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

Marius S. Knorr[1] & Jan P. Bremer[1], Renate B. Schnabel[1]

[1]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. Our one-fits-all algorithm uses cutmix and smooth labels to simultaneously train different tasks. The classifier achieved a weighted accuracy score of 0.68 for murmur detection (ranked X out of 80 teams) and a challenge cost score of 10104 (ranked X out of 100 teams) on the hidden testing set.

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 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. Pipeline design choices heavily rely on the fact that the dataset is comparably small for deep learning and therefore overfitting must be narrowed down. Further, audio recordings of different ausculations locations are often incomplete (e.g., only AV was recorded) and of varying length. To overcome these two hurdles, our pipeline incorporates 3 main concepts:

1. Cutmix and soft targets
2. Common feature embedder
3. Time-series based pooling

These main concepts are combined with a careful nested 6-fold cross-validation scheme.

2. Method

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

Approach

Our training-pipeline incorporates 3 main concepts to overcome overfitting in this competition:

1. Cutmix and soft targets: To prevent overfitting, the dataset was 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 (e.g. 0.75 Present, 0 Unknowns, 0.25 Absent).
2. Common feature embedder backbone: Although a single backbone was used for all 4 locations by collapsing the respective tensor dimension, information about the location was not lost as the

feature extractor output was concatenated location-wise, making the embedding vector 4 times larger.

3. Pooling: Murmurs are not present in every heart cycle. Thus if the time-series is cropped into multiple parts, one can not be sure that certain parts also include murmurs. Therefore, to allow the model to look at the whole audio sequence, while keeping the perks of a small and uniformed sized crop, the time-series of embedding vectors of mel-spectogram crops were pooled per PCG.

Raw audio data of up to 4 different auscultation locations were transformed to mel-spectrograms. These were fed into a 2d-CNN feature extractor. The feature extractor computed an embedding vector which was pooled per PCG so that for each patient, one feature vector per auscultation location remained. These were concatenated with tabular data. Using held out data, we found the optimal threshold for the final label.

Audio preprocessing

Raw audio files, sampled at 4000 Hz, were converted to a 2d-mel-spectrogram representation, which more closely resembles the sound volume heard by the human ear than a standard spectrogram. We choosed a number of mels of 200, num_fft of 256 and a hop length of 64. Mel-spectrogram's pixel values were z-transformed. We resized the mel axis (length 200) to 224 and cropped 224 long time windows without overlap from the spectrogram. 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. A crop usually contains 3-5 heart cycles.

Audio augmentation

During training, we applied coarse dropout and random time and frequency dropouts in order to increase the robustness of the model. During inference and/or validation, no augmentations were applied.

Model

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

bs, n_leads, n_crops, channel, x, y

with

bs = batchsize
n_leads = PCG locations (AV, MV, PV, TV)
n_crops = number of mel-spectrom crops from one PCG-location
x = number of mels of the mel-spectrogram
y = time-axis of the mel-spectrogram
with bs = 3, num_leads = 4, num_crops=5, 3 color channels and 244 points in time and 244 (200) frequency bands. A reshape operation collapses the first three dimensions. The resulting 4-dimensional tensor is forwarded into an Efficient Net-B1 feature extractor that outputs a feature vector of size 100 for each crop. Then, the first dimension of the tensors are reshaped back into the initial 3 dimensions. This process of collapse, feature embedding and reshaping resulted in the embedding of the mel-spectrograms through a common feature embedder, while containing the information of leads and crops:

*bs*n_leads*n_crops, 100 -> bs, n_leads, n_crops, 100*

Next, the tensor was 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, 100 -> bs, n_leads * 25*
=> bs, 100

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

Targets and Losses

Our model has 8 heads in total with varying output dimensions (see table 1). Depending on the task, we either used a softmax activation function and categorical cross entropy loss for multi-class classification tasks, or sigmoid activation and binary cross entropy 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 are auxiliary tasks. CCE = Categorical cross entropy, BCE = Binary cross entropy.

The losses are weighted based on their magnitude and relevance to the main task according to:

$$loss_{murmur\_aux} = (loss_{timing}, loss_{shape}, loss_{grading}, loss_{pitch}, loss_{quality}) / 5$$

$$loss = (loss_{murmur}*3 + loss_{outcome} + loss_{where\_hearable} + loss_{murmur\_aux} * 2) / 6$$

so that the model loss (figure 2) consists of several sub-tasks.

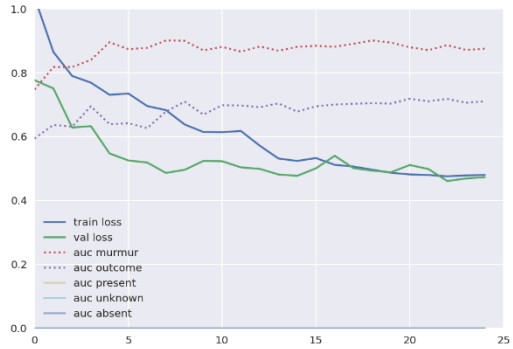

Figure 2. Model loss and AUC curves. The training loss (blue) and validation loss (green) are shown, and the area-under-curve (AUC) for the two competition tasks (red; murmur, violet; outcome). The horizontal axis indicates training epochs.

Training details

We used a batch size of 3, with an Adam optimizer with weight decay (0.0005), an initial learning rate (lr) of 0.0001, and a learning rate scheduler that reduces the lr by factor 0.5 after 3 epochs without validation loss improvement. Cut-Mix was randomly applied during training in 80% cases.

Pretraining on Cinc16

Pretrained weights were used for faster model convergence. Here, the backbone of our model was pretrained on CinC 2016 Challenge data. The dataset contains PCG recordings with absent and present labels. The pretraining was formulated as a binary classification task, but with only one crop simultaneously. However, pretraining was not used for the final entry.

Tabular data

Tabular data was concatenated with the mel-spectogram feature vector, followed by a dropout layer to regularize training in the low data regime. We used 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 lead 2 -> [0,1,0,0]).

Cut-Mix and Smooth labels

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 respectively: Say patient 1 had a murmur and patient 2 was healthy, and one lead of patient 1 was replaced by 1 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).

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 data was split in 6 parts. One of the 6 was used to imitate the *hidden* test set. From the other 5 splits, 3 were used for training, one for validation and one for threshold selection. These 'inner-folds' were shuffled and repeated 5 times. That is, a model submission consists of 5 individual models' average prediction. For the CV score, 6 individual scores are reported to approximate the hidden data.

Threshold selection

We initially set all patients to Present. Then, two thresholds were used: 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 in form of a heatmap (figure 3). 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.

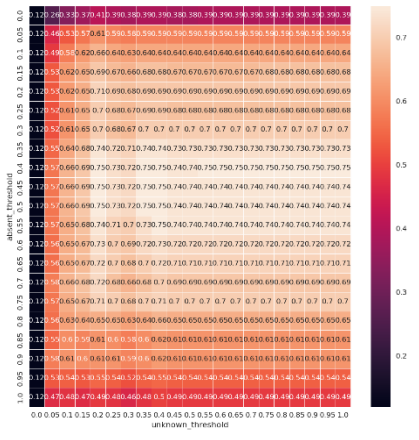

Figure 3. The heatmap shows the murmur competition metric for different thresholds on the threshold selection dataset. The final thresholds were chosen to be the average threshold value of the highest competition scores.

Here, the highest score is 0.75. The absent threshold therefore is at 0.45, since the best score of 0.75 occurs between 0.4 and 0.5. The best unknown threshold is 0.6.

4. Results

Results are reported solely for the murmur detection task, since we did not optimize for the outcome task.

Training	Validation	Test	Ranking
0.74±0.04	0.68	TBA	TBA

Table 2. Weighted accuracy metric scores (official challenge score) for our final selected entry (team uke-cardio) for the murmur detection task, including the ranking of our team on the hidden test set. We used 6-fold cross validation on the public training set, repeated scoring on the hidden validation set, and one-time scoring on the hidden test set.

5. Discussion and Conclusions

We approached the 2022 PhysioNet Challenge with deep learning methods, even though the dataset is relatively small. Since 2d-CNNs perform well on mel-spectograms 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. However, our local cross-validation (CV) resulted in superior performance scores than the 'validation data' which resembles the leaderboard (LB) during the official phase. Since the CV scores were relatively constant across different random states, we suspect (and hope) that the CV-LB-gap does not stem from bad generalization.

References

[1] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 2000;101(23):e215–e220.

[2] Reyna MA, Kiarashi Y, Elola A, Oliveira J, Renna F, Gu A, et al. Heart murmur detection from phonocardiogram recordings: The George B. Moody PhysioNet Challenge 2022. medRxiv 2022; URL https://doi.org/10.1101/2022.08.11.22278688.

[3] Oliveira J, Renna F, Costa PD, Nogueira M, Oliveira C, Ferreira C, et al. The CirCor DigiScope dataset: from murmur detection to murmur classification. IEEE Journal of Biomedical and Health Informatics 2021;26(6):2524–2535.

Address for correspondence:
marius.knorr@stud.uke.uni-hamburg.de