Using Mel-Spectrograms and 2D-CNNs to Detect Murmurs in Variable Length Phonocardiograms

Marius S Knorr, Jan P Bremer

Department of Cardiology, University Heart & Vascular Center Hamburg, University Medical Center Hamburg-Eppendorf, Hamburg, Germany

Abstract

As part of the George B. Moody PhysioNet Challenge 2022, we developed a computational approach for identifying abnormal cardiac valve function from phonocardiograms (PCGs). Our team, uke-cardio, developed a deep learning model that uses mel-spectograms of up to four different auscultation locations. CutMix and smooth labels were used to simultaneously learn different tasks. On the hidden test set, the classifier achieved a weighted accuracy score of 0.735 for murmur detection (ranked 15 out of 40 teams) and a challenge cost score of 11990 *(ranked 8 out of 39 teams) on the outcome detection task. The source code is available at: github.com/msknorr/cinc22-pcg*

1. Introduction

The phonocardiogram (PCG) refers to the audio recording of heart activity, usually obtained with an electronic stethoscope. PCGs can be used to uncover abnormal heart sounds, such as murmurs, related to heart conditions. The PCG is non-invasive and easy to obtain and thereby allows for accessible screening of murmurs in resource-constrained environments. The George B. Moody PhysioNet Challenge focuses on automated, open-source approaches for classifying abnormal cardiac function [1]. The goal of this year's challenge is to automatically classify abnormal heart function and clinical outcome from phonocardiograms [2] using data from mass screening campaigns of young individuals [3]. We approached the problem as a multi-class and multi-label learning task in order to build a robust one-fits-all algorithm. The two main hurdles of the challenge were a comparably small development dataset and dealing with the variable number of audio recordings of different ausculations locations (e.g., only AV was recorded) of varying length. To overcome these hurdles, our entry incorporated three main concepts:

1. CutMix and soft targets
2. Common feature extractor
3. Time-series based pooling

These main concepts were combined and carefully evaluated with a nested 6-fold cross-validation scheme.

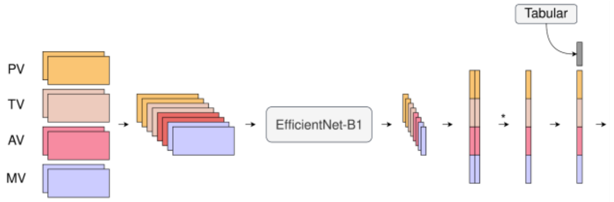

Figure 1. The feature extractor (EfficientNet-B1) takes as input crops of mel-spectrograms of up to four different auscultation locations (PV, TV, AV, MV). A reshape and pool (*) operation pools multiple crops from the same time-series of the same valve. The four resulting pooled embedding vectors were concatenated with tabular features and fed into a stack of linear layers for the competition tasks (not shown here).

2. Method

Briefly, raw audio data of up to 4 different auscultation locations were transformed to mel-spectrograms. These were cropped and passed into a 2D-CNN feature extractor. The resulting embedding vector was combined with tabular data, to finally retrieve the class probabilities for different classification tasks. Using held out data, we found the optimal threshold for the binarization of the predictions to retrieve the final prediction. The training dataset consisted of 942 unique patients. Our pipeline therefore incorporated 3 main concepts to overcome overfitting in this competition: **1. CutMix and soft targets.** To prevent overfitting, we employed a variation of the CutMix regularization technique [4] during training. Here, the dataset was artificially enlarged by mixing the PCGs of different patients. That is, a datapoint may be put together by three valves of patient one (e.g., Present murmur) and one valve of a different patient (e.g., Absent murmur). The targets were adjusted accordingly, resulting in soft targets (e.g., 0.75 Present, 0.0 Unknowns, 0.25 Absent). **2.**

Common feature extractor. Instead of training a CNN for each auscultation location individually, we employed a single common feature extractor. Although this single backbone was used for all four locations by collapsing the respective tensor dimension (2.2 Model), information about the auscultation location was not lost as the feature extractor output was concatenated location-wise, resulting in a four times larger embedding vector. **3. Time-series based Pooling.** Murmurs are present to varying extents in consecutive heart cycles. Thus, if the time-series is cropped into multiple parts, one cannot be sure that certain parts also include murmurs. Therefore, to allow the model to look at the whole audio sequence while keeping the benefit of a small and uniformed sized crop, the time-series of embedding vectors of mel-spectogram crops were pooled per PCG.

2.1. Audio preprocessing

Raw audio files, sampled at 4000 Hz, were converted to a 2D-mel-spectrogram representation, which more closely resembles the frequency distance heard by the human ear compared to a standard spectrogram. Using time-frequency representations, i.e., a mel-spectrogram, instead of directly applying a 1D-CNN on the audio data is expected to perform better as shown for electrocardiogram (ECG) data [5]. We chose the number of mels of 200 (yielding 200 individual frequencies), num_fft of 256 and a hop length of 64. Mel-spectrogram's pixel values were z-transformed. We resized the frequency axis (length 200) to 224 and cropped 224 long time windows without overlap from the spectrogram, equivalent to approximately 3 second windows. A crop usually contains 3-5 heart cycles. During training, 5 crops for each location were used. For inference and validation, we used 7 crops to account for longer sequences. Missing leads or short sequences were padded with zeros. During training, coarse dropout and random time and frequency dropouts were applied in order to increase robustness. During inference and validation, no augmentations were applied.

2.2. Model

The model (figure 1) receives a 6-dimensional vector:

[bs, n_leads, n_crops, c, x, y]

with
bs = batchsize
n_leads = PCG locations (AV, MV, PV, TV)
n_crops = number of mel-spectrogram crops from one auscultation location
c = color channels
x = number of mels of the mel-spectrogram
y = time-axis of the mel-spectrogram

with *bs* = 3, *n_leads* = 4, *n_crops* = 5 (inference: 7), 3 color channels and 244 points in time and 244 frequency bands. A reshape operation collapses the first three dimensions:

*[bs, n_leads, n_crops, c, x, y] -> [bs * n_leads * n_crops, c, x, y]*

The resulting 4-dimensional tensor is forwarded into an EfficientNet-B1 [6] feature extractor that outputs a feature vector of size 25 for each crop. Then, the first dimension of the tensor is reshaped back into the initial 3 dimensions. This process of collapse, feature embedding and reshaping results in the embedding of the mel-spectrograms through a common feature extractor, while preserving the information of leads and crops:

*[bs * n_leads * n_crops, 100] -> [bs, n_leads, n_crops, 25]*

Next, the tensor is average pooled along the time dimension (*n_crops*):

[bs, n_leads, n_crops, 25] -> [bs, n_leads, 25]

and reshaped so that for each patient a feature vector for all 4 leads remains:

*[bs, n_leads, 25] -> [bs, n_leads * 25]*
-> [bs, 100]

This embedding was concatenated with tabular data (2.4 Tabular data), followed by a dropout layer and an intermediate linear layer. Then, for each of our targets, a head of linear layers follows.

2.3. Targets and losses

In addition to the main task of predicting murmur and outcome probabilities, auxiliary tasks function as model regularization. These additional tasks are supposed to improve robustness, performance and data efficiency [7], [8] but could also harm the generalization performance (‘negative transfer’) [9] when weighted unfavorably. Our model predicts eight targets in total with varying output dimensions (table 1). Depending on the task, either a softmax activation function was applied and trained with categorical cross entropy (CCE) loss for multi-class classification tasks, or sigmoid activation and binary cross entropy (BCE) for multi-label classification.

Task	out_dim	loss
Murmur	3	CCE
Outcome	2	CCE
Where hearable	4	BCE
Timing	5	CCE
Shape	5	CCE
Grading	4	CCE
Pitch	4	CCE
Quality	4	CCE

Table 1. A composition of tasks the model learned. Murmur and outcome refer to the challenge objective with 3 and 2 output neurons respectively. The remaining tasks refer to murmur-related auxiliary tasks. CCE = Categorical

cross entropy, BCE = Binary cross entropy.

The losses were weighted based on their magnitude and relevance to the main task to form the loss $\mathscr{L}_{\text{total}}$ according to:

$$\mathscr{L}_{\text{aux}} = (\mathscr{L}_{\text{timing}} + \mathscr{L}_{\text{shape}} + \mathscr{L}_{\text{grading}} + \mathscr{L}_{\text{pitch}} + \mathscr{L}_{\text{quality}}) / 5$$

$$\mathscr{L}_{\text{total}} = (\mathscr{L}_{\text{murmur}} * 3 + \mathscr{L}_{\text{outcome}} + \mathscr{L}_{\text{where\_hearable}} + \mathscr{L}_{\text{aux}} * 2) / 6$$

During training, a batch size of 3 was used, with an Adam optimizer [10] with weight decay (0.0005), an initial learning rate of 0.0001, and a scheduler that reduces the learning rate by a factor of 0.5 after 3 epochs without validation loss improvement. CutMix was randomly applied during training in 80% samples.

2.4. Tabular data

Tabular data was concatenated with the mel-spectrogram feature vector, followed by a dropout layer to regularize training in the low data regime. The tabular data consisted of a one-hot encoded representation of sex and pregnancy status (e.g., male -> [0, 1], female -> [1, 0]), and a stairway encoded one-hot vector of 5 given age specifications (Neonate, Infant, Child, Adolescent, Young adult). Missing values were set to Child. Finally, available auscultation locations were encoded in a vector of length 4 (e.g., only valve 2 available -> [0, 1, 0, 0]).

2.5. CutMix and soft targets

The application of the CutMix augmentation method improves robustness and out-of-distribution detection performance [4]. During training, auscultation locations were mixed between patients. That is, a datapoint may be put together by three valves of a patient and one valve of a different patient. The targets were adjusted accordingly. For example: patient 1 had a murmur and patient 2 was healthy, and one lead of patient 1 was replaced by one lead of patient 2, the new label would be set to [0.75, 0, 0.25], as now the contribution of the murmur class was only 75% (3 of 4 leads).

2.6. Local validation routine

We used a nested stratified cross-validation procedure. The training data was split patient-wise and stratified by occurrences of murmur labels. The whole dataset was split in six folds. One of the six was used to simulate the hidden test set. From the other five splits, three were used for training, one for local validation (≠ validation data) and one for threshold selection. These 'inner-folds' were shuffled and repeated 5 times, yielding 5 models. A submission to the competition consists of the average prediction of these 5. For the cross validation (CV) score, 6 individual scores (5 models each) on the local test data are reported. Therefore, a full CV requires training of 30 models and a submission requires training 5.

2.7. Binarization

During inference, initially all patients were classified as Present in the murmur detection task. Then, two thresholds were applied: If the model probability for Unknown was greater than the Unknown threshold, the patient was set to Unknown. Afterwards, the same procedure was applied for Absent. That is, Unknown may override Present, and Absent may override Unknown. Thresholds were determined by iterating in steps of 0.05 over the Unknown and Absent threshold and reporting the weighted accuracy on the threshold selection dataset as a heatmap. Then, the scores were rounded to second decimal place. If two or more thresholds resulted in the same score, the 'final' threshold was determined by taking the average.

3. Results

Our team achieved a weighted accuracy score of 0.735 (ranked 15 out of 40 teams) and a challenge cost score of 11990 (ranked 8 out of 39 teams) for the outcome prediction task on the hidden test set. Murmur scores can be found in table 2 and 3, outcome scores are reported in table 4.

Training	Validation	Test	Ranking
0.74±0.04	0.68	0.735	15/40

Table 2. Weighted accuracy scores for the murmur detection task. We used 6-fold cross validation on the public training set (mean and SD), repeated scoring on the hidden validation set, and one-time scoring on the hidden test set.

AUROC	AUPRC	F-measure	Accuracy	Weighted Accuracy
0.89	0.735	0.597	0.79	0.735

Table 3. Advanced metrics for the murmur detection task on the hidden test dataset.

Training	Validation	Test	Ranking
10567	10105	11990	8/39

Table 4. Cost scores for the outcome detection task.

4. Discussion

We approached the 2022 PhysioNet Challenge with deep learning methodology, even though the dataset was relatively small. Since 2D-CNNs perform well on mel-spectrograms or spectrograms in general, we used them together with methods to prevent overfitting, namely CutMix of auscultation locations of different patients with soft targets, a small feature extractor and finally strong audio augmentation. The cross-validation score on the training dataset did not differ much from the final competition scores on the murmur task, implying that we could successfully circumvent generalization issues on unseen data and trust our local validation routine.

References

[1] A. L. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,” *Circulation*, vol. 101, no. 23, pp. E215-220, Jun. 2000, doi: 10.1161/01.cir.101.23.e215.

[2] M. A. Reyna *et al.*, “Heart Murmur Detection from Phonocardiogram Recordings: The George B. Moody PhysioNet Challenge 2022.” medRxiv, p. 2022.08.11.22278688, Aug. 16, 2022. doi: 10.1101/2022.08.11.22278688.

[3] J. Oliveira *et al.*, “The CirCor DigiScope Dataset: From Murmur Detection to Murmur Classification,” *IEEE J. Biomed. Health Inform.*, vol. 26, no. 6, pp. 2524–2535, Jun. 2022, doi: 10.1109/JBHI.2021.3137048.

[4] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features.” arXiv, Aug. 07, 2019. doi: 10.48550/arXiv.1905.04899.

[5] J. Huang, B. Chen, B. Yao, and W. He, “ECG Arrhythmia Classification Using STFT-Based Spectrogram and Convolutional Neural Network,” *IEEE Access*, vol. 7, pp. 92871–92880, 2019, doi: 10.1109/ACCESS.2019.2928017.

[6] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *CoRR*, vol. abs/1905.11946, 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

[7] L. Liebel and M. Körner, “Auxiliary Tasks in Multi-task Learning.” arXiv, May 17, 2018. doi: 10.48550/arXiv.1805.06334.

[8] B. Rafiee, J. Jin, J. Luo, and A. White, “What Makes Useful Auxiliary Tasks in Reinforcement Learning: Investigating the Effect of the Target Policy.” arXiv, Apr. 01, 2022. doi: 10.48550/arXiv.2204.00565.

[9] Z. Meng, X. Yao, and L. Sun, “Multi-Task Distillation: Towards Mitigating the Negative Transfer in Multi-Task Learning,” in *2021 IEEE International Conference on Image Processing (ICIP)*, Sep. 2021, pp. 389–393. doi: 10.1109/ICIP42928.2021.9506618.

[10] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. doi: 10.48550/arXiv.1412.6980.

Address for correspondence:

Marius S Knorr
Department of Cardiology, University Heart & Vascular Center Hamburg, University Medical Center Hamburg-Eppendorf, Hamburg, Germany
marius.knorr@stud.uke.uni-hamburg.de