BSP Project
Classifying Murmurs - Team L1-D

Denissza Incze (GIVO61)
incze.denissza@hallgato.ppke.hu

Noémi Nagy (FZEQG3)
nagy.noemi.6@hallgato.ppke.hu

Subham Shome (GPB0A8)
shome.subham@hallgato.ppke.hu

Location: 1083 Budapest, Práter str. 50./A
Pázmány Péter Catholic University - Faculty of Information Technology and Bionics

***Abstract* – Phonocardiogram (PCG) recording is a recording method that captures the heart sounds as the valves are closing and the blood flows though the heart. Beside being cost-effective, it is not the most accurate and most sensitive method, since hearing a sound is a perceptive characteristic of the brain. To create an objective evaluation computer science must be involved. Our task was to create a classification method that decides, whether the given signal corresponds to a normal or pathological recording. The George B. Moody PhysioNet Challange has the same goal with an enormous dataset, called the CirCor DigiScope dataset. In this project we had to work on a simplified dataset, that is based on the original one. After a brief introduction to the motivation behind this project and providing some theoretical background, the methods and applied tools are detailed. It is followed by the description about the structure of the project, how it is constructed with the help of different scripts and functions in MATLAB. In the end partial results are listed, however, the final evaluation will be carried out on a hidden dataset, which is unknown for us.**

Keywords – PCG, heart sounds, murmurs, signal processing, feature extraction, classification

I. Introduction

The task in this project was to create a method to classify heart sounds based on whether they contain heart murmurs or not. The basis of this project was the George B. Moody PhysioNet Challange: we have been working on the same dataset, however, the lecturers already pre-selected some of them to make our task easier.

Cardiac auscultation (characterising heart sounds and murmurs) is the most cost-effective tool for cardiac pre-screening, however, this method has limited accuracy and sensitivity, furthermore, years of experience is required to evaluate the acoustic signals, which is a rather non-objective decision due to clinical disagreements. [1]

Heart sounds are generated by the cardiac valves as they open and close during the cardiac cycle. The cardiac cycle is the following:

1) The Pulmonary Veins from the lungs bring oxygenated blood to the Left Atrium of the heart
2) Through the Mitral Valves the blood flows into the Left Ventricle
3) Through the Aortic Valves the blood is pumped to the Aorta and Systemic Arteries: this is how the oxygenated blood enters the systemic circulation
4) From the different tissues deoxygenated blood returns to the Right Atrium via the Systemic Vena cavas
   - Superior Vena cava: from the upper body
   - Inferior Vena cava: from the lower body
5) The blood is pumped through the Tricuspid Valve into the Right Ventricle
6) Through the Pulmonary Valves deoxygenated blood enters the pulmonary circulation thus CO_2 can be exchanged to O_2 in the capillaries of the lungs

Then the process restarts again from step 1.

During the cardiac cycle the blood pressure signal changes, most conspicuously in the Ventricles, in the Atria it is almost constant. When a BP signal originates from an artery, 6 main phases can be observed:

1) Systolic upstroke
2) Systolic peak pressure
3) Systolic decline
4) Dicrotic notch: this sound may arise from the closing of the Semilunar Valves, but this is not the complete explanation
5) Diastolic runoff
6) End-diastolic pressure

Phonocardiogram recordings (PCG) normally capture the first (S1: closure of the mitral and tricupsid valves) and second (S2: closure of the aortic and pulmonary valve), while further heart sounds (S3, even S4) indicate pathology. [2] S1 can be heard at the beginning of the systole, while S2 occurs at the beginning of diastole. Based on this the cardiac cycle can be segmented on the systolic phase (between S1 and S2) and the diastolic phase (between S2 and the next S1). If further sounds are heard, those are called murmurs (S3, S4, etc), each of them have their own characteristics. However, how a sound is heard is based on perceptiveness, thus it is hard to determine an objective evaluation for heart sounds.

The aim of the original Challenge was to develop fully automated processes to detect murmurs. In our project - on an easier dataset - by applying the adequate methods for each step in the process we tried to obtain a similar tool that classifies heart sounds based on whether they contain heart murmurs or not.

A. Heart Murmurs

Heart murmurs are heart sounds, which are generated when turbulent blood flow (not smooth) creates a sound that could be heard be a stethoscope. There are two main types of murmurs:

- Functional murmur: caused by physiological conditions
- Structural murmur: caused by alterations in the heart's structure

Furthermore, the noises can be classified based on their pitch, timing and shape. Considering the many different characteristics of them, they can be:

- Early-systolic or Early-diastolic
- Mid-systolic or Mid-diastolic
- Late-systolic or Late-diastolic
- Holo-systolic or Holo-diastolic

II. Methods

A. Dataset

The original dataset is the CirCor DigiScope dataset is by far the largest publicly available heart sound dataset (5282 recordings), including recordings collected from multiple auscultation locations on the body. Furthermore, the study resulted in a very detailed murmur characterization and classification database (including timing, pitch, grading, shape, quality, auscultation location, etc.), which can be used in future research [3]. The PCGs were recorded using an electronic auscultation device (Littmann 3200 stethoscope) from four prominent auscultation locations on the body: aortic valve, pulmonary valve, tricuspid valve, and mitral valve. However, some patients have recordings from fewer than four locations, and some have multiple recordings per location. Thus, the number of recordings, location, and duration varies between patients. For each patient, the PCGs were recorded by the same operator sequentially (not simultaneously) from different locations on the patient's body [1].

For this project we received a pre-selected and smaller dataset to make our work easier; 50 recordings of normal heart sounds and 50 recordings of murmurs. The PCG recordings' sample rate is 4 kHZ. Beside the recordings we also received the heart sound location files with corresponding IDs (1-S1, 2-systole, 3-S2, 4-diastole, 0-unknown), the measured heart rate (in bpm), a MATLAB function template and a MATLAB runner and evaluator script.

After we downloaded the dataset we read it in and created variables for the audio signals (.wav) and the corresponding labels (.tsv). In the label files the first and the second column is the start and end of the given event (seconds), the third column contains the IDs.

B. Preprocessing

1) Bandpass Filtering: Bandpass filtering is usually employed in heartbeat analysis to remove the low and high frequency noise from stethoscope signals. In our solution the bandpass filter was chosen to keep those components that are between 20 and 400 Hz, based on literary research [10].

Signal from the filtering stages is next fed to the normalization stage. Normalization removes the amplitude variation for inter- and intra-signal classification.

2) Shannon Energy: Shannon Energy is used as a calculation method for computing envelopes; however, this computation is done by adaptive windows. It is calculated as follows:

$$SE(t) = -signal^2(t) \cdot log(signal^2(t))$$

In order to evaluate envelopes average Shannon Energy is calculated in a specific time duration:

$$SE_{avg} = -\frac{1}{N}\sum_{t=1}^{N} SE(t)$$

where N is the window size [5].

Shannon energy envelope is most widely employed method in case of time-based techniques to localise S1 and S2 peaks.

3) Normalisation: In data processing tasks during the preprocessing phase it is essential to perform normalisation since it might not be known, whether all the data have their attributes on the same scale. After this step they will become comparable by adjusting the values to a common scale.

C. Peak Detection

To carry out this step the *findpeaks()* MATLAB function was applied; it returns the local peaks of the input vector. A point is considered as a local maxima if it is greater than its neighbours or equals to infinity [7]. The 'MinPeakHeight' is set to be 0.005 units of amplitude, while the 'MinPeakDistance' attribute is set to 0.2 times the sampling frequency.

When this step is carried out, the heart rate can be calculated, which will be evaluated later.

D. Binary Mask

Masking in general means that some part of the data gets hidden, while the others remain visible without corrupting the structure of the data. After the peaks are detected, binary mask in our model is used to determine systolic and diastolic phases of the recording.

E. Feature Extraction

After the signals are transformed into the frequency domain by a simple Fast Fourier Transform (FFT). FFT is a useful method to obtain frequency information of a given, time-varying signal. The maximum frequency component is extracted, and with the help of it, we obtained the power spectral density, as well. The power spectral density (PSD) shows at which frequencies variations are strong and at which frequencies variations are weak [4].

The next calculation considered the short-time Fourier transform (STFT), which is used to analyze how the frequency content of a non-stationary signal changes over time and helps with the spectral analysis, as well.

Another time-frequency connected step was calculating the wavelet decomposition of signals. It is a common method to retrieve time and frequency characteristics of a signal at the same time, thus it is useful in case of feature extraction [8].

The last step in feature extraction is to create a feature table which will be used later in the classification steps. The following features were extracted (from time and frequency domain, as well): mean, median, mean absolute deviation, skewness, kurtosis, inter-quantile range, entropy, sum of the systolic and diastolic regions, max frequency peak, maximum of the power spectrum density, sum of the STFT, sum of the 5th and 6th order wavelets the MFC coefficients (MFCC) and the 6th level of Empirical Mode Decomposition (EMD). MFCC is short for Mel-Frequency Cepstral Coefficients. Mel-frequency cepstrum (MFC) is the compressed representation of the Mel-scaled power spectrogram, which can be found by taking the Discrete Cosine Transform (DCT) of a log power spectrum on a nonlinear Mel-scale of frequency. The

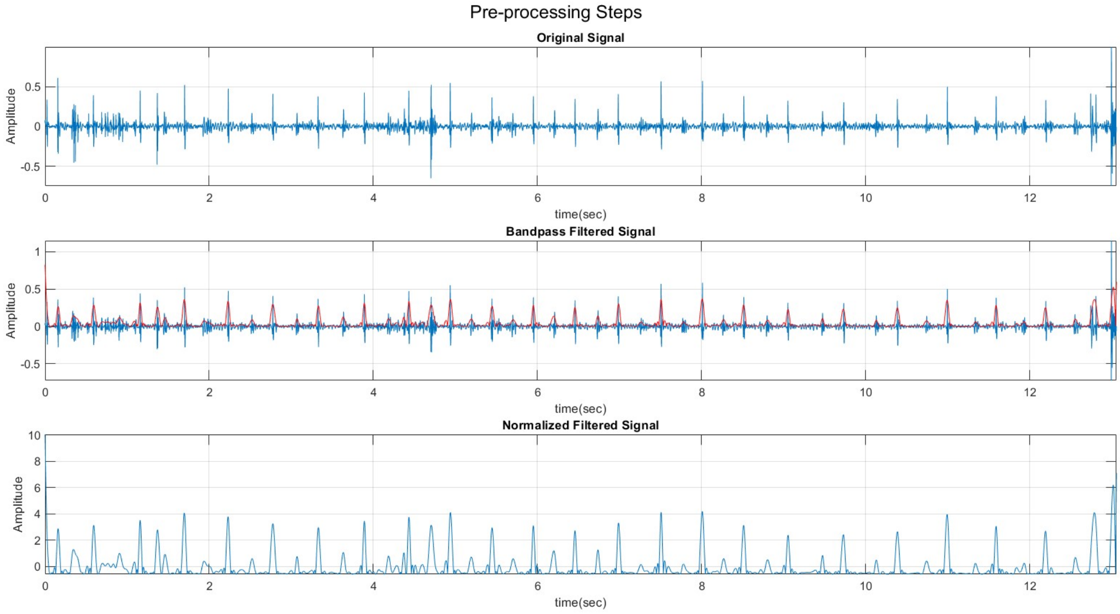

Figure 1: Preprocessing Steps

conversion from frequency to mel scale is given by the following equation:

$$mel(f) = 2595 \cdot log(1 + \frac{f}{700})$$

The Empirical Mode Decomposition method was developed so that data can be examined in an adaptive time–frequency–amplitude space for nonlinear and non-stationary signals.

F. Classification

This is a binary classification task: we have to decide about a signal, whether it is normal or not. To perform the prediction we labeled normal signals as '0' and murmurs as '1'. The sufficient model has to be used on the dataset to decide, based on the extracted features, if the signal is a normal or pathological one. In our case this network was chosen to be the linear SVM with 16 features.

Support Vector Machines are designed to find a hyperplane in the N-dimensional (N refers to the number of features, in our case it is 16) space that classifies the data points. The optimal solution is the one, when the maximal margin (the distance of the nearest element to the hyperplane) is the same for both classes. Support vectors in this algorithm are the data points that help adjusting the margin. To teach the network, whether it has done the classification well, it is penalised with loss, when makes a bad decision, but updated only with the regularisation parameter when the decision is good. This way the network will learn how to classify each data point based on its features [9].

We also used a 5-fold cross-validation before training the model to ensure lesser chances of overfitting.

III. Structure of the Project

The main and most important script that has to be run is called `runner.m`. This file will give the final scores that will help in the evaluation by calling many other functions, that has to downloaded in the same folder as the main script. To obtain the final results several processes are implemented in these helper functions, the steps and processes are listed and detailed in this section.

The first function called is the `run_for_folder.m`, that takes the file names (both .wav and .tsv) and runs the `project_run.m` function on all the read-in files. This is the main function of this task, since this is the one that sends the signals for further processing.

This separate function, called `processing.m`, takes the sampling frequency beside the signal as input, and the processing steps are carried out on the entire dataset. To make easier all the following steps and to avoid imprecisions the original signal is transposed. On this transposed signal the bandpass filtering is performed and then its wavelet entropy (which will be handled as a feature) calculated. Wavelet entropy (WE) is often used to analyze non-stationary signals. WE combines a wavelet or wavelet decomposition with a measure of order within the wavelet coefficients by scale. These measures of order are referred to as entropy measures. WE treats the normalized wavelet coefficients as an empirical probability distribution and calculates its entropy. This matlab function returns the Shannon wavelet entropy of the input [6].

Then the Shannon Energy envelope is calculated and then, normalised.

In the next step the peaks are searched on the Shannon Envelope is used to find the peaks, which will be considered as possible locations of heart sounds (S_loc). These peaks provide sufficient information to estimated the heart rate.

With the help of the peak's location we applied a binary mask on the signal so we could segment our data into systolic and diastolic phases.

After this step we transformed our signal with a simple FFT into the frequency domain. The first feature obtained

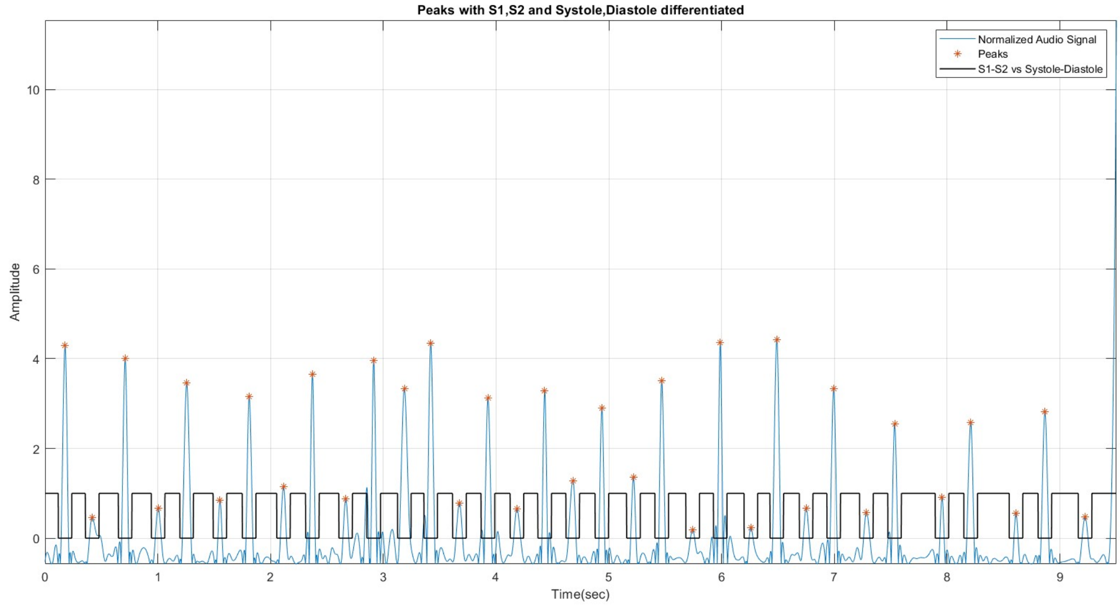

Figure 2: Peak Detection and Binary Mask of S1-S2 and Systole-Diastole Region difference

from this domain is the maximum frequency peak and then the maximum of the power spectral density as high_pwelch. On the normalised signal STFT was performed, as well.

To get a more extensive pool of features, wavelet transformation was applied on the signals. We created a five-feature wavelet decomposition, using the order 2 Daubechies wavelets. As this was the last step in feature extraction, all of them are collected into a table.

The output of this function is the locations of the detected heart sounds, the binary mask, the heart rate, the peaks and the feature table.

Another function calls `processing.m`, as well. This is the `pcg_plot.m` function, where the location of the heart sounds, the binary mask and the peaks are needed to provide a proper plot, which is needed for the evaluation.

As the `project_run.m` script finished with the `processing.m` function, it stores the results. The location of the heart sounds had to be multiplied with the sampling frequency to match with the calculations. The output of the `project_run.m` will be a structure, called properties, that will include the heart sound locations, the heart rate, the systolic and diastolic regions (the binary mask) and the decision, whether a signal is a normal or a pathological one.

Until this point the prediction structure contains only the heart sound locations, the heart rate and the binary mask. To upload the decisions, as well, the last and most important step of the processing has to be performed, the prediction.

For this part the built-in linear Support Vector Machine network model was used that handles 7 features from the feature table. With regularisation (when a decision is correctly made) and cost function (to penalise bad decision) the network learns how to decide on each data. These decisions are stored in the properties structure along with the other, previously obtained results.

IV. Results

We were given the ground truths for Heart Rate, the correct measures of hits and misses of the heart sound locations and the correct segmentation of the binary mask. We achieved an average hits of 84.0756%, an average heart rate difference of 12.1022% and an average segmentation difference of 14.0698%.

For the classification with 100 sets of data with 50 each for normal and murmur, we achieved a validation accuracy of 76% using Linear SVM. The confusion matrix is given by Table 1.

True class \ Predicted class	murmur	normal
murmur	28	22
normal	2	48

Table I: Confusion Matrix - Linear SVM

Calculating from the confusion matrix, we could get a 96% correct detection of normal signals and 56% correct detection of murmur signals. Figure 1 shows the RoC Curve for the Linear SVM classifier.

V. Conclusion

The main goal of the project was to find out the location of the peaks, the heart rate, differentiating the various heart sound regions and classifying each heart signal as a normal or a murmur signal. The high difference of our results from the expected values is largely due to the end-point noises that exist on almost all of the signals. The end-point noises are characterised by a lot of non-heart sound labelling at the beginning and at the end of the signal. We used 16 features for classification and that includes statistical, time domain, frequency domain and time-frequency domain features. Classification accuracy can be increased by extracting better features, particularly for murmur signals.

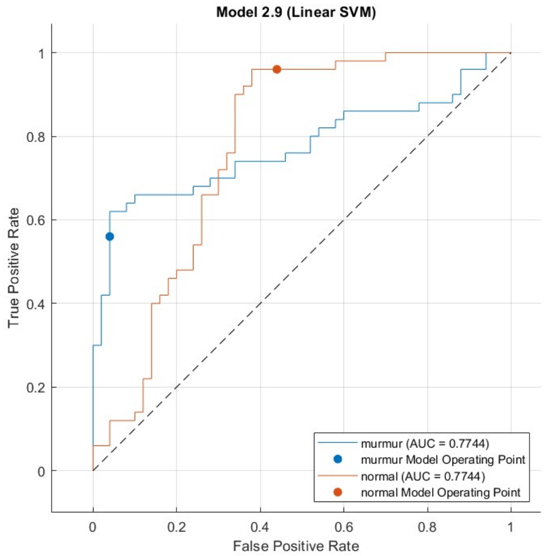

Figure 3: RoC Curve

REFERENCES

[1] Reyna, M. A., Kiarashi, Y., Elola, A., Oliveira, J., Renna, F., Gu, A., Perez-Alday, E. A., Sadr, N., Sharma, A., Mattos, S., Coimbra, M. T., Sameni, R., Rad, A. B., Clifford, G. D. (2022). Heart murmur detection from phonocardiogram recordings: The George B. Moody PhysioNet Challenge 2022. medRxiv, doi: https://doi.org/10.1101/2022.08.11.22278688

[2] Biomedical Signal Processing (P-ITJEL-0024) - 2nd Lecture at PPCU-FIT

[3] Oliveira, J., Renna, F., Costa, P. D., Nogueira, M., Oliveira, C., Ferreira, C., … & Coimbra, M. T. (2022). The CirCor DigiScope Dataset: From Murmur Detection to Murmur Classification. IEEE Journal of Biomedical and Health Informatics, doi: https://ieeexplore.ieee.org/document/9658215

[4] Ismail, S., Siddiqi, I. & Akram, U. Localization and classification of heart beats in phonocardiography signals —a comprehensive review. EURASIP J. Adv. Signal Process. 2018, 26 (2018). doi: https://doi.org/10.1186/s13634-018-0545-9

[5] Ali, Hussnain & Ahmad, Talha & Khan, Shoab. (2007). Heart Sound Signal Modeling and Segmentation based on Improved Shannon Energy Envelogram using Adaptive Windows. https://www.researchgate.net/profile/Hussnain-Ali-2/publication/262487927_Heart_Sound_Signal_Modeling_and_Segmentation_based_on_Improved_Shannon_Energy_Envelogram_using_Adaptive_Windows/links/00b49537d45234099b000000/Heart-Sound-Signal-Modeling-and-Segmentation-based-on-Improved-Shannon-Energy-Envelogram-using-Adaptive-Windows.pdf

[6] MATLAB documentation on Wavelet Entropy https://uk.mathworks.com/help/wavelet/ref/wentropy.html#d124e137143

[7] MATLAB documentation on finding local maxima https://uk.mathworks.com/help/signal/ref/findpeaks.html?searchHighlight=findpeaks&s_tid=srchtitle_findpeaks_1

[8] Li, F., Liu, M., Zhao, Y. et al. Feature extraction and classification of heart sound using 1D convolutional neural networks. EURASIP J. Adv. Signal Process. 2019, 59 (2019). https://doi.org/10.1186/s13634-019-0651-3

[9] Tang, H., Dai, Z., Jiang, Y. et al. PCG Classification Using Multidomain Features and SVM Classifier. Pattern Recognition in Medical Decision Support, 2018. https://doi.org/10.1155/2018/4205027

[10] M. Á. Goda and P. Hajas, "Morphological determination of pathological PCG signals by time and frequency domain analysis," 2016 Computing in Cardiology Conference (CinC), 2016, pp. 1133-1136.