Deep learning based electrocardiogram peak detector algorithm

Margarita Sharkova*

April 29, 2023

Abstract

The proposed approach takes as input an ECG signal and outputs a binary mask indicating the location of the R-peak.

1 Introduction

Electrocardiogram (ECG) signal describes the activity of the myocardium over the time. In a typical ECG setup, depending on the particular application, three to twelve signal receptive electrodes are attached to the patient's body. One of the methods is R-peak detection [Figure 1]. The correct analysis of patient's ECG records is essential for the diagnosis of cardiovascular diseases. This R-peak localization is critical for arrhythmia diagnoses such as atrial premature contraction, tachycardia, and bradycardia [2]. Also, it is common to use peak location as a step for extracting advanced features and estimation of the heart rate (HR).

The challenging part of working with ECG signal is unexpected noise, such as baseline drift, electrode motion and stretching, motion artifacts, and muscle noise [11]. Consequently, noise removal is the preliminary issue to consider in ECG signal processing.

Another important thing to consider is R-peak detectors vary in their effectiveness. R-peak is not always aligned in the center of QRS complex. This is because the R waves are not at constant distances and the the idea of finding the position of the R wave and then generalizing the peak globally for all signal leads could help [14].

1.1 Related works

The existing approaches could be divided into 3 groups: classical statistical approaches with thresholding, feature engineering approaches that could be extracted from signal and used as an input for classical machine learning algorithms, and NN approaches that use raw signals as an input with some modifications and utilizes the power of deep learning feature extraction.

*Corresponding author: shark.margaret@gmail.com

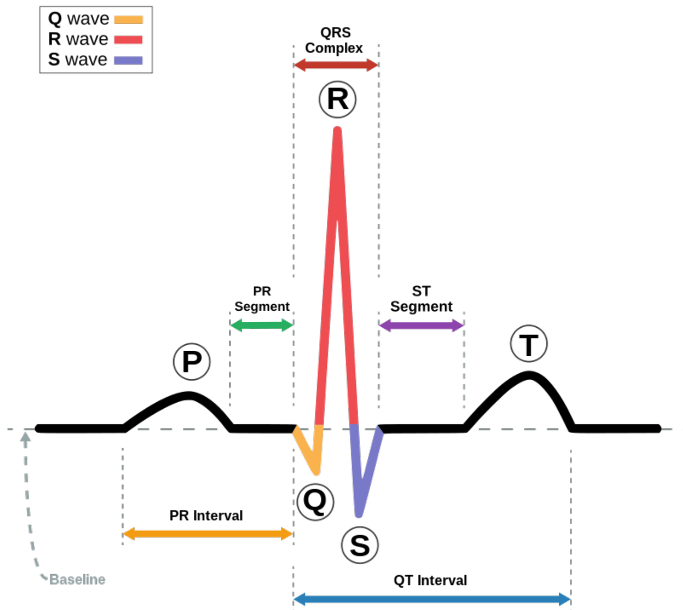

Figure 1: Schematic representation of a normal sinus rhythm ECG wave

1.1.1 Classical approaches

Thresholding methods suffer from the miscalculated threshold due to the noise events and require a certain number of previous data points to predict the threshold [1].

- *Peak detection method.* This method involves finding the local maxima in the ECG signal. The peak detection algorithm searches for the highest point in a certain window of the ECG signal and identifies it as an R-peak. This method is relatively simple and fast but can be sensitive to noise and baseline wander.
- Detection models, such as Pan-Tompkins [8], Hamilton [4], and Christov algorithms that consist of filtering noise, enhancing peaks, and adaptive thresholding. They are widely used due to their robustness and high accuracy.
- Transform-based methods (waveform derivatives, Hilbert transform, discrete wavelet transform (DWT), stationary wavelet transform (SWT) [10, 5]): The transforms to frequency domain are used to analyze the frequency content of the ECG signal, and the R-peaks are identified as the maximum values in the transformed signal. These methods are robust to noise and can handle ECG signals with varying morphologies but applied adaptive thresholding is tricky in tuning

Feature-based approaches also exist for this task. In Al ZM et al. (2021) [1] study have been calculated twenty-five features, i.e., four features related to the peak, two from a distance, eighteen statistical features from each peak's window, and one from estimated noise. Then Sequential Forward Selection (SFS) method was used to find relevant features for better classification. SFS is a wrapper-approach method based on the greedy search algorithm that removes irrelevant features based on criterion values. Further classification model like support vector ma- chine (SVM) could been employed for ECG signal classification [7]. The performance would be much lower than that by a cardiologist due to the high complexity of classification tasks and the poor feature extraction.

Template matching method: This method involves comparing the ECG signal with a predefined template of an R-peak like in [9]. The template matching algorithm searches for the highest correlation between the ECG signal and the template and identifies it as an R-peak. This method is computationally intensive and requires a high-quality template.

1.2 Deep Learning based Methods

Let's dive into the latest SOTA DL approaches for R-peak detection and disease classification on ECG data as they are similar. The task of detecting R-peaks can be approached as the creation of a one-dimensional segmentation map that identifies the location of R-peaks within the input. Most existing approaches for ECG signal classification use Recurrent Neural Network models, e.g., LSTM and GRU, which are unable to extract accurate features for such long sequences. Using the 1-Dimensional Convolutional Neural Network resolves this problem but most of the approaches use only 1 lead information from ECG signal. Zahid et al. (2021) [12] utilize 1D CNN integrated with a verification model to reduce the number of false alarms. Using F1-score as an evaluation metric, the network reaches a score of 99.83.

Mehmood et al. (2020) developed deep learning model for R-peak detection using stationary wavelet transform (SWT) and separable convolution. Also, they pointed out that previous approaches reported performance in a single database and had a bad generalization for different data. Thus to overcome these complications they used in training and testing MIT-BIH Arrhythmia, the Institute of Cardiological Technics [INCART], and the QT databases. Separable convolution with atrous spatial pyramidal pooling was selected as the model's architecture, and noise-augmented waveforms of 5.69 s duration (2048 size in 360 Hz) were used in training. The F1 score model was 90.7 with the MIT-BIH.

Since CNN models are effective in the spatial feature extraction, and RNN models are effective in the time series feature extraction, Murugesan et al. (2018) [6] propose to combine CNN and RNN modules for feature extraction. They apply wavelet-based filtering and split the ECG signal to a fixed-length based on R-peaks. And then put 1-D sequence data into classic CNN model and adopted RNN model to further learn features from the output.

Zhang et al. (2020) [13] proposed to treat the multi-lead signal as a 2-dimensional matrix and combine multi-scale 2-D convolution blocks with 1-D convolution blocks for feature extraction. Their proposed model achieves 99.2% F1-score in the MIT-BIH dataset on 5-class classification.

2 Methods

2.1 Database

Applying deep learning techniques (DL) in ECG signals was a big challenge due to the limited access to the data. Large publicly available electrocardiography PTB-XL dataset was published in April 2020. Nowadays for training such databases as Physionet, PhysioBank, and PhysioToolkit datasets could be used.

For new approach testing MIT-BIH Arrhythmia Database would be taking for conducting experiments. This dataset has been widely used as a benchmark and it is the most popular clinical ECG database for the evaluation of peak detection algorithms [12].

The MIT-BIH comprises 48 two-lead electrocardiograms (ECGs) sampled at 360 Hz and each record covers 30 minutes. The corresponding beat annotations are provided for the Lead II ECG signal. This lead is labeled as MLII in the database. The annotations include the beat type (such as N for normal beat, V for premature ventricular contraction, A for atrial premature beat).

2.2 Preprocessing step

For the case of using multiple databases that contained various types of noise filtering methods to remove unwanted features should be applied. In my approach, I propose to apply DWT to attenuate the baseline wandering by a decomposition down to level 9 and lowpass filter with a cutoff of 40 Hz to remove powerline noise. The main dilemma is that depending on the application and the nature of the signal, a certain level of noise may be acceptable or even desirable. In some cases, removing all noise from a signal may lead to overfitting or loss of important information. So, the model training and testing should be tried on both versions of data.

For validation, the quality of filtering Signal-to-noise ratio (SNR) could be applied. SNR is a measure of how much the signal power exceeds the noise power in a signal. The calculations are done on the original and filtered ECG signal with the added Gaussian noise (Table 1). The increase in SNR from 5.30 dB to 7.32 dB indicates that the signal has become stronger relative to the noise by reducing the level of noise relative to the desired signal. The smoothed and the original signal is displayed in Figure 1

SNR of noisy signal	5.30 dB
SNR of filtered signal	7.32 dB

Table 1: Comparison of SNR values between noisy and filtered signals

The signal normalization between -1 and 1 is applied before the training stage and used for the test inference. Additionally, the test set is resampled to prevent boundary effects on frames.

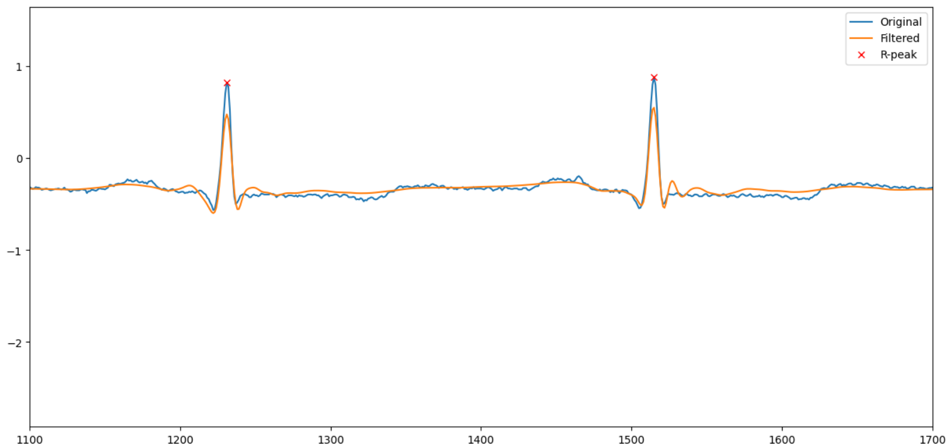

Figure 2: Comparing the original signal with filtered one(DWT with level 9 and lowpass with a cutoff of 40 Hz)

2.3 Augmentation

Noise augmentation technique should be applied to make training robust and reduce overfitting. Applying the dropout technique while training NN also should be tried during experiments to avoid overfitting. Possible methods of augmentation:

- add different types of noise (Gaussian noise, Shot noise, Salt-and-pepper noise, Speckle noise, Pink noise, Quantization noise)
- applying a time-stretching filter on waveform level [3]
- vary the frequency of waveforms and add Gaussian noise up to $\pm 10\%$ [3]
- multiplying of waveforms by sinusoidal waveforms to represent different peak amplitudes

Application of different noises with 50% probability allows model to avoid classification noises as peaks and reduce false-positive peaks. These techniques can help improve the robustness of the UNet network to different types of ECG signals.

2.4 Metrics

The approach performance can be evaluated with precision, recall and F1 score. The tolerance boundary of 150 ms was applied for truth peak location definition. Since the in focus of detection are the R-peaks and not non-R-peaks, True Negatives are not considered as a relevant performance measure for this task.

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{1}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{2}$$

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

2.5 Training

Records were divided into training, validation, and test data at the ratios of 70%, 15%, and 15%. Further, for improving the quality of the research cross validation should be applied. Weight initialization for better convergence was automatically applied automatically by the PyTorch framework with He Initialization method.

The determination of early stopping was based on detecting a non-decrease in the validation loss with patience of 5. Adam and binary cross-entropy (BCE) loss were selected as the training optimizer and loss function, respectively. The models are trained for 100 epochs with a batch size of 4.

2.6 Proposed DL model

The architecture is designed to learn a mapping from a 1D input signal to a 1D segmentation mask, with the intermediate convolutional layers extracting high-level features that are used to generate the mask in the up-stack. The network has 6 downsampling blocks and 5 upsampling blocks with skip connections between corresponding layers of both stages. The final layer of model takes a 32-channel input signal and produces a 1-channel output signal, passed through a sigmoid activation function.

Downsampling stage: each block consisting of a 1D convolutional layer with a specified kernel size, stride, and padding, followed by batch normalization. The first Downsample block takes a 1-channel input signal and produces a 16-channel output signal, while subsequent blocks double the number of output channels until the final block produces a 64-channel output.

Upsampling stage: each block consist of a transposed 1D convolutional layer, followed by batch normalization. The first Upsample block takes a 64-channel input signal and produces a 64-channel output signal, while subsequent blocks halve the number of input channels until the final block produces a 16-channel output. The detailed model summary is provided in Section 5 in Figure 3.

2.7 Postprocessing

For the reduction of False Positive, an additional validation model could be applied. The model would take detected R-peak locations as input and outputs a probability score for each detected R-peak location as it was realized in this reference paper (Zahid et al., 2021) [12]. The main principle is based on a timing criterion. The criterion checks if the predicted locations of two beats fall within a window of 300 milliseconds and if so, one of the beats is identified as a false alarm.

3 Results

Confusion matrix and perfomance metrics have been calculated on raw data without preprocessing, then with noise filtering.

Table 2: Correlation matrices for test set

Database	**TP**	**FN**	**FP**
test set	20520	25	24786
filtered test set	20545	0	42333

Table 3: Peak detection performances

Methods	**Database**	**Precision**	**Recall**	**F1-Score**
Zahid et al. (2021)	MIT-BIH	99.82	99.85	99.83
Mehmood et al. (2020)	MIT-BIH	99.69	83.19	90.7
UNet 1D	test set	46.16	99.89	62.53
UNet 1D	filtered test set	32.69	100.0	49.17

Since the methods for performance comparison proposed on 3 rely on similar 1D CNN architecture and implementation, we can utilize their scores as a benchmark to enhance the performance of our method.

4 Conclusion

Based on the results, the best model proposed model in this report gives a lot of false positives and should be revised additionally to avoid mislabeling. The different augmentation methods should be tried and the padding methods for the input signal should be checked additionally. Also, the less deep architecture should be checked and the opportunity of validation model for the CNN output should be tried.

5 Appendix

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv1d-1            [-1, 16, 3616]             160
        Downsample-2            [-1, 16, 3616]               0
            Conv1d-3            [-1, 16, 1808]           2,320
       BatchNorm1d-4            [-1, 16, 1808]              32
        Downsample-5            [-1, 16, 1808]               0
            Conv1d-6             [-1, 32, 904]           3,104
       BatchNorm1d-7             [-1, 32, 904]              64
        Downsample-8             [-1, 32, 904]               0
            Conv1d-9             [-1, 32, 452]           6,176
      BatchNorm1d-10             [-1, 32, 452]              64
       Downsample-11             [-1, 32, 452]               0
           Conv1d-12             [-1, 64, 226]           6,208
      BatchNorm1d-13             [-1, 64, 226]             128
       Downsample-14             [-1, 64, 226]               0
           Conv1d-15             [-1, 64, 113]          12,352
      BatchNorm1d-16             [-1, 64, 113]             128
       Downsample-17             [-1, 64, 113]               0
  ConvTranspose1d-18             [-1, 64, 226]          12,352
      BatchNorm1d-19             [-1, 64, 226]             128
         Upsample-20             [-1, 64, 226]               0
  ConvTranspose1d-21             [-1, 32, 452]          12,320
      BatchNorm1d-22             [-1, 32, 452]              64
         Upsample-23             [-1, 32, 452]               0
  ConvTranspose1d-24             [-1, 32, 904]          12,320
      BatchNorm1d-25             [-1, 32, 904]              64
         Upsample-26             [-1, 32, 904]               0
  ConvTranspose1d-27            [-1, 16, 1808]           6,160
      BatchNorm1d-28            [-1, 16, 1808]              32
         Upsample-29            [-1, 16, 1808]               0
  ConvTranspose1d-30            [-1, 16, 3616]           4,624
      BatchNorm1d-31            [-1, 16, 3616]              32
         Upsample-32            [-1, 16, 3616]               0
  ConvTranspose1d-33             [-1, 1, 7232]             289
          Sigmoid-34             [-1, 1, 7232]               0
  Conv1DTranspose-35             [-1, 1, 7232]               0
================================================================
Total params: 79,121
Trainable params: 79,121
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.03
Forward/backward pass size (MB): 6.51
Params size (MB): 0.30
Estimated Total Size (MB): 6.84
```

Figure 3: Model summary

References

[1] Abdullah Al ZM, Keshav Thapa, and Sung-Hwan Yang. Improving r peak detection in ecg signal using dynamic mode selected energy and adaptive window sizing algorithm with decision tree algorithm. *Sensors (Basel)*, 21(19):6682, 2021.

[2] Redouane Benali, Fatiha B Reguig, and Zohra H Slimane. Automatic classification of heartbeats using wavelet neural network. *Journal of Medical Systems*, 36(2):883–892, 2012.

[3] Mehrdad Gandomi, Farhad Pourpanah, Mohammad Kalantarzadeh, and Hamid Hassannejad. From environmental sound representation to robustness of 2d cnn models against adversarial attacks. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 558–562. IEEE, 2021.

[4] P. Hamilton. Open source ecg analysis. In *Computers in Cardiology*, pages 101–104, 2002.

[5] Mohamed Merah, Tahar Ali Abdelmalik, and Boubakar H Larbi. R-peaks detection based on stationary wavelet transform. *Computer Methods and Programs in Biomedicine*, 121:149–160, 2015.

[6] Balamurali Murugesan, Vignesh Ravichandran, Keerthi Ram, Preejith S.P., Jayaraj Joseph, Sharath M. Shankaranarayana, and Mohanasankar Sivaprakasam. Ecgnet: Deep network for arrhythmia classification. In *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6, 2018.

[7] Jalal A Nasiri, Mahmoud Naghibzadeh, H Sadoghi Yazdi, and Bahram Naghibzadeh. Ecg arrhythmia classification with support vector machines and genetic algorithm. In *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, pages 187–192. IEEE, 2009.

[8] Jiapu Pan and Willis J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985.

[9] Mohammad Ebrahimi Panah, Mehdi Yaghini, Arash Ghaderzadeh, and Mohammad Bagher Shamsollahi. Utilizing ecg waveform features as new biometric authentication method. In *2021 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–7. IEEE, 2021.

[10] Jong-Seob Park, Seong-Wook Lee, and Uikyum Park. R peak detection method using wavelet transform and modified shannon energy envelope. *Journal of Healthcare Engineering*, 2017, 2017.

[11] Qian Qin, Jianqing Li, Yang Yue, and Chengyu Liu. An adaptive and time-efficient ecg r-peak detection algorithm. *Journal of Healthcare Engineering*, 2017:5980541, 2017.

[12] Muhammad Uzair Zahid, Serkan Kiranyaz, Turker Ince, Ozer Can Devecioglu, Muhammad E. H. Chowdhury, Amith Khandakar, Anas Tahir, and Moncef Gabbouj. Robust r-peak detection in low-quality holter ecgs using 1d convolutional neural network. *IEEE Transactions on Biomedical Engineering*, 69(1):119–128, 2022.

[13] Haozhen Zhang, Wei Zhao, and Shuang Liu. Se-ecgnet: A multi-scale deep residual network with squeeze-and-excitation module for ecg signal classification, 2020.

[14] Sebastian Śmigiel, Kacper Pałczyński, and Dominik Ledziński. Deep learning techniques in the classification of ecg signals using r-peak detection based on the ptb-xl dataset. *Sensors*, 21(24):8174, 2021.