Semantic Segmentation of 12-Lead ECG Using 1D Residual U-Net with Squeeze-Excitation Blocks

Konrad Duraj, Natalia Piaseczna*, Paweł Kostka and Ewaryst Tkacz

Department of Biosensors and Processing of Biomedical Signals, Faculty of Biomedical Engineering, Silesian University of Technology, Roosevelta 40, 41-800 Zabrze, Poland; kduraj@polsl.pl (K.D.); pkostka@polsl.pl (P.K.); etkacz@polsl.pl (E.T.)
* Correspondence: npiaseczna@polsl.pl

Abstract: Analyzing biomedical data is a complex task that requires specialized knowledge. The development of knowledge and technology in the field of deep machine learning creates an opportunity to try and transfer human knowledge to the computer. In turn, this fact influences the development of systems for the automatic evaluation of the patient’s health based on data acquired from sensors. Electrocardiography (ECG) is a technique that enables visualizing the electrical activity of the heart in a non-invasive way, using electrodes placed on the surface of the skin. This signal carries a lot of information about the condition of heart muscle. The aim of this work is to create a system for semantic segmentation of the ECG signal. For this purpose, we used a database from Lobachevsky University available on Physionet, containing 200, 10-second, and 12-lead ECG signals with annotations, and applied one-dimensional U-Net with the addition of squeeze-excitation blocks. The created model achieved a set of parameters indicating high performance (for the test set: accuracy—0.95, AUC—0.99, specificity—0.95, sensitivity—0.99) in extracting characteristic parts of ECG signal such as P and T-waves and QRS complex, regardless of the lead.

Keywords: ECG; signal segmentation; deep learning

1. Introduction

Electrocardiography (ECG) is a golden diagnostic standard for measuring the electrical activity of the heart. This signal has its reflection in the mechanical action of the heart and can inform us about the physiological condition of this organ. The analysis of an electrocardiogram is a complex process that requires specific knowledge. Engineers try to help physicians by creating expert systems. This process requires transfer of the specialist’s knowledge to a computer by feeding it with examples and a set of rules about how to treat specific cases. There are many possibilities of creation of an expert system—one of the most developed is using machine-learning-based technologies.

One of the subfields of machine learning methods is deep learning (DL)—containing many hidden layers in the Artificial Neural Network (ANN) structure, which mimics the performance of the human brain [1].

The convolutional neural network, CNN for short, is a specialized type of neural network designed for working with two-dimensional data. It creates rich representations of the input data by sequentially stacking the convolution operations over the image. To reduce the data size, CNNs are often equipped with pooling layers. The filter values are being updated using a backpropagation algorithm. Figure 1 presents a sample of 1D CNN.

These structures can be applied for analyses of physiological signals, such as ECG. The conventional deep convolutional networks are designed to operate exclusively on two-dimensional data. As described in [2], 1D CNNs are more advantageous than their 2D counterparts in dealing with one-dimensional data.
Data annotation is a time-consuming activity, so especially in the era of big data, systems that automatically (but also reliably) label the data are absolutely necessary.

**Related Work**

The application of deep learning techniques in processing different physiological signals was summarized by Faust et al. They analyzed 53 publications from 1 January 2008 to 31 December 2017, 17 concerning ECGs [3]. Another review of using DL methods for ECG data was performed by Hong et al. in [4]. They evaluated 191 articles published between 1 January 2010 and 29 February 2020. The authors of this work confirm that the application of DL methods in ECG analysis (including signal annotation) is becoming an increasingly frequently discussed topic.

The most frequently discussed issue in relation to the ECG signal is undoubtedly the anomaly detection. Novotna et al. used DL methods for premature ventricular contractions (PVCs) localization. On the CPSC2018 (China Physiological Signal Challenge 2018) database, they achieved the dice coefficient 0.947 for the model with a max pooling layer [5].

Du et al. presented a fine-grained multilabel ECG framework to detect correlated cardiac abnormalities in clinical ECG data using convolutional neural network (CNN) and recurrent neural network (RNN) [6]. The authors of [7] used U-Net with bidirectional LSTM for the automated detection of QRS complexes. They achieved an accuracy of 78.73% and 98.29% on the CPSC2019 (China Physiological Signal Challenge 2019) and mitdb datasets, respectively.

In their work, Weinman and Conrad used transfer learning for improving CNNs classification of heart rhythm and atrial fibrillation (AF) from short ECG signals [8]. The authors explored both unsupervised and supervised pretraining of CNN on the Icentia11K dataset. Icentia11K contains ECG data from 11,000 patients, who wore a monitoring device for up to two weeks resulting in 630,000 hours of ECG signal with over 2,700,000,000 beats labeled by the device and then again by a specialist) dataset. They showed that pretraining improves CNN’s performance by over 6% and scored the maximum of $F_1$-score 0.926 for beat classification.

Another interesting issue is the analysis of the ECG signal in terms of detecting individual features in physiological biometrics. Zheng et al., in their work, used BP and DNN for ECG-based identification [9]. They created and tested their model on two databases: mitdb (MIT arrhythmia database) and a self-collected one. The database consisted of signals obtained during various emotions. The authors achieved a 94.39% recognition rate on the combined datasets.

An automatic segmentation of a signal, which is nothing more than finding the reference points within it, is crucial in order to perform automatic interpretation of the signal. There are many different algorithms for segmentation of the ECG signal that use many signal processing methods and work on different databases. In their work, Beraza and Romero compared algorithms for ECG segmentation [10]. The authors performed signal segmentation on ECG using nine different algorithms [11–19] and PhysioNet's
QT database [20]. They achieved the best results while using probabilistic methods and methods based on wavelet transform.

In our study, we propose an alternative approach for the segmentation of a one-dimensional signal segmentation such as ECG. We created a tool for semantic segmentation that used a one-dimensional U-Net containing squeeze-excitation blocks.

2. Materials and Methods

2.1. Dataset

In this study, we used the dataset provided by Lobachevsky University [21], available on the Physionet website. The dataset consists of 200, 10-second, and 12-lead ECG signals, representing different morphologies. Each of the signals comes with an annotation describing the starting, stopping, and peak points of the P and T waves and QRS complexes. The data are compatible with the popular WFDB toolbox. For a more detailed description, please refer to [21].

Data Preparation

In the provided dataset, some of the annotations were incorrect, containing an annotation without the peak point or with it being mislabeled. These samples were removed from the data along with its paired signals. In total, there were 2377 signals with a length of 5000 samples. As mentioned, the annotation came in the form of point collection. To create suitable masks for our model, we transformed the points collection into masks matching the shape of the input signal. For the specific fragments corresponding to the annotations, we applied a label in the form of an integer number as follows:

- “0”—background;
- “1”—QRS block;
- “2”—T-wave;
- “3”—P-wave.

Then, we took the generated sparse masks that are one-hot-encoded, and the final labels have the shape \( k \times 5000 \times 4 \), where \( k \) indicates the number of samples. The input signals were normalized using the min–max scaling technique, which is defined as:

\[
x' = \frac{x - \min(x)}{\max(x) - \min(x)}.
\]

(1)

Then, the dataset was split into training, validation, and testing sets as follows: first, we divided the data into a training and testing set with 80% and 20%; then, we extracted 20% of the training set into the validation set. The distribution of the dataset is presented in the Figure 2.

![Figure 2. Data distribution.](image)

The distributions between categories across the dataset are as follows:

- Background—7,309,154 points,
- QRS complex—894,721 points;
- P-wave—1,516,929 points;
- T-wave—814,196 points.

2.2. Model Architecture

For our study, we chose the U-Net architecture design proposed by Ronnenberger et al. [22] and adapted it for the task of 1D semantic segmentation. Inspired by the paper [23], we have decided to incorporate residual and squeeze-excitation blocks into our model. A diagram of the model architecture is presented in Figure 3. A full diagram of the model is available as supplementary material (File S1).

Figure 3. Diagram of the model.

2.2.1. Encoder

The encoder part of our network is a modified ResNet [24] convolutional neural network. It was designed with four sequentially connected blocks. The single block was made of 3 functional units:
1. Convolutional unit—this unit consists of a one-dimensional convolutional layer followed by batch normalization [25];

2. Squeeze-exciting unit—is a cell designed to improve the representational power of a network by enabling it to perform channel-wise feature calibration. The inputs to this block are the feature maps generated by the convolutional unit. Each of the channels is being “squeezed” into a single numeric value using global average pooling. This value will be passed through two feed-forward layers activated by the ReLU and sigmoid functions to add nonlinearity and give each channel a smooth gating function. Then, the output of the sigmoid function is weighted by the input feature maps to get the excitation [26];

3. Residual unit—this unit stacks two convolutional units, and on top of them, the squeeze-exciting unit is being stacked. The output is being added to the input feature maps.

The encoder consists of four of these blocks as a single convolutional unit followed by four consecutive residual blocks. The outputs of the second and third block are being concatenated with the averaged pooled input data.

2.2.2. Decoder

The decoder part consists of up-sampling layers followed by concatenation with encoder feature maps and convolutional units. In total, there were 34,305,156 parameters from which 34,816 were not trainable.

2.3. Training Process

The training procedure was conducted using the Adam optimization algorithm [27] and categorical cross-entropy as the loss function. We have examined two versions of our model—one with and one without squeeze-exciting blocks. Both models were trained with the following parameters:

- 32 mini-batch size;
- starting learning rate of 0.0005.

The hyperparameters were tuned empirically. For this procedure, we have defined a set of callbacks to prevent overfitting and optimize learning:

- Model checkpoint—saving the model that achieves the best validation loss score;
- Reduce learning rate—dynamic learning rate set to decrease by half each 3 epochs when the validation loss is not improving;
- Early stopping—stopping the training procedure when the model overfits the data (after 3 epochs).

The model was set to be trained for 100 epochs, but after 20 epochs, the early stopping callback stopped the learning process due to overfitting. The model with the best validation loss was saved.

The described model was trained on an NVIDIA RTX 2080Ti graphics card with 12 GB of video RAM. The model was designed with the help of Tensorflow and Keras deep learning libraries.

3. Results

This section provides numerical and visual results of the training, validation, and testing of the proposed models. We compared the models with and without the addition of squeeze-exciting blocks.

3.1. Numerical Results

To examine the overall model performance, we calculated standard metrics, such as precision, recall, and area under the ROC curve. The numerical results are presented for two versions of the model—with and without squeeze-exciting blocks. The comparison between these two models is shown in Table 1.
When evaluating on the test set, the Jaccard index \[28\] and the F1-score were also defined to properly test the trained models (see Table 2). The Jaccard index \(J_{(A,B)}\) and F1-score \(F1_{(A,B)}\) are defined as follows:

\[
J_{(A,B)} = \frac{|A \cap B|}{|A \cup B|},
\]

\[
F1_{(A,B)} = \frac{2 \cdot |A \cap B|}{|A| + |B|},
\]

where \(A \cap B\) is the intersection of sets \(A\) and \(B\), and \(A \cup B\) is the union of sets \(A\) and \(B\).

**Table 1.** Comparison between metrics achieved without and with squeeze-exciting blocks for the train, validation, and test datasets.

|                  | Without Squeeze-Exciting | With Squeeze-Exciting |
|------------------|--------------------------|-----------------------|
| Dataset          | Train | Validation | Test | Train | Validation | Test |
| Accuracy         | 0.97  | 0.92       | 0.92 | 0.97  | 0.95       | 0.95 |
| AUC              | 0.99  | 0.98       | 0.98 | 0.99  | 0.99       | 0.99 |
| Specificity      | 1.0   | 0.99       | 0.97 | 0.99  | 0.99       | 0.99 |
| Sensitivity      | 1.0   | 0.99       | 0.99 | 0.99  | 0.99       | 0.99 |
| Recall           | 0.97  | 0.92       | 0.92 | 0.98  | 0.95       | 0.95 |
| Precision        | 0.97  | 0.92       | 0.92 | 0.97  | 0.95       | 0.95 |
| Loss             | 0.06  | 0.25       | 0.39 | 0.06  | 0.14       | 0.14 |

**Table 2.** Comparison of the Jaccard index and F1-score achieved without and with squeeze-exciting blocks for the test dataset.

|                  | Jaccard Index | F1-Score |
|------------------|---------------|----------|
|                  | Macro         | Micro    | Weighted | Macro | Micro | Weighted |
| Without squeeze-exciting | 0.8           | 0.86     | 0.86     | 0.87  | 0.92  | 0.92     |
| With squeeze-exciting    | 0.87          | 0.91     | 0.91     | 0.93  | 0.96  | 0.96     |

As we are dealing with a multi-class segmentation problem, the aforementioned indicators were calculated in three different variants: micro, macro, and weighted. With the ‘micro’ variant, we calculate metrics globally, i.e., by counting the total true positives, true negatives, false negatives, and false positives. For the ‘macro’ variant, we calculate metrics for each label, and then, we calculate their unweighted mean. This does not take label imbalance into account. In case of the ‘weighted’ variant, we calculate metrics for each label and find their average, which is weighted by the number of true instances for each label. This alters ‘macro’ to account for label imbalance.

Based on the results presented in Table 2, we can see that the squeeze-excitation blocks enhance the overall performance of the model. They also provide a better generalization of the problem, which can be seen by analyzing the loss function across different sets.

The learning curves are presented for the model that contains squeeze-excitation blocks since it yields better results (see Figures 4–8).
Figure 4. Learning curves—accuracy and AUC among the epochs.

Accuracy is a percent of correct predictions. It describes the performance of the model among classes. Accuracy is given by the following formula [29]:

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN},
\]

where $TP$—true positive value, $TN$—true negative value, $FP$—false positive value, $FN$—false negative value (applies to Formulas (4)–(8)).

AUC, Area Under the ROC Curve, provides information on how well the model distinguishes between the classes. The desired value is an AUC as close as possible to 1.

Figure 5. Learning curves—specificity and sensitivity among the epochs.

Figure 6. Learning curves—precision and recall among the epochs.
Specificity is a metric that describes the model’s ability to correctly predict the true negative value. Specificity is given by the following formula [29]:

\[
\text{Specificity} = \frac{TN}{TN + FP}.
\] (5)

Sensitivity is a metric that describes the model’s ability to correctly predict the true positive value. Sensitivity is given by the following formula [29]:

\[
\text{Sensitivity} = \frac{TP}{TP + FN}.
\] (6)

In this case, to determine the sensitivity, we first compute specificity at 200 different thresholds \( threshold_n = \frac{threshold_{n-1}}{2}; n = 1, 2, 3, \ldots, 200; \) \( threshold_0 = 1 \). Then, the sensitivity is computed at the chosen threshold [30].

When the model achieves high specificity and sensitivity values after the learning cycle, that means that the results are reliable.

Precision is a ratio between the number of true positive samples and all samples classified as positive (i.e., the sum of true and false positive samples). This metric shows the model’s ability to correctly classify positive samples. Precision is given by the following formula [29]:

\[
\text{Precision} = \frac{TP}{TP + FP}.
\] (7)

Recall is a metric that defines how well the model detects the positive samples. It is calculated as a ratio between true positive samples and all positive samples (that is, the sum of true positive and false negative samples).

\[
\text{Recall} = \frac{TP}{TP + FN}.
\] (8)
Loss, in other words, penalty for a bad prediction, is a value that indicates how bad the model prediction was on a single sample. In simple terms—the lower the loss, the better the model.

We have calculated the percentage of correctly classified samples for each class and presented it in the form of confusion matrices. Figure 9 shows confusion matrices for each class for the test dataset.

The percentage of correctly classified samples can be calculated as the sum of the values on the main diagonal. For the test dataset, the model scored the following values for each class:

- background—95%;
- QRS block—99.3%;
- P wave—99%;
- T wave—98%.

3.2. Visual Results

To compare predictions of the created model with the original annotations, we chose three examples of signals from the test dataset. As the model containing the squeeze-exiting blocks achieved better results, only its predictions were visualized.

Figure 10 show signals with masks presented as follows:

- black—background (0);
- red—QRS block (1);
- blue—T wave (2);
- cyan—P wave (3).

In the figure, we can see that the model correctly segmented all the aforementioned ECG fragments regardless of the signal lead.
4. Discussion

In this work, we presented a solution for extracting P and T-waves and QRS complexes from 12-lead ECG signals using methods that are mostly used in the image processing domain. The created model achieves a high set of performance parameters (accuracy, AUC, specificity, and sensitivity) independently on all signal leads. Thanks to this, our solution can be used in any type of electrode configuration as a basis for heart rhythm and heart rate variability (HRV) classification as well as many other parameters resulting from specific fragments of ECG. Using our method, we obtained results comparable or better than those presented in the literature [3–19].

According to the World Health Organization (WHO), the top cause of death worldwide in 2019 was ischemic heart disease [31], also showing the biggest increase in deaths since 2000. This disease has its representation in ECG as ST-segment elevation or depression. Taking into account the growing popularity of wearable devices (e.g., a smart watch or a smart band equipped with a simple ECG recorder), we have a chance for earlier detection of cardiovascular disorders by applying real-time segmentation methods and ECG analysis on an everyday basis for high-risk patients.

The deep learning approach seems to be an effective solution to the problem of medical signal segmentation. Considering the ability of neural networks to capture highly nonlinear patterns and their efficient feature extraction process, it can be used to help diagnose patients based on the signals captured from their body. These solutions are yielding state-of-the-art results and can be adopted as preliminary diagnosis systems.

For solving the problem of semantic segmentation of ECG, we have chosen to use U-Net like architecture. This solution has a few advantages. First, convolutions are less computationally expensive than for example the transformer model, which adopts pointwise dependencies discovery, whose complexity increases quadratically with the length of the time series, which given a time series of a 5000-samples length can easily become
intractable. In turn, the lower computation cost can be advantageous when trying to incorporate neural network models onto microprocessors to work directly on hardware. Given the nature of the problem—recognizing sequentially repeating signals—using convolution seems to be a suitable solution since it is sensitive to a local neighborhood of values.

A significant challenge in creating a system for ECG segmentation is finding an appropriate database for training the model or classifier. For this purpose, signal generators can be used, such as the one proposed by Stabenau et al. [32]. Artificial signals can be used to pretrain models that then are tuned to real data.

In the future, we plan to replace the up-sampling layers of the decoder with one-dimensional transposed convolution layers that have learnable parameters. Based on the success of the transformer architecture [33] in natural language processing, we also consider replacing the convolution-based feature extractor with the mentioned architecture. In the future work, besides architectural changes, we also plan to extend our dataset with new categories that represent both healthy P and T-waves, ORS-complexes, and pathological. Another direction worth exploring in the future research is the use of a semantic segmentation approach to detect more complex, and non-sequential events in biomedical signals comparing a wider range of architectures, from 1D convolutional networks, Performer [34], and Linformer [35] and new architectures that seem to be a bridge between the two [36].

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app12073332/s1. File S1: Full diagram of the proposed model.

Author Contributions: Conceptualization, K.D. and P.K.; methodology, K.D. and N.P.; software, K.D.; validation, K.D. and P.K.; formal analysis, K.D. and N.P.; investigation, K.D. and E.T.; resources, K.D.; data curation, K.D.; writing—original draft preparation, K.D. and N.P.; writing—review and editing, K.D., N.P., P.K. and E.T.; visualization, K.D. and N.P.; supervision, P.K. and E.T.; project administration, N.P.; funding acquisition, K.D. and E.T. All authors have read and agreed to the published version of the manuscript.

Funding: This study was carried out under the project “InterPOWER—Silesian University of Technology as a modern European technical university”, co-financed by the European Union under Measure 3.5 Comprehensive programs of universities III Priority Axis Higher education for the economy and development of the Operational Program Knowledge Education Development 2014–2020.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

AF Atrial Fibrillation
ANN Artificial Neural Network
AUC Area Under the Curve
CNN Convolutional Neural Network
DL Deep Learning
ECG Electrocardiography
HRV Heart Rate Variability
PVC Premature Ventricular Contractions
ReLU Rectified Linear Unit
RNN Recurrent Neural Network
WHO World Health Organization

References
1. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; Adaptive Computation and Machine Learning; The MIT Press: Cambridge, MA, USA, 2016.
2. Kiranyaz, S.; Acvi, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. Mech. Syst. Signal Process. 2021, 151, 107398. [CrossRef]
3. Faust, O.; Hagiwara, Y.; Hong, T.J.; Lih, O.S.; Acharya, U.R. Deep learning for healthcare applications based on physiological signals: A review. Comput. Methods Programs Biomed. 2018, 161, 1–13. [CrossRef] [PubMed]
35. Wang, S.; Li, B.Z.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-Attention with Linear Complexity. *arXiv* 2020, arXiv:2006.04768.
36. Li, D.; Hu, J.; Wang, C.; Li, X.; She, Q.; Zhu, L.; Zhang, T.; Chen, Q. Involution: Inverting the Inherence of Convolution for Visual Recognition. *arXiv* 2021, arXiv:2103.06255.