Atrial Fibrillation Classification in Electrocardiography using Deep Learning

Christoph Reich[‡], Vladislav Kruk
Department of Electrical Engineering and Information Technology,
Technische Universität Darmstadt
[‡]*christoph.reich@stud.tu-darmstadt.de*

***Abstract*—The evaluation of electrocardiogram recordings is the most common approach to diagnose and monitor cardiac arrhythmia such as atrial fibrillation. The electrocardiogram evaluation typically requires expert knowledge, which is not always available. We present a novel approach for the automated classification of atrial fibrillation in electrocardiogram recordings with variable length. Our deep learning approach utilizes both input data in the time and frequency domain. The performance of the proposed dual network for ECG classification (ECG-DualNet) is showcased on the 2017 PhysioNet/CinC Challenge dataset. ECG-DualNet outperforms recent Convolutional Neural Network approaches in terms of classification accuracy. Code and trained models are available at https://github.com/ChristophReich1996/ECG_Classification.**

***Index Terms*—deep learning, attention, arrhythmia classification, atrial fibrillation classification, electrocardiography.**

I. Introduction

Electrocardiography (ECG) is the most important tool for the diagnosis and the monitoring of cardiac arrhythmia [1]–[3]. The first recorded human heart beat dates back to the late 19th century [3]. Today 12-lead ECGs are the common standard [3]. The analysis of ECG recordings, especially the detection of cardiac arrhythmia, requires expert knowledge [1]. This expert knowledge is sometimes not available. Since a fast and accurate diagnosis of cardiac arrhythmia can highly affect the chance of survival of a patient positively, an increasing interest for the automated detection of cardiac arrhythmia occurs [1], [4]–[6].

The most common human cardiac arrhythmia is atrial fibrillation (AF) (Fig. 1). [7] AF mostly affects patients at an advanced age and can lead to an increased risk for dementia, stroke, and even heart failure. High blood pressure and obesity are common factors for an increased risk for AF. During AF undirected electoral impulses (Fig. 1) occur in the atrium of the heart. This leads to a fast movement of the heart atrium resulting in an impaired blood flow. Additionally, blood clots can occur which potentially lead to thrombosis. [1], [7]

In this study, we propose a novel deep learning approach for classifying AF in single-lead ECG recordings with variable lengths. Our novel deep learning approach ECG-DualNet utilized both input data for the time and frequency domain, enabling the network to learn features in both domains. The proposed ECG-DualNet surpasses the classification accuracy of recent convolutional neural network (CNN) approaches which only utilizes input data from the frequency domain [5].

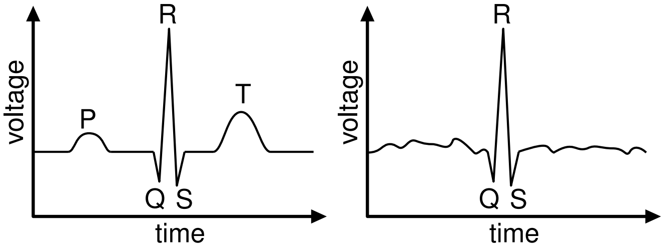

Fig. 1. Regular heart beat with QRS complex, P wave, and T wave on the left and atrial fibrillation on the right. AF can be detected by the noisy ECG at the P and T wave regions. Please note that this is only a theoretical illustration. Real ECGs of AF can be seen in the appendix.

II. Related Work

Recent approaches for the task of AF classification in ECG recordings can be clustered in two groups. First, classical machine learning approaches [4], [9], and second, deep learning approaches [5], [10]–[13]. In general, deep learning approaches achieve better classification accuracy, however, sacrifice explainability [5], [10], [14], [15].

Classical machine learning approaches typically extract features, first and classify them in a section learnable step. Hoog Antink *et al.* [4] proposed an approach that first extracts ECG timing features, robust interval features, and waveform features [4]. All extracted features are fed into a learned random forest for classification [4]. Other approaches perform similar feature extractions but utilize different learnable classification methods, such as support vector machines [9]. These approaches, however, require a lot of domain knowledge to extract relevant features. Additionally, hand-crafted feature extraction approaches are often complicated and error prune to implement.

Recently deep learning-based approaches have been applied to AF classification in ECG recordings [5], [10]–[13]. These approaches typically produce a spectrogram of the ECG recording and feed it into a deep neural network for classification [5], [11]. Deep learning AF classification approaches tend to outperform more classical machine learning approaches [10], [12]. Using deep neural networks, however, require a reasonable amount of data, or extensive data augmentation [16], and has higher computational requirements [14], [17]. Since a deep neural network is able to learn the extraction of

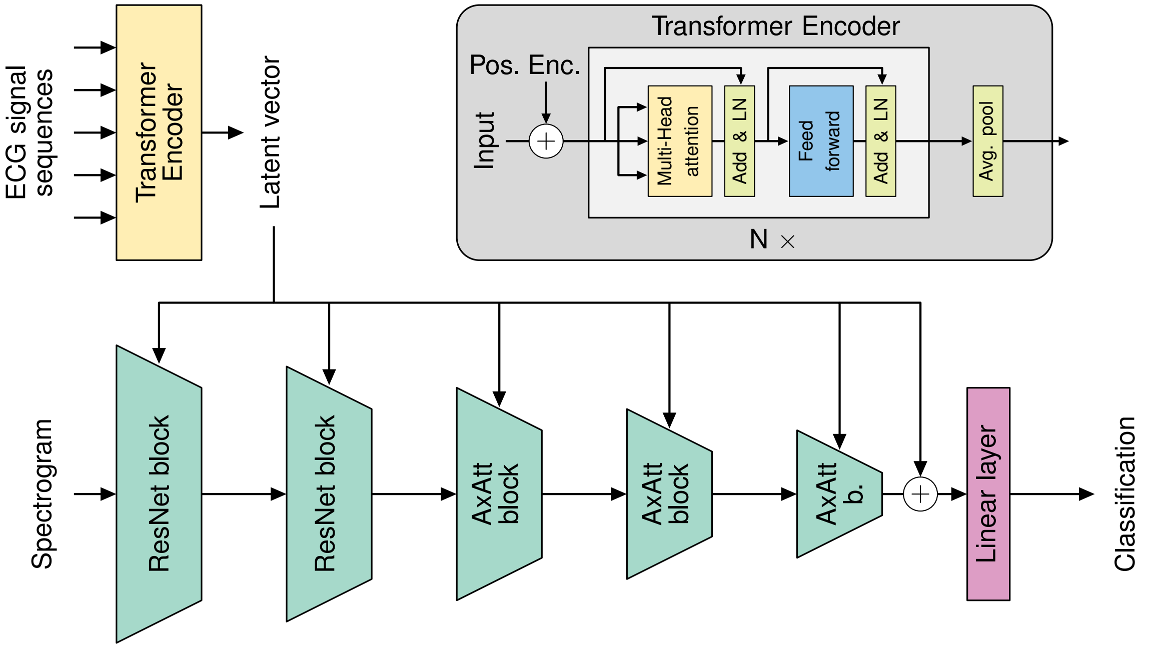

Fig. 2. ECG-DualNet++ architecture with spectrogram and ECG signal as inputs. The ECG signal sequence gets encoded by a Transformer encoder to a single latent vector. The spectrogram is encoded by multiple 2D blocks. The first two blocks are standard ResNet like blocks and the following three blocks are Axial-Attention blocks. All blocks of the spectrogram encoder utilize the latent vector by conditional batch normalization. Transformer encoder [8] architecture shown in the top right.

relevant features, heavy preprocessing is typically not required [5], [14], [17].

III. Method

In this section, we introduce our ECG-DualNet. We take inspiration from the model by Mousavi *et al.* which utilizes two separate network paths to encode both time and frequency domain data [10]. Our data augmentation pipeline builds on the work of Nonaka *et al.* which presented multiple augmentations for ECG data [13].

A. ECG-DualNet Architecture

Our ECG-DualNet (Fig. 2) utilizes two separate encoders. The first encoder takes the ECG signal as the input. We will refer to this encoder as the signal encoder. The second encoder is fed with a spectrogram. We call this encoder, spectrogram encoder. The final part of the network builds a simple softmax classifier to produce the final classification. We utilize two network settings, ECG-DualNet and ECG-DualNet++ in which we use different building blocks for both encoders.

In the standard-setting (ECG-DualNet), the signal encoder consists of a standard long-short-term-memory (LSTM) [18] module. The LSTM module encodes temporal agnostic feautres into a latent vector. This latent vector is incorporated into the spectrogram encoder and the final linear layer. In the ECG-DualNet++ we replace the LSTM module with a Transformer encoder [8], [19] (Fig. 2 top right), using learnable encodings [8], [20], Layer Normalization [21], and Gaussian Error Linear Units [22]. To encounter overfitting dropout [23] is used in both the LSTM and the Transformer.

The spectrogram encoder is, in the standard setting, comprised of five ResNet-like [24] blocks. Each block consists of two 2D convolutions, two Padé Activation Units [25], two Conditional Batch Normalization layers (CBN) [26], an average pooling layer, and a skip connection. CBN is utilized to conditionalize the spectrogram encoder on the latent vector of the signal encoder. For ECG-DualNet++ the three highest blocks of the spectrogram encoder are replaced with Axial-Attention blocks [27]. Still, CBN and Padé Activation Units are employed. Similar to the signal encoder, dropout [23] is also applied in each spectrogram encoder block.

To investigate the effect of the network size we employ different network sizes for both the ECG-DualNet and ECG-DualNet++. We vary the width and the depth of the signal encoder. For the spectrogram encoder, we diversify the width.

B. Training Approach

We train all networks on a weighted version of the cross-entropy loss [14]

$$\mathcal{L} = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{4}\alpha_i\, y_{ji}\, \log(\hat{y}_{ji}). \tag{1}$$

Where $\mathbf{y}_j \in \mathbb{R}^4$ is the ground truth one-hot label, $\hat{\mathbf{y}}_j \in \mathbb{R}^4$ the network softmax prediction, and $\alpha \in \mathbb{R}^4$ the class weighting. The cross-entropy loss is averaged over a mini-batch of the size N. The loss function (Eq. 1) is minimized by using the RAdam optimizer [28].

C. Validation Approach

To validate the performance of our networks we utilize the accuracy and the F1 score. The accuracy is computed over all classes by

$$\mathrm{ACC} = \frac{1}{n}\sum_{j=1}^{n}\delta\left(\arg\max(\mathbf{y}_j), \arg\max(\hat{\mathbf{y}}_j)\right). \quad (2)$$

Where $\delta(\cdot,\cdot)$ is the Kronecker delta, and $\arg\max(\cdot)$ estimates the position of the maximum value present in the input vector. n corresponds to the dataset size. The F1 score is computed as

$$\mathrm{F1} = \frac{1}{4}\sum_{i=1}^{4}\frac{2\mathrm{TP}_i}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}. \quad (3)$$

TP_i represents the true positive predictions of a class over the whole dataset, FP_i the false positive predictions, and FN_i the false negative predictions.

D. Preprocessing

We utilize a simple preprocessing composed of four steps. In the first steps, the ECG signal gets standardization to a mean of zero and unit variance. In the second step, various augmentations (Sec. III-E) are applied to the ECG signal. In the following step, a log spectrogram of the ECG signal. The log spectrogram is computed with a window length of 64, a hop size of 32, and 64 bins. Recent work showed, using the logarithmic spectrogram improves the classification accuracy of CNNs [5]. Finally, both the ECG signal and the spectrogram are zero-padded to a fixed length.

E. Data Augmentation

Our augmentation pipeline applies randomly multiple different data augmentations to the ECG signal. This improved the generalization of the trained network and prevents overfitting [13], [16], [29]. The following augmentations are used: dropping, cut-out, resampling, random resampling, scaling, shifting, sine addition, and band-pass filtering. The dropping augmentation sets random samples of the ECG signal to zero, the cut-out augmentation sets a random sequence to zero. In the resampling augmentation, the signal gets resampled to a different heartbeat rate. Random resampling is inspired by the random elastic deformation [30]–[32] used for image augmentation. The ECG signal gets resampled by smooth random offset, resulting in a changing heartbeat rate. In the scaling augmentation, the signal gets scaled by a random factor. The sine addition augmentation adds a sinusoidal signal with a random magnitude and phase to the ECG signal. The shift augmentation shifts the ECG signal by a random length. Finally, in the band pass filter augmentation, the ECG signal is filtered by a band-pass.

F. Implementation Details

ECG-DualNet is implemented in PyTorch [33]. For implementing the preprocessing SciPy [34], NumPy [35], and Torchaudio[1] were used, in addition to PyTorch. For the Padé Activation Unit we used the official PyTorch extension by the authors [25].

Different network sizes (S, M, L, and XL) for both ECG-DualNet and ECG-DualNet++ (Tab. I) were employed. A detailed overview of the network configurations can be found in the appendix or in the provided implementation.

100 training epochs with a fixed batch size of 24 were performed. All models except ECG-DualNet++ $130\mathrm{M}$ were trained on a single Nvidia 2080 Ti. Training took took between $30\mathrm{min}$ (ECG-DualNet S) and $3\mathrm{h}$ (ECG-DualNet++ XL). Our biggest model ECG-DualNet++ $130\mathrm{M}$ was trained on four Nvidia Tesla V100 (16GB) which took approximately 6h. The weights of the loss function (Eq. 1) were set to $\alpha = [0.4, 0.7, 0.9, 0.9]$, for counteracting the dataset class imbalance (Sec. IV-A). The initial learning rate of the RAdam optimizer was set to 10^{-3}. The learning rate were decreased after 25, 50, and 75 epochs by 0.1. The first and second-order momentum factors were set to 0.9 and 0.999, respectively. Each augmentation described in Section III-E was applied with a probability of 0.2. An overview of all hyperparameters is presented in the appendix.
We also consider pre-training on the Icentia11k dataset [36]. Similar to pre-training in computer vision [37]–[39], we first train on the very large Icentia11k dataset. Afterward, the pre-trained weights were used as the initial weights for training on the target PhysioNet dataset. For pre-training we perform 20 epochs with a batch size of 100 on a single Nvidia Tesla V100 (32GB). Training took approximately $24\mathrm{h}$. During pre-training, the same learning rate schedule as described earlier with steps after 5, 10, and 15 epochs was employed. The training on the target dataset was performed as a normal training run on the PhysioNet dataset, described earlier.

IV. Experiments

We conduct experiments on the 2017 PhysioNet/CinC Challenge dataset [6] (Sec. IV-A). We performed three training runs (if not stated different) for each model with different random seeds, where the best run, based on the validation accuracy, was reported. Additional results can be found in the appendix.

A. 2017 PhysioNet/CinC Challenge Dataset

We utilize 2017 PhysioNet/CinC Challenge dataset [6] for training. The dataset includes 8529 publicly available[2] labeled single lead ECG sequences. Each ECG sequence includes between 2714 and 18286 samples, recorded with a sampling frequency of 300. For each sequence, a ground truth classification label is provided. Four classes are given, namely, normal, indicating a normal cardiac rhythm, AF, indicating atrial fibrillation, other, for different rhythms, and noisy, for

[1] https://github.com/pytorch/audio

[2] https://physionet.org/content/challenge-2017

a noisy measurement. Each class includes 5050 (normal), 2456 (AF), 738 (other), and 284 (noisy) samples, respectively. Visualizations of multiple sequences and corresponding labels are provided in the appendix.

We split the data once randomly into a training and validation set. The training set includes 7000 samples and the validation set 1528 samples. We omit the use of k-fold cross-validation due to the computational expensiveness of each training run [14], [40]. All sequences are zero-padded, respectively, cropped to a fixed length of 18000 samples. For a detailed description of the preprocessing see Section III-D.

B. Icentia11k dataset

The Icentia11k dataset [36], used for pre-training, consists single lead ECG recordings of 11k patients. The dataset is partly labeled with six different heart rhythms including atrial fibrillation. Each dataset sample is a sequences of approximately $1\mathrm{h}$ with a sampling frequency of 250. We resample the ECG signal to match the sampling frequency of the PhysioNet dataset. Finally, we randomly crop each sequence to a random length of 9000 to 18000 samples. For unlabeled crops we utilize a dummy class. For validation we split the dataset patient-wise. The resulting training set includes 10k patients and the validation set 1k patients.

C. Results

Our archived classification results are presented in Table I, in which we present both the accuracy (Eq. 2) and the F1 score (Eq. 3). Table I also includes classification results of Zihlmann *et al.* [5]. The models trained by Zihlmann et al. are, however, trained on the full publicly available 2017 PhysioNet/CinC Challenge dataset and tested on the private test dataset [5]. Whereas our models are trained on a custom split of the publically available samples (Sec. IV-A). Additionally, the reported F1 scores by Zihlmann *et al.* are computed over three classes (excluding the noise class) and thus are not directly comparable to our F1 scores.

TABLE I
CLASSIFICATION RESULTS OF OUR PROPOSED APPROACHES AND BASELINES ON THE 2017 PHYSIONET VALIDATION SET.

Model	ACC ↑	F1 ↑	# Parameters
CNN baseline* [5]	0.812	*0.790*†	∼ 3.5M
CRNN baseline* [5]	0.823	*0.792*†	∼ 3.5M
ECG-DualNet S	0.8527	0.8049	1.8M
ECG-DualNet M	0.8560	0.7938	4.3M
ECG-DualNet L	0.8514	0.8038	6.2M
ECG-DualNet XL	**0.8612**	**0.8164**	20.7M
ECG-DualNet++ S	0.8174	0.7291	1.8M
ECG-DualNet++ M	0.8259	0.7730	2.6M
ECG-DualNet++ L	0.8449	0.7859	3.7M
ECG-DualNet++ XL	0.8593	0.8051	8.2M
ECG-DualNet++ 130M	0.8534	0.7963	128M

* Reported literature values.

† F1 score computed over three classes, thus not directly comparable.

Although training was performed with fewer data samples, all ECG-DualNet architectures outperformed the baselines from the literature in classification accuracy. We observe that larger network configurations tend to perform stronger compared to smaller configurations. Especially, ECG-DualNet++ profits from a larger network capacity.
Training on the Icentia11k dataset [36] yield the results presented in Table II. Only two models were trained on the Icentia11k dataset due to the immense computational requirements.

TABLE II
CLASSIFICATION RESULTS OF OUR PROPOSED APPROACHES ON THE ICENTIA11K VALIDATION SET. ONLY A SINGLE TRAINING FOR EACH MODEL RUN WAS CONDUCTED.

Model	ACC ↑	F1 ↑
ECG-DualNet XL	0.8989	0.4564
ECG-DualNet++ XL	0.8899	0.4970

Using the model weights trained on the Icentia11k dataset (Tab. II) for training on the PhysioNet dataset lead to the results presented in Table VII. Compared to the results, using no pre-trained weights (Tab. VI), the results, with pre-trained weights, lead to similar or slightly worse results.

TABLE III
CLASSIFICATION RESULTS OF OUR PROPOSED APPROACHES ON THE 2017 PHYSIONET VALIDATION SET AND PRE-TRAINED ON THE ICENTIA11K DATASET.

Model	ACC ↑	F1 ↑
ECG-DualNet XL	0.8534	0.8024
ECG-DualNet++ XL	0.8534	0.8004

Additional experimental results of each training run are presented in the appendix.

D. Ablation Study

We conducted multiple ablation training runs to investigate the effectiveness of each of the proposed components. In the presented results in table IV, we omit one core component of our approach in each training run. We only perform ablations with one network configuration, namely ECG-DualNet M. This is due to the large computational requirements for each training run.

TABLE IV
CLASSIFICATION RESULTS ON THE 2017 PHYSIONET VALIDATION FOR DIFFERENT ABLATIONS. ECG-DUALNET L CONFIGURATION UTILIZED.

Data aug. &	Signal	Spectrogram		
dropout	encoder	encoder	ACC ↑	F1 ↑
✗	✓	✓	0.8272	0.7493
✓	✗	✓	0.8440	0.7855
✓	✓	✗	0.7264	0.5813
✓	✓	✓	**0.8560**	**0.7938**

When utilizing no data augmentation (Sec. III-E) and dropout the training loss is minimized to approximately zero.

This indicates signs of overfitting, even though the achieved accuracy is 0.8272.

From the results in table 1, it can be observed that each main component is required to reach the best classification performance. The largest impact on the classification accuracy has the spectrogram encoder. The use of data augmentation and a signal encoder has a lesser impact on the performance.

V. Discussion

As observed in the ablations (Sec. IV-D) training without data augmentation leads to a zero valued loss. Experimental results without data augmentation are presented in Table IV. Utilizing the proposed augmentation pipeline improves generalization. Unfortionalty, the use of a sophisticated augmentation pipeline comes with additional hyperparameters. We set the hyperparameters of our augmentation pipeline empirically based on a few test training runs. To use the full potential of the proposed augmentation pipeline hyperparameter optimization [14], [41] or the development of an adaptive approach, such as [42], [43], may lead to improvements.

We also observe the phenomenon that an increase in the model size with and without data augmentation does not lead to overfitting but more generalization. We have no clear explanation for this behavior. A potential description could be the double decent phenomenon [44], [45], without an overfitting bump when regularization (data augmentation & dropout) is applied. A large-scale study with more network sizes may help to understand the observed phenomenon.

Surprisingly, performing pre-training on the Icentia11k does not lead to better classification performance on the target dataset. This could be due to the fact that the dataset used for pre-training is too out-of-domain transferring knowledge to the target dataset.

We also observe from the results in Table I that ECG-DualNet++ XL and 130M perform similarly to ECG-DualNet XL. For smaller models, the ECG-DualNet surpasses the classification accuracy of ECG-DualNet++. However, on the Icentia11k dataset ECG-DualNet++ outperforms ECG-DualNet in the F1 score. This may indicate that attention-based approaches need more data to surpass the performance of traditional CNNs and LSTMs for ECG classification.

VI. Conclusion

This work presented a novel deep neural network architecture for classifying AF in single-lead ECG signals with variable length. The introduced ECG-DualNet utilizes input data in the frequency and time domain. The time-domain ECG signal gets processed by a separate encoder. The resulting encoded latent vector gets employed in the spectrogram encoder by conditional batch normalization. We also extended ECG-DualNet by the use of Transformers and Axial-Attentions (ECG-DualNet++). With the use of both input data from the time and frequency domain, we were able to outperform a CNN baseline which only utilizes input data from the frequency domain.

References

[1] D. E. Becker, "Fundamentals of electrocardiography interpretation," *Anesthesia progress*, vol. 53, no. 2, pp. 53–64, 2006.

[2] R. H. Anderson, E. J. Baker, A. Redington, M. L. Rigby, D. Penny, and G. Wernovsky, *Paediatric Cardiology*. Elsevier Health Sciences, 2009.

[3] M. AlGhatrif and J. Lindsay, "A brief review: history to understand fundamentals of electrocardiography," *Journal of community hospital internal medicine perspectives*, vol. 2, no. 1, p. 14383, 2012.

[4] C. H. Antink, S. Leonhardt, and M. Walter, "Fusing QRS Detection and Robust Interval Estimation with a Random Forest to Classify Atrial Fibrillation," in *2017 Computing in Cardiology (CinC)*. IEEE, 2017, pp. 1–4.

[5] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional Recurrent Neural Networks for Electrocardiogram Classification," in *2017 Computing in Cardiology (CinC)*. IEEE, 2017, pp. 1–4.

[6] G. D. Clifford, C. Liu, B. Moody, H. L. Li-wei, I. Silva, Q. Li, A. Johnson, and R. G. Mark, "AF Classification from a Short Single Lead ECG Recording: the PhysioNet/Computing in Cardiology Challenge 2017," in *2017 Computing in Cardiology (CinC)*. IEEE, 2017, pp. 1–4.

[7] G. Herold, *Innere Medizin*. Walter de Gruyter GmbH & Co KG, 2019.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] R. Smíšek, J. Hejč, M. Ronzhina, A. Němcová, L. Maršánová, J. Chmelík, J. Kolářová, I. Provazník, L. Smital, and M. Vítek, "Svm based ecg classification using rhythm and morphology features, cluster analysis and multilevel noise estimation," in *2017 Computing in Cardiology (CinC)*, 2017, pp. 1–4.

[10] S. Mousavi, F. Afghah, A. Razi, and U. R. Acharya, "ECGNET: Learning Where to Attend for Detection of Atrial Fibrillation with Deep Visual Attention," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2019, pp. 1–4.

[11] F. R. Mashrur, A. D. Roy, and D. K. Saha, "Automatic identification of arrhythmia from ecg using alexnet convolutional neural network," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, 2019, pp. 1–5.

[12] L. Khriji, M. Fradi, M. Machhout, and A. Hossen, "Deep Learning-based Approach for Atrial Fibrillation Detection," in *International Conference on Smart Homes and Health Telematics*. Springer, 2020, pp. 100–113.

[13] N. Nonaka and J. Seita, "Data Augmentation for Electrocardiogram Classification with Deep Neural Network," *preprint arXiv:2009.04398*, 2020.

[14] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[15] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services*, vol. 1, no. 1, pp. 39–48, 2018.

[16] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," *preprint arXiv:1712.04621*, 2017.

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

[18] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *preprint arXiv:2010.11929*, 2020.

[20] T. Prangemeier, C. Reich, and H. Koeppl, "Attention-Based Transformers for Instance Segmentation of Cells in Microstructures," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020, pp. 700–707.

[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *preprint arXiv:1607.06450*, 2016.

[22] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *preprint arXiv:1606.08415*, 2016.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] A. Molina, P. Schramowski, and K. Kersting, “Padé Activation Units: End-to-end Learning of Flexible Activation Functions in Deep Networks,” in *International Conference on Learning Representations*, 2020.

[26] H. de Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. Courville, “Modulating early visual processing by language,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6597–6607.

[27] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, “Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation,” in *European Conference on Computer Vision*, 2020, pp. 108–126.

[28] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the Variance of the Adaptive Learning Rate and Beyond,” in *International Conference on Learning Representations*, 2020.

[29] F. N. Hatamian, N. Ravikumar, S. Vesal, F. P. Kemeth, M. Struck, and A. Maier, “The Effect of Data Augmentation on Classification of Atrial Fibrillation in Short Single-Lead ECG Signals Using Deep Neural Networks,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1264–1268.

[30] P. Y. Simard, D. Steinkraus, J. C. Platt *et al.*, “Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis,” in *Icdar*, vol. 3, no. 2003, 2003.

[31] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[32] T. Prangemeier, C. Wildner, A. O. Françani, C. Reich, and H. Koeppl, “Multiclass Yeast Segmentation in Microstructured Environments with Deep Learning,” in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2020, pp. 1–8.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[34] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[35] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[36] S. Tan, G. Androz, A. Chamseddine, P. Fecteau, A. Courville, Y. Bengio, and J. P. Cohen, “Icentia11k: An unsupervised representation learning dataset for arrhythmia subtype discovery,” *preprint arXiv:1910.09570*, 2019.

[37] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *European conference on computer vision*, 2014, pp. 818–833.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[39] K. He, R. Girshick, and P. Dollar, “Rethinking imagenet pre-training,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[40] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[41] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning Augmentation Strategies From Data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, “Adaptive data augmentation for image classification,” in *2016 IEEE international conference on image processing (ICIP)*, 2016, pp. 3688–3692.

[43] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training Generative Adversarial Networks with Limited Data,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 104–12 114.

[44] M. Belkin, D. Hsu, S. Ma, and S. Mandal, “Reconciling modern machine-learning practice and the classical bias–variance trade-off,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[45] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, “Deep double descent: Where bigger models and more data hurt,” in *International Conference on Learning Representations*, 2020.

VII. Appendix

This section includes additional information and results of this study work.

A. Dataset

This subsection includes visualizations and additional information of the 2017 PhysioNet/CinC Challenge dataset [6].

TABLE V
Class distribution of the 2017 PhysioNet/CinC Challenge dataset [6].

Normal	Other rhythm	AF	Noisy
5050	2456	738	284

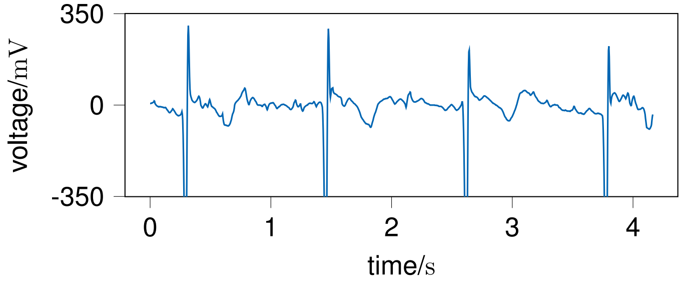

Fig. 3. ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] labeled as normal.

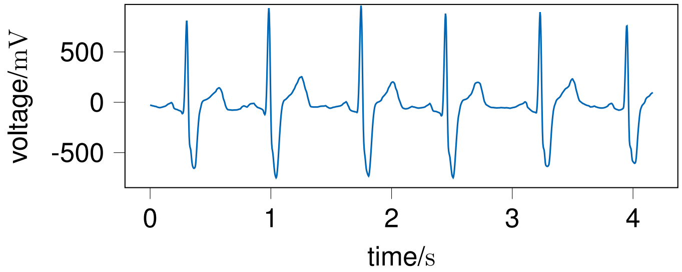

Fig. 4. ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] labeled as other rhythm.

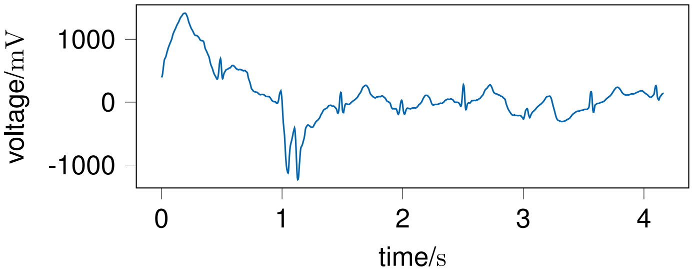

Fig. 5. ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] labeled as noisy.

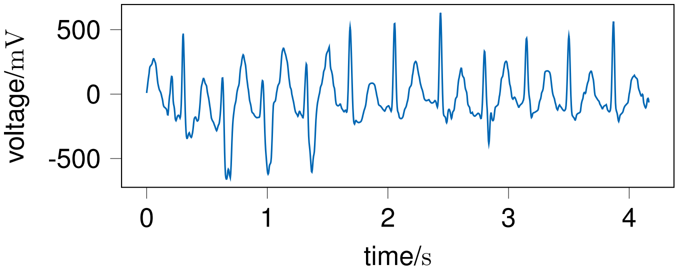

Fig. 6. ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] labeled as AF.

B. Augmentation Pipeline

This section includes visualization of the each augmentation included in our augmentation pipeline.

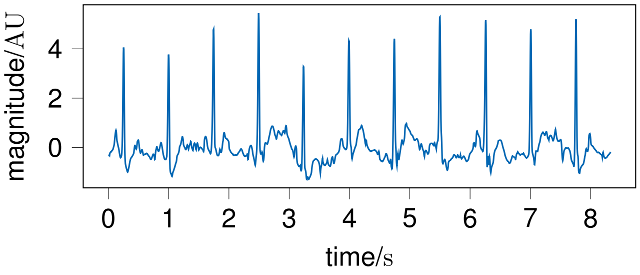

Fig. 7. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by scaling augmentation.

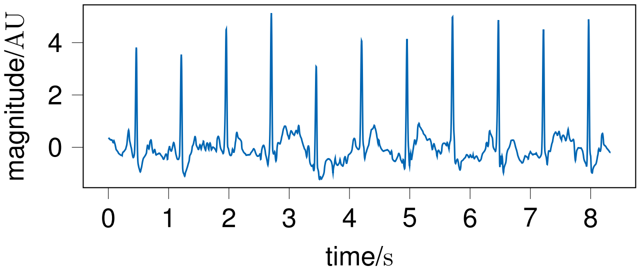

Fig. 8. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by shift augmentation.

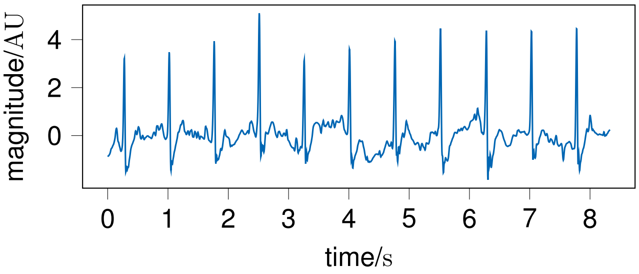

Fig. 9. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by band pass filter augmentation.

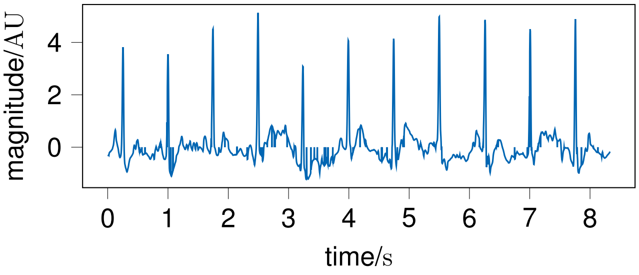

Fig. 10. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by dropping augmentation.

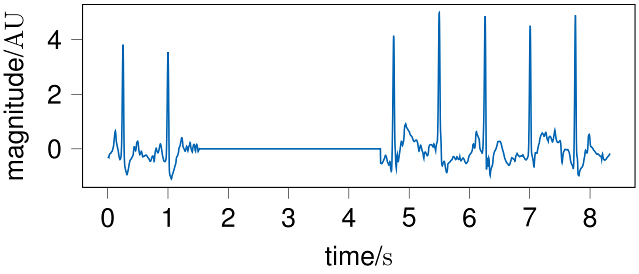

Fig. 11. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by cutout augmentation.

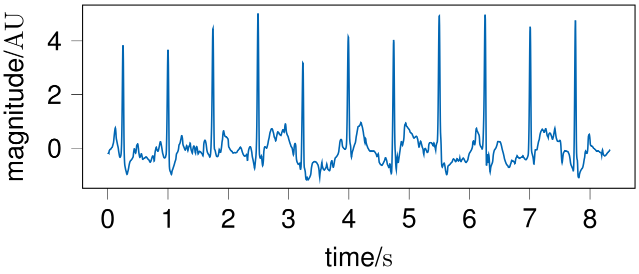

Fig. 12. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by sine augmentation.

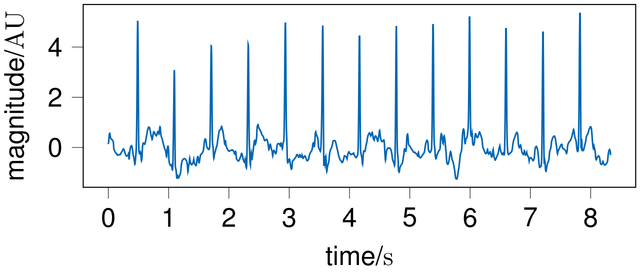

Fig. 13. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by resample augmentation.

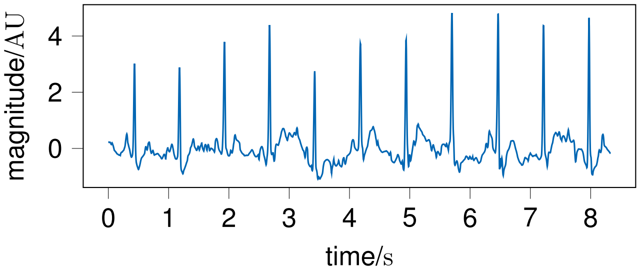

Fig. 14. Standardized ECG signal of the 2017 PhysioNet/CinC Challenge dataset [6] processed by random resample augmentation.

C. *Failed Experiments*

We experimented also with a full transformer-based architecture [8]. In which the ECG signal builds the input to the transformer encoder. The transformer decoder was feed with patches of the spectrogram. Unfortunately, this architecture achieved a weak classification accuracy (< 0.72) during early tests. Because of this, a full transformer-based architecture was not considered further.

D. *Experimental Results*

TABLE VI
Classification results of ECG-DualNet(++) on the PhysioNet 2017 validation set for three different training runs with different seeds.

	Run 1		Run 2		Run 3	
Model	ACC	F1	ACC	F1	ACC	F1
ECG-DualNet S	0.8527	0.8049	0.8410	0.7923	0.8455	0.7799
ECG-DualNet M	0.8560	0.7938	0.8442	0.7955	0.8495	0.7928
ECG-DualNet L	0.8508	0.8097	0.8213	0.7515	0.8514	0.8038
ECG-DualNet XL	0.7702	0.6899	0.8612	0.8164	0.7866	0.7162
ECG-DualNet++ S	0.7323	0.6239	0.8174	0.7291	0.7912	0.7127
ECG-DualNet++ M	0.8226	0.7544	0.8259	0.7730	0.7938	0.6947
ECG-DualNet++ L	0.8449	0.7859	0.8442	0.7750	0.8396	0.7671
ECG-DualNet++ XL	0.8593	0.8051	0.8051	0.7799	0.8501	0.7851
ECG-DualNet++ 130M	0.8475	0.7878	0.8534	0.7963	0.8462	0.7740

E. *Challenge Submission*

Our final challenge submission includes an ECG-DualNet XL. This model was pre-trained on the Icentia11k dataset [36] and fine-tuned on the 2017 PhysioNet/CinC Challenge dataset [6]. However, instead of the random training and validation split used in the reported results, an optimized split is used. Since it is known that a fraction of the final test set is taken from the publicly available samples of the 2017 PhysioNet/CinC Challenge dataset we detected these samples with the provided samples. Samples that are not included in the provided dataset but are included in the 2017 PhysioNet/CinC Challenge dataset are put into the training set. The final split includes 8000 training samples (including potential test data) and 528 samples to validate the network's performance during training. The achieved validation results are presented in Table VIII.

TABLE VII
Classification results of ECG-DualNet(++) on the PhysioNet 2017 validation set for three different training runs with pre-trained weights (Icentia11k dataset) and different seeds.

	Run 1		Run 2		Run 3	
Model	ACC	F1	ACC	F1	ACC	F1
ECG-DualNet XL	0.8534	0.8024	0.8167	0.7385	0.7663	0.5880
ECG-DualNet++ XL	0.8534	0.8004	0.8534	0.7844	0.8508	0.7952

TABLE VIII
Classification results of ECG-DualNet XL pre-trained on the Icentia11k dataset and fine-tuned on the PhysioNet dataset with optimizer split. Metric computed on the small validation set.

Model	ACC ↑	F1 ↑
ECG-DualNet XL	0.8840	0.8549

F. *Hyperparameters*

All important network and augmentation hyperparameters can be seen in the provided config file at: https://github.com/ChristophReich1996/ECG_Classification/blob/main/ecg_classification/config.py.
Additional hyperparameters can be found in the training script at: https://github.com/ChristophReich1996/ECG_Classification/blob/main/train.py

G. *Contributions*

CR implemented the PhysioNet and Icentia11k dataset and developed/implemented the augmentation pipeline and the ECG-DualNet(++) models (training and validation). CR conducted the experiments, wrote the report and produced Figure 1 & 2.
VK contributed the ECG visualization (Fig. 3 - 14) and implemented a XGBoost baseline (see implementation). Additionally, VK proofread the paper and helped with useful discussions regarding both the implementation and the report.