-19, 2015

[7] F. Plesinger, J. Jurco, J. Halamek, and P. Jurak, "SignalPlant: an open signal processing software platform," Physiol. Meas., vol. 37, no. 7, pp. N38– N48, 2016

[8] O. Ronneberger, P. Fischer, T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in ArXiv, 2015

[9] V. Moskalenko, N. Zolotykh, G. Osipov, "Deep learning for ECG segmentation," in Advances in Neural Computation, Machine Learning, and Cognitive Research III, Sep.2019

[10] G. Van Rossum, F.L. Drake, "Python 3 Reference Manual," in CreateSpace, 2009

[11] Paszke, Adam and Gross, Sam and Massa et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems 32, 2019

[12] J. Pan, W. J. Tompkins, "A real-time QRS detection algorithm," in IEEE Transactions of Biomedical Engineering, vol. BME-32, no. 3, Mar. 1985

Address for correspondence:

Zuzana Koscova
Kralovopolska 147, Brno, Czech Republic.
zkoscova@isibrno.cz