Developing a Platform for the Automatic Recording and Denoising of Multichannel PCG Data

Emmanuel Zerefa, Wanli Lu, Jiayi Gao, Lester Liu
Rice University

***Abstract*—Heart disease is the leading cause of death worldwide [1]. The early detection of heart diseases is thus a significant problem, yet traditional methods to do so face several limitations as they're generally time consuming, expensive, and rely on specialized labor or machinery. Our solution is to develop a handheld device, capable of performing auscultation on all four major heart valves in a matter of seconds, that can detect possible abnormalities in the heart's structure. Our work to date on this project involves a low-cost FPGA-based solution for converting Pulse Density Modulation (PDM) signals from MEMS microphones to Pulse Code Modulation (PCM), offering 40% cost reduction compared to high-end FPGAs and power consumption $<$ 0.1 W. We've also implemented several software-based methods, using wavelet transforms and neural networks, to automatically denoise data, with current average noise reduction of 46.5% based on before-after RMSE. This paper will summarize these recent developments.**

I. Introduction

PDM is commonly used in MEMS microphones, while PCM is the standard for digital audio processing. Existing PDM-to-PCM converters often rely on costly proprietary hardware. We are developing a modular, cost-effective conversion pipeline using open-source tools and custom IP blocks. Key innovations include a 1-to-2-bit converter for CIC compatibility and a low-latency CIC-FIR chain, optimized for real-time audio processing with minimal latency and power consumption.

On the software side, we've been developing techniques to denoise PCG recordings and measure signal quality, for final use in making the auscultation device more robust to real-world conditions. Our denoising pipeline makes use of neural networks, specifically the U-Net architecture for its proven effectiveness on denoising tasks.

II. Methods

A. Software - Neural Network Denoising Pipeline

A major component of the overall project was developing a pipeline written in PyTorch to automatically train neural networks to denoise heart sounds. This section will elaborate on the design of the six parts of this pipeline.

1) Data Loading

Part one of the pipeline involves locating and loading in datasets containing clean heart recordings and noises that commonly occur in hospitals. The following datasets were ultimately used:

- As a source of clean PCG recordings: PASCAL Challenge [2], Physionet 2016 [3] [4].
- As a source of common hospital noises: ESC-50 [5], HAN (Hospital Ambient Noise) [6], ARCA23K [7].

2) Normalization

Such that the various sources of PCG data and noise sounds can be standardized and combined, the pipeline contains a section devoted to normalization. This section contains functionality that is able to filter signals, downsample signals, combine two audio files at a specific SNR, and synthetically match the length of two signals. These operations are performed using the Python libraries Librosa, SciPy and NumPy.

3) Dataset Creation

The next step of the pipeline is to create training, validation and test data for use in the model. The pipeline performs the following process. Firstly, a total of 2919 PCGs (266 from PASCAL, and 2653 from Physionet 2016) are partitioned into training, validation, and test arrays. The ratio used is 70/15/15. For each PCG in each of those arrays, 8 random noises were selected (3 from ESC-50, 2 from ARCA-23K, 3 from HAN). At a variety of SNRs, each noise was then separately combined with a copy of the original PCG data. In this way the noisy inputs to the denoising model were made.

4) Model Definition

The pipeline includes a section where models can be defined and implemented. Two models are currently defined here, both one-dimensional variants of the U-Net [9].

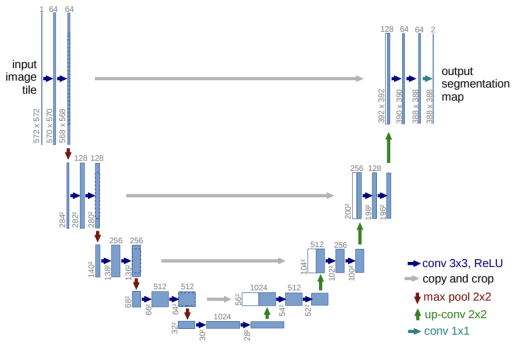

Fig. 1. U-net visual

The best performing U-Net variant ultimately had 8.05 million parameters, 18 convolutional layers, batch normalization after almost all layers, and GELU for nearly all activation functions.

5) **Model Training**

The pipeline includes a section where models are trained. Users are able to define a maximum number of epochs to train for, and the tolerances of various overfitting-prevention mechanisms. The model training functionality also measures and stores relevant information such as model loss over time and real-world time needed to train the model. This step also stores the best performing model to disk.

6) **Metrics and Visualizations**

The final step of the pipeline is to perform metrics and visualizations of the model's performance. For quantitative metrics, the primary tool is comparison of RMSE before and after model training. For visualization, PCG recordings of both normal and murmured heartbeats are plotted both after noise is added and after denoising occurs. An example:

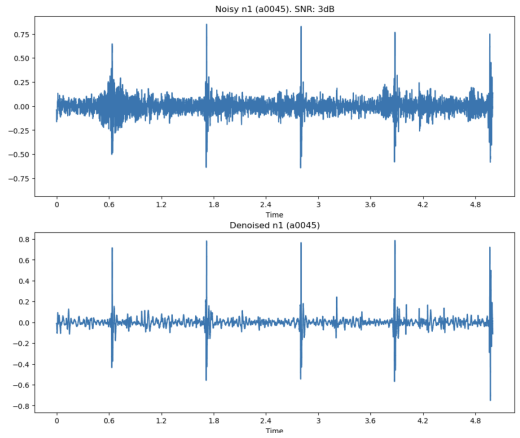

Fig. 2. Example of a noisy heartbeat before and after denoising

B. Software - Signal Quality Assessment

This section covers several machine learning and data science techniques to develop an automated, real-time method for PCG signal quality assessment.

1) **Signal quality assessment using classification**

The signal quality assessment pipeline for PCG signals is designed to automatically and efficiently distinguish clean heart sound recordings from noisy ones, even in real-time environments. To ensure consistency, all PCG signals are first down-sampled to 1000 Hz. Preprocessing includes the removal of spikes and baseline wandering, followed by normalization to zero mean and unit variance. The system then extracts a set of ten key features that capture various signal characteristics. These include the kurtosis of the heart sound signal, energy ratios across different frequency bands, and envelope-based features. Additionally, the pipeline calculates the autocorrelation of the envelope to assess signal consistency over time and evaluates the overall degree of periodicity in the signal. Together, these features provide a comprehensive basis for evaluating signal quality [10]. Distribution histograms for each feature across different classes (Fig. 3) were plotted to identify the feature most strongly associated with signal quality among the ten extracted features.

The data were categorized into two groups: samples labeled as 4 and 5 were considered acceptable, while those labeled as 1, 2, and 3 were deemed unacceptable. After performing feature extraction on the raw data, the extracted features were divided into a training set and a testing set (10%). The training set was then used to train an SVM classifier.

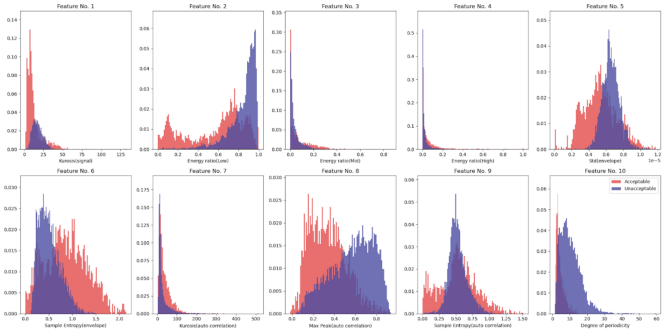

Fig. 3. Feature Occurrence

2) **Signal Quality Assessment using Quality Index**

This method introduces an automated, real-time approach for assessing the quality of PCG signals by computing a quality score that quantifies the signal's degree of periodicity. The evaluation begins by analyzing heart sound segments containing 4 to 10 continuous cardiac cycles. A time-varying autocorrelation function, $R(t, \tau)$, is computed to capture the signal's temporal structure. From this, the Fourier series coefficients $R(\alpha, \tau)$ are derived, followed by the computation of the cyclic spectral density $S(\alpha, f)$ via Fourier transform. The cycle frequency spectral density $\gamma(\alpha)$ is then obtained by integrating the cyclic spectral density. Finally, the quality score is determined as the ratio of the peak of $\gamma(\alpha)$ to its total integral, reflecting how strongly periodic the signal is [11].

C. Hardware - DDR3 Driver

The DDR3 driver, designated as `mig_7series_0`, must be configured accordingly. Upon receiving the reset and clock signals from the input board, this driver is capable of generating a reset signal and two clock signals to facilitate subsequent operations. Moreover, this intellectual property (IP) core grants access to the DDR3 memory.

D. Clocking Wizard

We connect the clock signals generated by the DDR3 driver to the Clocking Wizard IP to create two new clock frequencies, specifically 100 MHz and 24 MHz, which will be employed by the MicroBlaze soft processor and the CIC_compiler IP.

Fig. 4. Clocking Wizard IP block.

E. MicroBlaze

The MicroBlaze-based system includes the MicroBlaze processor, AXI Interrupt Controller, and the MicroBlaze Debug Module. This configuration facilitates efficient interrupt handling and debugging for FPGA-based systems. The AXI Interrupt Controller is responsible for managing interrupt signals from peripherals, while the MicroBlaze Debug Module (MDM) provides debugging capabilities and facilitates system resets.

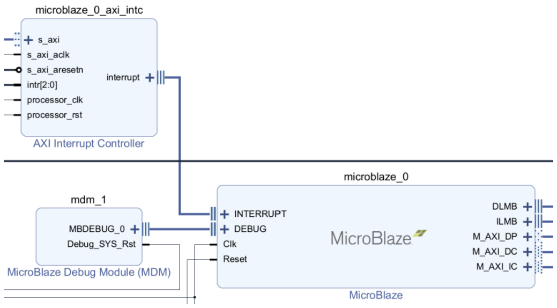

Fig. 5. The IP blocks: Microblaze, Interrupt Controller, and Debug Module.

F. PDM_MIC Driver and CIC Compiler

By configuring the user-defined PDM_MIC_DRIVER (Fig. 4, left) and the Cascaded Integrator Comb (CIC) Compiler (Fig. 4, right), we generate a 2.4 MHz signal to drive the PDM microphone. This allows the conversion of 2-bit PDM data into 8-bit data, and subsequently transforms the PDM signals into PCM signals using the CIC Compiler.

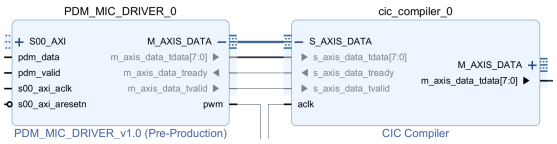

Fig. 6. The IP blocks: Custom PDM_MIC Driver and CIC Compiler.

G. Peripherals

We configure essential peripherals, including the timer and UART, to enable user interaction with the device.

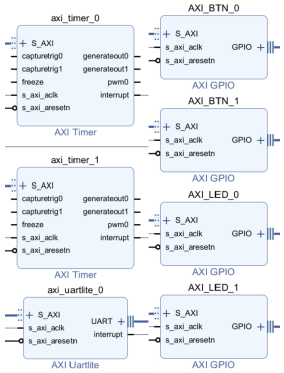

Fig. 7. Peripheral device.

H. Hardware platform

This represents the overall presentation of the final hardware platform.

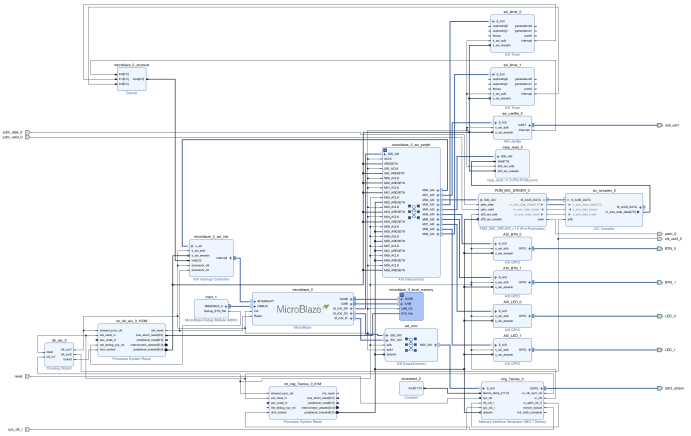

Fig. 8. Vivado hardware platform.

I. Software Function Flowchart

The software flowchart outlines a basic function implemented on the Vivado hardware platform. Users can switch between functions using Button1 and Button2, enabling operations such as recording PCM data and uploading it from DDR3 memory to a PC via UART transmission.

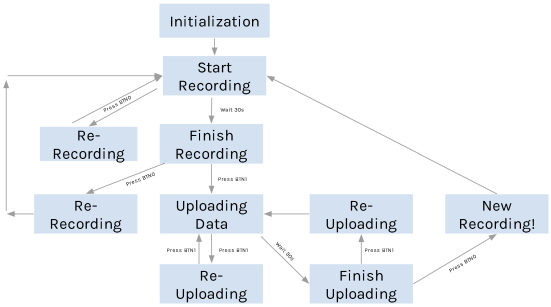

Fig. 9. Software Function Flowchart.

J. Hardware - Pipeline simulation and filter design

The PDM-to-PCM pipeline includes digital signal processing filters. Modeling and simulating this pipeline is crucial for optimizing filter configurations to convert the oversampled PDM signal into high-quality PCM output. This process ensures optimal frequency response, noise shaping, and aliasing suppression while preserving signal integrity. The pipeline system diagram is shown below.

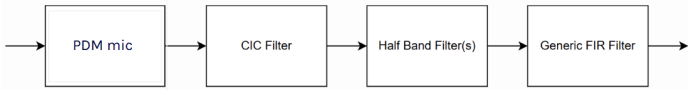

Fig. 10. Pipeline System Diagram

In order to simulate a system, first we need to define the design goals of the pipeline system, and here are the design goals retrieved from data sheet.

- 48kHz Output sample rate: 48kHz is universally supported and probably the most common sample rate for high quality audio.
- 2.304 MHz PDM sample rate: We expect to use half-band filter to decrease decimation ratio. The recommended frequency is 2.4MHz, but 2400/48=50, which is only divisable by 2. So let's choose 2.304MHz: 2304/48=48.
- 0 - 6kHz passband: The frequency response of the microphone is only flat between 100Hz and 5kHz. [21]
- 10kHz start of stop band: The specification of the microphone doesn't say anything about behavior above 10kHz.
- 87dB SNR of the output signal: The dynamic range of the microphone is the SNR plus the difference between the AOP and 94dbSPL.
- 0.1dB maximum passband ripple at the output: A typical requirement for high quality audio.

1) **PDM Microphone**

We use the MP34DT01 PDM MEMS Microphone, but its datasheet lacks detailed structural information. Based on similar products, we model the PDM mic as a sigma-delta modulator, with performance resembling a 4th-order sigma-delta modulator.

2) **CIC filters**

A Cascaded Integrator-Comb (CIC) filter is an efficient, multiplier-free digital filter used for decimation and interpolation, especially in PDM to PCM conversion. The impulse response of a CIC filter follows a binomial coefficient pattern and can be expressed as:

$$h[n] = \sum_{k=0}^{M-1} (-1)^k \binom{M}{k} \delta[n - kR]$$

With Simulink DSP toolbox, we can configure and simulate CIC filter. The simulation result is shown below.

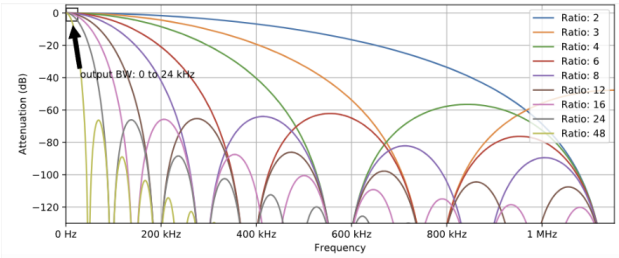

Fig. 11. CIC Frequency Spectrum

The pass band ripple gets progressively worse with increasing frequency and with an increasing number of cascade stages. And Anti-aliasing performance is also required as specified by the stop band attenuation. By combining both requirements, here are 3 best solutions that has the lowest number of CIC stages and highest decimation ratio.

- 6x decimation ratio with 3 stages
- 8x decimation ratio with 4 stages
- 12x decimation ratio with 4 stages

3) **FIR filters**

The half band filter is a simple low-pass FIR filter. In a PDM to PCM pipeline, a half-band filter is beneficial in reducing the oversampling rate while efficiently suppressing out-of-band noise. Resource usage is also reduced. The following figure shows the multiplication resources needed for different options. The power of half-band filter could also be observed.

CIC Config	HB1 mul/s	HB2 mul/s	HB3 mul/s	Decim FIR mul/s	Final FIR mul/s	Total mul/s
Decim:6 Stages:3	4 x 192k = 768k	6 x 96k = 576k	10 x 48k = 480k		47 x 48k = 2256k	4080k
Decim:8 Stages:4	6 x 144k = 864k			21 x 48k = 1008k	51 x 48k = 2448k	4320k
Decim:8 Stages:4	6 x 144k = 864k				141 x 48k = 6768k	7632k
Decim:12 Stages:4	6 x 96k = 576k	10 x 48k = 480k			51 x 48k = 2448k	3504k

Fig. 12. Total multiplication resources of each filter design solution

The solution with 12 stages has the least multiplication number per second. It is indeed the optimal one. Once the properties of the CIC filter has been decided, we can simply fill in the numbers and run the Parks-McClellan/Remez algorithm to come up with the exact filter coefficients of the half-band and generic FIR filter.

III. Results

A. Software - Neural Network Denoising Pipeline

The U-Net architecture with 18 convolutional layers currently achieves the best performance across all denoising models, with an average noise reduction of 46.5%, measured by the RMSE drop between noisy and denoised recordings. The neural network effectively distinguishes murmurs from noise, as the murmur detection rate in a downstream binary classifier (normal vs. murmur) remains unchanged for all types of murmurs after denoising heartbeats.

B. Software - Signal Quality Assessment

1) Dataset

There are 7893 recordings in the dataset, all of them are down sampled to 1000 Hz sample rate, and all of them has a duration longer than 6 seconds. Each recording is annotated with a quality label of 1-5. In the dataset, 319 recordings are "very bad," 2187 are "bad," 1880 are borderline quality, 1950 are "good," and 1557 are "excellent."

2) Results of Classification method

The final trained SVM binary classifier achieved an accuracy of 92.6% on the training set and 92.8% on the testing set. The classification's precision was 87.3%, and its sensitivity was 85.2%. The confusion matrix of the classification results is shown in the Fig.16 below.

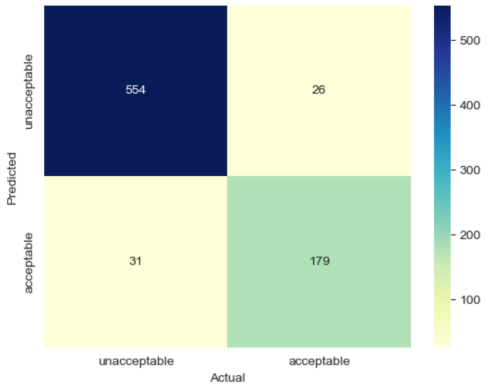

Fig. 13. Confusion Matrix of Binary Signal Quality Classification

3) Results of Quality Index method

The performance of this method is analyzed with a 25 second heart sound recording in which 2 segments are contaminated with simulated noise. The SNR was set to 5 dB. The quality score is evaluated with a sliding window covering 2 s moves in steps of 0.1 s, the time-varying quality score and the contaminated signal are shown below.

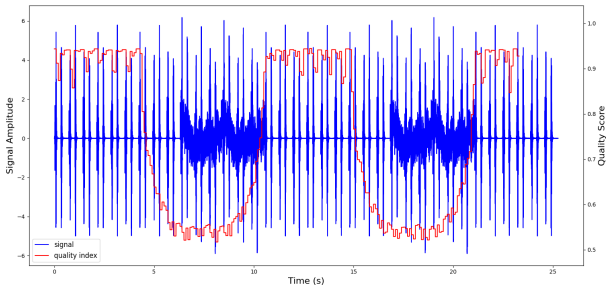

Fig. 14. Quality Score over PCG Signal with simulated noisy segments

C. Hardware - Test results based on the ARTY-S7-50

Upon completing the described operations, a .wav file containing the recorded PCM data is generated. This file can be transferred to a PC and opened in audio editing software like Audacity, where users can play, analyze, and manipulate the audio for signal processing or quality assessment.

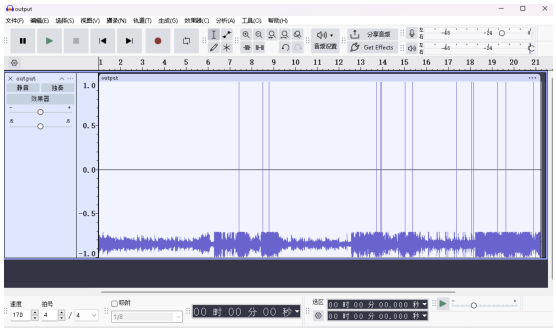

Fig. 15. A .wav file generated by converting PCM data.

D. Hardware - Pipeline simulation and filter design

By choosing 12 stages CIC filter and running the Parks-McClellan/Remez algorithm in python, we come up with the exact filter coefficients of the half-band and generic FIR filter and finish the pipeline design. The magnitude frequency plots of all the filter components are shown in the following figure.

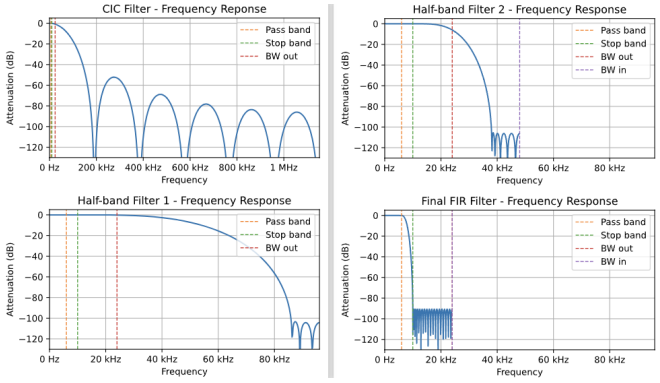

Fig. 16. Frequency response of all the stages

IV. Discussion

The current design offers cost-efficient PDM-to-PCM conversion but lacks full integration. The absence of FIR filtering affects signal fidelity, and the UART interface limits real-time throughput. While more customizable than commercial ICs like the AD1938, it requires further optimization, especially for power-sensitive applications. Future improvements, such as DDR memory integration and FreeRTOS, will enhance scalability and real-time performance.

Concerning denoising, the best performing model so far currently eliminates 46.5% of noise on average, as measured by RMSE reduction from before to after denoising. This figure will likely improve, as more state-of-the-art architectures such as transformers haven't been deployed and trained in the pipeline yet. However there's likely a fundamental limit to denoising performance, since the primary dataset for the denoising model, Physionet 2016, isn't exceptionally clean, containing many recordings collected in open air environments.

For the signal quality assessment part, the binary classification has achieved a relatively high accuracy and the classifier has the ability to identify noisy signals from clean ones. The quality index method can detect the changes in signal quality with a short response time. The future work of this part should include trying different classifiers and testing these pipelines on other datasets.

Acknowledgements

This project leverages the open-source development platforms Vivado and Vitis provided by AMD, in conjunction with the MEMS design kit from STMicroelectronics.

REFERENCES

[1] World Health Organization: WHO, "Cardiovascular diseases," Jun. 11, 2019. https://www.who.int/health-topics/cardiovascular-diseases

[2] Bentley, P.; Nordehn, G.; Coimbra, M.; Mannor, S.; Getz, R. Classifying Heart Sounds Callenge [online]. Available online: http://www.peterjbentley.com/heartchallenge/

[3] Liu C, Springer D, Li Q, Moody B, Juan RA, Chorro FJ, Castells F, Roig JM, Silva I, Johnson AE, Syed Z, Schmidt SE, Papadaniil CD, Hadjileontiadis L, Naseri H, Moukadem A, Dieterlen A, Brandt C, Tang H, Samieinasab M, Samieinasab MR, Sameni R, Mark RG, Clifford GD. An open access database for the evaluation of heart sound algorithms. Physiol Meas.2016 Dec;37(12):2181-2213

[4] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

[5] K. J. Piczak. ESC: Dataset for Environmental Sound Classification. Proceedings of the 23rd Annual ACM Conference on Multimedia, Brisbane, Australia, 2015.
[DOI: http://dx.doi.org/10.1145/2733373.2806390]

[6] Ali, S. N., Shuvo, S. B., Al-Manzo, M. I. S., Hasan, A., & Hasan, T. (2023). An end-to-end deep learning framework for real-time denoising of heart sounds for cardiac disease detection in unseen noise. IEEE Access.

[7] T. Iqbal, Y. Cao, A. Bailey, M. D. Plumbley, and W. Wang, "ARCA23K: An audio dataset for investigating open-set label noise," arXiv (Cornell University), Jan. 2021, doi: 10.48550/arxiv.2109.09227.

[8] C. Liu et al., "An open access database for the evaluation of heart sound algorithms," Physiological Measurement, vol. 37, no. 12, pp. 2181–2213, Nov. 2016, doi: 10.1088/0967-3334/37/12/2181.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-NET: Convolutional Networks for Biomedical Image Segmentation," arXiv (Cornell University), Jan. 2015, doi: 10.48550/arxiv.1505.04597.

[10] H. Tang, M. Wang, Y. Hu, B. Guo, and T. Li, "Automated signal quality assessment for heart sound signal by novel features and evaluation in open public datasets," *BioMed Research International*, vol. 2021, no. 1, p. 7565398, 2021. [Online]. Available: https://doi.org/10.1155/2021/7565398

[11] T. Li, T. Qiu, and H. Tang, "Optimum heart sound signal selection based on the cyclostationary property," *Computers in Biology and Medicine*, vol. 43, no. 6, pp. 607–612, Jul. 2013. [Online]. Available: https://doi.org/10.1016/j.compbiomed.2013.03.002

[12] S. Mallat, *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, no. 7, pp. 674–693, July 1989.

[13] S. R. Messer, J. Agzarian, D. Abbott, *Optimal wavelet denoising for phonocardiograms*, Microelectron. J., vol. 32, pp. 931–941, 2001.

[14] J. Singh, R. S. Anand, "Computer aided analysis of phonocardiogram," *J. Med. Eng. Technol.*, vol. 31, no. 5, pp. 319–323, 2007.

[15] G. Luo and D. Zhang, "Wavelet Denoising," in *Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology*, D. Baleanu, Ed. InTechOpen, 2012.

[16] Oliveira, J., Renna, F., Costa, P., Nogueira, M., Oliveira, A. C., Elola, A., Ferreira, C., Jorge, A., Bahrami Rad, A., Reyna, M., Sameni, R., Clifford, G., & Coimbra, M. (2022). The CirCor DigiScope Phonocardiogram Dataset (version 1.0.3). PhysioNet. https://doi.org/10.13026/tshs-mw03.

[17] STMicroelectronics, "MP54DT05-A: Omnidirectional digital microphone for high-performance applications," *Datasheet*, Rev. 5, Jan. 2021. [Online]. Available: https://www.st.com/resource/en/datasheet/mp54dt05-a.pdf

[18] Xilinx, "CIC Compiler LogiCORE IP Product Guide (v4.0)," *PG140*, Dec. 2021. [Online]. Available: https://docs.xilinx.com/r/en-US/pg140-cic-compiler

[19] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, NJ: Pearson, 2007.

[20] Y. Liu, J. Lanier, K. Fu, and A. Sample, "PDM-to-PCM microphone signal conversion on FPGA using CIC and FIR filters," Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep., 2023. [Online]. Available: https://yatian-liu.github.io/public/PDM_PCM_signal_conversion_FPGA.pdf [Accessed: Oct. 1, 2023].

[21] STMicroelectronics, "MEMS audio sensor omnidirectional digital microphone," MP34DT01-M datasheet, Sep. 2014 [Revised Sept. 2002].Available: https://www.st.com/resource/en/datasheet/mp34dt01-m.pdf