Heart Murmur Detection from Phonocardiogram Recordings: George B. Mood PhysioNet Challenge 2022

Enoch Guo
Stony Brook School, New York

Abstract

We present a machine learning method for heart murmur detection from phonocardiogram (PCG) recordings. Our approach consists of three steps: (1) Split recordings into 3-second waveforms; (2) Transform one-dimensional waveforms into two-dimensional time-frequency heat maps using Mel-Frequency Cepstral Coefficients (MFCC); (3) Classify MFCC using deep convolutional neural networks (CNN). We also use denoising autoencoder classification to improve the basic CNN.

1. Introduction

The goal of the Challenge is to identify the presence, absence, or unclear cases of murmurs and the normal vs. abnormal clinical outcomes from heart sound recordings collected from multiple auscultation locations on the body using a digital stethoscope.

The Challenge data contain one or more heart sound recordings for 1568 patients as well as routine demographic information about the patients from whom the recordings were taken. The Challenge labels consist of two types:

1. Murmur-related labels indicate whether an expert annotator detected the presence or absence of a murmur in a patient from the recordings or whether the annotator was unsure about the presence or absence of a murmur.
2. Outcome-related labels indicate the normal or abnormal clinical outcome diagnosed by a medical expert.

The Challenge data is organized into three distinct sets: training, validation, and test sets. The organizers have publicly released 60% of the dataset as the training set of the 2022 Challenge, and have retained the remaining 40% as a hidden data for validation and test purposes.

Therefore, we are given the training set that contains 3163 recordings from 942 patients. The public training set contains heart sound recordings, routine demographic information, murmur-related labels (presence, absence, or unknown), outcome-related labels (normal or abnormal), annotations of the murmur characteristics (location, timing, shape, pitch, quality, and grade), and heart sound segmentations. The private validation and test sets only contain heart sound recordings and demographic information.

2. Split into 3-second waveforms

The PCG recordings have different lengths, varying from 20608 to 258048 data points; with daterate being 4000 samples/sec, they vary from 5 seconds to about one minute.

PCG consists of cycles; each cycle consists of four states: S1, S2, systole and diastole. We split each recording into 3-second long overlapping segments, which is long enough to determine abnormality of heart sound. Each split wave consists of 12000 data points.

3. MFCC

Mel-Frequency Cepstral Coefficients (MFCC) [1] has been widely used in speech recognition. We extract 4 frequency bands, because if we extracted more frequency bands, other frequency bands would be mostly empty. Thus each one-dimensional wave (12000 long) is transformed into a two-dimensional heat map, of size $(4, 201)$, using $\texttt{win\_length} = 100$ and $\texttt{hop\_length} = 60$.

4. Basic CNN

We use three layers of convolution.

1. Encode:
   (a) Conv2d
   (b) BatchNorm2d
   (c) ReLU
   (d) Conv2d
   (e) BatchNorm2d
   (f) ReLU
   (g) Conv2d
   (h) BatchNorm2d
   (i) ReLU
   (j) Flatten
2. Classify:
   (a) Dense
   (b) ReLU
   (c) Dense

(d) Softmax

5. Denoising Autoencoder Classification

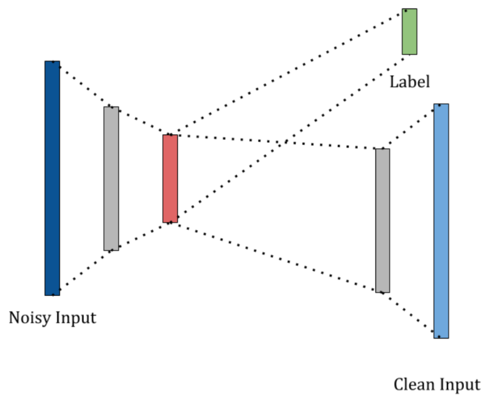

Figure 1. Denosing Autoencoder Classification Model

Training algorithm: Let X be input points (cyan), of size $(C, W, H) = (1, 4, 201)$, and Y be the classification labels (green).

1. Add noise: $X' = add\_noise(X)$. X' is blue. We have two methods to add noise: blackout and random. Blackout method chooses a small sample points (10%) to zero out their values; Random method changes the values by randomly with $\mu = 0$ and a small σ, say, 0.1.
2. Denoising autoencoder decode: Let Z be the latent layer (red). We train $X' \to Z \to (Y, X)$, where $X' \to Z$ is encode; $Z \to Y$ is classify; and $Z \to X$ is decode (decode consists of the reversed steps of encode). In this step, the loss function for classify is zero. The model is called regularized after this step.
3. Classify from the regularized model: continue to train using cross entropy loss function for classify: $Z \to Y$.

Each step in Decode is the inverse function of the corresponding Encode step, in reverse order.

3. Decode
   (a) Unflatten
   (b) ConvTranspose2d
   (c) BatchNorm2d
   (d) ReLU
   (e) ConvTranspose2d
   (f) BatchNorm2d
   (g) ReLU
   (h) ConvTranspose2d
   (i) Sigmoid

References

[1] T. Ganchev, N. Fakotakis, G. Kokkinakis, “Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task,” *Proc. of the SPECOM-2005*, October 17-19, 2005. Patras, Greece. Vol. 1, pp.191-194.