Automated Classification of Electrocardiograms Using Wavelet Analysis and Deep Learning

Andrew Demonbreun, Grace M Mirsky

Benedictine University, Lisle, IL, USA

Abstract

For the 2020 PhysioNet/Computing in Cardiology Challenge, we applied wavelet analysis to develop multiple deep learning models, creating a unique model for each lead. This approach leverages the ability of different leads, based upon their anatomical placement, to better observe different arrhythmias. A voting scheme is implemented amongst the leads, allowing for confirmation of arrhythmia diagnosis from multiple leads, thereby increasing confidence in the diagnosis while also allowing for diagnosis of multiple concurrent arrhythmias. We leverage transfer learning to simplify training our deep learning network by utilizing a modified version of SqueezeNet for training. Since SqueezeNet is designed for image classification, the ECG signals are converted to scalograms prior to training. Using this method, our team, Eagles, achieved a challenge validation score of 0.214 and a full test score of 0.205, placing us 20th out of 41 in the official ranking. While this method has shown promise, improvements are needed to improve classification accuracy in order to make it a clinically viable technique.

1. Introduction

The standard 12-lead electrocardiogram (ECG) is a non-invasive diagnostic tool for measuring and recording the electrical activity of the heart. The ECG is commonly used in the diagnosis of cardiac arrhythmias and abnormalities [1]; however, the accurate interpretation of the ECG requires highly skilled practitioners [2]. Therefore, automated diagnostic classification of ECGs can greatly assist clinicians, particularly when a shortage of such specialized personnel exists. In recent years, there has been increased interest in this research topic; however, these studies tend to be limited in the number of samples and/or diversity of the datasets. The 2020 PhysioNet/Computing in Cardiology Challenge, Classification of 12-lead ECGs, facilitates the development of robust classification algorithms over a large, diverse dataset in order to overcome limitations of previous studies [3-7]. Details of the 2020 Challenge may be found at [8].

2. Methods

Transfer learning with SqueezeNet requires images for training. To accomplish this, we converted the ECG signals to scalograms, which are time-frequency representations of the absolute value of the continuous wavelet transform coefficients plotted over time and frequency. Examples of ECG signals (short snippets are used for clarity) and their corresponding scalograms are shown in Figure 1.

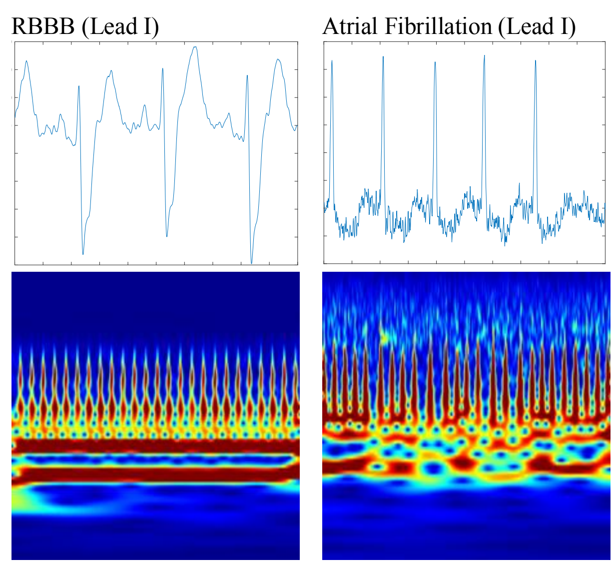

Figure 1. Example of ECGs and associated scalograms of two different patients: one patient with right bundle branch block (RBBB) and another patient with atrial fibrillation.

In order to leverage the ability of different leads, based upon their anatomical placement, to better observe different arrhythmias, we created twelve separate models, one for each lead. By examining the scalograms generated from the 12 lead positions, we can qualitatively observe differences that distinguish them from one another, which should be able to be exploited through deep learning. An example is shown in Figure 2.

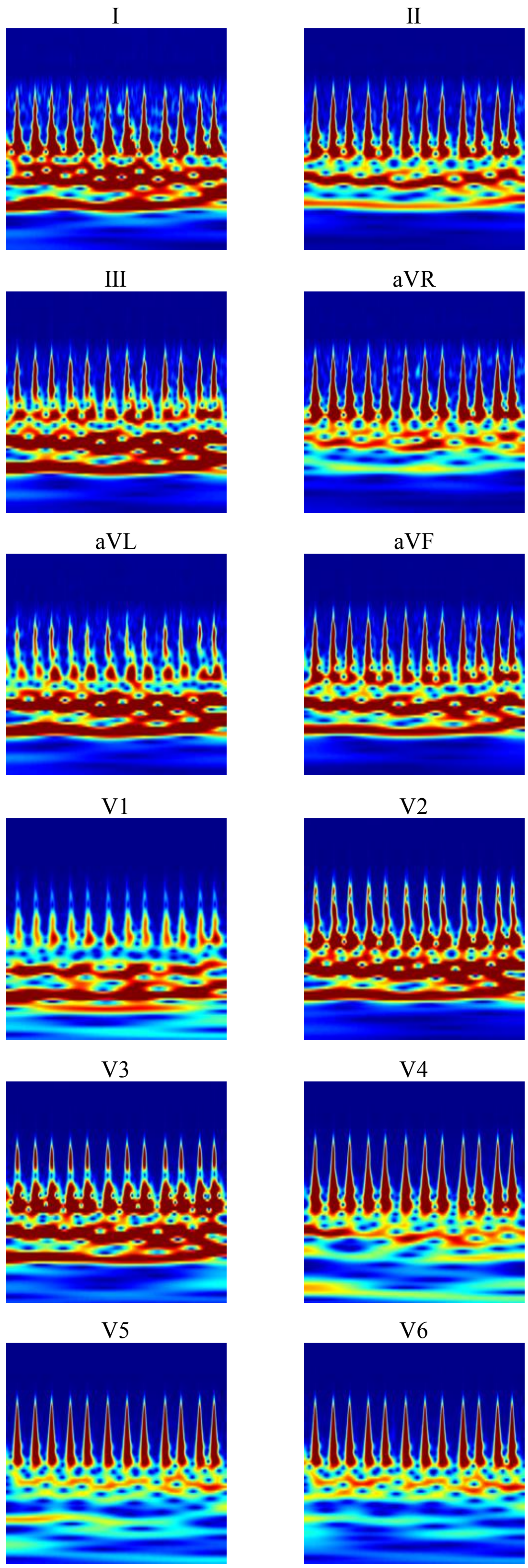

Figure 2. Scalograms from all 12 leads for a patient with left bundle branch block (LBBB).

2.1. Data Reduction

We reduced the diagnosis categories in the training set to the 27 individual diagnoses that were designated for classification as well as combinations of diagnoses that had at least 50 instances in the training set. This resulted in 76 possible diagnoses, and we ignored any data that did not contain at least one of these 76 options. The combined diagnoses used for training are shown in Table 1.

Table 1. Diagnosis code combinations used for training.

#	SNOMED codes	#	SNOMED codes
1	39732003,426783006	18	426177001,428750005
2	164934002,426783006	19	426783006,698252002
3	164873001,426783006	20	164889003,55930002
4	426783006,427393009	21	164934002,427084000
5	426783006,713426002	22	284470004,284470004
6	164865005,426783006	23	164867002,427084000
7	426177001,426783006	24	164909002,426783006
8	164889003,59118001	25	270492004,426177001
9	164951009,426783006	26	164873001,426177001
10	427084000,428750005	27	284470004,426783006
11	426783006,427084000	28	426627000,428750005
12	426783006,55930002	29	164889003,428750005
13	164861001,426783006	30	164867002,426627000
14	164889003,164934002	31	270492004,426783006
15	164934002,425623009	32	164947007,426783006
16	111975006,164930006	33	284470004,59118001
17	164884008,426783006	34	164884008,59118001
35	164865005,164951009,426783006		
36	39732003,426783006,445118002		
37	164861001,164873001,426783006		
38	164934002,39732003,426783006		
39	164909002,39732003,426783006		
40	164865005,39732003,426783006		
41	164865005,164917005,426783006		
42	111975006,164930006,428750005		
43	164861001,164873001,164889003		
44	39732003,426177001,426783006		
45	164873001,164934002,426783006		
46	39732003,426783006,713426002		
47	164865005,164951009,39732003,426783006		
48	164865005,39732003,426783006,445118002		
49	164865005,164951009,39732003,426783006,445118002		

By using this set of predefined combinations, we intentionally limited the different possible outcomes, rather than allowing all possible variations. We selected ten of each of these signals for training in order to have a balanced dataset that could be processed in a reasonable timeframe within our limited processing capabilities.

2.2. SqueezeNet

SqueezeNet is a small convolutional neural network that has been demonstrated to have accuracy similar to AlexNet on ImageNet data with significantly less parameters [9]. It further provides several advantages because of its smaller size, including reduced communication for distributed servers during training, reduced bandwidth for model export, and increased variety of possible platforms for deployment. SqueezeNet is available in MATLAB as part of the Deep Learning Toolbox; additionally, the R2020a version of the Deep Learning Toolbox provides the use of SqueezeNet without having to install a support package [10].

The last six layers of the SqueezeNet model in MATLAB [11] are shown in Table 2.

Table 2. Final six layers of SqueezeNet.

Layer description	Additional details
Dropout	50% dropout
Convolution	1000 1x1x512 convolutions with stride [1 1] and padding [0 0 0 0]
ReLU	
Global Avg Pooling	
Softmax	
Classification Output	crossentropyex with ‘tench’ and 999 other classes

For our Challenge submission in the official phase, we modified the layers shown above as follows. We replaced the last dropout layer in the network with a dropout layer with 60% probability rather than 50% [11]. The 1-by-1 convolutional layer, which is not a fully connected layer, was replaced with a convolutional layer with the number of filters set to the number of potential output classes. The final layer was replaced with a classification layer without class labels. This modified model is shown in Figure 3, and training parameters for the model are shown in Table 3.

As shown in Table 3, the stochastic gradient descent with momentum (SGDM) optimizer was selected as the solver for our model. It is a very commonly used solver in machine learning applications that has been shown to yield fast convergence [12]. It has been successfully utilized for training deep learning networks under both convex and non-convex settings for smooth objectives [12,13].

Table 3. Model training parameters [11].

Parameter	Value
Solver	Stochastic gradient descent with momentum (SGDM) optimizer
Initial learning rate	3e-4
Mini batch size	10
Max epochs	15
Validation frequency	Total # training samples / Mini batch size

2.3. Voting Scheme

Using the 12 models, one for each lead, a voting scheme is implemented for classifying new samples. Namely, each model can potentially assign a value of 0.083 to each diagnosis code. After each of the 12 models have made their individual predictions, the scores for each diagnosis across all of the leads are summed together. Any diagnosis code that has a score greater than 0.3, meaning at least four votes, is labeled as one of the diagnoses for that sample. If there is no score greater than 0.3, the diagnosis or diagnoses with the maximum score are used for the classification label. By using a voting scheme, the confidence of the resulting classification labels should be increased [14]. There are different potential options for combining the results of multiple classifiers, but we chose the straightforward one described above as a starting point for the purposes of the Challenge.

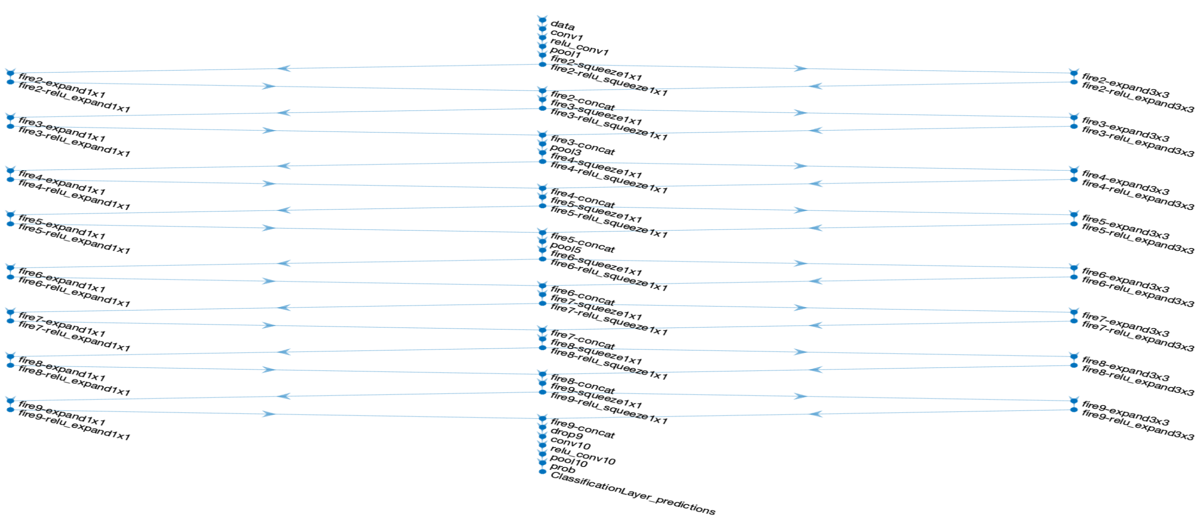

Figure 3. SqueezeNet Layer Graph

3. Results

In the unofficial phase of the Challenge, the best performing entry for team Eagles received F_beta score = 0.310 and G_beta score = 0.170. Validation accuracy ranged from 83-98% for the different arrhythmias during training. In the official phase of the Challenge, our team placed 20th out of 41 teams. Our score on the validation set was 0.214, and our score on the full test set was 0.205. Details of the test dataset and scoring algorithms used in the Challenge can be found in [8].

4. Discussion and Conclusions

Benefits of this approach include fast training time both from leveraging transfer learning, as well as the small size of SqueezeNet. However, while our results show some promise, there is noticeably significant room for improvement. One of the difficulties encountered during the Challenge was that our intended pre-trained model, in which we had expended considerable development time, was GoogLeNet [15], but due to limitations in the test environment, we were unable to obtain results for this model. GoogLeNet is similar to SqueezeNet in the sense that both are pretrained models used for image classification, but GoogLeNet is a deeper convolutional neural network capable of more complex classification. GoogLeNet has a depth of 22 layers with parameters, whereas SqueezeNet has 18. We hope to test out our original design using GoogLeNet in the future.

Another limitation of our work is related to lack of sufficient computing resources, which affected the sophistication of training we were able to accomplish. In particular, using only ten signals per arrhythmia was an unfortunate limitation of our available computing power and should be increased significantly to potentially improve classification accuracy.

References

[1] Kligfield P, Gettes LS, Bailey JJ, et al., "Recommendations for the standardization and interpretation of the electrocardiogram: part I: the electrocardiogram and its technology a scientific statement from the American Heart Association Electrocardiography and Arrhythmias Committee, Council on Clinical Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society endorsed by the International Society for Computerized Electrocardiology," *Journal of the American College of Cardiology*, vol. 49, no. 10, pp. 1109-1127, 2007.

[2] Bickerton M and Pooler A, "Misplaced ECG electrodes and the need for continuing training," *British Journal of Cardiac Nursing*, vol. 14, no. 3, pp. 123-132, 2019.

[3] Tuncer T, Dogan S, Pławiak P and Acharya UR, "Automated arrhythmia detection using novel hexadecimal local pattern and multilevel wavelet transform with ECG signals," *Knowledge-Based Systems*, vol. 186, 104923, 2019.

[4] Sharma M, Tan RS and Acharya UR, "Automated heartbeat classification and detection of arrhythmia using optimal orthogonal wavelet filters," *Informatics in Medicine Unlocked*, vol. 16, 100221, 2019.

[5] Oh SL, Ng EY, San Tan R and Acharya UR, "Automated beat-wise arrhythmia diagnosis using modified U-net on extended electrocardiographic recordings with heterogeneous arrhythmia types," *Computers in Biology and Medicine*, vol. 105, pp. 92-101, 2019.

[6] Yildirim O, Baloglu UB, Tan RS, et al., "A new approach for arrhythmia classification using deep coded features and LSTM networks," *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 121-133, 2019.

[7] Acharya UR, Fujita H, Oh SL, et al., "Deep convolutional neural network for the automated diagnosis of congestive heart failure using ECG signals," *Applied Intelligence*, vol. 49, no. 1, pp. 16-27, 2019.

[8] Alday EAP, Gu A, Shah A, et al., "Classification of 12-lead ECGs: The PhysioNet/Computing in Cardiology Challenge 2020," *Physiol. Measurement*, In Press.

[9] Iandola FN, Han S, Moskewicz MW, et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv preprint*, arXiv:1602.07360, 2016.

[10] Deep Learning Toolbox Model for SqueezeNet Network. MathWorks Deep Learning Toolbox Team. Available from: https://www.mathworks.com/matlabcentral/fileexchange/67005-deep-learning-toolbox-model-for-squeezenet-network [Accessed 23 August 2020].

[11] Classify Time Series Using Wavelet Analysis and Deep Learning (MATLAB Help Center Tutorial). Available from: https://www.mathworks.com/help/wavelet/examples/classify-time-series-using-wavelet-analysis-and-deep-learning.html [Accessed 23 August 2020].

[12] Liu Y, Gao Y and Yin W, "An improved analysis of stochastic gradient descent with momentum," *arXiv preprint*, arXiv:2007.07989, 2020.

[13] Li X and Orabona F, "A high probability analysis of adaptive SGD with momentum," *arXiv preprint*, arXiv:2007.14294. 2020.

[14] Battiti R and Colla AM, "Democracy in neural nets: Voting schemes for classification," *Neural Networks*, vol. 7, no. 4, pp. 691-707, 1994.

[15] Szegedy C, Liu W, Jia Y, et al., "Going deeper with convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.

Address for correspondence:

Dr. Grace Mirsky
Benedictine University
5700 College Road
Lisle, IL 60532, USA
gmirsky@ben.edu