Universidad Autónoma de Baja California
Facultad de Ciencias
Maestría y Doctorado en Ciencias e Ingeniería

Técnicas de eliminación de ruido real en fonocardiogramas mediante U-Nets y representaciones en tiempo-frecuencia

TESIS

que para cubrir parcialmente los requisitos para obtener el grado de

MAESTRO EN CIENCIAS

presenta

Cristóbal González Rodríguez

Directora de tesis:

Dra. Eloísa del Carmen García Canseco

Co–director de tesis:
Dr. Miguel Ángel Alonso Arévalo

Ensenada, B. C., marzo 2025.

Universidad Autónoma de Baja California

Facultad de Ciencias

Técnicas de eliminación de ruido real en fonocardiogramas mediante U-Nets y representaciones en tiempo-frecuencia

TESIS DE MAESTRÍA EN CIENCIAS

QUE PRESENTA

Cristóbal González–Rodríguez

APROBADO POR:

Dra. Eloísa del Carmen García Canseco

Directora de tesis

Dr. Miguel Ángel Alonso Arévalo

Co-director de tesis

Dr. Everardo Inzunza González

Sinodal

Dr. Enrique Efrén García Guerrero

Sinodal

Ensenada B.C. *Marzo, 2025*

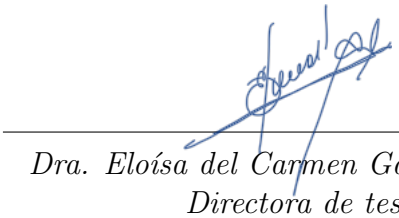
RESUMEN de la Tesis de Cristóbal González Rodríguez, presentada para obtener el grado de Maestro en Ciencias. Ensenada, Baja California, México, marzo del 2025.

Técnicas de eliminación de ruido real en fonocardiogramas mediante U-Nets y representaciones en tiempo-frecuencia

Las enfermedades cardiovasculares siguen siendo la principal causa de muerte en todo el mundo. El análisis automático del sonido cardíaco es una herramienta de diagnóstico económica y muy prometedora, pero su exactitud puede verse dificultada por el ruido durante las grabaciones. Las técnicas actuales de separación ciega de fuentes se basan en modelos de aprendizaje profundo, en particular arquitecturas U-Net, junto con la transformada de Fourier de tiempo corto (STFT) para la representación en tiempo-frecuencia de la señal de audio. Sin embargo, las representaciones alternativas de tiempo-frecuencia permanecen inexploradas. Esta tesis propone un nuevo método de eliminación de ruido de fonocardiogramas, en el que la STFT, la transformada de ondeleta continua (CWT), la transformada de ondeleta sincronizada y la transformada S, en combinación con una arquitectura U-Net, mejoran la eliminación de ruido de las señales de fonocardiograma. La aplicación de estas técnicas plantea diversos retos, como la búsqueda de bases de datos de fonocardiogramas adecuadas y de señales de ruido similares a las del mundo real para simular entornos de ruido realistas, todos los cuales se abordan aquí. El algoritmo de eliminación de ruido se evaluó en señales de sonido cardiaco contaminadas con cuatro tipos de ruido, incluidas perturbaciones no estacionarias del mundo real, con relaciones señal-ruido (SNR) de -5 dB y 0 dB. Los resultados y análisis presentados incluyen la introducción de una métrica de eliminación de ruido más estricta que la mayoría de las métricas utilizadas en los estudios de eliminación de ruido más avanzados, junto con un análisis de densidad espectral de potencia y una evaluación del rendimiento del algoritmo. Se demostró la eficacia del modelo en los análisis de dominio temporal, tiempo-frecuencia y densidad espectral de potencia, lo que establece un enfoque de eliminación de ruido que supera a los métodos actuales más avanzados.

Palabras clave: Representación tiempo-frecuencia, fonocardiograma, ondeleta, eliminación de ruido, red neuronal, U-Net.

Resumen aprobado por:



Dra. Eloísa del Carmen García Canseco
Directora de tesis

Dr. Miguel Ángel Alonso Arévalo
Co-director de tesis

ABSTRACT of the thesis by Cristóbal González Rodríguez, presented to obtain the Master of Science degree. Ensenada, Baja California, México. March 2025.

Phonocardiogram denoising techniques using U-Nets and time-frequency representations

Cardiovascular diseases remain the leading cause of death globally. Automatic heart sound analysis provides an economical and highly promising diagnostic tool, yet its accuracy can be hindered by noise during recordings. Current state-of-the-art blind source separation techniques often rely on deep learning models, particularly U-Net architectures, in conjunction with short-time Fourier transform (STFT) for audio signal time-frequency representation. However, alternative time-frequency representations remain unexplored. This thesis proposes a novel method of phonocardiogram denoising, in which the STFT, continuous wavelet transform (CWT), wavelet synchrosqueezed transform, and S-transform, in pair with a U-Net architecture, enhances the denoising of phonocardiogram signals. The implementation of these techniques involves different challenges, including the search for appropriate phonocardiogram databases and real-world resembling noise signals to simulate realistic noise environments, all of which are tackled here. The denoising algorithm was evaluated on heart sound signals mixed with four types of noise, including non-stationary real-world disturbances, at signal-to-noise ratios (SNRs) of -5 dB and 0 dB. The results and analysis presented involve the introduction of a stricter denoising metric than most common metrics used in state-of-the-art denoising research, paired with a power spectral density analysis, and a performance evaluation of the algorithm.

Keywords: Time-frequency representation, phonocardiogram, wavelet, denoising, neural network, U-Net.

Abstract approved by:

Dr. Eloísa del Carmen García Canseco
Thesis Director

Dr. Miguel Ángel Alonso Arévalo
Thesis Co-director

Agradecimientos

Quisiera expresar mi gratitud a la Universidad Autónoma de Baja California por proporcionar el ambiente académico y los recursos necesarios para la realización de esta investigación. Un agradecimiento especial al CONAHCYT por su generoso apoyo financiero, que hizo posible este trabajo.

Agradezco a mi directora de tesis, Eloísa del Carmen García Canseco, por su inquebrantable guía, perspicaz retroalimentación y constante aliento a lo largo de este viaje. También me gustaría dar las gracias a mi codirector, Miguel Ángel Alonso Arévalo, por sus inestimables consejos, experiencia y apoyo, que han contribuido significativamente al desarrollo de este trabajo.

Estoy en deuda con los miembros de mi comite de tesis, Everardo Inzuna González y Enrique Efrén García Guerrero, por su atenta evaluación, comentarios constructivos y contribuciones esenciales que ayudaron a refinar mi investigación.

A todos los mencionados, y a otros que me apoyaron en diversas capacidades durante este proyecto, extiendo mi más sincero agradecimiento y aprecio.

Contents

List of Figures

List of Tables

Nomenclature

Variables

α	Synthetic noise type
β	Morse Beta parameter
γ	Morse Gamma parameter
κ	Scaling parameter
μ	Morlet wavelet scale parameter
ω	Angular frequency
ω_l^k	Sinusoidal amplitude parameter
ω_s	Instantaneous frequency
ϕ	Phase
ϕ_l^k	Sinusoidal amplitude parameter
σ^2	Variance
τ	Time delay
a	Wavelet frequency scale
A_l^k	Sinusoidal amplitude parameter
b	Wavelet time shift
f	Frequency
f_k	Frequency bins

f_s	Sampling frequency
H	Hop length
K	Number of frequency frames
L	Number of sinusoidal components
L_w	Window length
M	Number of time frames
N	Number of time samples
N_v	Number of voices
N_x	Frame length
P	Signal's power
S	Number of samples
S_w	Number of windowed segments
T	Time duration
t	Time

Constants

C_α	Synthetic noise constant
c_μ	Morlet wavelet normalization constant

Fundamental Mathematical Constants

$\imath$	Imaginary unit
π	Pi
e	Euler's number

Number Sets

$\mathbb{C}$	Complex numbers
$\mathbb{R}$	Real numbers

$\mathbb{Z}$ Integer numbers

Functions and Tensors

$\eta(t)$ Noise signal

$\lambda(n)$ Time-varying gain envelope function

$\mathscr{F}(\omega, \tau)$ Time-frequency function

$\mathscr{I}$ Image matrix

$\mathscr{K}$ Kernel

$\psi(t)$ Wavelet

$\tilde{s}$ Denoised signal

ε Noise

$\varphi_W(\omega)$ Periodogram

$p(\eta)$ Noise probability density distribution

PSD_n Synthetic noise power spectral density

$s(t)$ Time signal

$s_k(b)$ Short-time signal frames

$v(t)$ Tapering window

w Window function

Time-Frequency Representations

$F_s(\omega, \tau)$ Continuous short-time Fourier transform

$S_s(f, \tau)$ *S*-transform

$T_s(\omega_{s,\ell}, b)$ Wavelet synchrosqueezed transform

$W_s(a, b)$ Wavelet transform

$X_s(k, m)$ Discrete short-time Fourier transform

Indexes

i Sum index

j Octave index

k index

n index

Operations

$\langle \mathfrak{f}, \mathfrak{g} \rangle$ Inner product between functions $\mathfrak{f}$ and $\mathfrak{g}$.

$\mathfrak{f} * \mathfrak{g}$ Convolution between $\mathfrak{f}$ and $\mathfrak{g}$

$\widehat{s}(\omega)$ Fourier transform of $s(t)$

r Pearson correlation coefficient

fit Fit coefficient

Other Symbols

$\mathbf{L^2}(\mathbb{R})$ Finite energy functions

Acronyms / Abbreviations

APGN Additive Pink Gaussian Noise

ARGN Additive Red Gaussian Noise

AS Aortic Stenosis

AWGN Additive White Gaussian Noise

CC Cardiac Cycle

CNN Convolutional Neural Network

CWT Continuous Wavelet Transform

DFT Discrete Fourier Transform

DL Deep Learning

DWT Discrete Wavelet Transform

ECG Electrocardiogram

EGM eGeneralMedical database

FNN Feedforward Neural Network

HLS Heart-Lung Sound

HSS Heart Sound Signal database

LHS Littman Heart Sound database

LMS Least Mean Square

LSTM Long Short-Term Memory network

MAE Mean Absolute Error

MHSM Michigan Heart Sound and Murmur database

MRI Magnetic Resonance Imaging

MR Mitral Regurgitation

MSE Mean Squared Error

MS Mitral Stenosis

MVP Mitral Valve Prolapse

NMSE Normalized Mean Squared Error

N Normal

NRMSE Normalized Root Mean Squared Error

PCG Phonocardiogram

PRD Percentage Root-mean-square Difference

PSNR Peak Signal-to-Noise Ratio

RMSE Root Mean Squared Error

SI-SDR Scale-Invariant Signal-to-Distortion Ratio

SNR Signal-to-Noise Ratio

SR Sample Rate

STFT Short-time Fourier transform

WSST Wavelet Synchrosqueezed Transform

Chapter 1

Introduction

1.1. Problem

In Mexico and worldwide, the main cause of death is cardiovascular diseases (INEGI, 2021; OECD, 2017; Ritchie et al., 2018; World Health Organization, 2021). A main factor that contributes to this is the inaccessibility, both for the diagnosis and treatment of some of these diseases. Currently there are different studies that carry out the objective of finding and identifying pathologies in ill patients, examples of which are echocardiogram, electrocardiogram (ECG), and magnetic resonance imaging (MRI), among others. These studies essentially consist in obtaining information about the heart activity of the patient; however, due to the time and machines that are required to conduct these studies, they can be very expensive. This results in patients without sufficient economic resources unable to be diagnosed and treated in time for the disease they suffer from, which in turn contributes to the problem initially mentioned. An alternative to these methods is cardiac auscultation, the main advantage of which is that it is a low-cost, non-invasive method. Cardiac auscultation is a physical examination method that can provide a reliable and rapid diagnosis when performed by a trained physician (Pahlm and Wagner, 2011). Heart sound signal analysis has gained interest due to recent advances in computers and the ever-increasing processing capacity and diminishing size of electronic devices. The primary objective of automated cardiac sound analysis is to accurately classify the presence or absence of pathological events in the cardiac cycle (Mahnke, 2009). Nowadays there are digital stethoscopes capable of recording cardiac audio to be analyzed in more detail by obtaining its phonocardiogram (PCG) (Tavel, 2006). A PCG is a graphic representation of the information provided by the heart sound (Chowdhury et al., 2020), however, the main disadvantages of cardiac auscultation are in the audio recording, since this type of diagnosis is usually performed in very noisy environments (Mallinson, 2018), and because the stethoscope can also record the audio produced by other

physiological processes, this results in obtaining very noisy PCGs, which greatly hinders diagnosis by the physician (Gradolewski and Redlarski, 2014).

A healthy PCG is typically conformed of two main heart sounds: S1 and S2. The first is generated from the closure of the atrioventricular valves, while the second is generated from the semilunar valves during a cardiac cycle (CC). With the Systole (contraction phase) for S1 and the Diastole (relaxation phase) for S2, the CC exhibits quasi-periodic activity. In healthy individuals, these noises are limited to the corresponding phases of the CC, resulting in two calm intervals: diastolic silence (s-Dia) and systolic silence (s-Sys). People with cardiac issues may typically exhibit additional noises, such as S3 and S4, along with murmurs, frictions, or clicks (Abbas and Bassam, 2022).

The task proposed in this thesis is to train a neural network model to identify a variety of cardiac sounds available in databases, not only including healthy cardiac sounds, but also pathological. This can be achieved with a similar approach taken in blind source separation problems (see for instance the works of Andreas et al. (2017) and Hennequin et al. (2020)). A U-Net is used to breakdown an audio signal, which is then reconstructed into a mask that highlights the components of the audio that are to be separated; in this case, the signal is a PCG that is separated from the noise.

1.2. Justification

The primary goal of this work is to clean the PCGs of noise in the recordings without affecting the part of the signal that contains the information of the cardiac pathologies, thereby opening up new avenues for diagnosing cardiac diseases, not only with better audio quality in the PCGs but also in a timely and cost-effective manner. This would increase opportunities for the low-income population.

The expected outcome is an adapted model of an artificial intelligence algorithm for phonocardiograms based on existing models in the literature, trained with various noise databases that simulate realistic scenarios in which the audio is contaminated with noise from physiological processes and environmental noise, allowing the algorithm to identify and separate cardiac audio signals with different pathologies more accurately.

The findings of this study lay the groundwork for further investigation and applications in conjunction with PCG classification models. While the results obtained from the work and experimentation presented in this thesis provide state-of-the-art advancements, they provide a robust foundation to extend this work into related fields. Potential areas for further exploration are physiological sound denoising, cardiac sound classification, blind source separation, and similar topics associated to PCG denoising.

1.3. Hypothesis

Deep neural networks trained for sound source separation through mask estimation maintain consistent performance across diverse noise types, including complex environmental disturbances, provided that the training data encompasses a sufficiently representative spectrum of acoustic interference patterns. This independence from specific noise characteristics would enable robust audio separation in real-world applications where noise conditions are unpredictable and non-stationary.

1.4. Objectives

The problem presented in this thesis is the reduction of noise in PCGs to the greatest possible extent with the aid of neural networks. The following objectives define the key points to systematically address in order to verify the hypothesis.

1.4.1. General objective

To analyze and evaluate the performance of deep learning algorithms for the robust separation of pathological and non pathological PCGs from different types of noise, including those found in more realistic situations, such as environmental noise or noise produced by physiological processes.

1.4.2. Specific objectives

- To study techniques that allow a time-frequency representation of phonocardiograms.
- To study the training of different deep learning algorithms by means of the relationship between their architecture, training parameters, and performance.
- To evaluate how the model is affected by using different types of noise for training.
- To determine the best-fit model for the elimination of all types of noise used and at different levels.

1.5. Methodological framework of the thesis

The contents of research and experimentation of this thesis consist of implementing modern deep learning based techniques for PCG sound denoising. This involves utilizing

publicly available databases to train, process, and evaluate neural network models. The audio files from these databases are contaminated with noise, sourced either from other databases or artificially generated. All data processing and machine learning tasks are conducted in the Python program language with the aid of libraries that will be specified later in this thesis. The general structure of the methodology is presented in Figure 1.1. The methodology can be divided in two main parts: data pre-processing, in which the clean PCG audio is processed, contaminated, and from which time-frequency representations are obtained. The second part consists of neural network training and evaluation, specifically employing a U-Net architecture, based on the resulting time-frequency representations generated in the pre-processing stage.

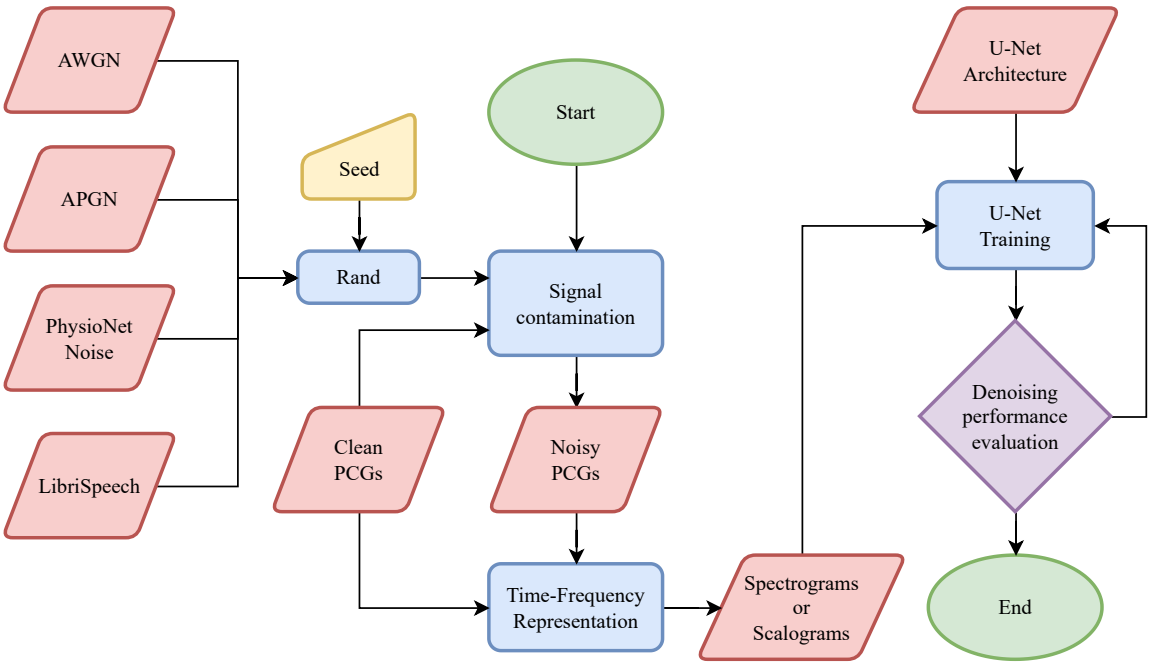


Figure 1.1 Flowchart of the methodology used to pre-process and train the denoising neural network models.

Deep learning methods generally require a large amount of data to achieve satisfactory training, this creates a limitation where a small training dataset can hinder the learning capability of neural networks. A popular approach to overcome this issue is extending the dataset with data augmentation techniques. An implementation of this approach is applied in §4.3. While working with medical data, that can impact people's health, it is necessary that both the data and results are validated by qualified experts in the field. While professional medical supervision can extend this method to real-world applications, this study is limited to quantitative results that can be compared with state-of-the-art research. Lastly, many tables and graphs are included to substantiate the reliability of the findings.

1.6. Structure of the thesis

This thesis is organized as follows: Chapter 2 presents a review of the literature relevant to the three primary areas of research that preceded the development of the proposed method: heart auscultation, traditional denoising techniques, and deep learning-based approaches for sound signal separation. Chapter 3 outlines the mathematical foundations necessary for this study, with a focus on signal analysis and processing, particularly the time-frequency transforms employed in the methodology, alongside an overview of the databases used and the fundamentals of the deep learning methodology. Chapter 4 details the application of these tools in the development of the proposed method. Chapter 5 presents the experimental procedures and testing conducted to identify the most effective denoising method, accompanied by various evaluation metrics to validate the robustness of the PCG denoising approach. Chapter 6 provides a detailed analysis and interpretation of the results, offering justifications for observed behaviors and explanations for any unexpected outcomes. Chapter 7 concludes the thesis with a summary of the proposed method and suggestions for future research aimed at improving the results and extending its application to other algorithms in the field. Finally, Chapter 8 presents the conference and journal publications that have resulted from this research.

Chapter 2

Literature review

2.1. Heart auscultation in cardiac diagnosis

Heart problems remain the leading cause of death worldwide, according to data from the World Health Organization (WHO) (2023). Heart auscultation is one of the least expensive, most useful, and fastest non-invasive diagnostic techniques now available, despite the existence of numerous other techniques like electrocardiography (ECG), magnetic resonance imaging (MRI), and echocardiography. Auscultation is the practice of using a stethoscope to listen to the body's internal sounds. Auscultation is an effective method for assessing the respiratory, circulatory, and gastrointestinal systems (breath sounds, heart sounds, and bowel sounds). Real-world ambient noise—that is, noises made by people and technology—as well as other internal body sounds frequently tamper with heart sounds captured with a stethoscope. The noises and murmurs the heart produces during its operation cycles are graphically represented by the phonocardiogram (PCG). The PCG records these noises over time, allowing medical professionals to examine the acoustic features of cardiac function. In most cases, a typical PCG signal consists of the two main cardiac sounds, S1 and S2. The Cardiac Cycle (CC) produces these noises when the semilunar valves close in S2 and the atrioventricular valves close in S1. With the Systole (contraction phase) for S1 and the Diastole (relaxation phase) for S2, the CC exhibits quasi-periodic activity (Figure 2.1). In healthy individuals, these noises are limited to the corresponding phases of the CC, resulting in two calm intervals: diastolic silence (s-Dia) and systolic silence (s-Sys). Individuals with cardiac conditions often have clicks, frictions, and murmurs in addition to other noises like S3 and S4 (Abbas and Bassam, 2022).

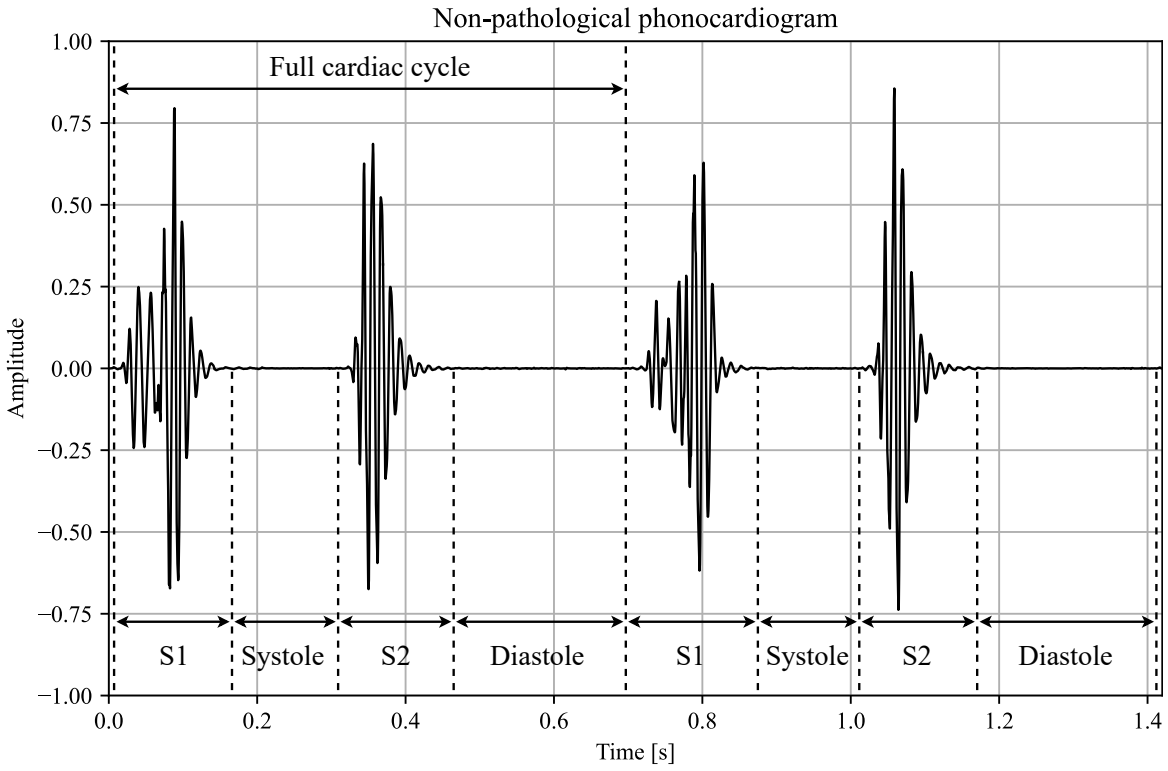


Figure 2.1 Normal phonocardiogram conformed of two cardiac cycles and their respective parts.

2.2. Traditional denoising techniques

The primary challenge to tackle here is the reduction of noise to the greatest possible extent that inevitably arises during stethoscope audio recordings. Signal denoising is a task that has been studied and improved over the years. One of the earliest and widely used methods for audio denoising, including PCGs, is spectral subtraction, initially proposed by Boll (1979). This technique involves estimating the noise spectrum and subtracting it from the noisy signal. Other common approaches to denoising involve filtering techniques. The Wiener filter, introduced by Chen et al. (2006), aims to estimate the original signal by minimizing the mean squared error (MSE), assuming that both the original signal and noise characteristics are known. Another widely used method is the median filter, which replaces data points in the signal with the median of neighboring points, as described by Yin et al. (1996). The least mean square (LMS) algorithm is another denoising method that shares similarities with the Wiener filter. However, unlike the Wiener filter, the LMS algorithm does not require knowledge of the statistical properties of the noise. Instead, it models the problem to fit the given data to minimize the MSE (Lines and Treitel, 1984). Recently, the LMS filter has been applied to PCG denoising, as demonstrated by Pauline and Dhanalakshmi

(2022). While this method successfully reduces a significant portion of the noise, its focus on synthetic and stationary noise, and leaves environmental noise unaddressed, indicating limitations in its effectiveness for more complex real-world noise scenarios.

Other denoising techniques include wavelet-based approaches, initially proposed by Donoho (1995) and later applied to PCGs by Messer et al. (2001). This method involves decomposing the signal into frequency bands using the discrete wavelet transform (DWT) to separate noise from the PCG signal before reconstructing the denoised signal, see Figures 2.2 and 2.3. Further research has expanded on this technique in recent years. For instance, Gradolewski and Redlarski (2014) explored various thresholding coefficients and evaluated the signal-to-noise ratio (SNR) on signals contaminated with additive white Gaussian noise (AWGN), additive pink Gaussian noise (APGN), and Brownian noise.

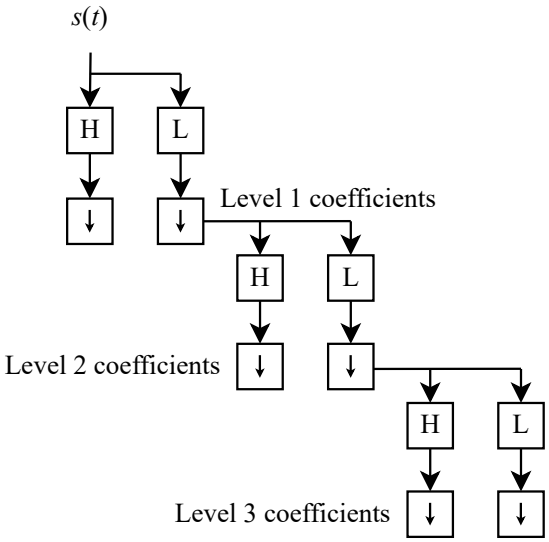


Figure 2.2 Discrete wavelet transform diagram of a signal $s(t)$, the 'H' and 'L' symbols represent a high and low-pass filters respectively, while the down arrow symbol represents down-sampling the resulting signal.

Further advancements have been made to the wavelet denoising technique (Table 2.1), including the exploration of different wavelet types and decomposition levels to enhance performance (Ali et al., 2017; Ghosh et al., 2020). Additionally, neural networks have been integrated into the process, enabling the reconstruction of the cardiac signal with an adaptive thresholding mechanism based on the wavelet coefficients (Gradolewski et al., 2019). These improvements have contributed to more effective noise reduction in phonocardiograms, leading to clearer signal outputs. However, despite these improvements, the performance of these methods remains heavily dependent on the selection of specific parameters, such

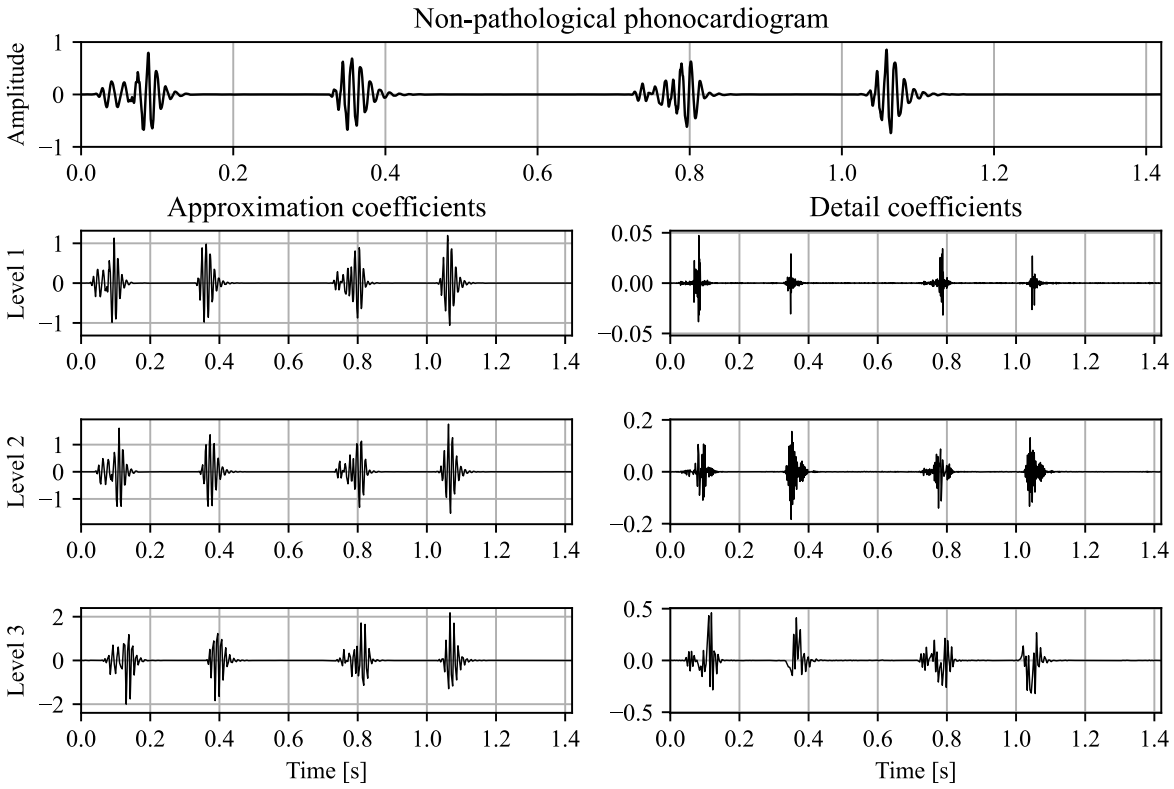


Figure 2.3 First 3 levels of decomposition with the discrete wavelet transform of the signal from Figure 2.1, using the Daubechies 10 (db-10) wavelet, commonly used when denoising PCGs. The approximation coefficients correspond to the low-pass filter while the detail coefficients correspond to the high-pass filter.

as the choice of wavelet, the level of decomposition, and the thresholding technique. This dependency is a critical limitation, as highlighted by Pauline and Dhanalakshmi (2022).

2.3. Deep learning-based methods

Notable progress has been made in artificial intelligence, and notably machine learning, in the last few years. Artificial neural networks are the main tool used in deep learning, a subfield of machine learning (Janiesch et al., 2021). Different neural networks architectures are developed to approach different kinds of problems, the most known architectures, to list a few, are: feedforward neural networks (FNNs) which are generally used for regression and basic classification tasks, its main disadvantage however is in the difficulty to store long-term information (LeCun et al., 2015). Convolutional neural networks (CNNs) mainly stand out in image and video processing (Krizhevsky et al., 2012; Simonyan, 2014), recurrent neural networks (RNNs) are usually utilized for data sequences, which is why they stand out in

speech recognition and natural language processing (Hochreiter and Schmidhuber, 1997; Rumelhart et al., 1986), long short-term memory networks (LSTMs) have its advantage in long-term dependencies in data sequences, making them more adequate for sequence learning (Lipton et al., 2015). The architecture to highlight here are CNNs, this is because although the problem posed here is on audio denoising, generally represented as a 1-dimensional data sequence over time, both time and frequency are conjugate variables that hold important information about the signal. Techniques to approach blind source separation problems for audio signals, such as Andreas et al. (2017); Hennequin et al. (2020) mainly focusing on separating songs into the individual instruments audio sources, generally use a time-frequency representation to feed into the neural network, in particular, a U-Net. This architecture is named after the letter 'U' in the western alphabet for its architecture shape and has shown outstanding results on biomedical image segmentation (Ronneberger et al., 2015). A recent survey on the application of U-Nets for audio processing in speech, music, biomedical, and other sounds is presented by Gul and Khan (2023). In recent years, significant advancements in computing power have facilitated the growing adoption of deep learning (DL)-based methods for PCG enhancement. Due to their effectiveness in handling non-stationary noise, DL-based approaches for PCG denoising have gained attention. These methods can be broadly categorized into time-domain (or 1-D signal) techniques (Ali et al., 2023; Nikbakht et al., 2024) and time-frequency domain (2-D image) approaches (Al-Zaben et al., 2024). A closely related challenge is the blind source separation of heart-lung sounds (HLS), particularly during clinical auscultation, where heart and lung sounds frequently overlap in both the frequency and time domains. Researchers have explored various methods for HLS separation (Nersisson and Noel, 2017), with recent studies increasingly investigating DL techniques (Sun et al., 2024; Wang et al., 2023).

2.4. Proposed approach

Most traditional PCG denoising methods have been evaluated using synthetic noise, typically AWGN and occasionally APGN. However, real-world PCG recordings often contain non-stationary noise, highlighting the limitations of these conventional approaches. While traditional filter-based methods have shown this limited effectiveness, recent advancements in deep learning models have demonstrated superior capabilities in addressing more complex denoising challenges. Consequently, the application of deep learning techniques to PCG denoising is proposed as the main subject to discuss in this thesis.

Table 2.1 PCG denoising literature review.

Authors	**Methodology**	**Noise type**	**Noise level**	**Results and Conclusions**	**Quantity/Length of audio and databases utilized**
Messer et al. (2001)	Discrete Wavelet Transform (DWT) with an evaluation of various wavelet families, decomposition levels, and thresholding methods.	AWGN	1, 5, 10 dB	Best results from three trials of contaminating a signal to an SNR of 1 dB were obtained with the Rigrsure thresholding method resulting on 13.65, 13.78, 13.7 dB.	3 audios were recorded and utilized on 3 trials with 1, 5, and 10 dBs of noise added for 1 to 10 decomposition levels.
Gradolewski and Redlarski (2014)	DWT with an evaluation of various wavelet families, decomposition levels, and thresholding methods.	AWGN, APGN, ARGN*	5, 10, 15 20 dB	Best results obtained with the Coiflet 5 wavelet at 10th decomposition level, obtaining an SNR (fit coefficient) of 15.3 dB (96.01%), 21.5 dB (87.73%), 26.7 dB (75.43%), 33.3 dB (58.72%) from audio at 5, 10, 15, 20 dB initial noise levels respectively.	Quantity not specified. LHS** database was utilized to find the optimal wavelet parameters. The performance of the algorithm was tested on 4 different pathological heart sounds.
Ali et al. (2017)	DWT with an evaluation of various wavelet families, decomposition levels, thresholding methods.	AWGN	5 dB	Best results were obtained at the 4th decomposition level, of 15.35 and 15.24 dB for the Discrete Meyer wavelet from soft and hard thresholding methods respectively and 15.43 and 15.6 dB for Daubechies 10.	Quantity not specified. PASCAL database.
Gradolewski et al. (2019)	DWT with a Time Delay Neural Network.	AWGN, APGN.	5, 10, 15, 20 dBm***	Results were evaluated with a correctness percentage, obtaining an average of 95.7%, 90.4%, and 83.7 on the signals contaminated with pink noise for the noise levels of 5, 10, and 15 dBm respectively.	1,647 cardiac cycles utilized with various pathologies from MHSM, EGM, LHS, Washington, and Thinklabs databases. One half is for training and the other half for testing.

Table 2.1 (continued).

Author	Methodology	Noise type	Noise level	Results and Conclusions	Quantity/Length of audio and databases utilized
Ghosh et al. (2020)	DWT with an evaluation of various wavelet families, and thresholding methods.	AWGN	0, 5, 10, 15 dB	Best results obtained from Coiflet 5 wavelet, on a normal PCG at an SNR of 15 dB denoised to 22.11 dB and from an SNR of 0 dB to 10.01 dB.	Types of heart sounds utilized: Normal: 69.22s, Mitral Stenosis: 61.17s, Aortic Stenosis: 61.8s, Pulmonary Valve Stenosis: 75.18s, duration from MHSM database.
Mohan et al. (2020)	Group Sparsity, Synchrosqueezing Wavelet Transform.	AWGN	-5, 0, 10, 20 dB	Comparisons from an SNR of -10 and 15 dB obtained an output SNR of 15.35 and 35.33 dB respectively.	Quantity not specified. EGM, MHSM, LHS, HSS databases were utilized.
Pauline and Dhanalakshmi (2022)	Adaptive Least Mean Square Filter.	AWGN, APGN	-1 and 4 dB	Best results obtained with the proposed model with an SNR of 42.6545 and 45.6264 for -1 and 4 dB of initial SNR respectively.	Quantity not specified. A normal and abnormal signal from the PhysioNet database, both with a duration of 2s sampled at 8 kHz, and a synthetic PCG signal of 1s sampled at 1 kHz.
Andreas et al. (2017)	STFT, U-Net.	Vocals, Instruments	Does not apply	Results evaluated with NSDR, obtaining 11.09 dB for vocals and 14.43 dB for the instruments	Model trained with commercially available recordings of original songs plus their instrumental version.

Table 2.1 (continued).

Author	Methodology	Noise type	Noise level	Results and Conclusions	Quantity/Length of audio and databases utilized
Hennequin et al. (2020)	STFT, U-Net.	Vocals, Instruments	Does not apply	Results evaluated with SDR, SIR, SAR, ISR, obtaining better results than the previous state-of-the-art performances on SDR, SIR, SAR for vocals, SDR and SIR for bass, all four metrics for drums and other.	Model trained with Deezer internal datasets.

*Additive Red Gaussian Noise (ARGN).

**eGeneralMedical (EGM), Michigan Heart Sound and Murmur (MHSM), Littman Heart Sound (LHS), Heart Sound Signal (HSS).

***Noise measured as the power of noise introduced to the signal.

Chapter 3

Theoretical framework

3.1. Time-frequency signal analysis and processing

Signals are a practical tool in which many physical phenomena with time variations can be mathematically represented. Seismic signals represent earth vibrations over time, caused by many factors (Díaz, 2016); in electronics, signals general refer to the voltage variation over time in an electrical circuit (Vasudevan, 2018); acoustic signals display the time-varying air pressure that we interpret as sound (Boashash, 2003). This time representation facilitates the analysis for different kinds of signals, they can be mathematically expressed as a time-dependent function $s(t)$ and represented graphically in an amplitude-time plane where the energy can be localized throughout time in the signal.

While this type of representation allows for this kind of analysis, an important piece of information contained in the signal is found in its frequency. The frequency of a signal visually represents how elongated the wave is along the amplitude-time plane; along with the energy, it describes the shape of the signal. The frequency is what characterizes a signal, a sound signal with high frequency will be percieved as having higher pitch than a low frequency signal; however, frequency is not often trivial to estimate at plain sight from a 1-dimensional time sequence, but a frequency-dependent function can be obtained with the Fourier transform:

$$\widehat{s}(\omega) = \int_{-\infty}^{+\infty} s(t) e^{-\imath\omega t} dt, \tag{3.1}$$

from the integral it can be seen that time is the integration variable to localize the frequency and therefore is lost; this creates a trade-off of information between time and frequency to localize one or the other, also known as the Heisenberg uncertainty principle (Mallat, 2008).

The signal $s(t)$ (and any signal referred to in this thesis) is a function in the space $s(t) \in \mathbf{L^2}(\mathbb{R})$, this means the signal has finite energy $\int_{-\infty}^{+\infty} |s(t)|^2 dt < +\infty$ and therefore $s(t)$

is square-integrable. The signal's Fourier transform also pertains to this space $\widehat{s}(\omega) \in \mathbf{L^2}(\mathbb{R})$ and has finite energy. As consequence of $\mathbf{L^2}(\mathbb{R})$ being a Hilbert space, its inner product $\langle s, s' \rangle = \int_{-\infty}^{+\infty} s(t)s'(t)^* dt$ exists, where $s'(t)^*$ is the complex conjugate of any signal $s'(t) \in \mathbf{L^2}(\mathbb{R})$. The Parseval's theorem holds, where (Mallat, 2008):

$$\int_{-\infty}^{+\infty} |s(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |\widehat{s}(\omega)|^2 d\omega. \tag{3.2}$$

A signal can contain essential information that may not be apparent or available to work with in a standard time-dependent representation, as this information is hidden in its frequencies. As seen previously, working in the frequency domain loses the time localization of the signal's given frequencies; yet, it is often necessary to know where those frequencies are present in time, i.e., a time-frequency function $\mathscr{F}(\tau, \omega)$ would be desirable. The most basic example of this is with the short-time Fourier transform, defined in the next section.

As mentioned, the signal's localization in time is lost when going to the frequency domain. This opens the possibility of having two completely different signals in the time domain that look identical in the frequency domain (Figure 3.1). Therefore, being able to retain the localization information of the signal is important in order to go back from the frequency to the time domain, and the signal phase allows this (Boashash, 2003). The phase represents an angle in the complex plane for each frequency in $\widehat{s}(\omega)$ and can be defined as:

$$\phi(\omega) = \arctan\left[\frac{\mathrm{Im}(\widehat{s}(\omega))}{\mathrm{Re}(\widehat{s}(\omega))}\right], \tag{3.3}$$

multiplying the magnitude $\|\widehat{s}(\omega)\|$ by the angle $e^{i\phi(\omega)}$ and applying the inverse Fourier transform allows for a perfect reconstruction of the signal. Similarly, in a time-frequency representation with a function of the form $\mathscr{F}(\tau, \omega)$, the time-frequency trade-off exists. The frequencies are generally calculated for a short time window that is sampled over the entire duration of the signal. Calculating the magnitude $\|\mathscr{F}(\tau, \omega)\|$ loses the time localization for each sampled window, therefore also requiring the phase to achieve a perfect reconstruction.

3.2. Short-time Fourier transform

The STFT is a popular method used over any other time-frequency representation given its simplicity, computational speed and highly satisfactory performance. In this work we process thousands of audios and images which makes computational speed a crucial factor. In other works where U-Nets are utilized for blind source separation, such as Andreas et al. (2017); Hennequin et al. (2020), the STFT is used to provide a time-frequency representation

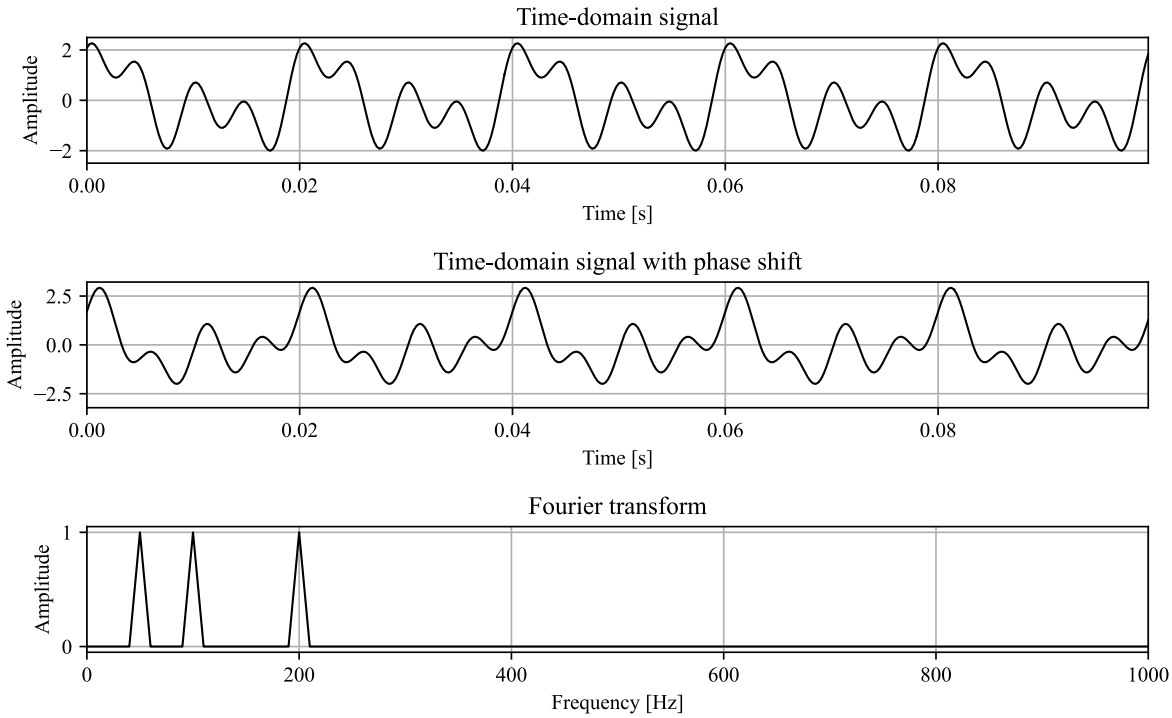


Figure 3.1 Example of two different signals with the same Fourier transform. Each signal is conformed of 3 sine waves with the same frequency, shifting the phase on each sine wave for the second signal.

of the audio signals, the spectrograms then are utilized as images to train the network. This approach includes an accurate representation of the signal's time and frequency information and it has demonstrated good performance on source separation difficulties; these are the two main reasons why the current work uses the same approach.

The calculation of the STFT is based on the translation and modulation of a *window* function along a signal (Mallat, 2008). A real and symmetric window $w(t)$ is translated in time by τ and modulated by the frequency ω: $w_{\omega,\tau}(t) = w(t-\tau)e^{\imath\omega t}$, where $\|w_{\omega,\tau}\| = 1$ for any $\omega, \tau \in \mathbb{R}^2$. The resulting windowed Fourier transform for a signal $s(t)$ is then:

$$F_s(\omega, \tau) = \int_{-\infty}^{+\infty} s(t)w(t-\tau)e^{-\imath\omega t}dt, \tag{3.4}$$

the STFT localizes the energy of the Fourier integral in the neighborhood of $t = \tau$ (Mallat, 2008).

In this work discrete sampled PCG signals are utilized. These signals are obtained by recording a value for the amplitude a fixed amount of times every second, this would be the sample rate (SR). To analyze and process these signals the discrete STFT is required, for this the discrete Fourier transform (DFT) is applied to a time frame formed by a group of samples

where the result is a complex vector that is added as a column to a matrix, this matrix records the magnitude and phase for all time frames and frequency bins. The discrete STFT of a signal s is computed as follows (Smith, 2011):

$$X_s(k,m) = \sum_{n=-\infty}^{+\infty} s_l(n)\, w(n-mH)\, e^{-\imath 2\pi nk/N}, \tag{3.5}$$

here m represents each time frame along the original signal s, with $m = 0,1,2,\ldots,M-1$, where M is the total number of time frames the original signal is divided in and n represents each sample across the local signal $s_l(n)$. For each time frame m the DFT is calculated locally across a signal $s_l(n)$, resulting in a vector of frequencies $X_{s_l}(k)$. The set of frequency bins along this vector is defined as $f_k = kf_s/N$, for $k = 0,1,2,\ldots,N/2$, with f_s being the sampling frequency of the signal and N the number of time samples in the DFT. Finally, H is the hop length in between two consecutive time frames m and $m+1$ that delimit $s_l(n)$.

Although there are many window functions, any that complies with the constant overlap-add (COLA) constraint:

$$\sum_{m=-\infty}^{+\infty} w(n-mH) = 1,\ \forall n \in \mathbb{Z}, \tag{3.6}$$

allows the STFT to have an inverse (iSTFT). The Hann window is utilized here and is defined as follows:

$$w(n') = 0.5 - 0.5\cos\left(\frac{2\pi n'}{L_w - 1}\right), 0 \leq n' \leq L_w - 1, \tag{3.7}$$

the window length is denoted by L_w. With this, the shape of the resulting matrix $X_s(k,m)$ would be (K,M), with $K = N/2+1$, defining $R = (-(S-L_w)\,\%H)\,\%L_w$ with $\%$ being the division remainder, then $M = \lceil (S+R+1)/H \rceil$, where $\lceil\cdot\rceil$ represents the ceiling function; $S = T f_s$ being the total number of samples that conform the original signal s, where T is the time duration of the signal (Virtanen et al., 2020). Testing different windows that fulfilled this constraint did not show any noticeable changes in the denoising performance.

3.3. Wavelet transform

Heisenberg's uncertainty principle states that there is a trade-off between time and frequency resolution when employing this type of representation for signal analysis. It is easier to identify the frequency of the signal when working with high frequencies given a shorter time window. This characteristic is utilized by the continuous wavelet transform (CWT) by generating a time-frequency representation with a variable resolution for different values of time and frequency, this is commonly referred to as a scalogram.

3.3.1. Continuous wavelet transform

Given a signal $s(t)$ and a wavelet $\psi_{a,b} \in \mathbf{L^2}(\mathbb{R})$, the CWT is defined as follows:

$$W_s(a,b) = \langle s, \psi_{a,b} \rangle = \int_{-\infty}^{+\infty} s(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-b}{a} \right) dt, \tag{3.8}$$

the main difference it has with the STFT is that the CWT utilizes a wavelet dictionary instead of a window function. This dictionary is conformed by wavelets with variable time-frequency resolution, defined by the parameter a, shifted in time by b. One of the requirements to apply the CWT is that it is has an inverse transform, for this the wavelet dictionary has to be based on a mother wavelet:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right), \tag{3.9}$$

with average of zero:

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0. \tag{3.10}$$

The time-frequency resolution of each wavelet is controlled the parameter $a = 2^{j/N_v}$, this can be interpreted as how many frequency bands will be set up in the wavelet dictionary, j is the index representing the octaves, dividing by N_v (the number of voices), it essentially defines how many subdivisions or intermediate scales will be set for each octave, see Figure 3.2. Thus, increasing N_v improves the frequency resolution by employing a larger set of wavelets to analyze a wider range of intermediate frequency values (Mallat, 2008).

Wavelets are mainly classified into real and analytic wavelets. While real wavelets are used to detect sudden shifts in signals (Mallat, 2008) or analyze mathematical singularities (Tu et al., 2005), analytic wavelets allow the study of frequency variations along time in signals by having the capacity to separate the amplitude and phase information of the signal (Mallat, 2008). Examples of both real and analytic wavelets are presented in Figure 3.3, the first three real wavelets are common examples of wavelets utilized with the DWT for PCG denoising, while the analytic Morlet wavelet, along with other analytic wavelets, are commonly used with the CWT for their representation of complex signals, this wavelet is illustrated in the complex plane in Figure 3.4.

There are different types of analytic wavelets, in particular, the Gabor wavelets which have been studied and experimented through many research works, have provided greater performance on PCG analysis (Ibarra-Hernández et al., 2017). For this reason the Morlet wavelet was chosen in this work, a particular case of the Gabor wavelet. The Morlet wavelet

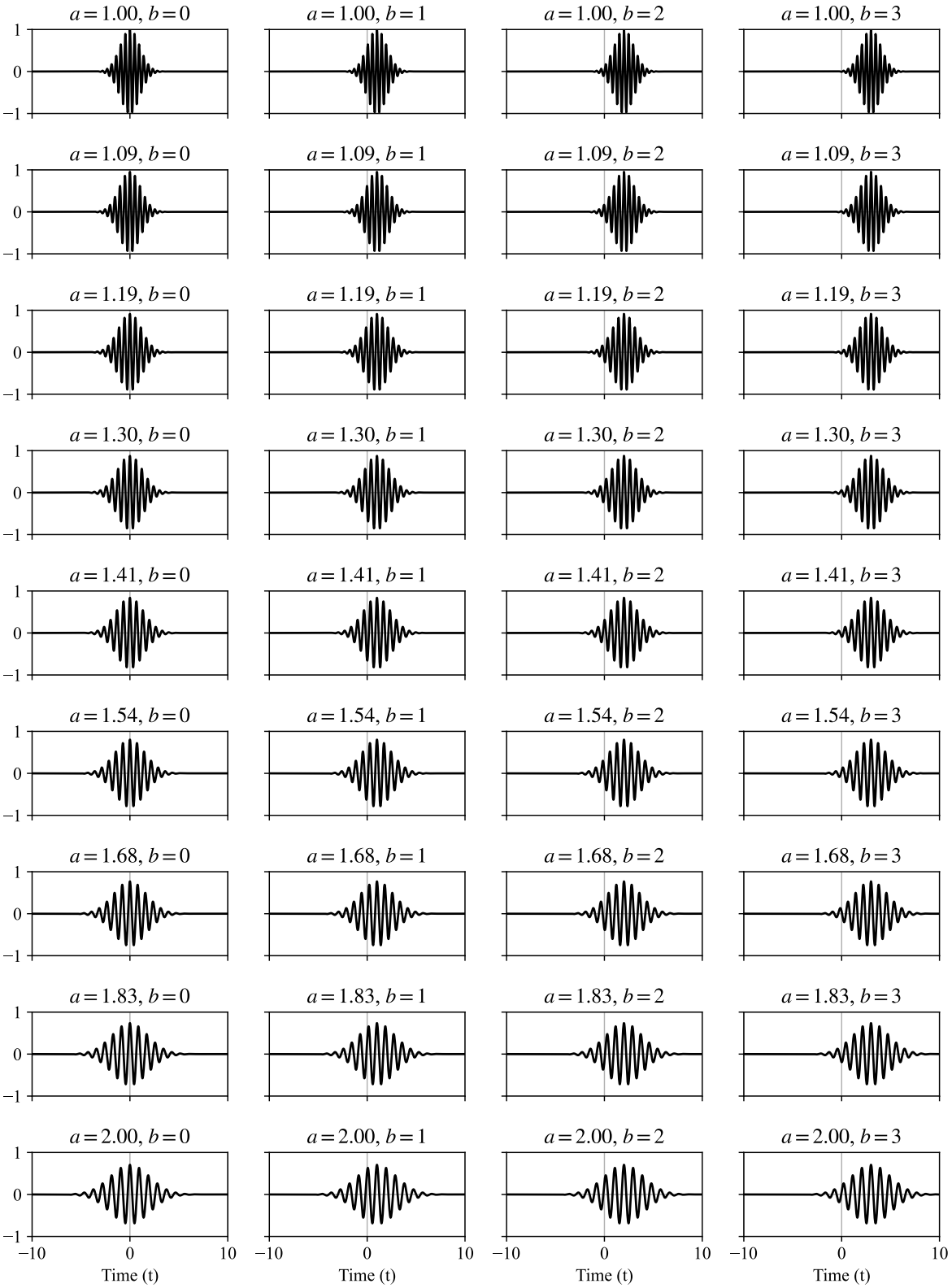


Figure 3.2 Example of a wavelet dictionary with the real part of the Morlet mother wavelet, with $\mu = 13.4$, displaced in time by b and scaled by a. Here $N_v = 8$, which corresponds to 8 wavelets scaled in between the interval $a = [1,2)$.

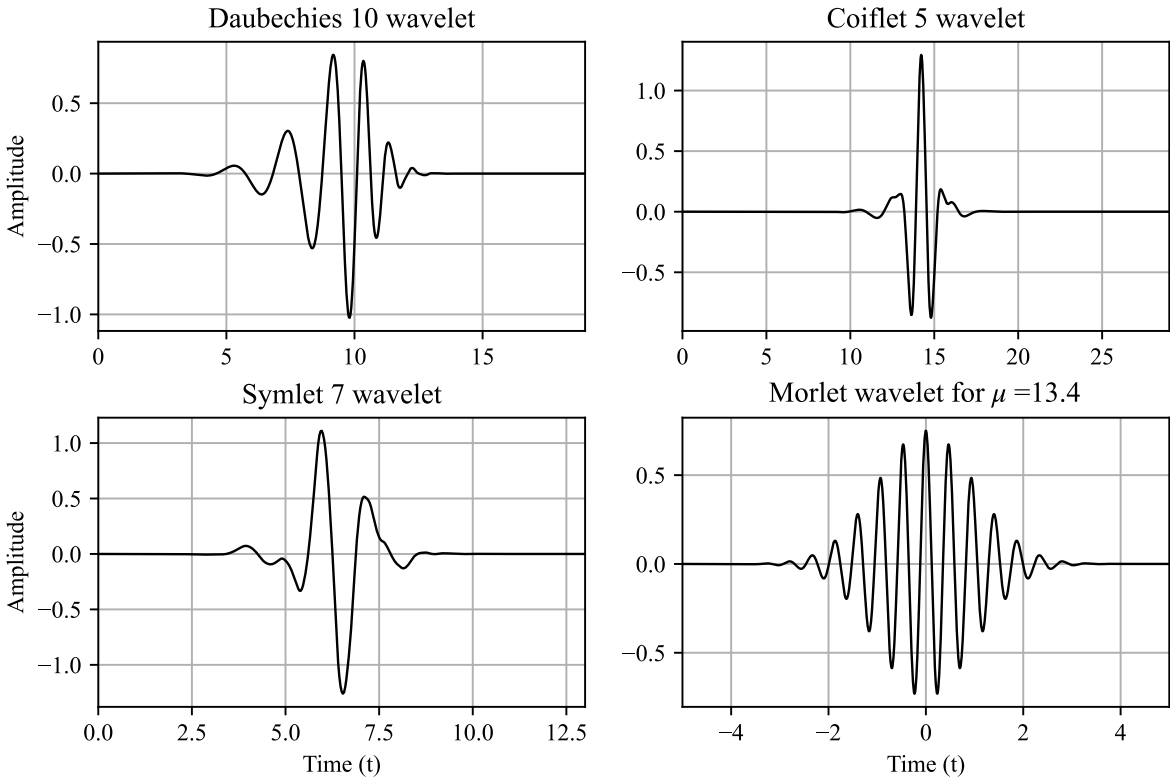


Figure 3.3 Example of wavelets commonly used for PCG denoising. The first three being real wavelets and the fourth illustrating the real part of the analytic Morlet wavelet.

is defined in the time domain as:

$$\psi_\mu(t) = c_\mu e^{-\frac{1}{2}t^2}\left(e^{\imath\mu t} - e^{-\frac{1}{2}\mu^2}\right), \tag{3.11}$$

with the normalization constant c_μ:

$$c_\mu = (1 + e^{-\mu^2} - 2e^{-\frac{3}{4}\mu^2})^{-\frac{1}{2}}, \tag{3.12}$$

in particular, when taking $\mu = 13.4$, this wavelet matches the the generalized Morse wavelets family (Muradeli, 2020), defined in the frequency domain as:

$$\psi_{\beta,\gamma}(\omega) = C(\omega)a_{\beta,\gamma}\omega^\beta e^{-\omega^\gamma}, \tag{3.13}$$

with $\beta = 3$ and $\gamma = 60$, here $C(\omega)$ is a normalization constant; these wavelets are analyzed in more detail in Lilly and Olhede (2012).

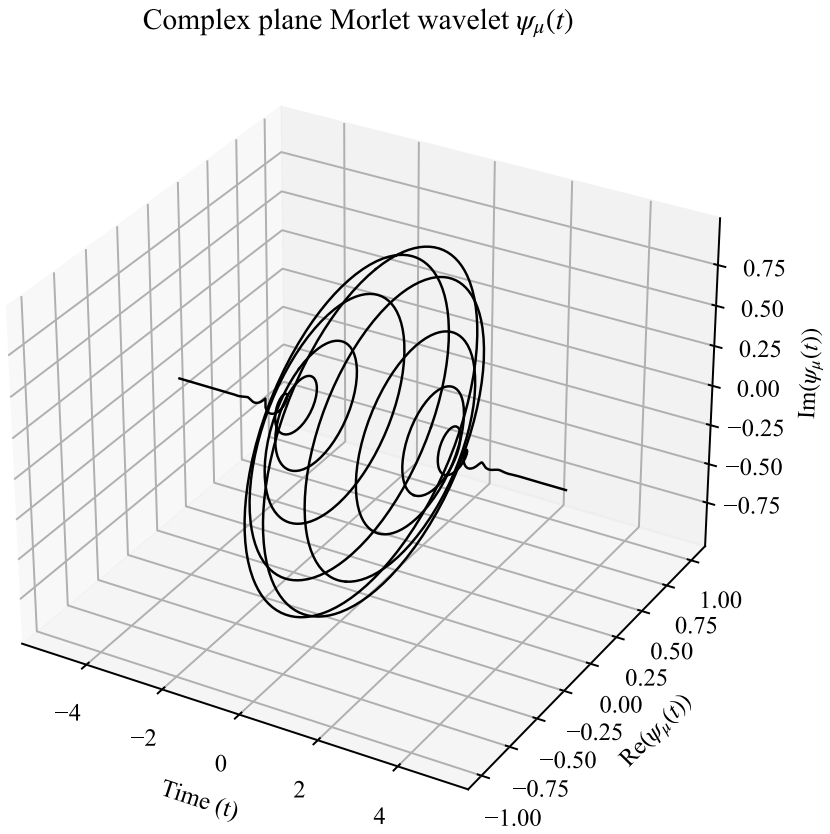


Figure 3.4 Morlet wavelet in the complex plane for $\mu = 13.4$.

To take a closer look at the time-frequency resolution of the wavelet transform, employing the Morlet wavelet with $\mu = 13.4$, its Fourier transform is calculated:

$$\widehat{\psi}_\mu(\omega) = c_\mu \sqrt{2\pi} \left(e^{-\frac{1}{2}(\omega-\mu)^2} - e^{-\frac{1}{2}\mu^2} e^{-\frac{1}{2}\omega^2} \right), \tag{3.14}$$

this results on a function of the frequency spectrum of the wavelet which can be used to visualize the Heisenberg boxes for various values of a and b. As μ increases, the term $e^{-\frac{1}{2}\mu^2} e^{-\frac{1}{2}\omega^2}$ approaches zero, simplifying the function to a Gaussian form where μ represents the central frequency of the wavelet, here μ is large enough to use this approximation. The wavelet can be scaled by a factor a, modifying its shape. The frequency representation is inversely proportional to a, dilating its size by $1/a$ and changing the localization of the central frequency to μ/a. Conversely, the time representation of the wavelet is directly proportional to a, resulting on a wider representation as a increases. The parameter b simply represents a displacement in time, see Figure 3.5 (Mallat, 2008).

This can also be observed by calculating the Fourier transform after scaling the wavelet by a and displacing it by b, for simplicity the Morlet wavelet is taken as:

$$\psi_\mu(t) = c_\mu e^{-\frac{1}{2}t^2} e^{\imath\mu t}, \tag{3.15}$$

with $t = (t' - b)/a$ the variance of the time-domain representation becomes a, scaling the width of the Heisenberg box by that factor. Integrating with respect to t' gives its Fourier transform:

$$\widehat{\psi}_\mu(\omega) = c_\mu a\sqrt{2\pi} e^{-\imath\omega b} e^{-\frac{1}{2}(a\omega-\mu)^2}, \tag{3.16}$$

here the exponent can be represented as $-\frac{1}{2}\left(\frac{\omega-\mu/a}{1/a}\right)^2$ where the new central frequency is μ/a and the height scaling factor is $1/a$. While Eq. (3.16) has an oscillating term $e^{-\imath\omega b}$, Figure 3.5 represents the magnitude $\left|\widehat{\psi}_\mu(\omega)\right|$ along the frequency axis. It is important to

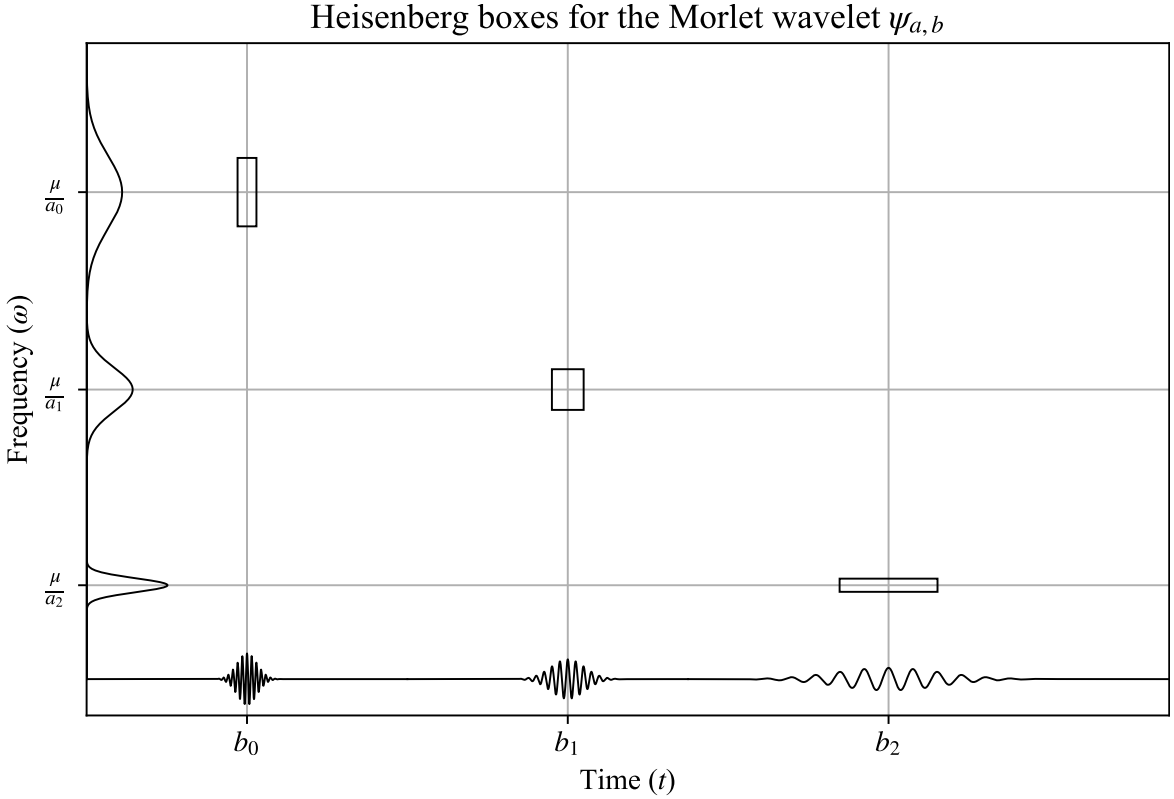


Figure 3.5 Heisenberg boxes for the Morlet wavelet at $\mu = 13.4$ given different values for a and b. For visibility purposes, the width and height of the boxes are scaled to twice their original size.

note that when computing scaled wavelets, the factor $1/\sqrt{a}$ from Eq. (3.9) must be added to

both the time and frequency representations so the Parseval's theorem Eq. (3.2) holds and to preserve the wavelet energy across different scales.

In contrast to the short-time Fourier transform, where each window is of uniform size regardless of its position in the time-frequency plane, the wavelet transform employs a variable that adjusts the size of each box according to the frequency range the signal is being analyzed at. This results in enhanced frequency resolution at lower frequencies, where it takes more time to localize the frequency of a signal. Conversely, at higher frequencies that are localized in a shorter amount of time, the wavelet transform provides a lower frequency resolution in trade for a higher time resolution.

The implementation of the different Heisenberg boxes shape can be observed by comparing the STFT and CWT of the same PCG signal in Figure 3.6 where each box is highlighted. In the STFT representation each box has the exact same shape, displaying the same time-frequency resolution across the entire time-frequency plane. The CWT however, analyzes a significantly larger frequency range at higher frequencies over a shorter period of time, while the frequency resolution increases at the low frequency values in trade for an analysis over a longer time window.

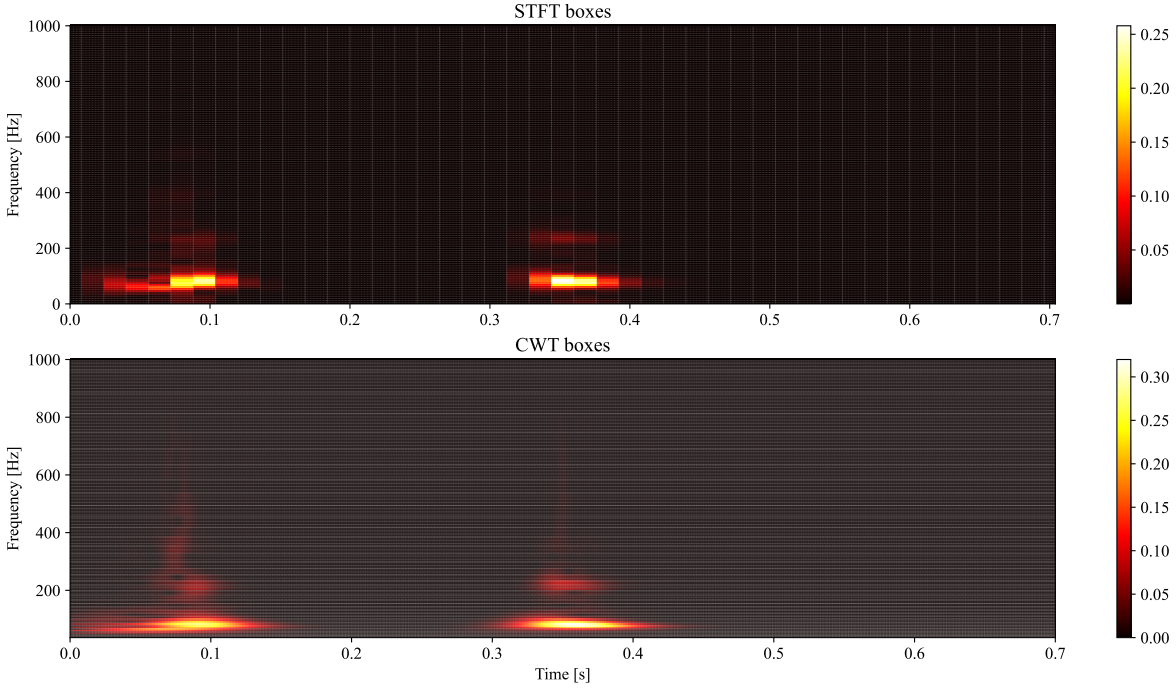


Figure 3.6 Heisenberg boxes displayed on the STFT and CWT representations of a PCG signal. This figure can be appreciated better in the digital version of this document which allows for magnification.

3.3.2. Wavelet synchrosqueezing transform

While the CWT proposes a new approach to obtain a time-frequency representation, the resolution trade-off is still present. In the wavelet synchrosqueezing transform (WSST), a method introduced by Daubechies et al. (2011), the energy of the signal is reassigned to finer frequency bins based on the instantaneous frequency of the signal at each point in time, aiming to generate a more precise representation (Figure 3.7).

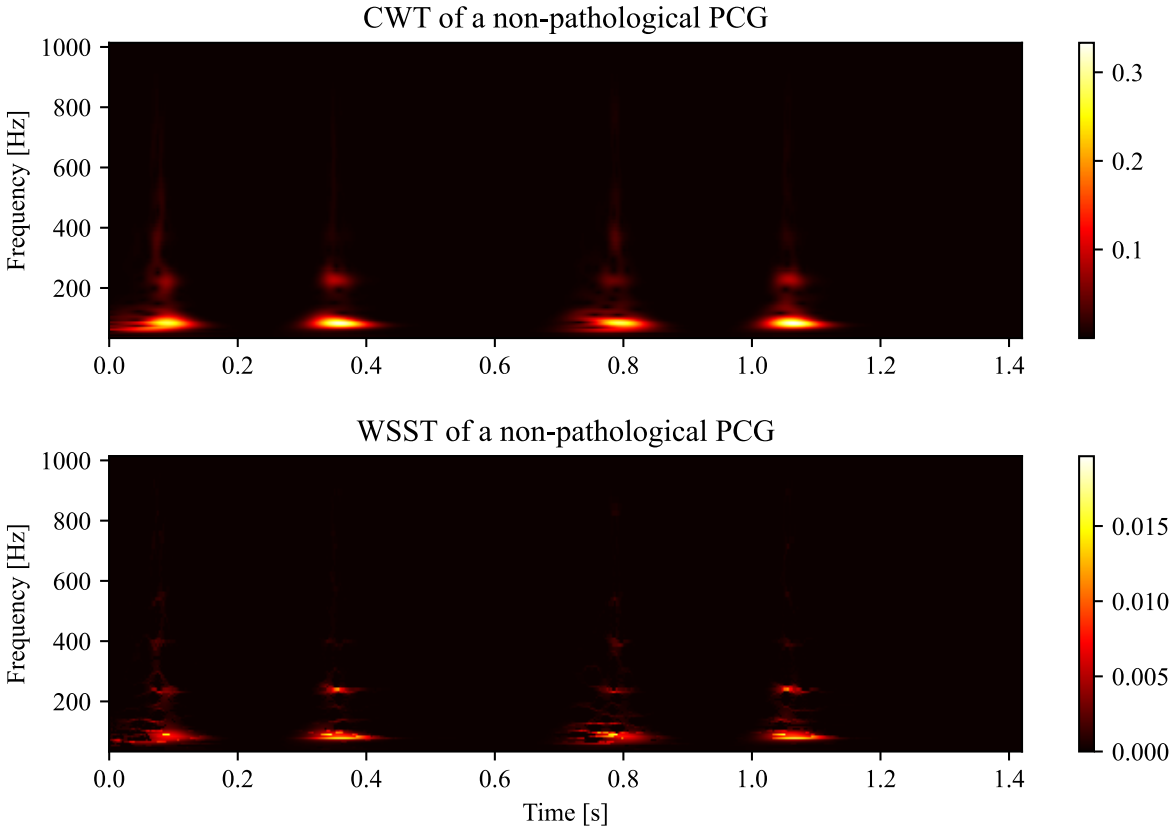


Figure 3.7 Comparison between the CWT and WSST of the signal from Figure 2.1.

This is done utilizing the instantaneous frequency $\omega_s(a,b)$ defined as:

$$\omega_s(a,b) = \frac{-\imath}{W_s(a,b)} \frac{\partial W_s(a,b)}{\partial b}. \tag{3.17}$$

While the CWT spreads signal energy across different scales, the WSST concentrates this energy in precise frequency bands, improving the frequency resolution while maintaining the time resolution. This refinement allows for a more accurate and interpretable time-frequency representation, particularly in signals with rapidly changing frequency components. The

synchrosqueezed wavelet transform $T_s(\omega_{s,\ell}, b)$ is given by:

$$T_s(\omega_{s,\ell}, b) = \frac{1}{\Delta\omega_s} \sum_{a_k : \left|\omega_s(a_k,b) - \omega_{s,\ell}\right| \leq \Delta\omega_s/2} W_s(a_k, b) a_k^{-3/2} \Delta a_k, \tag{3.18}$$

determined at centers ω_ℓ at discrete values Δa_k. The coefficients $T_s(\omega_{s,\ell}, b)$ are computed by concentrating the CWT coefficients $W_s(a_k, b)$ into specific frequency bins $\omega_{s,\ell}$ at each point b in time, discretized by the scale interval Δa_k (Daubechies et al., 2011).

A key feature of the WSST is its ability to accurately reconstruct the original signal from the synchrosqueezed time-frequency representation. The energy reassignment occurs only in the frequency domain, ensuring that the time domain resolution is preserved and that an inverse transform can accurately retrieve the signal.

3.4. *S*-transform

Another time-frequency representation method is the *S*-transform, this method has gained popularity in recent years due to its simplicity and the advantage it takes over the Heisenberg's uncertainty principle to analyze signals. Its mathematical definition is similar to that of the STFT and CWT, a signal s moved into the frequency domain with a window w:

$$S_s(f, \tau) = \int_{-\infty}^{+\infty} s(t) w(f, \tau - t) e^{-\iota 2\pi f t} dt, \tag{3.19}$$

the main difference is the type of window that is chosen, which depends on the frequency. Generally a Gaussian window is utilized (Stockwell et al., 1996):

$$w(f, \tau - t) = \frac{|f|}{\sqrt{2\pi}} e^{-\frac{f^2(\tau-t)^2}{2}}. \tag{3.20}$$

By introducing a frequency-dependent window, a parameter κ can be added (Eq. (3.21)) to vary the time-frequency resolution, in a similar way the shape of a single wavelet would change according to the parameter a to analyze a specific Heisenberg box; in the case of the *S*-transform a single window would be used to analyze the entire signal.

$$w(f, \tau - t) = \frac{|f|}{\kappa\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{f(\tau-t)}{\kappa}\right)^2} \quad \forall \kappa > 0. \tag{3.21}$$

The *S*-transform can be expressed as the convolution:

$$S_s(f, \tau) = s(t) e^{-\iota 2\pi f t} * w(f, \tau - t), \tag{3.22}$$

then, as demonstrated by Ventosa et al. (2008), the Fourier transform theorem can be applied to be able to calculate and compute the *S*-transform as the following inverse Fourier transform:

$$S_s(f,\tau) = \int_{-\infty}^{+\infty} \widehat{s}(\xi + f) e^{-2\left(\frac{\pi k \xi}{f}\right)^2} e^{\imath 2\pi\xi\tau} d\xi, \tag{3.23}$$

where ξ is an integration variable.

3.5. Power spectral density

Having a thorough understanding of the power distribution of the frequency from the different sounds that are being studied is essential for its analysis. Unlike the FFT, which also gives a frequency representation of the signal, the power spectral density (PSD) provides a distribution of the signals' energy across all its frequencies, thus allowing to view in which frequencies the PCGs' are mainly localized and what type of noise would result more useful to train with, that is, where traditional methods would fail to filter that noise.

The computation of the average PSD is carried out using Welch's averaging method (Stoica and Moses, 2005). This consists of averaging the periodograms $\varphi_j(\omega)$ of multiple overlapping windowed segments of the signal. Each segment, or windowed signal, is represented by $s_j(t) = s((j-1)K_s + t)$ for $t = 0,1,2,\ldots,M_s - 1$ and $j = 0,1,2,\ldots,S_w - 1$. Here, M_s represents the number of samples in each windowed segment, while K_s indicates the shift between the start points of consecutive segments, the overlap between segments is then given by $\frac{K_s}{M_s}$ and the total number of windowed segments is S_w. In order to perform the calculation, the periodogram of a windowed signal segment is computed first:

$$\varphi_j(\omega) = \frac{1}{M_s P} \left| \sum_{t=1}^{M_s} v(t) s_j(t) e^{-\imath\omega t} \right|^2, \tag{3.24}$$

where $v(t)$ is a tapering window function used to smooth the data and reduce spectral leakage. The power of the tapering window is given by $P = \frac{1}{M_s}\sum_{t=1}^{M_s} |v(t)|^2$. The PSD for one complete signal is obtained as follows (Stoica and Moses, 2005):

$$\varphi_W(\omega) = \frac{1}{S_w} \sum_{j=1}^{S_w} \varphi_j(\omega). \tag{3.25}$$

3.6. Acoustic sources used in this study

The type of input required to train the neural network is derived from both clean and noisy PCG signals. The easiest and most reliable approach to generate such data involves introducing noise into clean PCG signals. This process requires, first and foremost, a database of clean PCG recordings, and second, various types of noise to ensure robust denoising performance across different noise conditions. These noise sources can be both synthetically generated and obtained from external databases, providing a comprehensive set of training examples for the network to learn from and generalize effectively.

3.6.1. Natural sound sources

Three databases were selected to carry out the methodology of this thesis: one for clean PCG signals and two for noise sources. The first database is the Son and Kwon (2018) dataset, which contains 1,000 PCG recordings categorized into five classes: aortic stenosis (AS), mitral stenosis (MS), mitral regurgitation (MR), mitral valve prolapse (MVP), and normal (N), with 200 recordings in each category. Each recording consists of 3 full cardiac cycles, with an average duration of 2.44 seconds each, and a total combined duration of 40 minutes and 32 seconds. What characterizes this database is its lack of noise, unlike most cardiac sound databases. This absence of noise makes it ideal for obtaining clean target signals, which can then be artificially contaminated for the purpose of neural network training.

The second dataset used is the PhysioNet database (Liu et al., 2016), a collection of nine distinct datasets compiled for the PhysioNet/Computing in Cardiology (CinC) Challenge. It was created to address the lack of rigorously validated heart sound datasets for segmentation and classification algorithms. A subset of this dataset, specifically the training sets labeled alphabetically from *a* to *f*, comprises 3,153 heart sound recordings, forming the basis of the database used in this study. From datasets *a* to *c*, recordings of normal heart sounds were selected. In clean heart sound recordings, the systole and diastole phases typically correspond to periods of silence. However, in noisy recordings, these phases capture ambient noise typical of real-world environments, providing an authentic representation of PCG noise. Isolating these sections results in a total combined duration of 73 minutes and 48 seconds, primarily consisting of physiological noise.

The third database is LibriSpeech (Panayotov et al., 2015). The full database consists of around 1,000 hours of English-language speech, derived from read audiobooks, sampled at 16kHz and stored in .wav format. A duration of 5 hours, 23 minutes and 16 seconds was taken from this database to use as a different type of noise source for the PCG signal contamination part of this thesis. While this database is not closely related to heart sound

signals analysis as the previous two, it serves as a valuable source of non-stationary noise. This allows for simulating extreme noise conditions in PCG recordings, providing a rigorous test of the denoising method's robustness.

A further analysis into the PSD of the first two databases shows where the majority of the signal energy is located for each pathology along the frequency spectrum (Figures 3.8 and 3.9), this will be relevant later when analyzing the spectral contents of different types of noise.

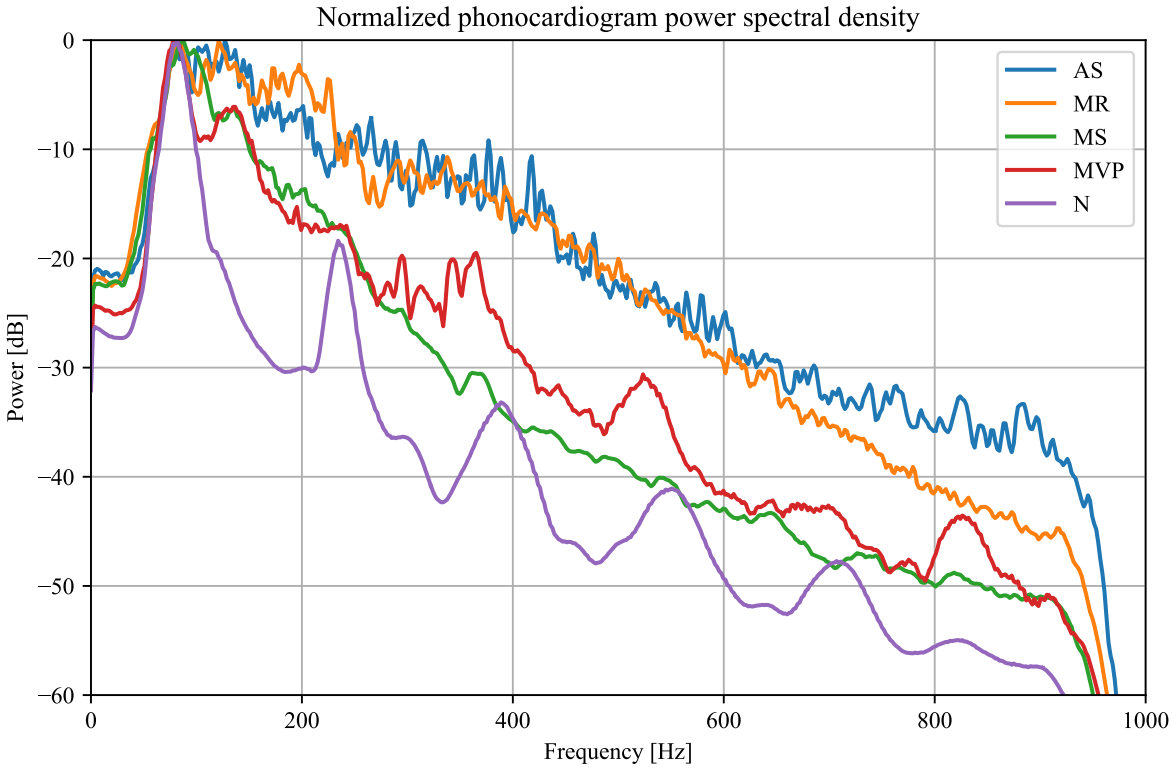


Figure 3.8 Distribution of the normalized power spectral density for the pathological cardiac sounds: aortic stenosis, mitral regurgitation, mitral stenosis, murmur in systole, and normal PCGs, averaged over 200 audios per class.

3.6.2. Synthetic noise sources

To ensure an effective method for denoising phonocardiograms, it is essential that the approach is rigorously tested in conditions closely resembling real-world scenarios. Additionally, the denoising method must produce outputs of adequate quality to be able to accurately identify the pathology in the sound. Achieving both objectives requires carefully choosing the type of noise to be employed.

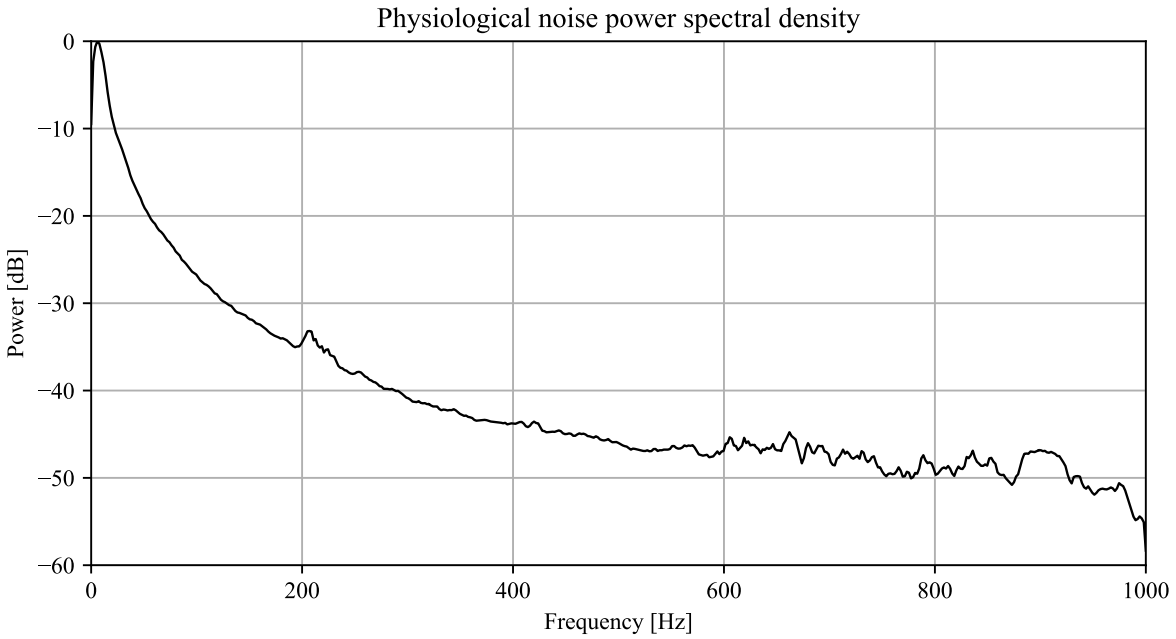


Figure 3.9 PSD of physiological noise extracted from the PhysioNet database (Liu et al., 2016).

Noise in the context of signal processing represents any altered component of the data sequence that does not match the original signal, in this case, the signal is the PCG. In PCGs, the noise is typically produced by four primary sources: physiological sounds, ambient noise, sensors, vocalizations (Gradolewski and Redlarski, 2014). Physiological sounds include noises from organs such as the lungs or the throat when swallowing. Ambient noise encompasses sounds from machinery and footsteps. Sensors noise can be caused by the movement of the devices used or the friction with the skin. Vocalizations include coughing and speech, among other sounds. Knowing what kind of noise can be found in real-world PCGs will allow a better construction of artificial noise to improve the method's performance.

Artificial noise can be generated with the following mathematical representation:

$$PSD_n(f) = \frac{C_\alpha}{|f|^\alpha}, \tag{3.26}$$

where f is the noise frequency, α defines which type of noise it represents and C_α is a constant for each type of noise. In particular, when $\alpha = 0$ the result is additive white noise, for pink noise $\alpha = 1$, for Brownian noise $\alpha = 2$, for blue noise $\alpha = -1$, and for violet noise $\alpha = -2$ (Sejdić and Lipsitz, 2013).

Figure 3.10 illustrates the power along the frequency spectrum for each type of noise; comparing with Figure 3.9, both the pink noise and the physiological noise match on a

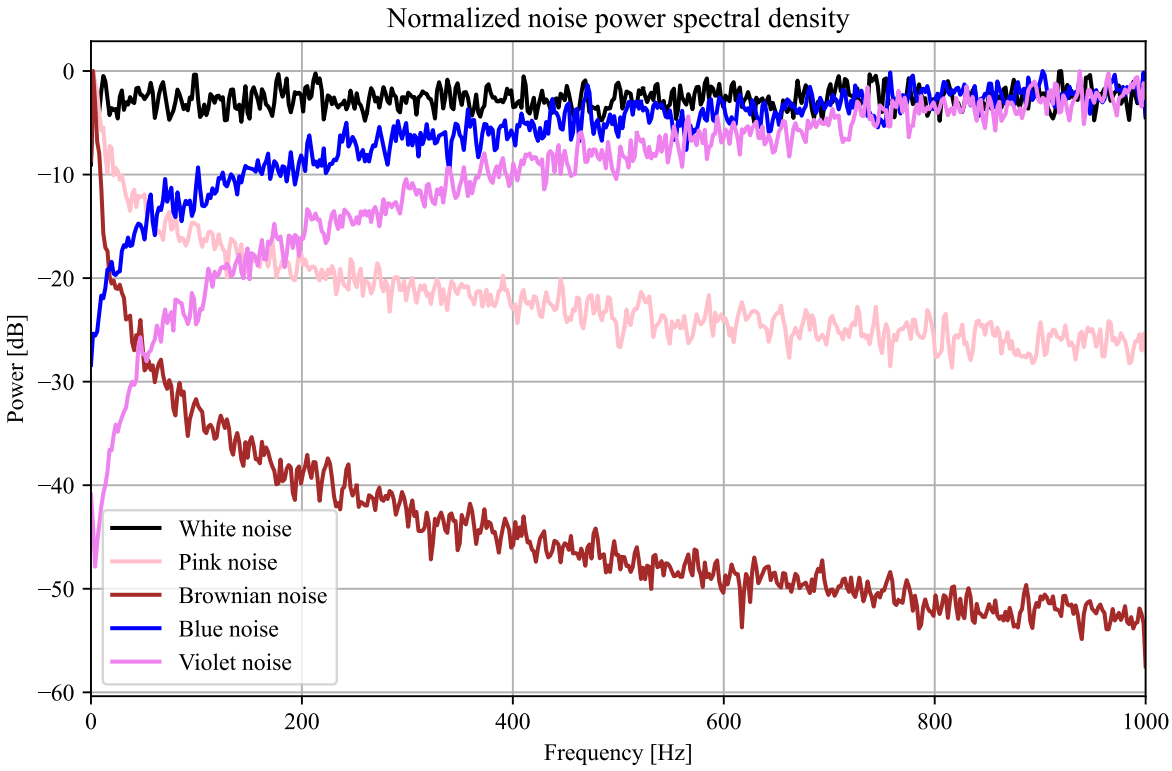


Figure 3.10 Distribution of white, pink, Brownian, blue, and violet noise normalized power spectral density sampled at 2 kHz.

predominance in power at the lower frequencies. Studies have also demonstrated that pink noise is often found in physiological processes (Sejdić and Lipsitz, 2013) and PCG denoising methods have also utilized pink and white noise to evaluate their models (Gradolewski et al., 2019; Gradolewski and Redlarski, 2014). In addition, the cardiac sound PSD also has the vast majority of its power on the lower frequencies (Figure 3.8), this makes the denoising task more challenging for filter methods, but also pink noise a good noise candidate to utilize.

3.7. Neural networks

In recent years, there have been noticeable advancements in the areas of artificial intelligence, and more specifically, machine learning. Deep learning, a branch of machine learning, focuses on the usage of artificial neural networks (Janiesch et al., 2021). The basics of the functionality of neural networks is on the multi-layered structure with weighted cells that are calibrated, during the training phase, to yield a probabilistic output from a given input and the network training. Each layer decomposes the input into different representations that can highlight different features of the given input, the initial input is generalized as a tensor,

but can be a vector to represent data sequences, a matrix for a grid of numbers or grey-scale images, or a 3D tensor generally used for colored images (Chollet, 2021).

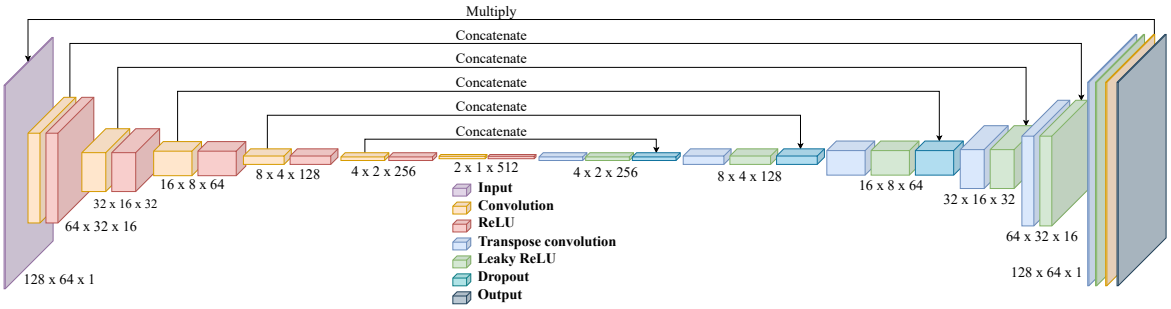


Figure 3.11 U-Net architecture taken from Hennequin et al. (2020), employed in this work, exemplified on a 128×64 input image dimensions.

Many different network architectures have been developed in order to tackle different kinds of problems, as seen in chapter 2. One architecture that is outstanding for the problem presented here is the U-Net architecture proposed by Ronneberger et al. (2015), its results exalt on biomedical image segmentation. This architecture consists of multiple layers which decompose, in this case a 2D array input, through multiple convolution, and max pooling operations, then multiple transposed convolution operations are carried out to retrieve the original image dimensions. In the context of image processing with neural networks, the input shape is typically defined as (B,H,W,C) where B is the batch size, H and W the height and width of the image respectively, and C the number of channels, for colored images, usually $C = 3$, while for grayscale images this is generally 1 and omitted (Bishop, 2006). Based on this architecture, applications in blind source separation problems have been developed, such as Andreas et al. (2017) and Hennequin et al. (2020). Its implementation aims to train a model to recognize a specific type of sound in a time-frequency representation, through the short-time Fourier transform, to isolate the different sources–instruments–conforming the music audio. The evaluation is carried out for the multiple datasets used and compared with previous state-of-the-art models.

3.7.1. Layers functionality

2-D Convolution

This architecture is composed of multiple key layers listed in Figure 3.11. The convolution layer consists of a 2-D convolution function; however, despite being named convolution, it

performs a cross-correlation of an input image $\mathscr{I} \in \mathbb{R}^{H \times W}$ with a kernel $\mathscr{K} \in \mathbb{R}^{k_h \times k_w}$:

$$(\mathscr{I} * \mathscr{K})(x, y) = \sum_{i=0}^{k_h-1} \sum_{j=0}^{k_w-1} \mathscr{I}(x+i, y+j)\mathscr{K}(i, j). \tag{3.27}$$

While the cross-correlation operation is generally defined with the complex conjugate of the function $\mathscr{I}$, each input used in this thesis is conformed of the magnitude of the time-frequency representation and therefore remains unchanged. The corresponding layer on the reconstruction second half, transpose convolution, is often referred to as deconvolution or inverse convolution layer; however, it does not perform the inverse of a convolution, the operation computed is the convolution of the transpose input function $\mathscr{I}^T$ with the kernel. The network training process consists of adjusting the different kernels from each layer to accurately predict a desired output. Each kernel selected has a specific size, a bigger kernel manages to capture more data points from the input at the same time at the cost of more computation time and resulting on a smaller output. Another parameter defined when computing this operation is the strides. The strides refer to how many steps the kernel advances through the input function, this can be defined for both the height and width along the input image, and likewise, the output size is reduced in the same way as the kernel affects it. Generally, the input image can be padded with zeros to have better control of the output dimensions and to compensate for the reductions the kernel size and strides introduce.

Activation function

A crucial component in the design of network architectures is the activation function. Most components of the neural network have a linear behavior; however, many real-world phenomena not necessarily show this, introducing the necessity for a method to predict non-linear behaviors—the main purpose of activation functions. Here the rectified linear unit (ReLU), leaky rectified linear unit (Leaky ReLU), and Sigmoid (not listed in Figure 3.11 but implemented only on the final 2-D convolution layer) activation functions are used (Figure 3.12), defined as:

$$\text{ReLU}(x) = \max(0, x), \tag{3.28}$$

$$\text{Leaky ReLU}(x) = \max(\alpha x, x), \tag{3.29}$$

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}, \tag{3.30}$$

taking as input the normalized output batch from the 2-D convolution and transpose 2-D convolution functions. The term α added to the Leaky ReLU function is usually a small constant value with the purpose of allowing negative values.

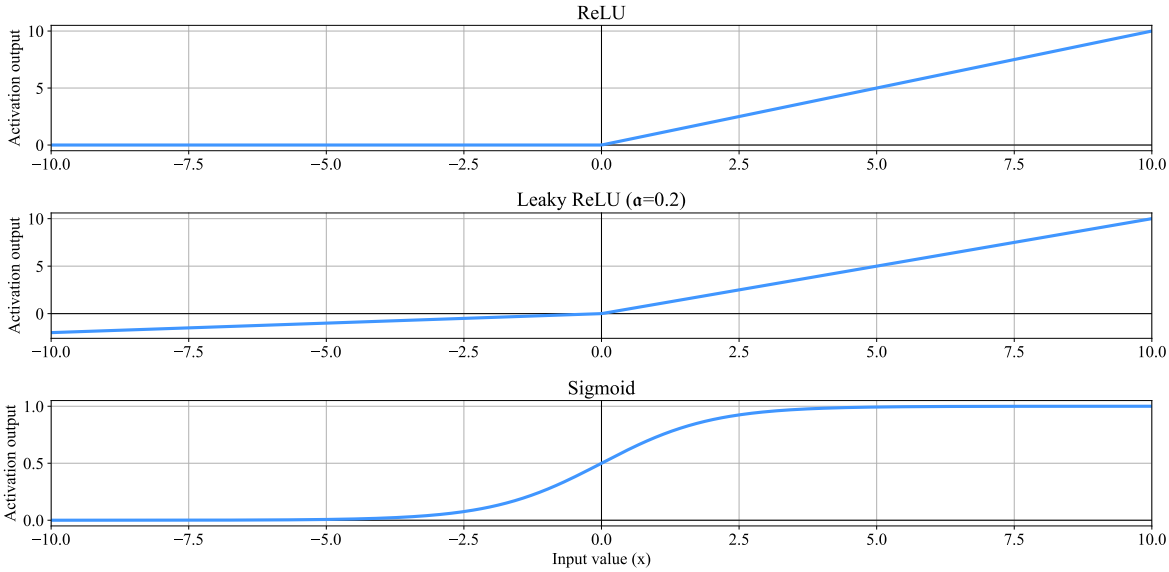


Figure 3.12 Activation functions examples.

Pooling

After running the input through the non-linear activation function, generally a pooling step follows, although not listed in Figure 3.11. In this step a new output is generated based on the nearby values around a pooling window throughout different locations of the input. The most common types of pooling are max pooling and average pooling, in which the output is based on the nearby max values or average values of the windowed input respectively. The main purpose of performing this operation is to downsample the previous output, modifying its size (Goodfellow et al., 2016).

Dropout

Throughout the training process of the neural network, the kernel weights are adjusted through each epoch with the aim of providing a prediction that not only minimizes the error for the given dataset, but generalizes to data not seen by the model to have a real-world application. A common issue presented during this process is overfitting, which is when the weights are over adjusted, focusing on very specific features from the network target images but fail to identify more general features. Dropout consists of dropping out neurons so the network does not rely on specific neurons to provide a good prediction. A parameter is introduced here, the dropout rate, which indicates the percentage of neurons dropped out after the specified layer (Chollet, 2021).

Tensor operations

The last operations presented in Figure 3.11 are concatenation and an element-wise multiplication. The concatenation operation combines two tensors along the channel axis, that is, if the tensors have shapes (B, H, W, C_1) and (B, H, W, C_2), the output would have the shape $(B, H, W, C_1 + C_2)$. In the element-wise multiplication each element in one tensor is multiplied by the corresponding from the other tensor. Since this is the final layer in the network, it is expected to filter and differentiate noise from the PCG audio, refining the input time-frequency image by highlighting relevant features.

Training metrics

During a neural network model training a loss function is selected to minimize the error between the target and training data. An accuracy metric is also often selected to evaluate the prediction by the model. For the loss function the mean absolute error was selected, defined as:

$$\text{MAE} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left| Y_{i,j} - X_{i,j} \right|, \tag{3.31}$$

where $Y_{i,j}$ represents each predicted data point by the model and $X_{i,j}$ each data point from the target data. Representing the model's accuracy requires a metric that evaluates the noise in the predicted image, for this the peak signal-to-noise ratio (PSNR) was chosen, defined as (Tabatabaeefar et al., 2020):

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right), \tag{3.32}$$

where MAX_I^2 represents the maximum possible pixel value in an image I, or matrix element in this case, given that the matrix elements are normalized from 0 to 1, $\text{MAX}_I^2 = 1$. The mean squared error MSE for one dimension is defined in Eq. (4.8), presented in the next chapter.

Chapter 4

Methodology

The overall methodology for training and evaluating each neural network involves several key steps. Initially, a dataset of PCG signals is randomly divided into a training and testing data set, 90% and 10% respectively with a controlled seed. The training dataset is first contaminated with various types of noise. Next, time-frequency representations of both the clean and noise-contaminated datasets are obtained. The neural network is then trained using these time-frequency representations. Following training, the network is evaluated on the testing dataset, which has not been exposed to the training data, to assess its performance. This is performed ten times to cross-validate the results, mitigate data bias, and verify the consistency of the results.

4.1. Noise contamination

Section 3.6 provides a general explanation of the databases from which the selected noise types are extracted and how synthetic noise is generated, including the reason behind their usage. This section continues with this topic, focusing on the generation and extraction of the selected noise types, as well as detailing their incorporation into the PCG signals.

4.1.1. Artificial noise

Given that the predominance in the PSD of PCG signals is on the lower frequencies below 400 Hz, from the artificial noise types only AWGN and APGN were selected. While Brownian noise could be a good candidate, in comparison with APGN its PSD falls off too quickly after 200 Hz. Its PSD is too similar to the one from physiological noise which would be redundant to use and therefore was discarded, meanwhile blue and violet noises have the complete opposite behavior and are not useful for this analysis.

From Eq. (3.26), $\alpha = 0$ for AWGN and $\alpha = 1$ for APGN. While their PSD is already represented in Figure 3.10. The amount of desired noise to contaminate a signal with can be calculated as follows. The signal power of a PCG discrete signal in time $s(t)$ is $P_s = \frac{1}{N}\sum^{N}|s(t)|^2$. The SNR is defined as $\text{SNR} = 10\log(P_s/P_\eta)$, the noise signal power with an arbitrary SNR difference between $s(t)$ and the noise $\eta(t)$ would be obtained by $10\log(P_\eta) = 10\log(P_s) - \text{SNR}$. Additionally, AWGN is obtained from a normal probability density distribution $p(t)$ where the variance σ^2 is the power of the noise signal, then, for an arbitrary SNR, the noise power can be calculated as:

$$\sigma^2 = P_\eta^{\text{SNR}} = 10^{\frac{10\log(P_s)-\text{SNR}}{10}}, \tag{4.1}$$

with the AWGN probability density distribution at a given SNR being:

$$p_{\text{SNR}}(\eta) = \frac{1}{\sqrt{2\pi P_\eta^{\text{SNR}}}} e^{-\frac{\eta^2}{2P_\eta^{\text{SNR}}}}. \tag{4.2}$$

Lastly, the resulting contaminated signal $s_\eta(t)$ is simply obtained by the sum of the original signal with the SNR-scaled noise signal $n_{\text{SNR}}(t)$:

$$s_\eta(t) = s(t) + \eta_{\text{SNR}}(t). \tag{4.3}$$

For pink noise, given a distribution obtained from Eq. (3.26), the noise power can be calculated from it and scale it to a desired SNR relative to $s(t)$. An important step here is to ensure that $\eta(t)$ is normalized to its power, which is done by dividing the signal by the square root of its power. Then multiplying $\eta(t)$ by the square root of the desired power at a given SNR $\sqrt{P_\eta^{\text{SNR}}}$, gives the desired SNR-scaled pink noise distribution. Example of signals contaminated at different SNR of AWGN are illustrated in Figure 4.1, while both AWGN and APGN with their respective spectrograms are presented in Figure 4.2.

4.1.2. Non-artificial noise

The PhysioNet database (Liu et al., 2016) is a phonocardiograms database that was recorded in a very noisy environment. This database being labeled makes it a good option to extract noise from that closely simulates a realistic scenario in which PCGs would be contaminated with ambient noise. Its PSD was already presented in Figure 3.9 and is similar to the PSD of pink noise from Figure 3.10, a noise that is often assimilated with physiological noise (Sejdić and Lipsitz, 2013).

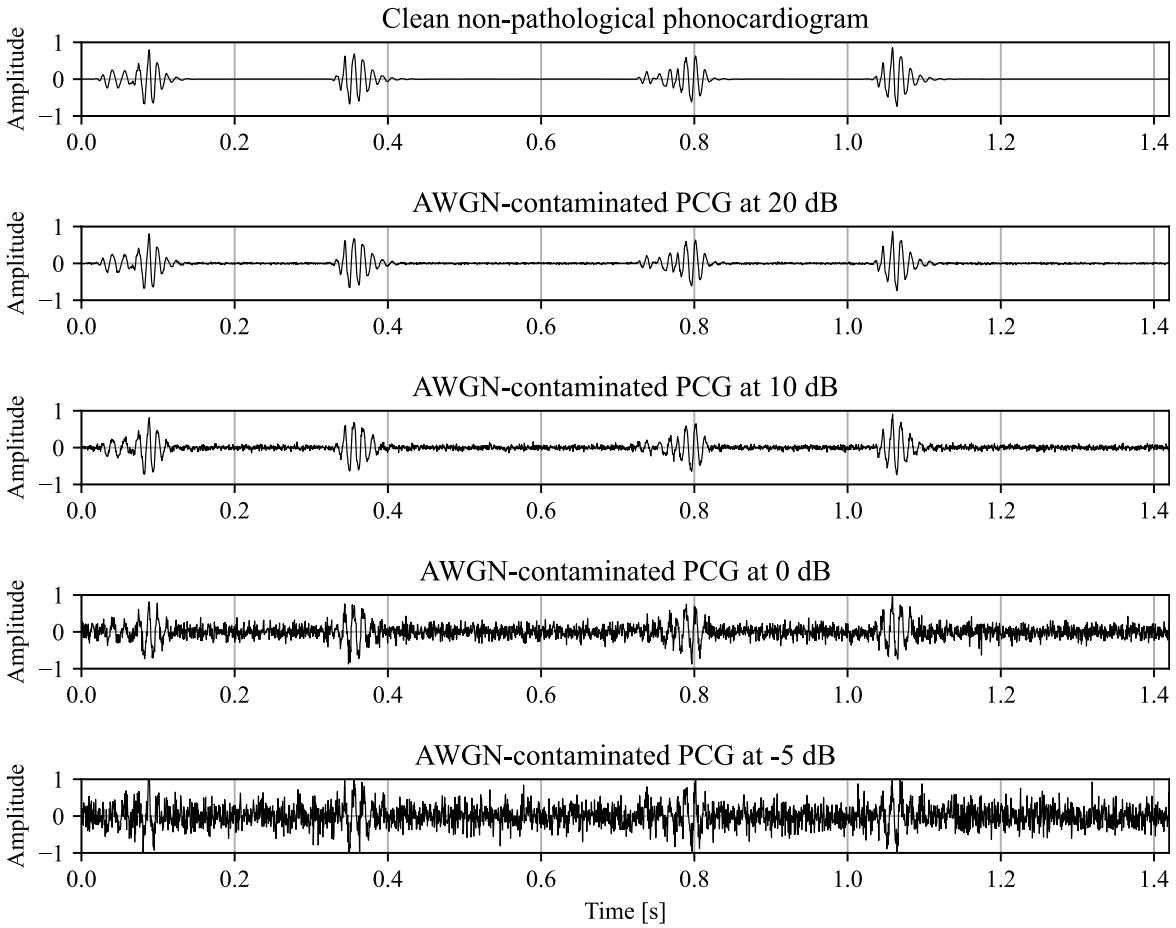


Figure 4.1 Example of a non-pathological PCG signal contaminated with AWGN at 20, 10, 0, and -5 dB of SNR.

The other noise proposed is speech noise, obtained from the LibriSpeech database (Panayotov et al., 2015). This database contains around 1,000 hours of English-language speech, and although PhysioNet noise already simulates a realistic real-world environment, it will serve as an example of an extremely non-stationary noise to reinforce the training of the network against all kinds of unpredictable noise.

The contamination process is similar to that of white and pink noise. The power from both signals is obtained $P = \frac{1}{N}\sum^{N}|s(t)|^2$, the noise signal is normalized with respect to its power $\frac{\eta(t)}{\sqrt{P_\eta}}$ and then scaled to the desired SNR relative to the clean PCG signal, the noise signal at an arbitrary SNR would then be calculated as follows:

$$\eta_{\text{SNR}}(t) = \eta(t)\sqrt{\frac{10^{\log(P_s)-\text{SNR}/10}}{P_\eta}}. \tag{4.4}$$

The respective extracted noises applied on a PCG signal are illustrated in Figure 4.2.

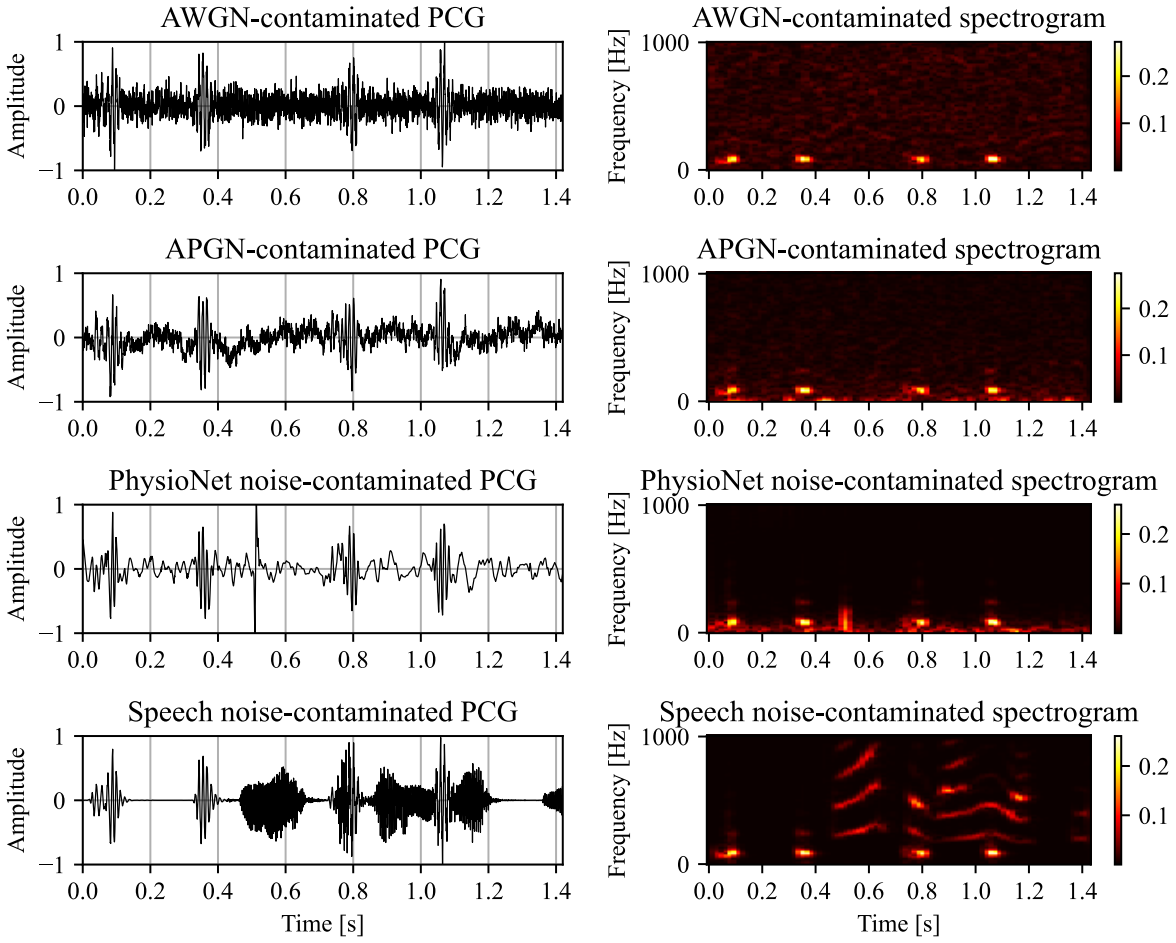


Figure 4.2 Example of a non-pathological PCG signal contaminated with AWGN, APGN, PhysioNet noise, and Speech noise at -5 dB of SNR and their respective spectrograms.

4.2. Signal processing

The signal processing component of the methodology is comprised of two primary parts: signal contamination and the generation of time-frequency representations. The types of noise utilized in the first part and the contamination process have been previously detailed; however, there are various ways by which the PCG signals can be contaminated with these four distinct types of noise. In addition, for the generation of time-frequency representations, four different techniques will be employed, each providing a distinct approach towards the signal representation for denoising performance comparison.

4.2.1. Signal contamination

While the noise contamination process was explained in the previous section, the noise arrangement is an important step that has a noticeable impact in the network training. This would include whether all signals are contaminated with one or multiple types of noise to train multiple or only one network, the randomness for each type of noise on combined noise types contamination, the way the PCG audios are arranged and contaminated.

4.2.2. Time-frequency representations

While generating the time-frequency representations—whether using the STFT, CWT, WSST, or *S*-transform—can be a straightforward process, two key considerations must be addressed. Firstly, the dimensions of the resulting time-frequency representation must be compatible with the input shape required by the neural network. Secondly, it is important to control the size of the resulting array, as processing large volumes of data can significantly increase computational complexity, thereby substantially slowing down the processing speed.

STFT partitions

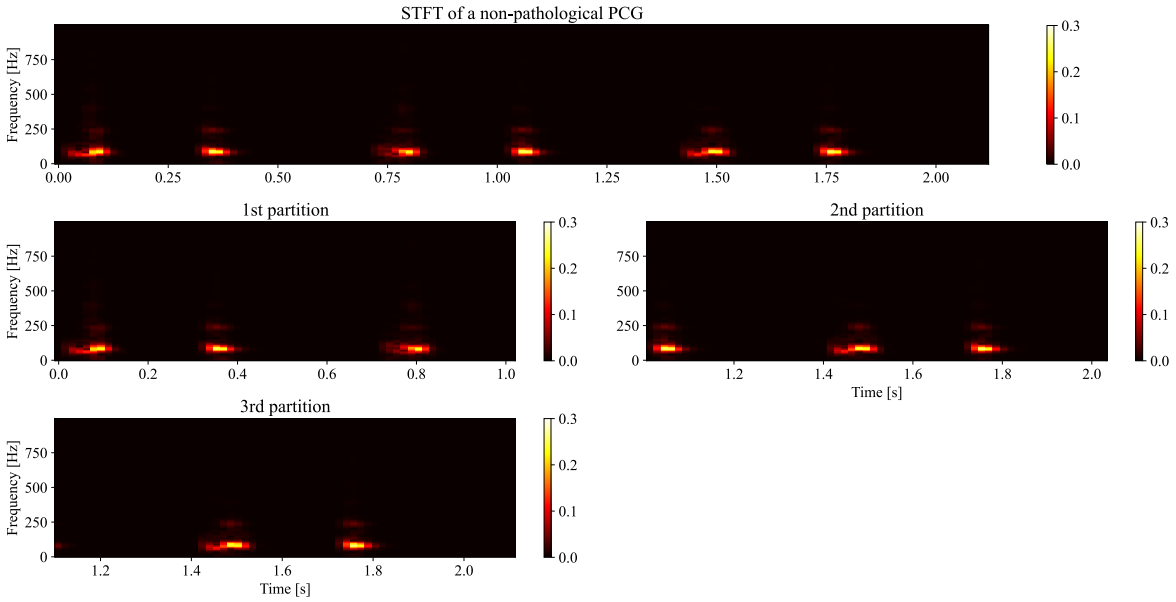


Figure 4.3 Partitions at a time window of 64 samples of the STFT from the 3 full cardiac cycles signal from Figure 2.1, with a sample rate of 2,000 Hz.

To address the first of these two requirements, the STFT representations are partitioned to fit the specific network input shape. For a given spectrogram of shape (K, M), the partitions

are carried out in the time axis from start to end of the spectrogram, leaving only the last partition from $M - W$ to M, where W is defined in §3.7. Depending on the SR, the STFT parameters, and the duration of each PCG the total number of partitions can vary. As an example, with the STFT parameters of 256 samples for the FFT size, hop length of 32 samples, and a window length of 128 samples, then taking a PCG with a duration of 2.105 seconds, sampled at 2,000 Hz, utilizing the expression for M in §3.2, the time dimension of the spectrogram would be of size 133, this would result in 2 full partitions, with an additional last partition taken from columns 69 to 133 for a network input shape of (128,64) (Figure 4.3).

Wavelet transform partitions

Both CWT and WSST are obtained in a similar way, this means defining the same parameters for the CWT will give a WSST representation with the same shape. For both of these transforms the time scale from the signal to scalogram is preserved to a 1:1 ratio, while for the frequency dimension it depends on the number of voices N_v. Choosing a high number of voices increases the frequency resolution and therefore the computational complexity, 8, 16, and 24 voices are a good middle point between resolution and computational complexity trade-off. Examples of partitions from the CWT and WSST of a signal are presented in Figures 4.4 and 4.5 respectively.

Besides defining a value for N_v to regulate the frequency resolution, the wavelet dictionary can be represented as a filter bank, in which one can have better control of the behavior of different wavelet scales as bandwidth filters. To generate a filter bank, the length of the wavelet is first defined, while this value generates a more detailed wavelet representation, it comes at a higher computational cost, for which it was kept low. Next, minimum and maximum scale values are generated in a way that they adjust for the best behavior of the chosen wavelet, from here the intermediate scales are generated with a logarithmic spacing in function of N_v. Most of these operations are carried out with the ssqueezepy python library for wavelet analysis and implementations of the CWT and WSST (Muradeli, 2020) .

S-transform partitions

As a result of the definition of the window from the *S*-transform, in theory there is a direct dependency between the time and frequency dimensions by the factor κ from Eq. (3.21); however, because it is implemented through the Fourier transform, the frequency bins will always be $N/2+1$, where N is the number of time samples of the signal. While the factor κ remains and influences the frequency resolution, for simplicity it will be taken as $\kappa = 1$.

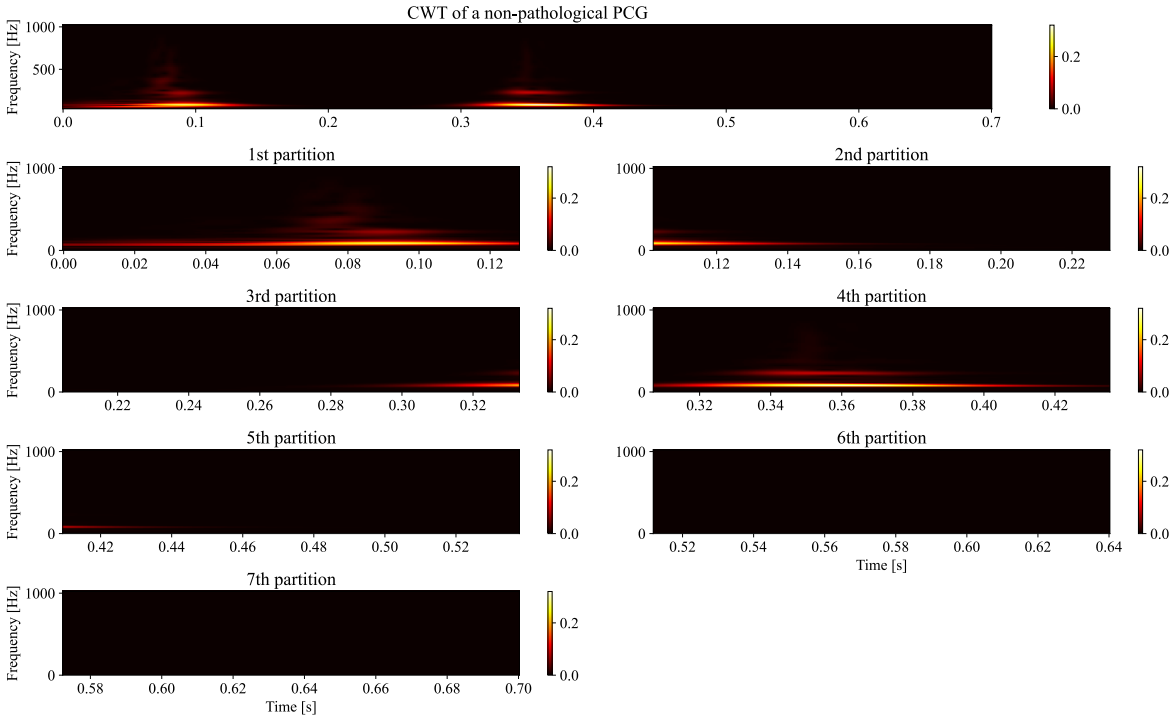


Figure 4.4 Partitions at a time window of 256 samples of the CWT from the first cardiac cycle signal from Figure 2.1, with a sample rate of 2,000 Hz and $N_v = 16$.

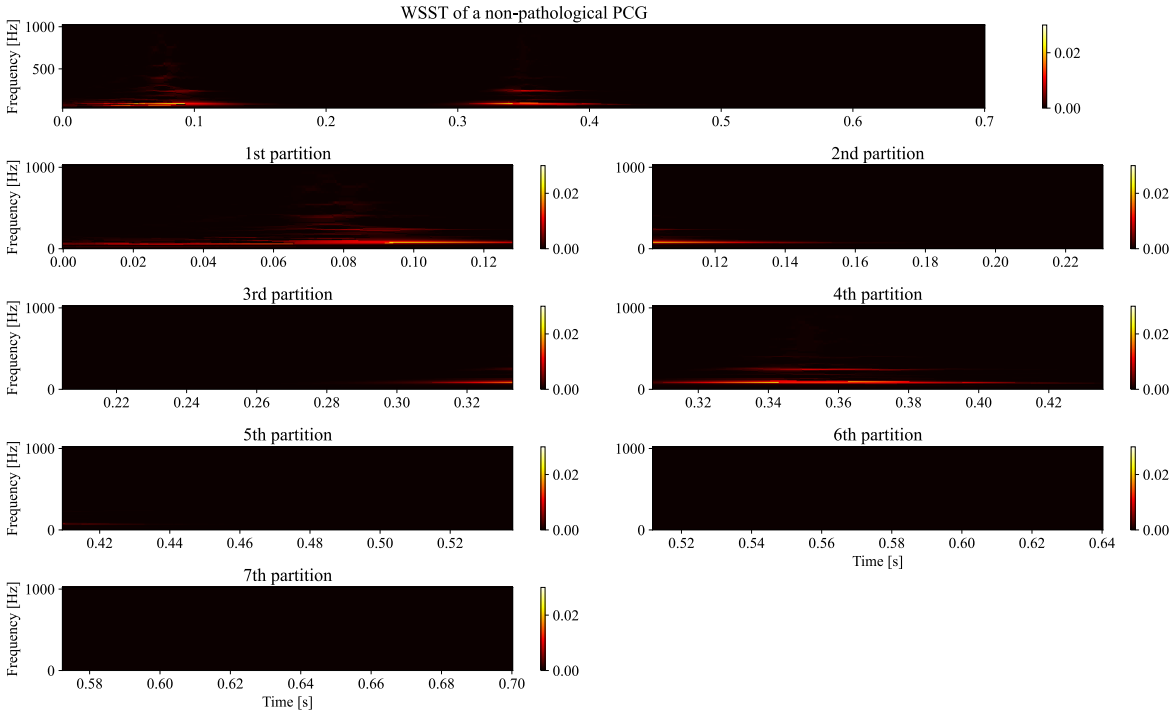


Figure 4.5 Partitions at a time window of 256 samples of the WSST from the first cardiac cycle signal from Figure 2.1, with a sample rate of 2,000 Hz and $N_v = 16$.

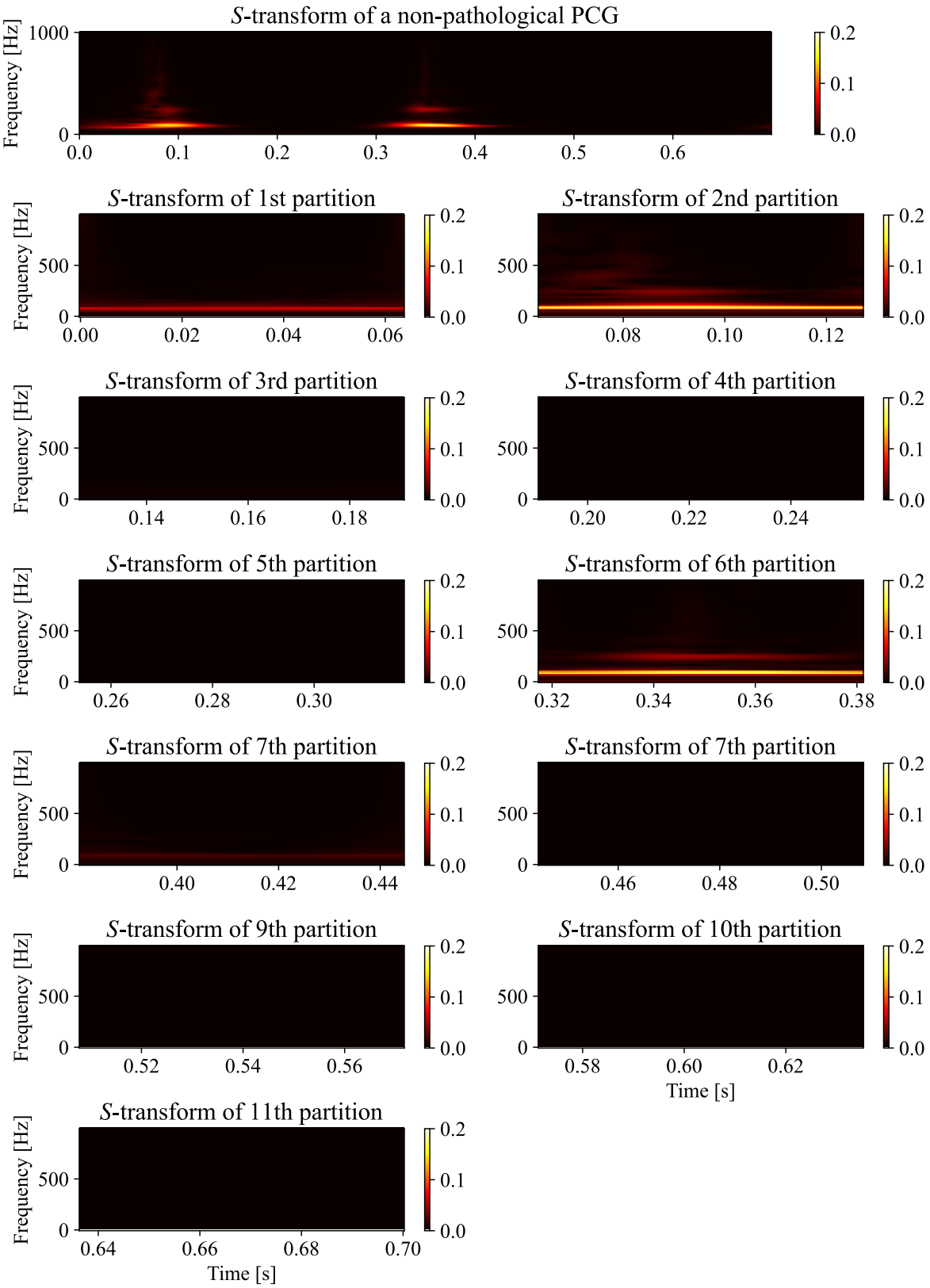


Figure 4.6 Partitions at a time window of 128 samples of the *S*-transform from the first cycle of the signal from Figure 2.1, with a sample rate of 2,000 Hz.

In summary, this means that the frequency dimension of the resulting matrix will only depend on the time dimension of the signal. Therefore, when applying the S-transform to a signal with duration t_1, the frequency resolution will be higher than for a signal with duration t_2 such that $t_1 > t_2$.

To carry out the partitions, the signal is segmented into time windows of 128 and 256 samples before performing the $S-$transform. This results in frequency windows of 65 and 129 samples, respectively, matching the required input shape for the network, which ideally should be divisible by 2. To achieve this, the highest frequency bin, corresponding to the Nyquist frequency, is removed. The loss of this bin is negligible since, due to the original signal's sampling process, the Nyquist frequency does not contain significant information necessary for either frequency mapping or signal reconstruction. Examples of partitions with these time windows are illustrated in Figures 4.6 and 4.7.

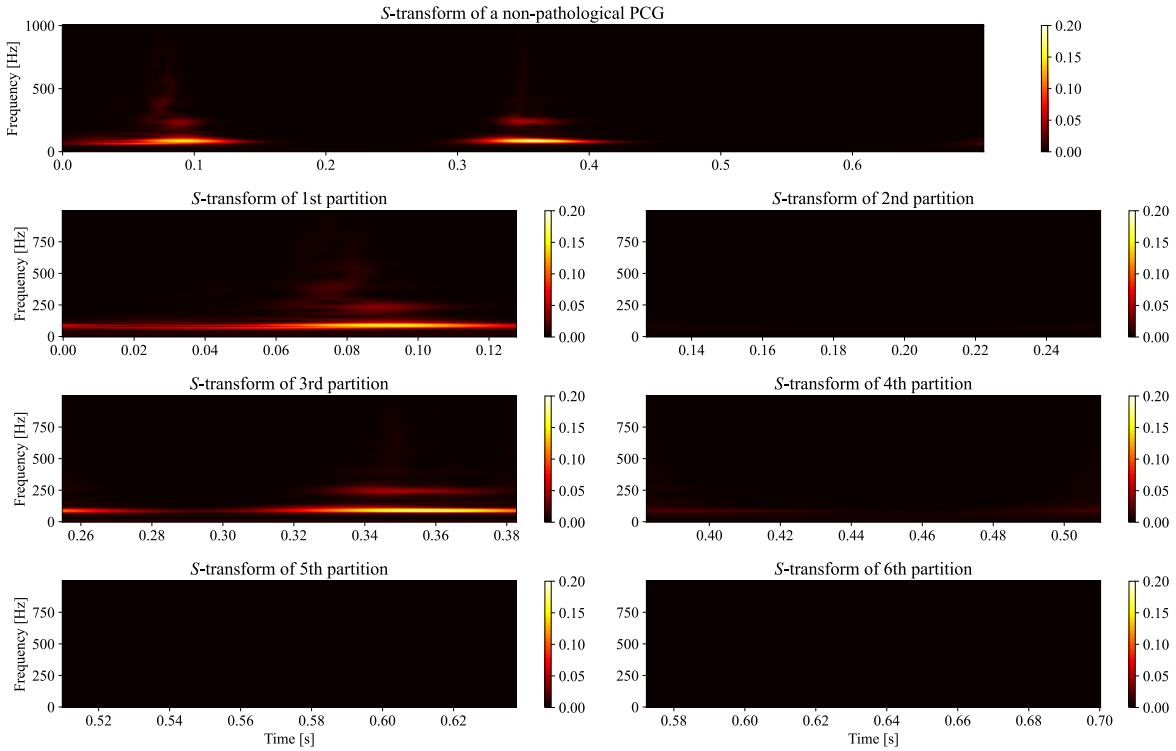


Figure 4.7 Partitions at a time window of 256 samples of the S-transform from the first cycle of the signal from Figure 2.1, with a sample rate of 2,000 Hz.

When comparing both figures, while the first figure, where the signal is divided into 11 partitions, may suggest that more data is being processed, each matrix has dimensions of 64×128, capturing nearly the same amount of time-domain samples as the six matrices from Figure 4.7, which have dimensions of 128×256. However, the frequency resolution in the

second figure is twice as high as that of the first, indicating that the second set of matrices involves more data processing due to the increased frequency dimension.

4.3. Data augmentation

A common problem in deep learning is the limited amount of data available to carry out the neural network training, an extensive database generally provides a better training. Although in this work the database used only has 1,000 audio files, from which 900 are used for training, the neural network actually receives more data as an input given that each spectrogram and scalogram is partitioned. The CWT, WSST, and *S*-transform preserve the time resolution of the time signal which results in a considerable amount of images available to train the network with; however, the STFT provides a significant less amount of partitions per PCG recording which can result into a network with poor training.

A novel method in PCG signal processing is proposed to artificially increase the dataset. It leverages the established Analysis-by-Synthesis/Overlap-Add (ABS/OLA) technique, as outlined by Macon and Clements (1996) and George and Smith (1997). This technique is grounded in a sinusoidal model, enabling high-quality pitch modification of sound signals. The ABS/OLA model represents the input audio signal, $s(n)$, as a sum of overlapping short-time signal frames, $s_k(b)$, mathematically expressed as:

$$s(n) = \lambda(n) \sum_k w(n - kN_x) s_k(n). \tag{4.5}$$

Here, N_x is the frame length, $w(n)$ is a window function, $\lambda(n)$ is a time-varying gain envelope, and $s_k(n)$ is the k-th frame contribution to the synthesized signal. Each signal frame, $s_k(n)$, is described by a sum of sinusoidal components:

$$s_k(n) = \sum_{l=0}^{L-1} A_l^k \cos\left(\omega_l^k n + \phi_l^k\right), \tag{4.6}$$

where L is the number of sinusoidal components, and $A_l^k, \omega_l^k, \phi_l^k$ are the sinusoidal parameters for amplitudes, frequencies, and phases, respectively. The synthesis process is performed using inverse fast Fourier transform (FFT), allowing pitch shifting by altering these components without significantly distorting the original signal's structure. In this thesis, ABS/OLA was implemented using the PySox library to augment the PCG dataset. The pitch of each recording was randomly shifted within a ± 10 Hz range, increasing the original dataset from 1,000 to 5,000 PCG recordings. This range was heuristically selected to ensure realistic

sound without introducing artifacts, maintaining the integrity of the augmented data. Pushing forward this method and further increasing the dataset beyond this amount was considered too risky to avoid running into over-fitting problems or compromising the network's ability to generalize due to a high amount of similar PCG sounds.

4.4. Network processing

After obtaining the time-frequency representations of both the clean and noise-contaminated PCG recordings, the network training can be carried out. The labeled data allows for a supervised type of learning where the network receives one on one feedback for each image processed. The network training is comprised of 900 PCG recordings in the case of no data augmentation and 4,500 with data augmentation, depending on the time-frequency representation, this corresponds to a range in between 20,000 to 80,000 images.

4.4.1. Network input

While section 4.2.2 already describes how each spectrogram and scalogram is partitioned, it is important to remark the shape each input image has. Figure 4.8 illustrates partition examples for the four different transforms with subdivisions for each Heisenberg box. It is important to note that, in a standard image, pixels are uniformly spaced, resulting in equal weighting for each pixel. However, due to the wavelet scaling, which produces varying Heisenberg box shapes throughout the scalogram, there is a notable emphasis on lower frequencies where the majority of the PCG sound is concentrated. Unlike the STFT, which has its boxes uniformly spaced along the frequency axis, using the wavelet-based approach matrix as the input image results in these low frequency boxes stretching, enhancing the analysis of this frequency range during the network training.

From an input shape of the form (B, H, W, C), defined in §3.7, the batch size is taken as 64 and all matrices can be considered as single-channel images. Each image then has a shape (H, W), where H corresponds to the frequency axis and W to the time axis. The values for H and W vary for each transform and are the following:

- STFT: $H = 128$, $W = 64$
- CWT/WSST: $H = 64, 128$, $W = 64, 128, 256$
- S-transform: $H = 64, 128$, $W = 128, 256$

These values are a result of the parameters chosen for each transform, adjusted where needed to fit a power of 2.

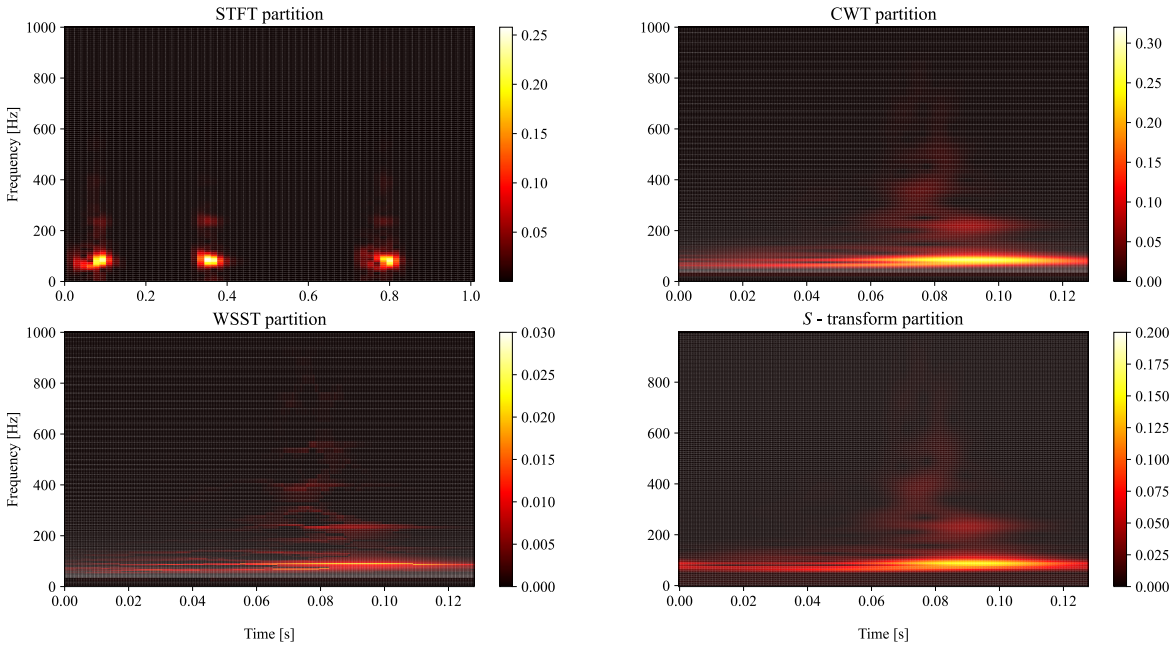


Figure 4.8 Partition examples with their respective Heisenberg box subdivisions. The STFT partition corresponds to a shape of $(128, 64)$, both CWT and WSST partitions have a shape of $(128, 256)$, and the S-transform partition has a shape of $(128, 256)$. This figure can be appreciated better in the digital version of this document which allows for magnification.

4.4.2. Architecture

Based on the type of problem desired to approach, network architectures are adapted to optimize the performance outcomes. This is the case for U-Nets as well, while different architectures have been designed to tackle a variety of problems, such as in Andreas et al. (2017); Hennequin et al. (2020); Ronneberger et al. (2015), to prioritize time efficiency and focus on exploring different time-frequency representations, the architecture proposed by Hennequin et al. (2020) was adopted in this thesis due to its application similarity, remarkable results and ease of implementation (Figure 4.9). This architecture, originally designed for blind source separation (BSS) tasks, aims to separate distinct instruments from music tracks. The network was trained using time-frequency representations derived from the STFT, achieving state-of-the-art performance.

This architecture is composed of six 2-D convolution layers with a kernel size of 5×5, strides of 2×2, each followed by zero padding and a leaky rectifier linear unit (Leaky ReLU) layer with $\alpha = 0.2$. The second part of the architecture is conformed of six transpose 2-D convolution layers with a kernel size of 5×5, strides of 2×2, zero padding, a Leaky ReLU layer, and a concatenation operation with its respective convolution output as shown in Figure 3.11, the first three transpose convolution layers have a dropout rate of 50% before

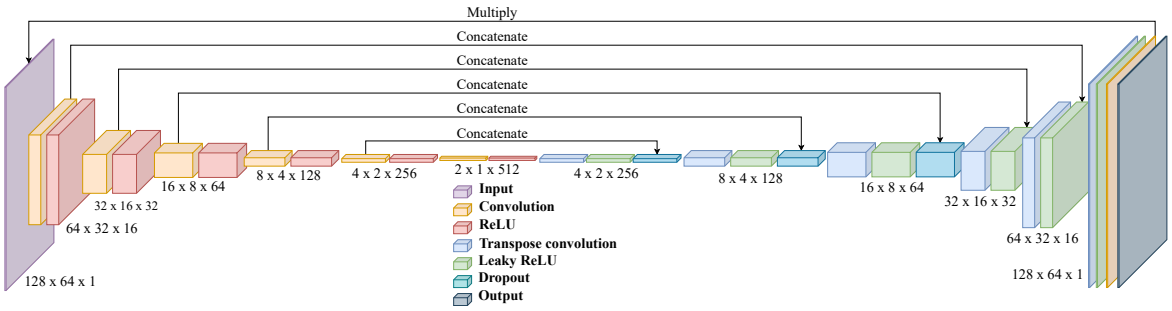


Figure 4.9 U-Net architecture taken from Hennequin et al. (2020), employed in this work, exemplified on a 128×64 input image dimensions.

the concatenation operation, a final layer then follows with a kernel size of 4×4, dilation rate of (2,2), and a sigmoid activation function. The output is expected to retrieve a mask with a weighted signaling of the noise, doing an element-wise multiplication by the input results in the denoised time-frequency representation.

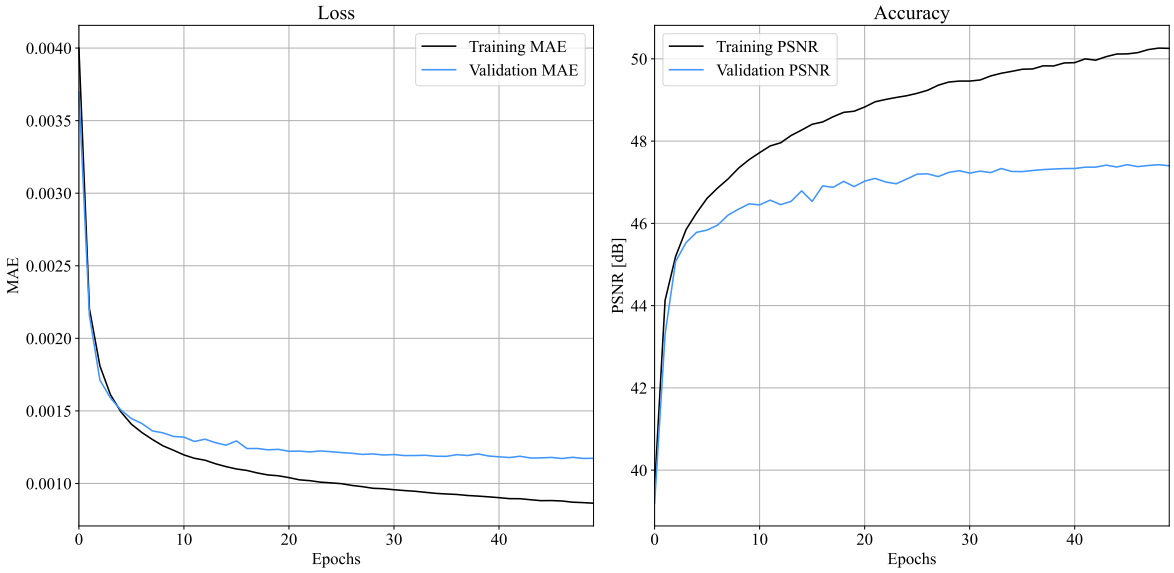


Figure 4.10 Loss and accuracy curves example obtained for the CWT-trained model at 50 epochs.

4.4.3. Network training

Employing the training metrics discussed in §3.7.1, loss and accuracy curves were obtained in which visual monitoring was conducted to ensure the training phase proceeded without issues, here the training parameters were arbitrarily adjusted to reach specific values that overall minimized the loss and maximized the accuracy. Figure 4.10 presents an example

of the MAE and PSNR curves obtained for 50 training epochs. Since the validation curves for both loss and accuracy showed no significant improvement beyond approximately 20 epochs, the training was concluded at 25 epochs.

4.5. Evaluation metrics

4.5.1. Objective denoising metrics

Common metrics used for evaluating sound denoising, including those specifically applied to PCG denoising in the literature, often involve the comparison between the clean signal s and the method-processed denoised signal $\tilde{s}$, both comprising N samples.

Signal-to-noise ratio

Among these metrics, the signal-to-noise ratio (SNR) is frequently utilized (Benesty et al., 2011):

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\|s(n)\|^2}{\|s(n)-\tilde{s}(n)\|^2}\right)\ \mathrm{dB}, \tag{4.7}$$

this is also commonly referred to as signal-to-distortion ratio (SDR), is a relative comparison between the original signal with the difference between both signals, scaled logarithmically.

Mean squared error

The mean squared error (MSE) (Aloorravi, 2024):

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}\left[s(n)-\tilde{s}(n)\right]^2, \tag{4.8}$$

is a metric that estimates the squared difference between each data point between both signals. Based on this there are a few similar metrics: including the normalized mean squared error (NMSE):

$$\mathrm{NMSE} = \frac{\sum_{n=1}^{N}\left[s(n)-\tilde{s}(n)\right]^2}{\sum_{n=1}^{N}\left[s(n)\right]^2}. \tag{4.9}$$

While the MSE can vary depending on the amplitude of the signal, the NMSE being relative to the power of the clean signal makes the comparison between both signals more consistent;

the root mean squared error (RMSE) (Aloorravi, 2024):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left[s(n)-\tilde{s}(n)\right]^2}, \tag{4.10}$$

is a metric similar to the MSE, it makes emphasis on larger errors due to the root operation applied to it and can be interpreted as a standard deviation; the normalized root mean squared error (NRMSE):

$$\text{NRMSE} = \sqrt{\frac{\sum_{n=1}^{N}\left[s(n)-\tilde{s}(n)\right]^2}{\sum_{n=1}^{N}\left[s(n)\right]^2}}, \tag{4.11}$$

provides the normalized version the RMSE, normalizing it makes it more appropriate for comparison where the signal power could have notable variations; the percentage root mean square difference (PRD):

$$\text{PRD} = \sqrt{\frac{\sum_{n=1}^{N}\left[s(n)-\tilde{s}(n)\right]^2}{\sum_{n=1}^{N}\left[s(n)\right]^2}} \times 100, \tag{4.12}$$

is a metric in which the point-to-point error resulting from the NRMSE is expressed as a percentage.

Fit coefficient

This metric was proposed by Gradolewski and Redlarski (2014) to evaluate PCG denoising:

$$\text{fit} = 100 \times \left(1 - \frac{\sum_{n=1}^{N}\left[\tilde{s}(n)-s(n)\right]^2}{\sum_{n=1}^{N}\left[s(n)-(1/n)\sum_{n=1}^{N}s(n)\right]^2}\right), \tag{4.13}$$

it calculates the ratio between the sum of squared differences and the total variance, expressed as a percentage

Pearson correlation coefficient

The Pearson correlation coefficient is defined as:

$$r = \frac{\sum_{n=1}^{N}\left[s(n)-\bar{s}\right]\left[\tilde{s}(n)-\bar{\tilde{s}}\right]}{\sqrt{\sum_{n=1}^{N}\left[s(n)-\bar{s}\right]^2}\sqrt{\sum_{i=1}^{N}\left[\tilde{s}(n)-\bar{\tilde{s}}\right]^2}}, \tag{4.14}$$

where the bar symbol ¯ represents the mean, and the function r is defined in the interval of [-1, 1], with $r = -1$ representing a perfect negative correlation, $r = 0$ as having no linear correlation, and $r = 1$ perfect positive correlation.

Scale invariant signal-to-distortion ratio

The scale invariant signal-to-distortion ratio (SI-SDR), proposed by Le Roux et al. (2019):

$$\text{SI-SDR} = 10\log_{10}\left(\frac{\left\|\frac{\tilde{s}^T(n)s(n)}{\|s(n)\|^2}s(n)\right\|^2}{\left\|\frac{\tilde{s}^T(n)s(n)}{\|s(n)\|^2}s(n)-\tilde{s}(n)\right\|^2}\right)\ \text{dB}, \tag{4.15}$$

describes a scale invariant logarithmic comparison between the signal and the distortion between both signals, unlike the SNR or SDR, this metric is invariant to signal scaling discrepancies that may arise when adding noise to the signal. This invariance against signal amplitude variations makes the evaluation more robust, ensuring that the metric reflects the true distortion unaffected by scaling inconsistencies.

4.5.2. Subjective denoising metrics

During the development of denoising or classification methods applied in the healthcare industry, one of the final and most important steps is the validation by an expert individual, in this case, an expert cardiologist with the adequate training to identify pathologies in heart sound signals. While it would be ideal to validate the denoising method, it is out of scope for this thesis and further analysis and evaluation relies solely on objective denoising metrics.

4.5.3. Denoising metrics selected

The metrics selected for the evaluation of this method are the SI-SDR, given its robustness over the commonly used SNR. The SI-SDR metric offers a more accurate representation of noise characteristics in sound denoising applications than traditional error metrics such as MSE or RMSE, defined in Eqs. (4.8) and (4.10) respectively.

Furthermore, while the SI-SDR metric assesses the distortion between a scaled version of the original signal and the residual distortion in the noise-processed signal, it does not adequately capture the denoising performance across various frequency ranges. To address this limitation, the PSD of the original, noise-contaminated, and noise-processed signals is computed. This analysis facilitates a visual comparison across the frequency spectrum,

highlighting the extent of noise reduction and the degree to which the noise-processed signal approximates the original signal.

Lastly, the Pearson correlation coefficient is calculated for the final results to provide an additional measure of the linear relationship between the original and processed signals.

Chapter 5

Results

5.1. STFT-based denoising

The first step in developing the PCG denoising algorithm involved utilizing the STFT representation. This representation has been proven to be effective at blind source separation (BSS) tasks in which a U-Net architecture is employed. The database provided by Son and Kwon (2018), originally consisting of 1,000 PCG recordings, was expanded through the data augmentation technique described in §4.3, resulting in a total of 5,000 recordings, each sampled at a rate of 4,000 Hz. Of this dataset, 90% was allocated for training and 10% for testing, ensuring that both subsets contained a proportional distribution of each PCG pathology type. During the training phase, 4,500 PCG recordings were artificially contaminated with various types of noise, including additive AWGN, APGN, PhysioNet noise, and speech noise at a SNR of 0 dB, giving the noise the same amount of energy as the PCG signal. Their time-frequency representations were then computed and split into respective partitions of shape $(128, 64)$ for the image height and width respectively, yielding approximately 24,250 partitions per iteration for network training.

For evaluation, the remaining 500 PCG recordings were similarly contaminated at SNRs of -5, 0, 5, and 10 dB, and the trained network model was applied to process their time-frequency representations and remove the noise. The SI-SDR was then calculated between the original and noise-processed signals. This entire process, including both training and evaluation, was repeated over 10 iterations, with the dataset randomly rearranged using a different seed for each iteration. The average SI-SDR values across the 10 iterations for each type of noise and noise level are presented in Tables 5.1, 5.2, 5.3, and 5.4 respectively.

Table 5.1 Average SI-SDR performance across 10 AWGN-trained networks for four different noise types at -5, 0, 5, and 10 dB.

AWGN Training				
Noise\SNR (dB)	**-5**	**0**	**5**	**10**
AWGN	9.3890	13.4112	17.2102	20.7089
APGN	-0.5680	4.3751	9.2031	13.2146
PhysioNet noise	-4.9312	0.0970	5.1318	10.1677
Speech noise	-4.9738	0.0968	5.1960	10.2631

Table 5.2 Average SI-SDR performance across 10 APGN-trained networks for four different noise types at -5, 0, 5, and 10 dB.

APGN Training				
Noise\SNR (dB)	**-5**	**0**	**5**	**10**
AWGN	1.7585	9.3568	15.2897	19.1538
APGN	7.2792	11.2817	14.8853	18.1790
PhysioNet noise	-1.8340	3.1566	7.9598	12.5518
Speech noise	-5.1485	0.1960	5.7840	11.5848

Table 5.3 Average SI-SDR performance across 10 PhysioNet noise-trained networks for four different noise types at -5, 0, 5, and 10 dB.

PhysioNet noise Training				
Noise\SNR (dB)	**-5**	**0**	**5**	**10**
AWGN	-5.1809	0.3959	5.8445	10.9261
APGN	-0.4906	5.0125	10.0556	14.5618
PhysioNet noise	11.4415	14.7594	17.2943	19.4510
Speech noise	-5.5146	-0.1229	5.0278	10.0635

Table 5.4 Average SI-SDR performance across 10 Speech-trained networks for four different noise types at -5, 0, 5, and 10 dB.

Speech Training				
Noise\SNR (dB)	**-5**	**0**	**5**	**10**
AWGN	6.2885	10.8868	14.6264	18.1483
APGN	2.8917	7.5144	11.4857	15.3985
PhysioNet noise	2.4587	7.1001	10.8411	14.6463
Speech noise	13.1278	15.9485	18.7854	21.6813

5.1.1. Combined noise

An analysis of the previous tables reveals several noteworthy results. In Table 5.1, where the network was trained using AWGN, its denoising performance for AWGN-contaminated signals is significantly higher than for the other three types of noise. Although the SI-SDR values remain consistent for both PhysioNet and speech noise, there is a notable improvement of approximately 3 to 5 dB when denoising signals contaminated with APGN.

In the case of training with APGN (Table 5.2), while this model does not perform well when denoising signals are contaminated with speech noise, it still shows a notable improvement in denoising AWGN and PhysioNet noise, the latter being particularly relevant, as it is commonly found in noisy PCG recordings. When the model was trained with PhysioNet noise (Table 5.3), the denoising performance was effective not only for PhysioNet noise itself but also for APGN, though the model struggled to denoise signals contaminated with AWGN and speech noise.

Table 5.4, which corresponds to the training using speech noise, presents the best overall performance. While the denoising performance for each noise type is not outstanding, the model is capable of addressing all four noise types to some extent. Given the non-stationary and unpredictable nature of speech noise, this suggests that training with this type of noise enabled the network to focus more on identifying PCG signals within the time-frequency representations, rather than learning to suppress a specific type of noise, as appeared to be the case in the other training scenarios.

These observations led to the development of a combined noise training strategy. In this approach, during the training phase, PCG signals were randomly contaminated with all four types of noise, each with equal probability. The goal of this combined training was to strengthen the network's ability to identify patterns associated with PCG signals and their pathologies, rather than overfitting to a particular noise type.

Table 5.5 Average SI-SDR performance across 10 combined noise-trained networks for four different noise types at -5, 0, 5, and 10 dB.

Combined Noise				
Noise\SNR (dB)	**-5**	**0**	**5**	**10**
AWGN	9.1611	13.2334	16.9955	20.4679
APGN	6.8464	11.0278	14.6008	17.8378
PhysioNet noise	12.6011	15.7543	18.1299	20.0806
Speech noise	12.1227	15.2298	18.2220	21.0995

Table 5.5 presents the results obtained from the combined noise training approach. A significant improvement in denoising performance is observed across all four noise types, with the highest performance seen in the removal of PhysioNet noise. While these results demonstrate an average SNR increase of 11 to 17 dB at the noisiest levels, they were achieved using a pre-existing network architecture with the STFT time-frequency representation. Several strategies could be explored to further enhance this method, such as experimenting with alternative network architectures, adjusting the training parameters, or utilizing other time-frequency transforms that may provide a more precise representation of the PCG signal. This thesis focuses on the latter approach, investigating the potential of different time-frequency representations to improve denoising performance.

5.2. Comparative denoising across time-frequency representations

The implementation of alternative time-frequency transforms involved introducing the CWT, WSST, and the *S*-transform. These transforms offer improved representation sharpness at the cost of higher computational demands. As a result, the network was trained on the original dataset of 1,000 PCG recordings, sampled at 2,000 Hz, maintaining the 90% training and 10% testing split. Since using the augmented dataset in the previous experiment could have enhanced the denoising evaluation, the STFT-based evaluation was repeated with this smaller dataset for consistency. Separate networks were trained for each of the partition shapes defined in §4.4.1, aiming to determine which transform yields the best performance and to identify the optimal parameters for maximizing denoising efficacy. Given the time that this process would take from the extended amount of models to be trained and the focus here is to evaluate the model in highly noisy scenarios, only SNRs of -5 and 0 dB were evaluated in the following stage.

5.2.1. Short-time Fourier transform

The STFT-based training was repeated without data augmentation to establish a reference point for comparison with the three newly introduced time-frequency representations. Furthermore, a comparison of Tables 5.6 and 5.5 highlights the significant impact of the database expansion, particularly in processing PCG signals contaminated with speech noise.

The results presented were obtained using STFT partitions with the same input shape as the previous STFT network training, yielding approximately 2,670 partitions in total. This

Table 5.6 Average SI-SDR performance across 10 STFT-trained networks for four different noise types at -5 dB and 0 dB.

STFT Processing		
Noise\SNR (dB)	**-5**	**0**
AWGN	4.6350	8.9248
APGN	4.4806	8.5175
PhysioNet noise	9.9808	12.4665
Speech noise	6.9246	11.0371

Table 5.7 Average correlation across 10 STFT-trained networks for four different noise types at -5 dB and 0 dB.

STFT Processing		
Noise\SNR (dB)	**-5**	**0**
AWGN	0.9048	0.9578
APGN	0.9083	0.9601
PhysioNet noise	0.9658	0.9851
Speech noise	0.9360	0.9688

contrasts with the 24,250 partitions generated from the data-augmented training, underscoring the influence of training dataset size on performance.

5.2.2. Wavelet transform

For both the CWT and WSST trainings, time windows of 64, 128, and 256 samples were used, yielding approximately 86,200, 43,200, and 21,700 resulting input images, respectively. These were combined with N_v values of 24, 16, and 8 voices, corresponding to frequency windows of 128 samples for the first two configurations and 64 samples for the third. Each combination of time and frequency windows produced a trained model, which was evaluated across all noise types at -5 dB and 0 dB SNR (Tables 5.8, 5.9, 5.10, and 5.11).

Both the CWT-trained and WSST-trained models demonstrated improved denoising performance compared to the control STFT model (Table 5.6), and in some cases even outperformed the data-augmented STFT model (Table 5.5). While the improvements in denoising PCG signals contaminated with AWGN and APGN were moderate, the most significant gains were observed for signals contaminated with PhysioNet and speech noise. Specifically, there was an increase of approximately 5 to 8 dB for PhysioNet noise and around 3.5 to 7 dB for speech noise.

5.2.3. *S*-transform

In the final proposed time-frequency transform, the *S*-transform utilizes input shapes of $(64, 128)$ and $(128, 256)$, where the height corresponds to the frequency window and the width to the time window. These configurations result in approximately 34,800 and 17,600 input images for each respective time window. Accordingly, two distinct models were trained for each input shape, as presented in Table 5.12.

Table 5.8 Average SI-SDR performance across 10 CWT-trained networks for four different noise types at -5 dB and 0 dB, highlighting in bold text the highest results for each N_v value.

		CWT Processing					
		Window=64		Window=128		Window=256	
Noise\SNR (dB)		-5	0	-5	0	-5	0
AWGN	N_v**=24**	6.8879	11.0194	7.0102	11.0099	**7.1642**	**11.0855**
APGN		7.0093	11.3115	7.0385	11.2775	**7.2647**	**11.3831**
PhysioNet noise		17.4748	19.8214	17.4860	19.8594	**17.6138**	**20.0158**
Speech noise		13.9385	16.3247	13.9835	16.2811	**14.1036**	**16.4003**
AWGN	N_v**=16**	6.8774	11.0107	7.0283	10.9845	**7.0723**	**11.0412**
APGN		6.9706	11.2437	7.0754	11.2810	**7.1917**	**11.2821**
PhysioNet noise		17.4234	19.7412	17.3954	19.7600	**17.5219**	**19.9238**
Speech noise		13.8521	16.2045	**14.0719**	**16.2814**	13.8967	16.2011
AWGN	N_v**=8**	6.7649	10.9055	6.8845	10.9507	**7.0475**	**11.0068**
APGN		6.8705	11.1694	6.9562	11.1848	**7.0880**	**11.2382**
PhysioNet noise		17.1984	19.5284	17.3088	19.6996	**17.4111**	**19.8320**
Speech noise		13.5206	15.8708	13.6419	**15.9300**	**13.6574**	15.8821

Table 5.9 Average correlation across 10 CWT-trained networks for four different noise types at -5 dB and 0 dB, highlighting in bold text the highest results for each N_v value.

		CWT Processing					
		Window=64		Window=128		Window=256	
Noise\SNR (dB)		-5	0	-5	0	-5	0
AWGN	N_v**=24**	0.9012	0.9584	0.9037	0.9582	**0.9066**	**0.9591**
APGN		0.9070	0.9623	0.9074	0.9618	**0.9115**	**0.9629**
PhysioNet noise		0.9885	0.9935	0.9883	0.9082	**0.9887**	**0.9938**
Speech noise		0.9616	0.9799	0.9616	0.9797	**0.9626**	**0.9801**
AWGN	N_v**=16**	0.9018	0.9584	0.9044	0.9581	**0.9056**	**0.9591**
APGN		0.9069	0.9617	0.9082	0.9619	**0.9107**	**0.9621**
PhysioNet noise		0.9883	0.9932	0.9882	0.9934	**0.9886**	**0.9937**
Speech noise		0.9613	0.9795	**0.9634**	**0.9798**	0.9614	0.9792
AWGN	N_v**=8**	0.8990	0.9574	0.9017	0.9578	**0.9049**	**0.9587**
APGN		0.9044	0.9610	0.9066	0.9613	**0.9087**	**0.9619**
PhysioNet noise		0.9872	0.9925	0.9877	0.9931	**0.9878**	**0.9933**
Speech noise		0.9585	0.9777	0.9594	**0.9779**	**0.9596**	0.9777

Table 5.10 Average SI-SDR performance across 10 WSST-trained networks for four different noise types at -5 dB and 0 dB, highlighting in bold text the highest results for each N_v value.

WSST Processing							
		Window=64		**Window=128**		**Window=256**	
Noise\SNR (dB)		**-5**	**0**	**-5**	**0**	**-5**	**0**
AWGN	N_v**=24**	**6.7668**	**10.3851**	6.7263	10.2154	5.9935	9.5131
APGN		6.9320	**10.8106**	**6.9383**	10.6938	6.5486	10.2484
PhysioNet noise		**16.4555**	**18.5092**	16.2454	18.3450	15.5813	17.7410
Speech noise		**13.3688**	**15.4060**	13.0285	15.1152	12.5718	14.6090
AWGN	N_v**=16**	6.9205	10.6174	**6.9957**	**10.6317**	6.7161	10.3681
APGN		7.0367	10.9604	**7.1676**	**11.0307**	7.1008	10.9670
PhysioNet noise		16.8386	19.0877	**16.9140**	**19.2463**	16.8047	19.1515
Speech noise		**13.5672**	**15.5742**	13.3078	15.4582	13.0694	15.1995
AWGN	N_v**=8**	6.8552	10.6179	**6.8644**	**10.6534**	6.8088	10.5955
APGN		6.8399	10.8751	**6.9630**	**10.9252**	6.9468	10.9158
PhysioNet noise		16.6947	19.0375	**16.8636**	**19.2804**	16.8030	19.2039
Speech noise		**13.3374**	**15.5308**	13.2874	15.4947	13.1233	15.2987

Table 5.11 Average correlation across 10 WSST-trained networks for four different noise types at -5 dB and 0 dB, highlighting in bold text the highest results for each N_v value.

WSST Processing							
		Window=64		**Window=128**		**Window=256**	
Noise\SNR (dB)		**-5**	**0**	**-5**	**0**	**-5**	**0**
AWGN	N_v**=24**	**0.8943**	**0.9492**	0.8929	0.9472	0.8729	0.9370
APGN		**0.9022**	**0.9555**	0.9015	0.9542	0.8907	0.9488
PhysioNet noise		**0.9848**	**0.9903**	0.9838	0.9900	0.9820	0.9888
Speech noise		**0.9597**	**0.9766**	0.9570	0.9752	0.9549	0.9727
AWGN	N_v**=16**	0.8983	0.9526	**0.8999**	**0.9529**	0.8915	0.9495
APGN		0.9049	0.9574	**0.9078**	**0.9585**	0.9047	0.9577
PhysioNet noise		0.9862	0.9919	0.9862	**0.9922**	**0.9865**	0.9921
Speech noise		**0.9608**	**0.9772**	0.9593	0.9769	0.9576	0.9755
AWGN	N_v**=8**	0.8989	0.9542	**0.8991**	**0.9545**	0.8976	0.9536
APGN		0.9029	0.9581	**0.9056**	**0.9584**	0.9051	0.9583
PhysioNet noise		0.9853	0.9915	0.9859	0.9920	**0.9860**	**0.9921**
Speech noise		**0.9575**	**0.9765**	0.9571	0.9762	0.9564	0.9754

Unlike the results from CWT-trained models in Table 5.8, where the time window did not significantly influence performance, in the *S*-transform, the time window directly impacts the frequency resolution, leading to clear differences in denoising effectiveness. Both configurations outperformed the STFT-trained model in Table 5.6; however, the model using the 256-sample time window, which offers higher frequency resolution, demonstrated superior denoising performance compared to the 128-sample window.

Table 5.12 Average SI-SDR performance across 10 *S*-transform-trained networks for four different noise types at -5 dB and 0 dB, highlighting in bold text the highest results.

***S*-transform Processing**				
	Window=128		**Window=256**	
Noise\SNR (dB)	**-5**	**0**	**-5**	**0**
AWGN	5.4843	9.6238	**6.1746**	**10.2165**
APGN	5.2032	9.2549	**6.0060**	**9.9436**
PhysioNet noise	10.9345	13.0674	**12.1449**	**13.2947**
Speech noise	11.7657	14.6630	**12.8791**	**15.8927**

Table 5.13 Average correlation across 10 *S*-transform-trained networks for four different noise types at -5 dB and 0 dB, highlighting in bold text the highest results.

***S*-transform Processing**				
	Window=128		**Window=256**	
Noise\SNR (dB)	**-5**	**0**	**-5**	**0**
AWGN	0.8759	0.9458	**0.8900**	**0.9520**
APGN	0.8726	0.9432	**0.8893**	**0.9505**
PhysioNet noise	0.9562	0.9728	**0.9670**	**0.9754**
Speech noise	0.9444	0.9718	**0.9512**	**0.9757**

5.2.4. Best performing model

Considering a N_v value above 8 significantly increases the computation time, without increasing the denoising performance in any meaningful way, the top overall results here were obtained with the CWT-trained model at time window of 256 samples and 8 voices. The highest denoising performance was achieved with PhysioNet and speech noise contamination, examples of PCG signals contaminated with these noise types are illustrated in Figures 5.1 and 5.2. These figures show that, despite significant noise overlapping with the pathological

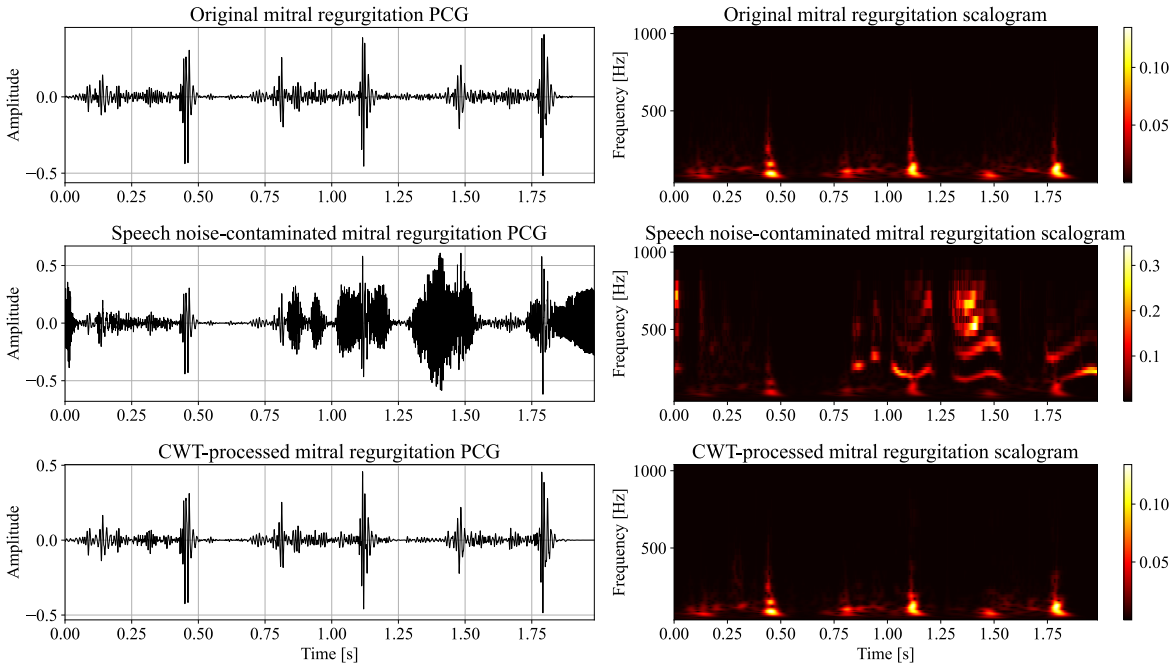


Figure 5.1 MR PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with speech noise.

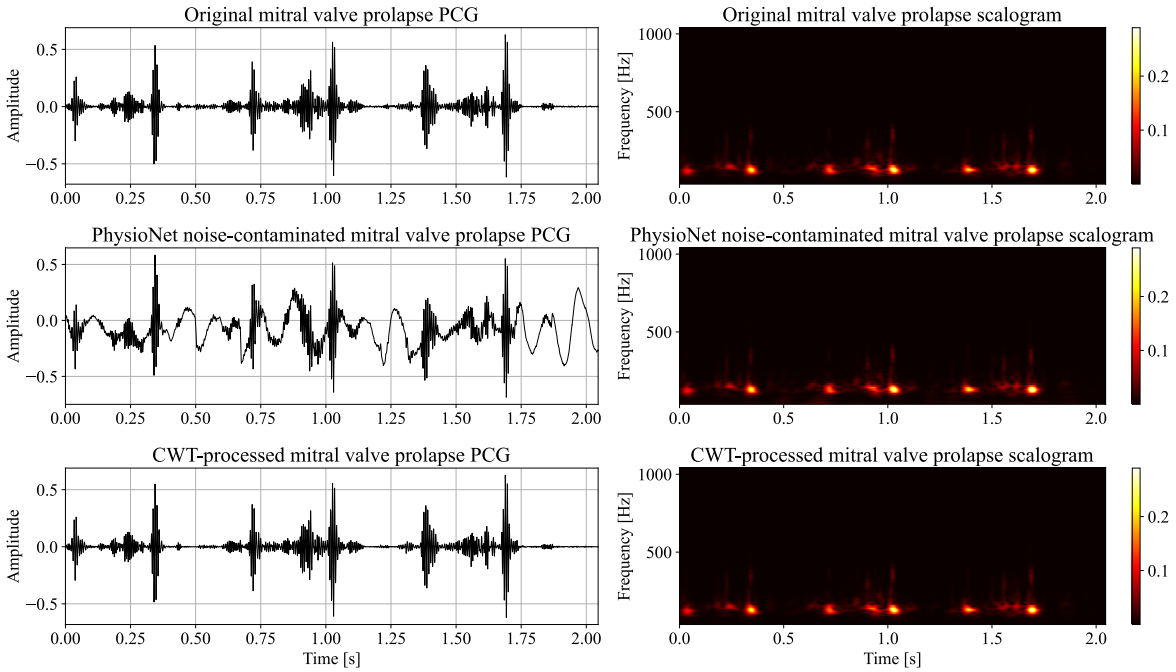


Figure 5.2 MVP PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with PhysioNet noise.

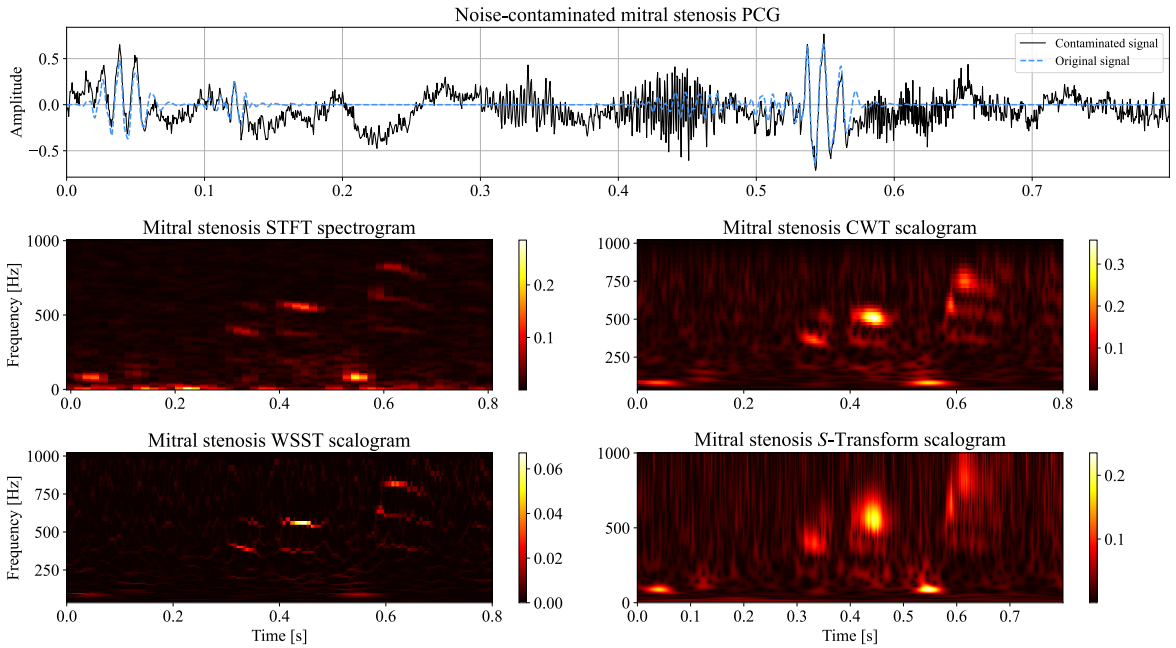


Figure 5.3 Composite noise-contaminated pathological PCG signal at -5 dB of SNR example with its 4 respective time-frequency representations. For visibility purposes, the CWT, WSST, and *S*-transform are illustrated in higher resolution than they were processed at.

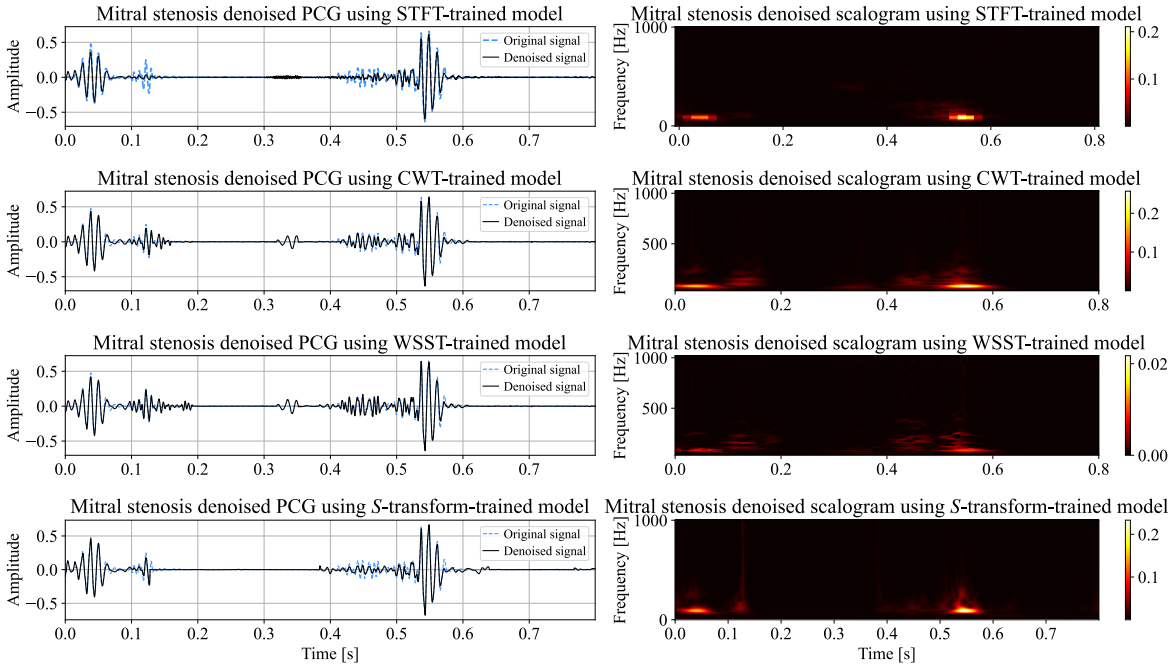


Figure 5.4 Denoised PCG signals from composite noise at -5 dB of SNR with their respective time-frequency representation. For visibility purposes, the CWT, WSST, and *S*-transform are illustrated in higher resolution than they were processed at.

features or the S1 and S2 components of the PCG, the model successfully eliminates most of the noise, leaving the time-domain signal nearly visually indistinguishable from the original PCG.

Figure 5.3 presents a cardiac cycle of a pathological PCG corrupted by a composite noise signal, which combines APGN, PhysioNet noise, and speech noise to contaminate the heart sound at -5 dB SNR. This figure allows for a comparison of the denoising performance of models trained using each of the four time-frequency transforms. The top plot in Figure 5.3 shows the contaminated signal, while the denoised outputs are depicted in Figure 5.4. The selected networks in this evaluation are those that demonstrated the highest performance. For the *S*-transform, the model trained with a time window of 256 samples was chosen. Similarly, the CWT model selected was trained with a 256-sample time window and $N_v = 8$. To ensure a consistent comparison between the CWT and the enhancement from the WSST, regardless of the variations in performance across different time windows and N_v values, a WSST model trained with the same parameters was also selected.

All four methods demonstrate strong denoising capabilities, but detailed analysis reveals important distinctions. The STFT approach effectively removes most of the noise but also distorts pathological features around 0.125 s and 0.450 s, failing to fully reconstruct the S1 and S2 heart sounds. The CWT method shows substantial improvements, more closely tracking the oscillations of the pathological waveform, although it introduces an artifact around 0.350 s. The WSST method, while nearly matching the CWT's performance, also introduces extraneous noise, such as around 0.190 s. The *S*-transform improves overall performance but continues to struggle with the accurate reconstruction of the pathological sounds. Overall, although both the CWT-trained and WSST-trained models exhibit strong noise suppression, the CWT-trained model consistently delivers superior denoising and signal reconstruction, outperforming the other time-frequency transforms on average.

5.3. Performance evaluation: Algorithm execution times

In practice, eliminating noise from a PCG takes a certain amount of time depending on the computational efficiency of the method; even in cases where the denoising speed difference is a fraction of a second, this number escalates as the number of PCGs denoised increases. In addition, training a network takes more time, to the extent where it limits the capabilities of generating a properly trained model. Therefore, to ensure a fair comparison of the performance between the four time-frequency transforms, it is necessary to take into account the computation times of the signal processing part, neural network model training,

Table 5.14 First numeric column presents the combined execution times of the signal processing, network training, and SI-SDR evaluation from each transform; the second numeric column shows the average time that takes to process each PCG that contains 3 full cardiac cycles given an already trained network, last column exhibits the ratio of the processing times with respect to the STFT time. The CWT/WSST time is evaluated on the best performing model, with a window of 256 samples and $N_v = 8$.

Execution times			
	Training (s)	**Execution (s)**	**Ratio**
STFT	109.71	0.0642	1
***S*-Transform (W=128)**	731.16	0.1179	1.84
***S*-Transform (W=256)**	2323.11	0.1475	2.30
CWT/WSST	820.9	0.1150	1.79

and the trained model execution. Testing the feasibility of the method's execution and its potential for further development.

The execution times are presented in Table 5.14, with an additional column for the ratio with respect to the STFT execution time. The training times cover from the signals pre-processing where the PCGs are processed into each respective time-frequency transform, to generating the dataset used next for the network training and evaluation; the execution times are exclusively the average time that it takes to process a single 3 full cardiac cycles PCG signal, given an already trained network with the respective time-frequency transform.

The two different models based on the *S*-transform are presented in separate rows, each with a different window size which resulted in a significant difference in processing time; however, despite the large difference in training time, the execution time does not reflect that big of a difference, the model with a 128-sample window being only 1.8 times slower than the STFT model while the 256-sample window model is only 2.3 times slower. The CWT and WSST were implemented with the Python library from Muradeli (2020), this calculates both transforms at the same time, which results in the same processing time. While the training time here is slightly higher than the training time for the *S*-transform at $W = 128$, the execution time results faster.

An important observation from the analysis of Tables 5.8 and 5.10 is that both signal pre-processing and network training times grow substantially as the number of voices N_v increases. This computation time increase is attributed to the need for a bigger frequency window, which consequently results in a matrix of larger size to process. However, it is noteworthy that the denoising performance improvement does not justify the computational burden. Moreover, although the time window does not significantly increase the computation time, a slight increase in performance is noticeable as this window increases. This result

suggests that experimenting with longer time windows could provide insightful information and possibly improve performance even further.

Chapter 6

Discussion

6.1. Denoising performance analysis

The incorporation of new frequency-transform representations established expectations for their performance based on their mathematical formulations and the visual quality of the generated representations. Given its foundational nature and simplicity in calculation, STFT was anticipated to underperform relative to the other three transforms. In contrast, the *S*-transform, which is computed similarly to the STFT but incorporates a variable time-frequency resolution parameter, suggested a potential for enhanced performance. The CWT generates a new dictionary of wavelets at varying scales, facilitating a more specialized analysis of each frequency band within the time intervals of the windowed PCG signal. This approach optimizes the time-frequency trade-off provided by the time representation of the signal, thereby enabling more effective training of the network. The WSST, being an enhancement of the CWT, refines the time-frequency representation further, leading to expectations of superior results among the previous time-frequency transforms.

The majority of the signal's energy is concentrated in the fundamental heart sounds, S1 and S2, within the frequency range below 250 Hz. Both the CWT and WSST are recognized for offering superior frequency resolution at lower frequencies, at the expense of time resolution. This feature is particularly beneficial for accurately capturing the energy distribution in these critical frequency bands, which likely explains the superior performance of the wavelet-based representations and the *S*-transform in this context.

While all three newly proposed representations, which capitalize on the Heisenberg uncertainty principle, demonstrated improved performance compared to the STFT, an unexpected finding was that the CWT-trained model outperformed the WSST-trained model (Tables 5.8 and 5.10). This outcome suggests two potential explanations. First, the time-frequency representation may have reached the limits of the current network architecture, indicating that

improvements in the architecture could lead to better performance. Second, the additional refinement introduced by the WSST may be redundant compared to the CWT in differentiating the components relevant to a PCG signal.

Another unexpected result was the subtle improvement in denoising performance associated with increasing the frequency resolution of the signal; i.e., increasing N_v. Although this adjustment creates a larger dictionary of wavelets, each focused on analyzing narrower frequency bands, only subtle differences were observable in both representations (Figure 6.1). Furthermore, the energy of both S1 and S2 is predominantly concentrated within a rectangular region where the height is significantly smaller than the width. Given this characteristic of PCGs, each Heisenberg box within this area benefits from improved frequency resolution at low frequencies, even with smaller N_v values. Consequently, increasing N_v does not substantially enhance the denoising quality and fails to justify the associated increase in computational complexity. This observation could explain the lack of expected improvements in denoising performance with higher N_v values.

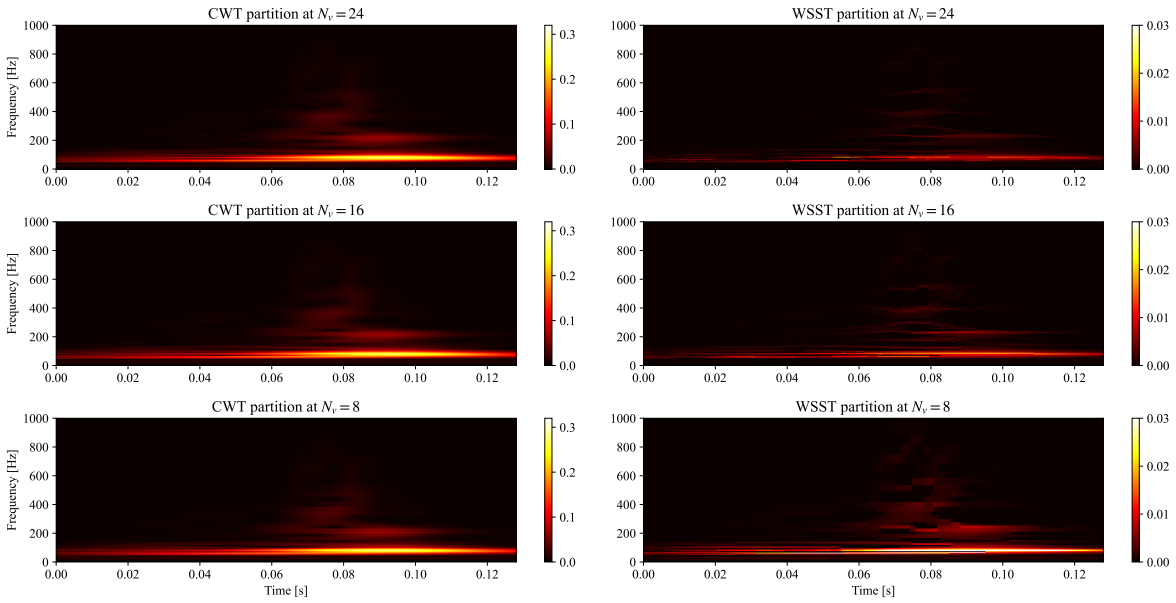


Figure 6.1 Partition examples of the CWT and WSST computed at N_v values of 24, 16, and 8.

A detailed inspection of individual signals for each pathology reveals that the best performance, unsurprisingly, was achieved on normal PCG signals, which consist primarily of S1 and S2 sounds. In contrast, pathological signals exhibited lower performance due to the presence of artifacts, in which each is only present in 20% of the training data. Analyzing the average denoising performance for each type of noise across different PCG pathologies reveals a noticeable trend, where specific noise types influence the denoising results for different pathologies in varying ways. For instance, AWGN has the greatest negative impact

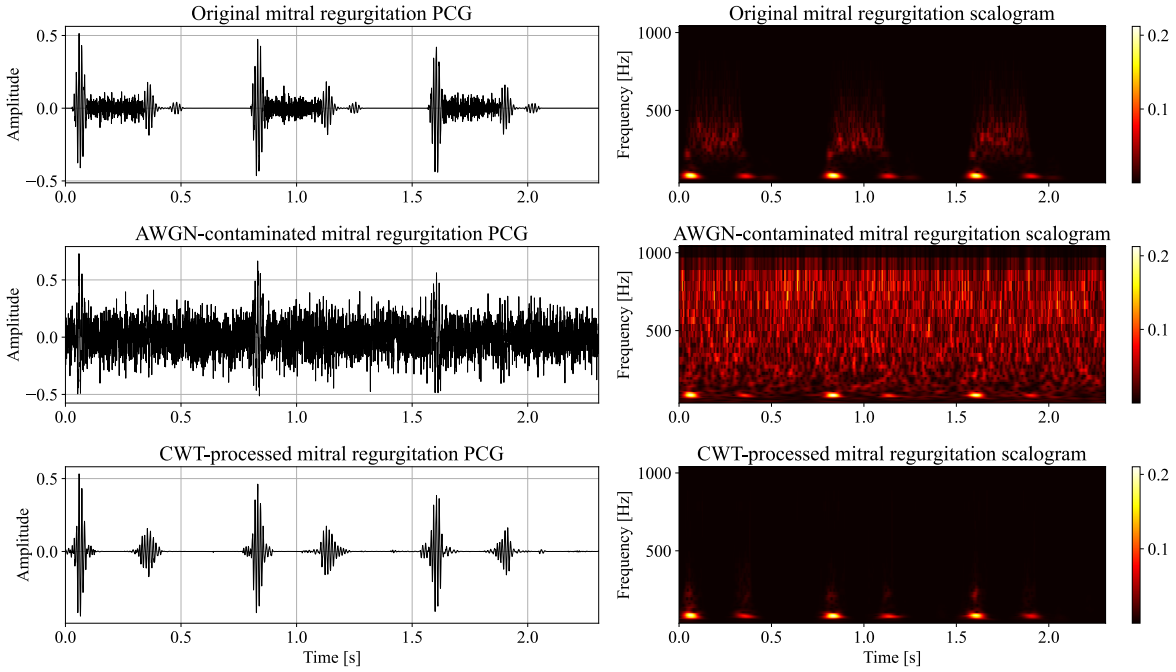


Figure 6.2 MR PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with AWGN.

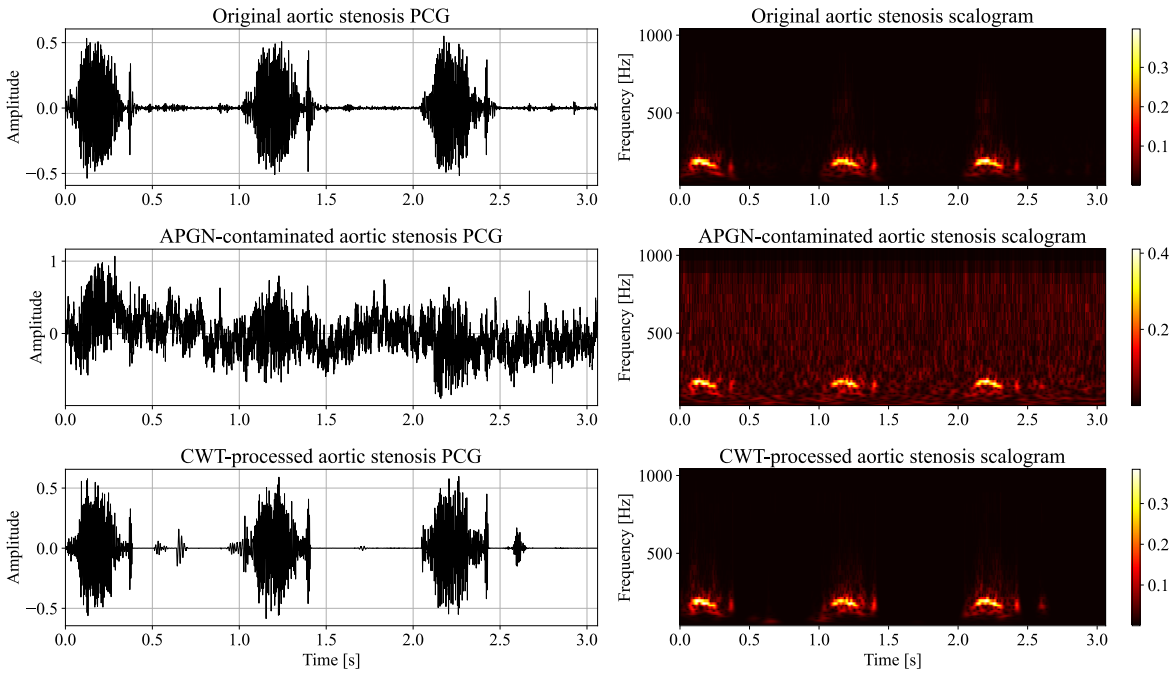


Figure 6.3 AS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with APGN.

on MR, followed by a slightly lesser effect on AS, and an even smaller impact on both MS and MVP, with normal PCGs yielding the best denoising performance. In the case of APGN, both AS and MR exhibited the poorest denoising performance, followed by MS and MVP, which performed similarly. As with AWGN, normal PCGs showed the highest performance. For PhysioNet noise, MS performed significantly worse than the other pathologies, while for speech noise, there was a gradual decline in performance from MS to MR and finally to AS with the lowest resulting SI-SDR values. All these values were obtained with relatively low standard deviations, indicating that the type of noise has a consistent and measurable impact on the denoising performance for different pathologies. Figures 6.2 and 6.3 present two examples for AWGN and APGN, in which both had the highest effect for denoising aortic stenosis and mitral regurgitation PCGs respectively. Figure 6.2 specifically, highlights a strong visual similarity between the pathology and the noise the signal was contaminated with. This similarity, in combination with the limited size of the training dataset, likely explains the model's inability to preserve the pathology. An interesting observation is that in the APGN-contaminated signal (Figure 6.3), with a randomized noise distinguished by its similarity to physiological sounds, the model interprets part of that noise as a PCG-like waveform in between 0.5 s and 0.7 s.

It is important to remark that the previously illustrated results represent highly unlikely scenarios in which the noise is more dominant than the PCG signal. While the model does not achieve a fully accurate reconstruction in these cases, these outcomes were produced by a CWT-trained model constrained by the limited dataset size. As demonstrated in earlier comparisons between the STFT-trained model and its data-augmented counterpart, incorporating data augmentation or utilizing a larger clean PCG dataset would likely result in improved denoising and more accurate reconstruction of the PCG signals, even in these extreme scenarios.

6.2. Power spectral density analysis of noise-reduced signals

Up to this point, the method has been evaluated using the SI-SDR, Pearson correlation coefficient metrics and through visual inspection and comparisons of the corresponding spectrograms or scalograms. This section extends the analysis by comparing the average PSDs of original PCG signals, their respective noise-contaminated versions at -5 dB of SNR, and the denoised signals. This comprehensive analysis was conducted using a test dataset completely isolated from the training dataset, comprising 100 PCG sounds, 20 of those

representing each of the 5 types of cardiac sound present in the dataset. For the sake of simplicity, this analysis focuses exclusively on the CWT-trined model, which produced the most favorable results, utilizing a time window of 256 samples and $N_v = 8$.

Figure 6.4 illustrates the average PSDs. Although the network demonstrates overall robust performance, there are segments where the spectral waveforms do not align perfectly. The most significant discrepancies are observed within the first 50 Hz and the last 100 Hz of the spectrum. Conversely, the 50-250 Hz interval exhibits near-identical alignment between the clean and denoised spectra. This behavior is thought to arise from the neural network's focus primarily on the frequency bands with the highest energy.

Figure 6.5 displays the normalized average PSD for all PCG signals and the various noise types used to contaminate the audio samples. It is noteworthy that, unlike the recorded sounds, the artificially generated noises (AWGN and APGN) were not produced by an anti-aliasing filter, leading to a substantial portion of signal energy concentrated in the high-frequency range. Furthermore, the denoised signals following AWGN and APGN contamination, as illustrated in Figure 6.4, do not exhibit an anti-aliasing effect. In contrast, signals contaminated with PhysioNet and speech noise do demonstrate this effect. Although a correlation is observed between the type of noise and the anti-aliasing behavior of the trained network, the underlying cause of this phenomenon remains unclear.

A potential enhancement to the trained model could involve training specialized networks targeting specific frequency bands where denoising performance is undesired, such as the 0–50 Hz and 400–1,000 Hz ranges. Although this strategy may improve performance in these challenging bands, it would significantly increase computational complexity due to the requirement for multiple specialized networks. As such, this approach is beyond the scope of this study.

6.3. Method-induced noise analysis

Although the SI-SDR evaluation demonstrates strong denoising performance, applying a denoising method to a clean signal introduces the potential for not recovering the original signal with sample-to-sample accuracy, thereby adding residual noise from the denoising process itself. Evaluating this residual noise is crucial for determining the method's reliability and overall quality, as it establishes an upper bound on the achievable denoising performance.

The original test signals were processed without added noise using both the CWT-trained and STFT-trained U-Net models. The steps for each method can be described as follows: $s \rightarrow \text{CWT} \rightarrow \text{UNet}_{\text{CWT}} \rightarrow \text{iCWT} \rightarrow \tilde{s}_{\text{CWT}}$ and $s \rightarrow \text{STFT} \rightarrow \text{UNet}_{\text{STFT}} \rightarrow \text{iSTFT} \rightarrow \tilde{s}_{\text{STFT}}$, respectively. In both cases, the signal s undergoes a time-frequency transform and its

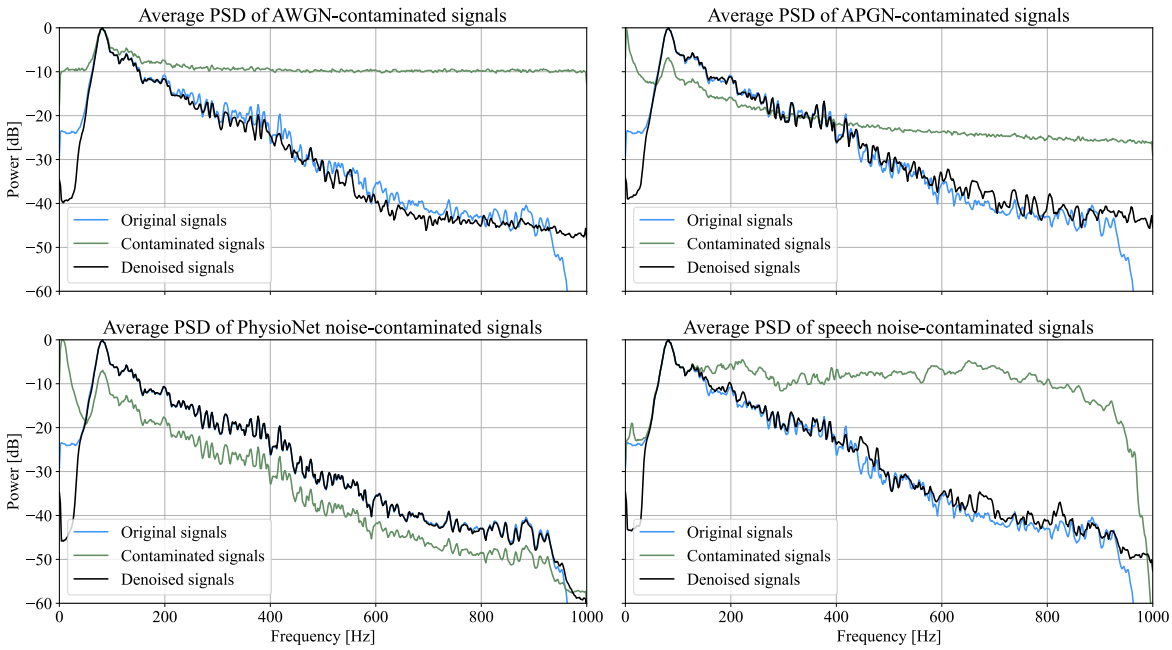


Figure 6.4 Average power spectral density comparison of the test dataset of original, contaminated at -5 dB, and denoised signals from four different types of noise by the CWT-trained model at 8 voices with a time window size of 256 samples.

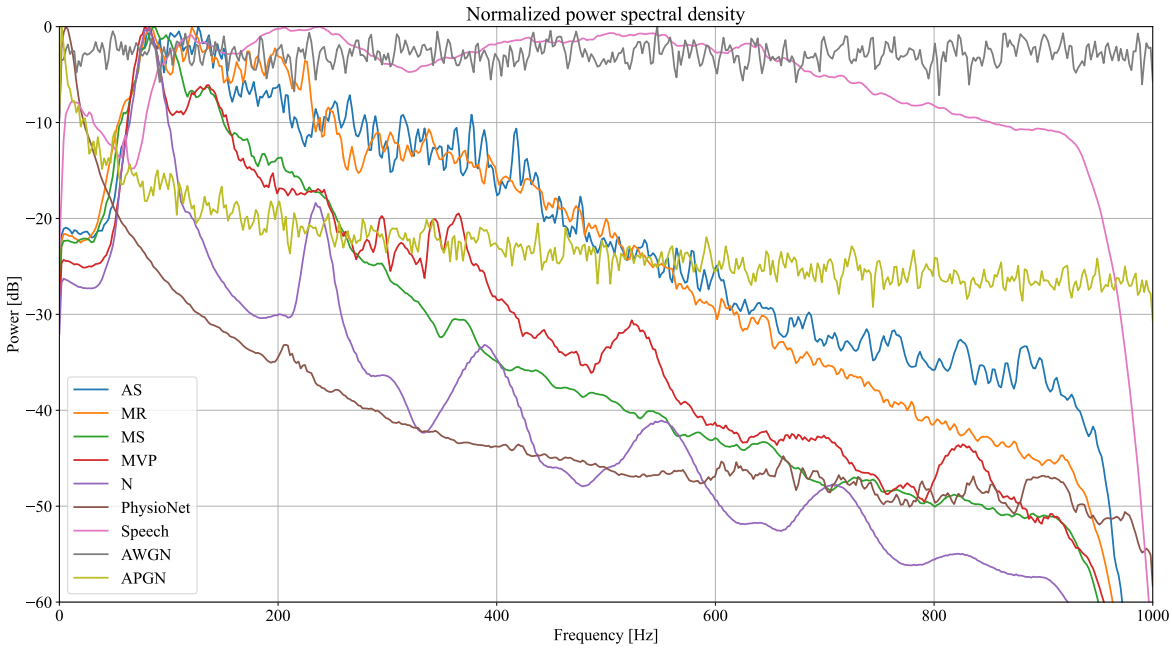


Figure 6.5 Normalized average power spectral density of 200 signals per PCG type and four different types of noise.

corresponding inverse, introducing transform-specific noise ε_{TF}. Additionally, the U-Net processing introduces network-specific noise $\varepsilon_{\text{UNet}}$, the general expression for $\tilde{s}_{\text{TF}}$ can be expressed as:

$$\tilde{s}_{\text{TF}} = s + \varepsilon_{\text{TF}} + \varepsilon_{\text{UNet}}, \tag{6.1}$$

defining the total noise introduced by each method as $\varepsilon_{total} = \varepsilon_{\text{TF}} + \varepsilon_{\text{UNet}}$.

Table 6.1 SI-SDR in dB of non-contaminated signals processed by the CWT-trained and STFT-trained models. ε_{TF} represents the noise introduced by the respective time-frequency transform, $\varepsilon_{\text{UNet}}$ the noise by each respective trained network, and ε_{total} the total amount of noise.

		Noise (dB)			
	Type of noise	**Avg**	**Std**	**Min**	**Max**
CWT	ε_{TF}	25.14	2.99	18.39	33.30
	$\varepsilon_{\text{UNet}}$	39.27	8.45	13.78	52.47
	ε_{total}	24.34	3.61	12.83	31.30
STFT	ε_{TF}	312.71	3.03	301.53	316.24
	$\varepsilon_{\text{UNet}}$	24.38	4.64	14.24	33.51
	ε_{total}	24.38	4.64	14.24	33.51

Table 6.1 displays the noise obtained by this process, comparing the CWT-trained model, which obtained the results with highest denoising performance, and the STFT which is known for being its computating being extremely efficient. The key observation from Table 6.1 is the average total noise ε_{total} generated by the CWT-based method. Although this approach demonstrated the best overall performance, the table reveals that its primary limitation stems from the noise ε_{TF} introduced by the time-frequency transform itself. In contrast, for the STFT-based method, this transform-related noise is negligible. Employing a time-frequency transform that introduces less noise but performs comparably to the CWT would shift the primary noise contribution to $\varepsilon_{\text{UNet}}$, as seen in the STFT method, where this value reflects a strong denoising capability. However, a significant issue with $\varepsilon_{\text{UNet}}$ in the CWT-based approach is the high standard deviation, indicating that the U-Net can sometimes alter otherwise perfectly denoised signals. While this is not ideal, it is likely due to the relatively small training dataset, which constrains the network's learning capacity. As shown by the comparison of two STFT-trained models, with and without data augmentation, the network performance can be substantially improved with an extended training dataset.

Chapter 7

Conclusions

Cardiovascular diseases remain the leading cause of death worldwide, and early diagnosis through automatic heart sound analysis offers a promising, low-cost solution. However, one of the main challenges in this field is the presence of noise during the recording process, which can significantly hinder the accuracy of phonocardiogram (PCG) analysis. While many existing PCG denoising algorithms have been developed, they often rely on idealistic assumptions about the nature of the corrupted signals, such as targeting synthetic white or pink noise. These assumptions fail to address the complexity of real-world noise that occurs in practical scenarios.

7.1. Contributions

In this thesis, a novel method for denoising heart sounds contaminated by real-world noise is proposed. This method leverages different time-frequency transforms—namely, the Short-Time Fourier Transform (STFT), Continuous Wavelet Transform (CWT), Wavelet Synchrosqueezed Transform (WSST), and *S*-transform—combined with a U-Net deep learning architecture. The use of this U-Net, which has shown exceptional performance in blind source separation tasks, was particularly effective in this context. According to the research conducted in this thesis, this is the first study that systematically compares the performance of different time-frequency transformations when paired with U-Net architectures for PCG denoising, making a significant contribution to the field.

The central approach involves transforming noisy PCG signals into two-dimensional time-frequency representations, i.e. images, which serve as inputs to the U-Net model. The U-Net then generates a mask to isolate and remove the noise components from the heart sound signals. The effectiveness of this denoising approach was evaluated using the Scale-Invariant Signal-to-Distortion Ratio (SI-SDR), a robust metric that measures the quality of

the output signal in relation to the original, the Pearson correlation coefficient, and a Power Spectral Density (PSD) inspection of the sounds used. In addition, the method was tested on PCG signals contaminated at -5 dB and 0 dB of Signal-to-Noise Ratio (SNR) using four types of noise: Additive White Gaussian Noise (AWGN), Additive Pink Gaussian Noise (APGN), and highly non-stationary disturbances such as real PCG background noise from the PhysioNet database and speech.

The findings in this thesis indicate that the CWT outperformed the other time-frequency transforms, achieving denoising gains of over 18 dB for PCG signals contaminated with speech at -5 dB SNR. The PSD analysis conducted revealed the frequency bands where the model performed most effectively. This insight allowed the refinement of the denoising model to handle real-world, non-stationary noise more effectively than state-of-the-art methods. Importantly, the proposed model can be seamlessly integrated into future heart sound classification systems, addressing a key issue in existing models where denoising often diminishes classification accuracy by removing critical pathological features. In contrast, this approach does a better denoising job and has the potential to improve in preserving these features, ensuring that they are retained during the denoising process.

Despite the promising results, one notable limitation of this study is the relatively small size of the training dataset, which consisted of only 1,000 signals. Future work should focus on expanding the dataset and employing data augmentation techniques to improve the model's robustness and generalizability. A data augmentation technique was proposed here, consisting of modifying previously existing signals; however, there is a diverse amount of options in which the dataset can be expanded further. A visual inspection of individual pathological PCGs in Appendix A shows that the best denoising performance is presented on non-pathological signals, likely due to the repeatability of the S1 and S2 components of the PCG during network training, in contrast to each individual pathology that is presented in only 20% of the signals. A larger, more diverse dataset would not only enhance denoising performance but also increase the model's applicability across a wider range of heart sound recordings. This step is essential for advancing the current state of PCG denoising and ensuring that the method can be reliably used in clinical settings.

In conclusion, this thesis presents a novel, effective method for denoising heart sounds using time-frequency representations and a U-Net architecture. The approach demonstrates state-of-the-art performance, particularly with real-world, non-stationary noise, and sets the foundation for future improvements in heart sound analysis and classification tasks.

7.2. Future work

The work presented in this thesis has the potential for further expansion in similar research fields. The extraction of various types of physiological sounds is a critical area of focus in biomedical signal processing. Beyond cardiac audio, other types of signals present challenges similar to those tackled here. While the physiological noise processed in this thesis was considered noise, in other research contexts, such sounds may represent the primary signals of interest, with cardiac sounds classified as noise—pulmonary audio serves as an example. Although the proposed method demonstrated strong performance in cardiac audio denoising, it is not limited to this type of sound and could be adapted for other audio types, given an appropriate training dataset. Additionally, the PCG denoising model can be refined based on insights gained from the results analysis, enabling its integration into cardiac sound classification systems. Since audio signals are processed in time-based segments and the processing of each segment is relatively fast, this method has the potential for near real-time denoising in electronic stethoscopes.

This approach extends beyond the domain of biomedical signal processing, offering a broad spectrum of potential applications. Initially inspired by blind source separation techniques applied to music signal processing, the methodology demonstrates versatility that could find applications in diverse fields. For instance, in speech enhancement, where the goal is to isolate and improve speech quality in noisy environments. Similarly, it can be employed in music processing tasks, such as separating instrumental components or enhancing audio quality, employing better time-frequency representations.

Chapter 8

Publications

This thesis has contributed to the following publications:

1. Conference articles:
    - C. González-Rodríguez, M. A. Alonso-Arévalo and E. García-Canseco, "Eliminación de ruido en sonidos cardíacos mediante técnicas de aprendizaje profundo," in Research in Computer Science, vol. 152(9), pp. 161-173, 2023.
2. Journal articles:
    - C. González-Rodríguez, M. A. Alonso-Arévalo and E. García-Canseco, "Robust Denoising of Phonocardiogram Signals Using Time-Frequency Analysis and U-Nets," in IEEE Access, vol. 11, pp. 52466-52479, 2023, doi: 10.1109/ACCESS.2023.3280453.
    - C. González-Rodríguez, E. García-Canseco and M. A. Alonso-Arévalo, "Enhancing Heart Sound Signal Denoising: Unveiling the Impact of Time-Frequency Transformation in U-Net Performance," (***Submitted***)

ISSN 1870-4069

Eliminación de ruido en sonidos cardíacos mediante técnicas de aprendizaje profundo

Cristóbal González Rodríguez[1], Miguel A. Alonso Arévalo[2], Eloísa García Canseco[1]

[1] Universidad Autónoma de Baja California,
Facultad de Ciencias,
México

[2] Centro de Investigación Científica y de Educación Superior de Ensenada,
Departamento de Electrónica y Telecomunicaciones,
División de Física Aplicada,
México

{a351269, eloisa.garcia}@uabc.edu.mx,
aalonso@cicese.edu.mx

Resumen. Las enfermedades cardiovasculares son la principal causa de mortalidad en todo el mundo. La auscultación cardiaca es un método de diagnóstico prometedor; sin embargo, uno de sus principales inconvenientes es que es altamente propensa al ruido durante la grabación del sonido, lo que dificulta el diagnóstico. En este trabajo proponemos un algoritmo de eliminación de ruido para las señales de audio cardíaco. El ruido se elimina en la representación tiempo–frecuencia de la señal. Específicamente, calculamos la transformada de Fourier de tiempo corto (STFT) de la señal de FCG contaminada y entrenamos una red neuronal de tipo U-Net para que reconozca los sonidos cardíacos, ya sean normales o patológicos, del ruido. En nuestras pruebas, el método propuesto muestra un alto desempeño incluso en escenarios altamente desfavorables, ya que puede eliminar el ruido de una señal FCG contaminada con una relación señal a ruido (SNR) de -5 dB con mejoras promedio del orden de $\approx$15 dB.

Palabras clave: Fonocardiograma, transformada de Fourier, red neuronal convolucional, separación de fuentes.

Noise Removal in Heart Sounds Using Deep Learning Techniques

Abstract. Cardiovascular diseases are the leading cause of mortality worldwide. Cardiac auscultation is a promising diagnostic method; however, one of its main drawbacks is that it is highly susceptible to noise during sound recording, which hinders diagnosis. In this study, we propose a noise removal algorithm for cardiac audio signals. The noise is eliminated in the time-frequency representation of the signal. Specifically, we calculate the Short-Time Fourier Transform (STFT)

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 April 2023, accepted 17 May 2023, date of publication 26 May 2023, date of current version 2 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280453

RESEARCH ARTICLE

Robust Denoising of Phonocardiogram Signals Using Time-Frequency Analysis and U-Nets

CRISTÓBAL GONZÁLEZ-RODRÍGUEZ[1], MIGUEL A. ALONSO-ARÉVALO[2], (Member, IEEE), AND ELOÍSA GARCÍA-CANSECO[1], (Member, IEEE)

[1]Faculty of Sciences, Autonomous University of Baja California, Ensenada, Baja California 22860, Mexico
[2]Applied Physics Division, Department of Electronics and Telecommunications, Ensenada Centre for Scientific Research and Higher Education, Ensenada, Baja California 22860, Mexico

Corresponding author: Eloísa García-Canseco (eloisa.garcia@uabc.edu.mx)

The work of Cristóbal González-Rodríguez was supported by the Mexican National Council for Science and Technology (CONACYT) through the Graduate Research Fellowship under Grant 1203286.

ABSTRACT Cardiovascular diseases are the cause of many deaths worldwide every day. Automated cardiac auscultation is a promising diagnosis method; however, one of its main disadvantages is that it is prone to receiving too much noise during sound recording, which hinders diagnosis. In the available literature, most phonocardiogram (PCG) denoising methods have been evaluated using only synthetic sources such as white noise; this work proposes a more realistic approach. In this paper, the denoising process occurs in the time–frequency domain. More specifically, we compute the Short-Time Fourier Transform (STFT) of the contaminated PCG signal and train a U-Net to recognize normal and pathological cardiac sounds from noise. We are unaware of previous attempts to develop a robust PCG-denoising algorithm capable of simultaneously removing noise signals from four different sources: additive white Gaussian noise (AWGN), additive pink Gaussian noise (APGN), speech, and real PCG background noise. Since we are limited by the relatively small number of clean PCG signals available, we also propose a method for high-quality phonocardiogram data augmentation. In our tests, the proposed method exhibits high performance even in unfavorable scenarios since it can denoise a PCG signal contaminated with a signal-to-noise ratio (SNR) of -5 dB with average improvements ranging from 11.85 dB up to 17.60 dB, depending on the noise type used to degrade the cardiac signal. This method could significantly improve the performance of automatic cardiac sound classification algorithms in noisy environments but could also be used in electronic stethoscopes.

INDEX TERMS Denoising, cardiac sounds, short-time Fourier transform, convolutional neural networks, U-Net.

I. INTRODUCTION

Cardiovascular diseases are the cause of many deaths worldwide every day [1], [2]. Although there are many diagnosis methods, such as electrocardiograms (ECG), magnetic resonance imaging (MRI), or echocardiograms, cardiac auscultation is one of the cheapest, most practical, and quickest non-invasive methods that exist to date. Recent advancements in computing, together with the shrinking size and ever-increasing processing power of electronic devices, have sparked an interest in the automatic analysis of heart sound signals. The main goal of automatic heart sound analysis is to precisely classify the presence or absence of pathological events in the cardiac cycle [3]. If the presence of such an event is confirmed, an automated system should ideally also identify the type of pathology. In addition, nowadays, many hospitals and clinics have electronic stethoscopes whose main advantage over more traditional ones is their ability to record the cardiac sound as a phonocardiogram (PCG), which contains valuable information about the state of the heart that can be later analyzed more carefully to determine whether a patient is healthy or if she/he has a pathology and identify it.

The presence of noise in the PCGs is, nevertheless, one of cardiac auscultation's most prominent problems. This noise can be either caused by the environment, such as speech and nearby machinery; by physiological sounds, such as gastric

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Piccialli.

Bibliography

Abbas, A. K. and Bassam, R. (2022). *Phonocardiography signal processing*. Springer Nature.

Al-Zaben, A., Al-Fahoum, A., Ababneh, M., Al-Naami, B., and Al-Omari, G. (2024). Improved recovery of cardiac auscultation sounds using modified cosine transform and LSTM-based masking. *Medical & Biological Engineering & Computing*, pages 1–13.

Ali, M. N., El-Dahshan, E.-S. A., and Yahia, A. H. (2017). Denoising of heart sound signals using discrete wavelet transform. *Circuits, Systems, and Signal Processing*, 36(11):4482–4497.

Ali, S. N., Shuvo, S. B., Al-Manzo, M. I. S., Hasan, A., and 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*.

Aloorravi, S. (2024). *Mastering Time Series Analysis and Forecasting with Python: Bridging Theory and Practice Through Insights, Techniques, and Tools for Effective Time Series Analysis in Python (English Edition)*. Orange Education Pvt Limited.

Andreas, J., Eric, H., Nicola, M., Rachel, B., Aparna, K., and Tillman, W. (2017). Singing voice separation with deep u-net convolutional networks. In *18th International Society for Music Information Retrieval Conference*, pages 23–27.

Benesty, J., Chen, J., and Habets, E. (2011). *Speech Enhancement in the STFT Domain*. SpringerBriefs in Electrical and Computer Engineering. Springer Berlin Heidelberg.

Bishop, C. M. (2006). Pattern recognition and machine learning. *Springer google schola*, 2:1122–1128.

Boashash, B. (2003). *Time Frequency Signal Analysis and Processing: A Comprehensive Reference*. Elsevier.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing*, 27(2):113–120.

Chen, J., Benesty, J., Huang, Y., and Doclo, S. (2006). New insights into the noise reduction wiener filter. *IEEE Transactions on audio, speech, and language processing*, 14(4):1218–1234.

Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.

Chowdhury, T. H., Poudel, K. N., and Hu, Y. (2020). Time-frequency analysis, denoising, compression, segmentation, and classification of pcg signals. *IEEE Access*, 8:160882–160890.

Daubechies, I., Lu, J., and Wu, H.-T. (2011). Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and computational harmonic analysis*, 30(2):243–261.

Donoho, D. L. (1995). De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627.

Díaz, J. (2016). On the origin of the signals observed across the seismic spectrum. *Earth-Science Reviews*, 161:224–232.

George, E. B. and Smith, M. J. (1997). Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model. *IEEE transactions on speech and audio processing*, 5(5):389–406.

Ghosh, S. K., Tripathy, R. K., and Ponnalagu, R. (2020). Evaluation of performance metrics and denoising of pcg signal using wavelet based decomposition. In *2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–6. IEEE.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gradolewski, D., Magenes, G., Johansson, S., and Kulesza, W. J. (2019). A wavelet transform-based neural network denoising algorithm for mobile phonocardiography. *Sensors*, 19(4):957.

Gradolewski, D. and Redlarski, G. (2014). Wavelet-based denoising method for real phonocardiography signal recorded by mobile devices in noisy environment. *Computers in biology and medicine*, 52:119–129.

Gul, S. and Khan, M. S. (2023). A survey of audio enhancement algorithms for music, speech, bioacoustics, biomedical, industrial and environmental sounds by image U-Net. *IEEE Access*.

Hennequin, R., Khlif, A., Voituret, F., and Moussallam, M. (2020). Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ibarra-Hernández, R. F., Alonso-Arévalo, M. A., Cruz-Gutiérrez, A., Licona-Chávez, A. L., and Villarreal-Reyes, S. (2017). Design and evaluation of a parametric model for cardiac sounds. *Computers in biology and medicine*, 89:170–180.

INEGI (2021). Características de las defunciones registradasen méxico durante 2020. *COMUNICADO DE PRENSA NÚM. 592/21*.

Janiesch, C., Zschech, P., and Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31:685 – 695.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Le Roux, J., Wisdom, S., Erdogan, H., and Hershey, J. R. (2019). Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630. IEEE.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lilly, J. M. and Olhede, S. C. (2012). Generalized morse wavelets as a superfamily of analytic wavelets. *IEEE Transactions on Signal Processing*, 60(11):6036–6041.

Lines, L. and Treitel, S. (1984). A review of least-squares inversion and its application to geophysical problems. *Geophysical prospecting*, 32(2):159–186.

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Liu, C., Springer, D., Li, Q., Moody, B., Juan, R. A., Chorro, F. J., Castells, F., Roig, J. M., Silva, I., Johnson, A. E., et al. (2016). An open access database for the evaluation of heart sound algorithms. *Physiological measurement*, 37(12):2181.

Macon, M. W. and Clements, M. A. (1996). Speech concatenation and synthesis using an overlap-add sinusoidal model. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 361–364. IEEE.

Mahnke, C. B. (2009). Automated heart sound analysis/computer-aided auscultation: a cardiologist's perspective and suggestions for future development. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3115–3118. IEEE.

Mallat, S. (2008). *A wavelet tour of signal processing: the sparse way*. Academic Press.

Mallinson, T. (2018). A qualitative exploration of current paramedic cardiac auscultation practices. *Journal of Paramedic Practice*, 10(9):387–393.

Messer, S. R., Agzarian, J., and Abbott, D. (2001). Optimal wavelet denoising for phonocardiograms. *Microelectronics journal*, 32(12):931–941.

Mohan, N., Kumar, S., and Soman, K. (2020). Group sparsity assisted synchrosqueezing approach for phonocardiogram signal denoising. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5. IEEE.

Muradeli, J. (2020). ssqueezepy. *GitHub. Note: https://github.com/OverLordGoldDragon/ssqueezepy/*.

Nersisson, R. and Noel, M. M. (2017). Heart sound and lung sound separation algorithms: a review. *Journal of medical engineering & technology*, 41(1):13–21.

Nikbakht, M., Chan, M., Lin, D. J., Gazi, A. H., and Inan, O. T. (2024). A residual U-Net neural network for seismocardiogram denoising and analysis during physical activity. *IEEE Journal of Biomedical and Health Informatics*, pages 1–12.

OECD (2017). Obesity Update. https://www.oecd.org/health/health-systems/Obesity-Update-2017.pdf.

Pahlm, O. and Wagner, G. S. (2011). *Multimodal cardiovascular imaging: principles and clinical applications*. McGraw-Hill Medical New York, NY, USA:.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Pauline, S. H. and Dhanalakshmi, S. (2022). A robust low-cost adaptive filtering technique for phonocardiogram signal denoising. *Signal Processing*, 201:108688.

Ritchie, H., Spooner, F., and Roser, M. (2018). Causes of death. *Our World in Data*. https://ourworldindata.org/causes-of-death.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Sejdić, E. and Lipsitz, L. A. (2013). Necessity of noise in physiology and medicine. *Computer methods and programs in biomedicine*, 111(2):459–470.

Simonyan, K. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, J. O. (2011). *Spectral audio signal processing*. W3K.

Son, G.-Y. and Kwon, S. (2018). Classification of heart sound signal using multiple features. *Applied Sciences*, 8(12):2344.

Stockwell, R. G., Mansinha, L., and Lowe, R. (1996). Localization of the complex spectrum: the s transform. *IEEE transactions on signal processing*, 44(4):998–1001.

Stoica, P. and Moses, R. L. (2005). *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ.

Sun, W., Zhang, Y., and Chen, F. (2024). Research on heart and lung sound separation method based on dae–nmf–vmd. *EURASIP Journal on Advances in Signal Processing*, 2024(1):59.

Tabatabaeefar, M., Mostaar, A., et al. (2020). Biomedical image denoising based on hybrid optimization algorithm and sequential filters. *Journal of biomedical physics & engineering*, 10(1):83.

Tavel, M. E. (2006). Cardiac auscultation: a glorious past—and it does have a future! *Circulation*, 113(9):1255–1259.

Tu, C.-L., Hwang, W.-L., and Ho, J. (2005). Analysis of singularities from modulus maxima of complex wavelets. *IEEE Trans. Inf. Theory*, 51:1049–1062.

Vasudevan, K. (2018). Signals and systems. *Analog Communications*.

Ventosa, S., Simon, C., Schimmel, M., Dañobeitia, J. J., and Mànuel, A. (2008). The *s*-transform from a wavelet point of view. *IEEE Transactions on Signal Processing*, 56(7):2771–2780.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Wang, W., Wang, S., Qin, D., Fang, Y., and Zheng, Y. (2023). Heart-lung sound separation by nonnegative matrix factorization and deep learning. *Biomedical Signal Processing and Control*, 79:104180.

World Health Organization (2021). World health statistics 2021: monitoring health for the SDGs, sustainable development goals. Technical report, World Health Organization.

World Health Organization (WHO) (2023). World health statistics 2023: monitoring health for the SDGs, sustainable development goals. Technical report, World Health Organization.

Yin, L., Yang, R., Gabbouj, M., and Neuvo, Y. (1996). Weighted median filters: a tutorial. *IEEE Transactions on circuits and systems II: analog and digital signal processing*, 43(3):157–192.

Appendix A

Additional figures

This appendix presents figures illustrating the original, noise-contaminated, and CWT-processed signals. These figures incorporate a combination of all five types of cardiac sound signals used in this thesis, along with the four different types of noise, resulting in a total of 20 figures.

PCG signal denoising examples

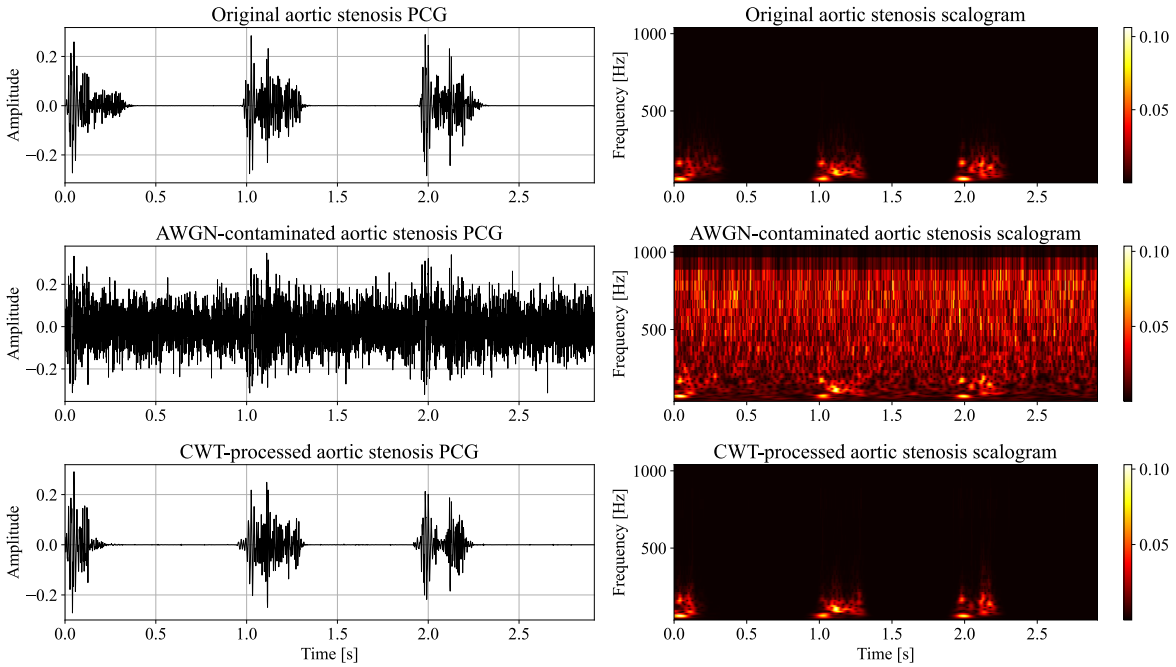


Figure A.1 AS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with AWGN.

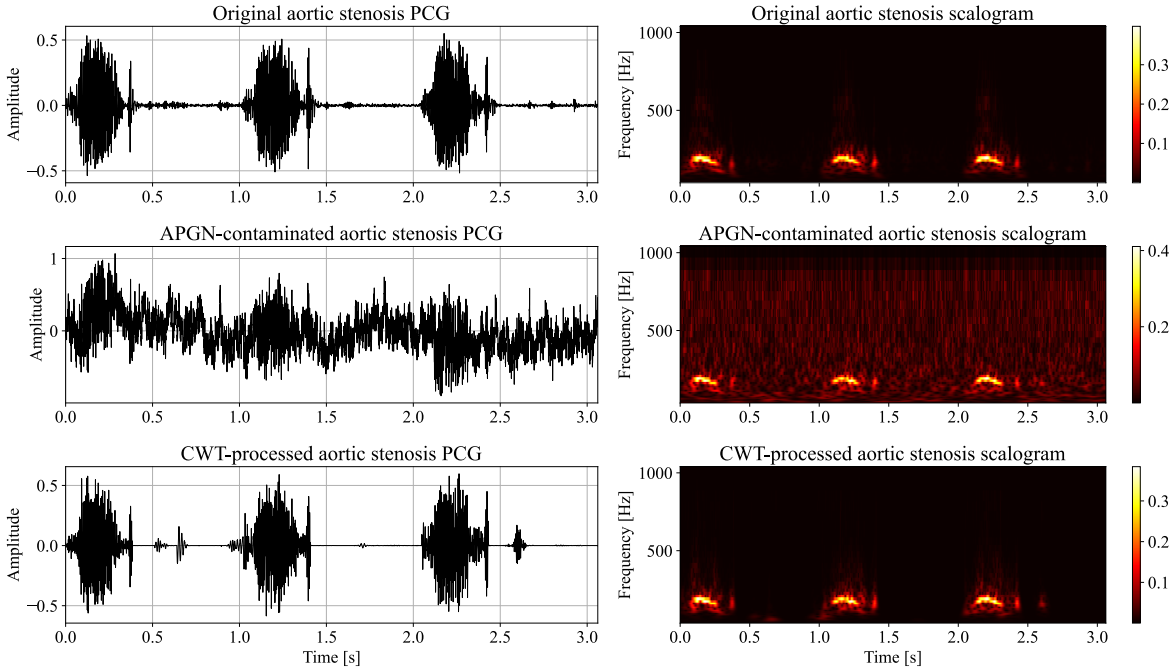


Figure A.2 AS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with APGN.

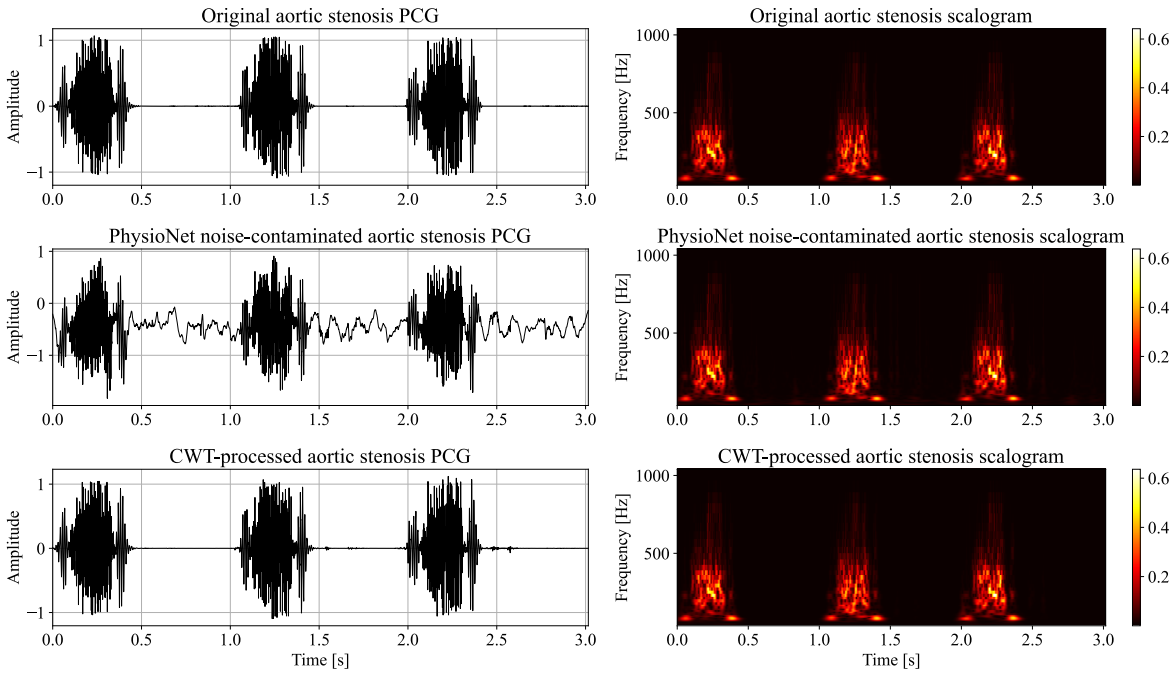


Figure A.3 AS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with PhysioNet noise.

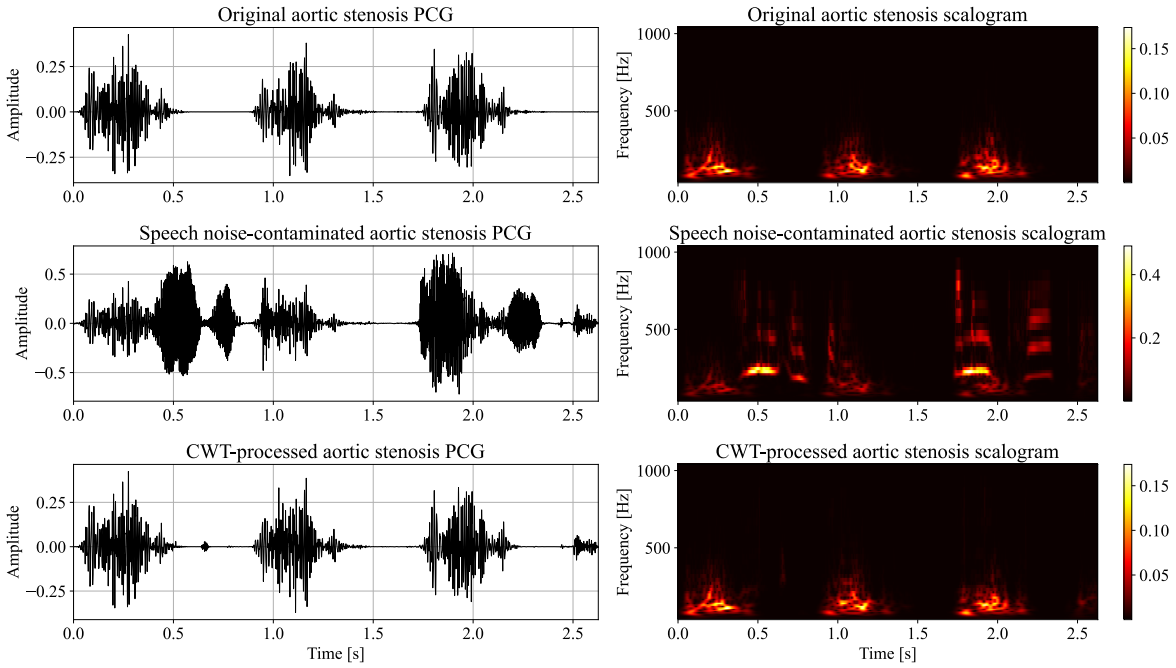


Figure A.4 AS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with speech noise.

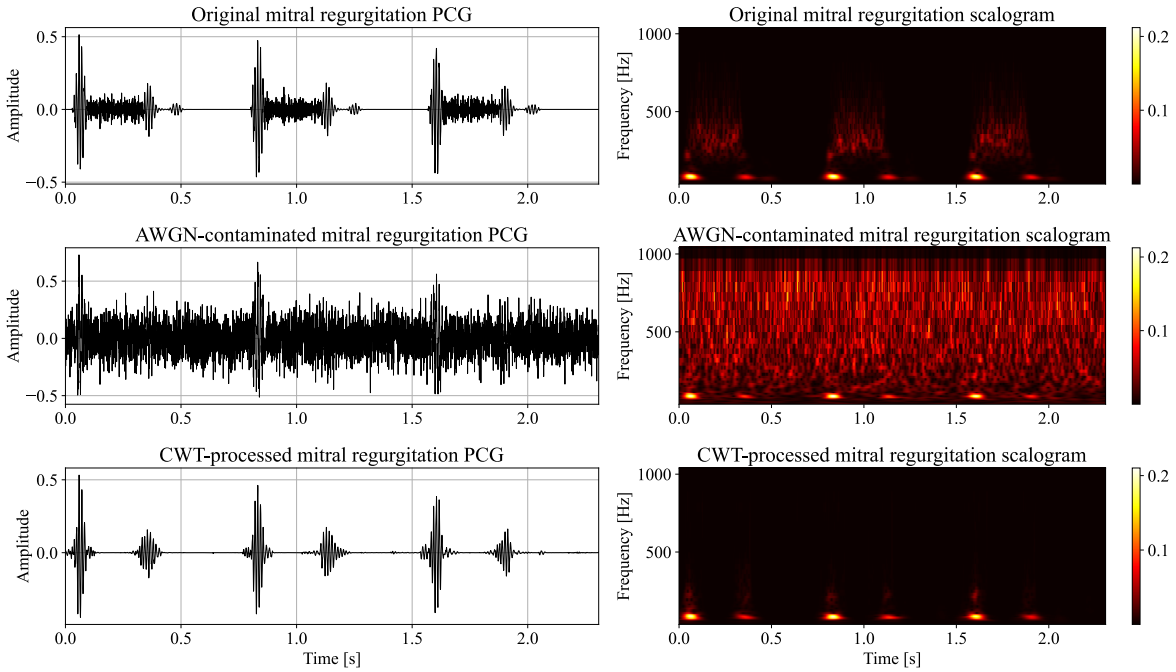


Figure A.5 MR PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with AWGN.

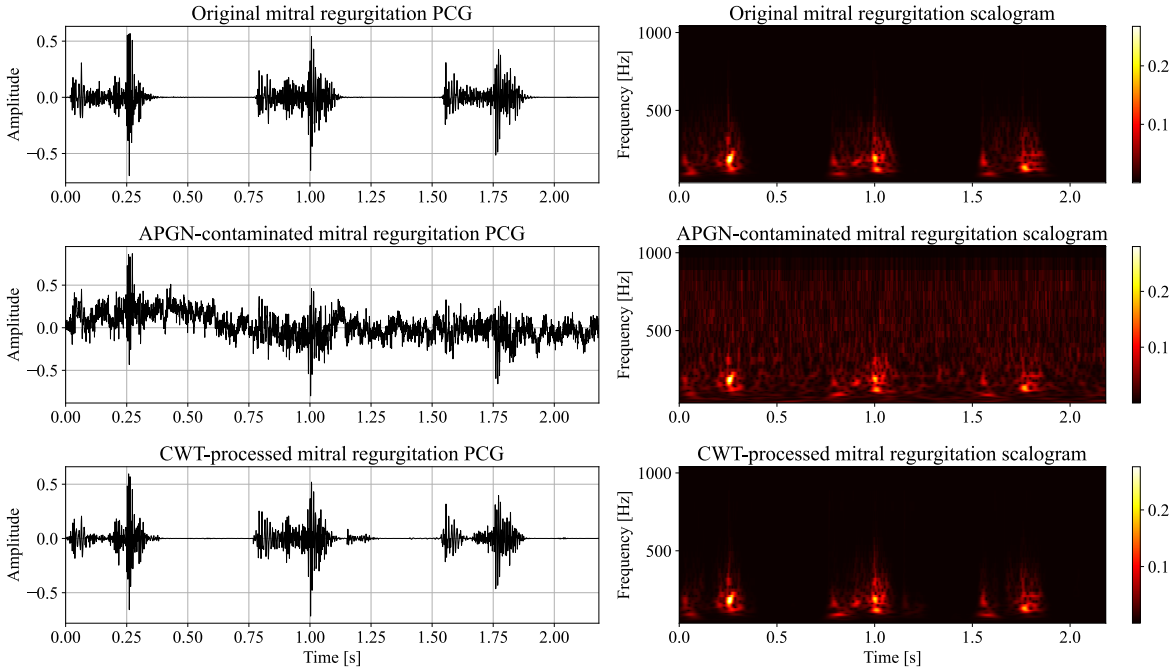


Figure A.6 MR PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with APGN.

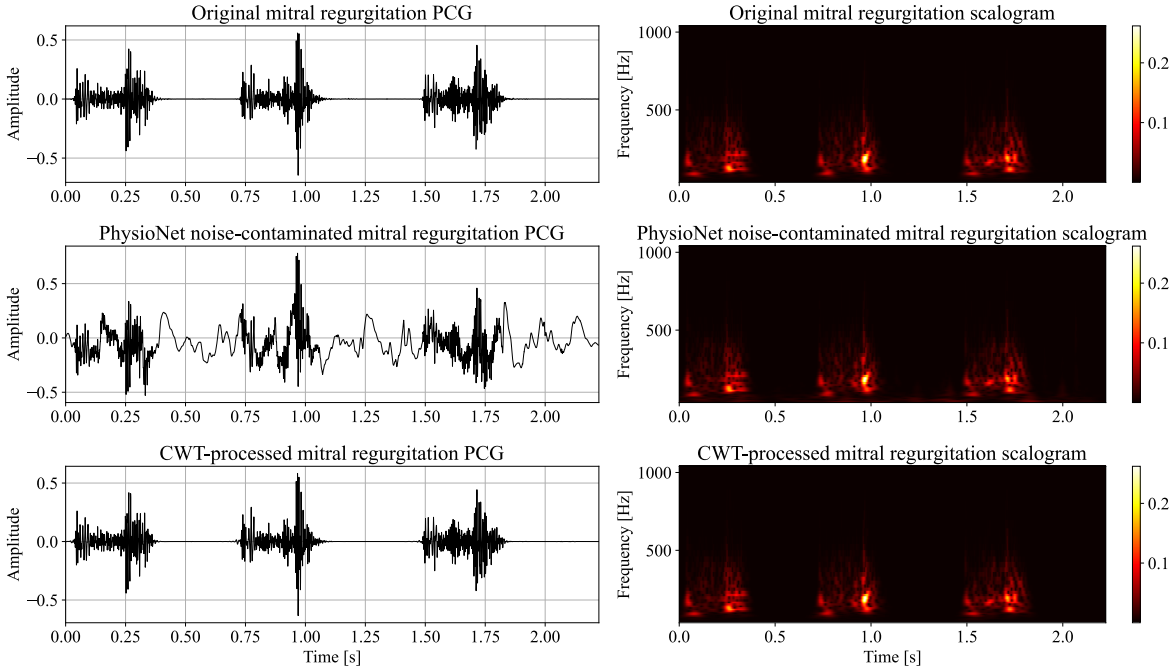


Figure A.7 MR PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with PhysioNet noise.

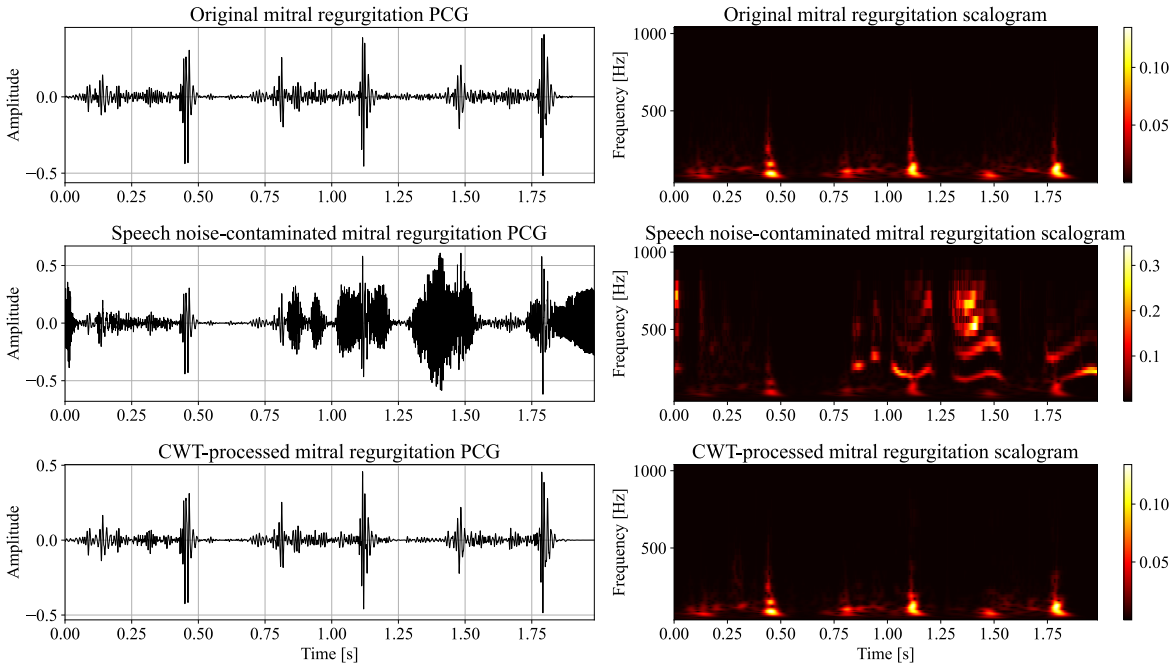


Figure A.8 MR PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with speech noise.

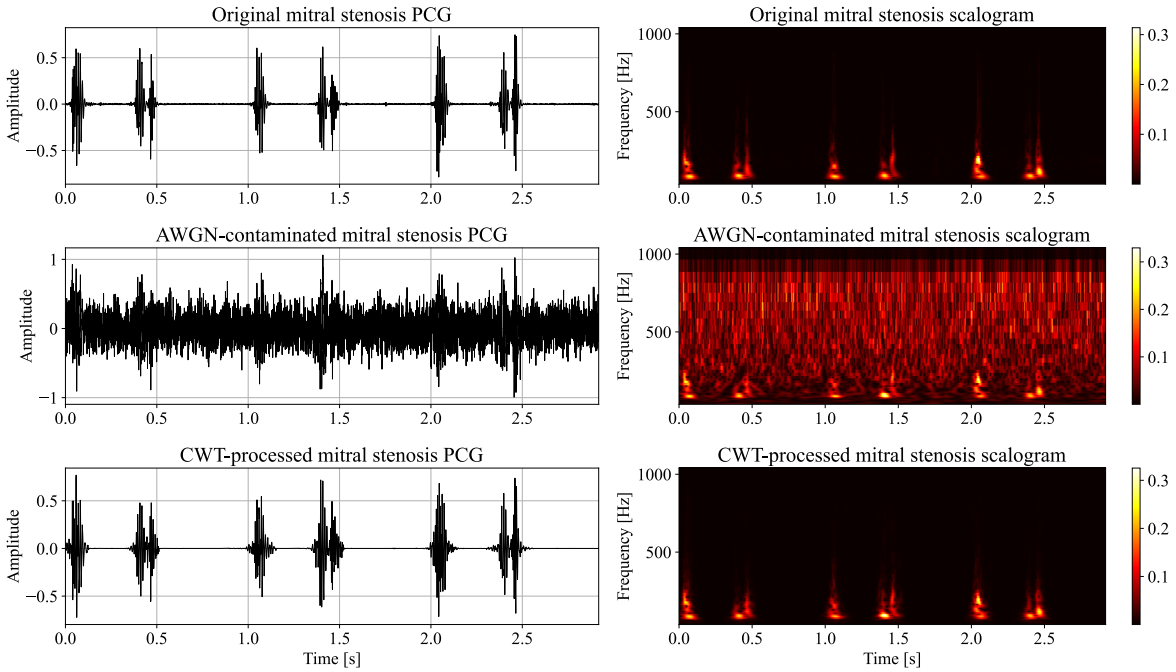


Figure A.9 MS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with AWGN.

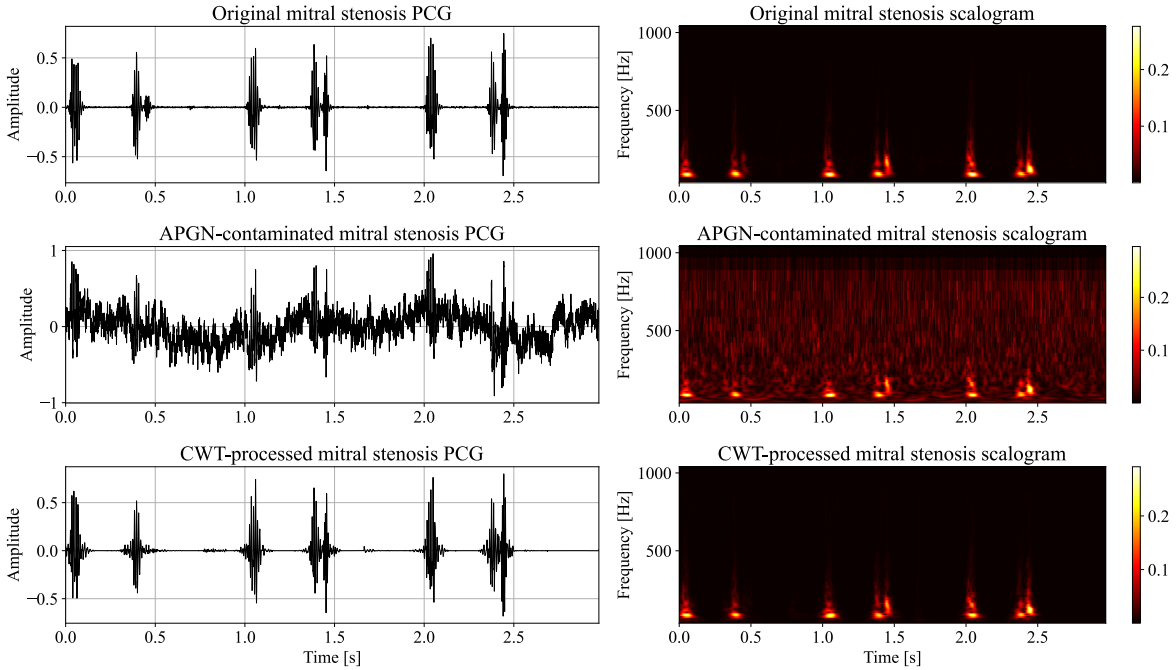


Figure A.10 MS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with APGN.

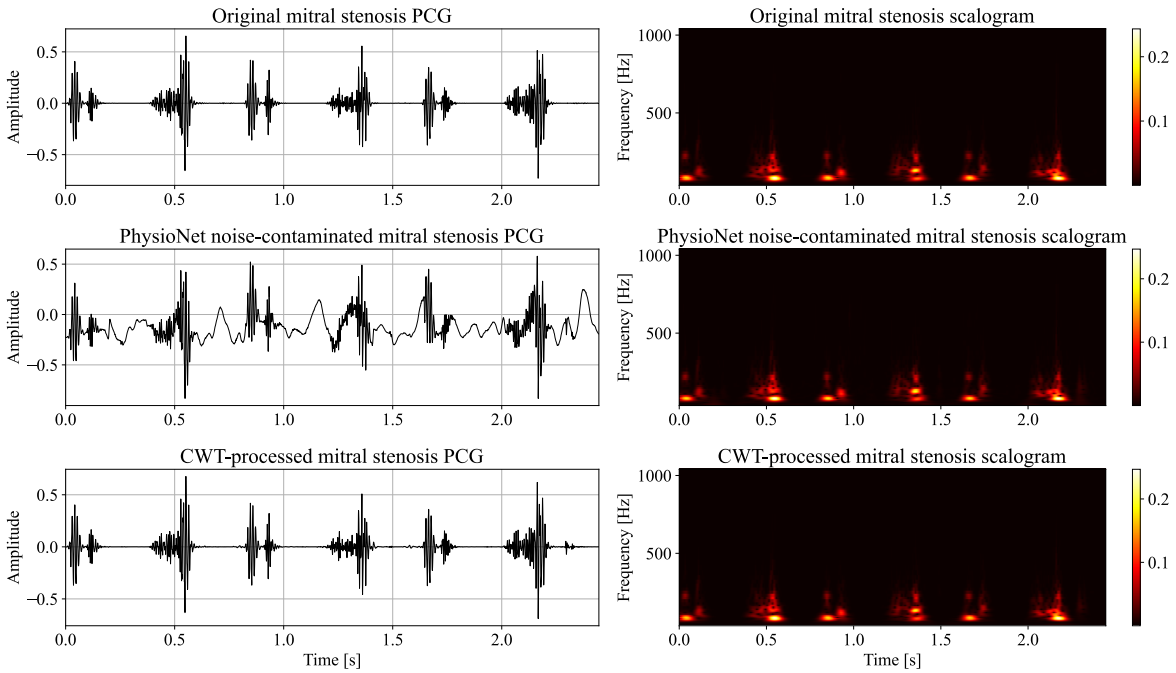


Figure A.11 MS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with PhysioNet noise.

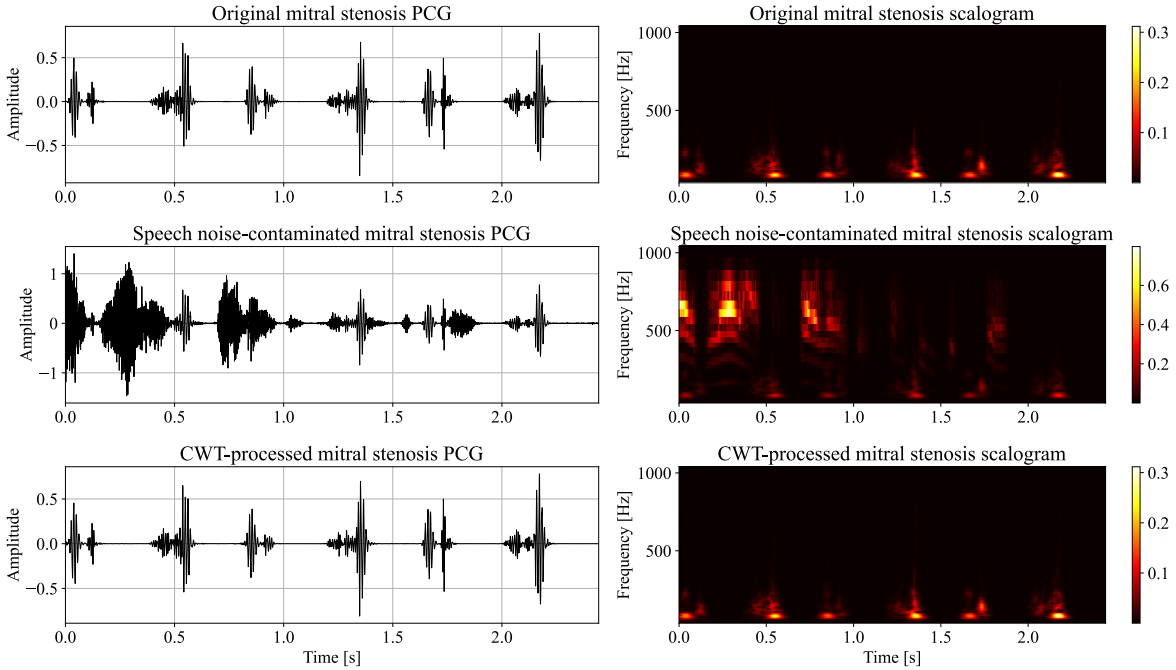


Figure A.12 MS PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with speech noise.

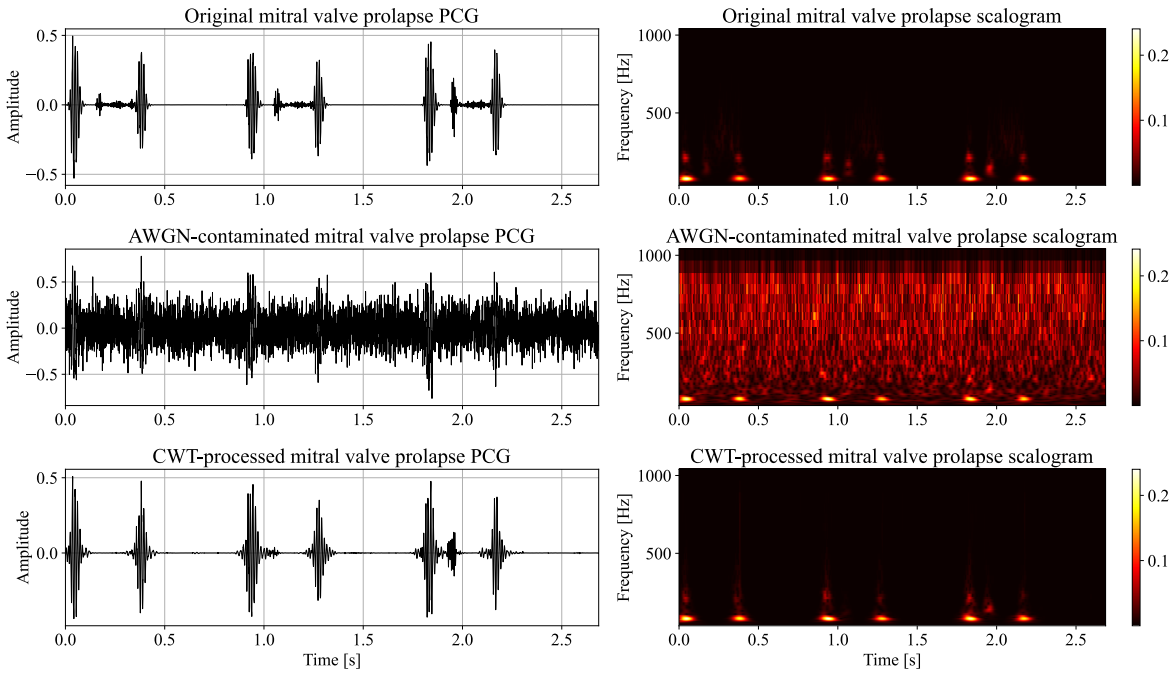


Figure A.13 MVP PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with AWGN.

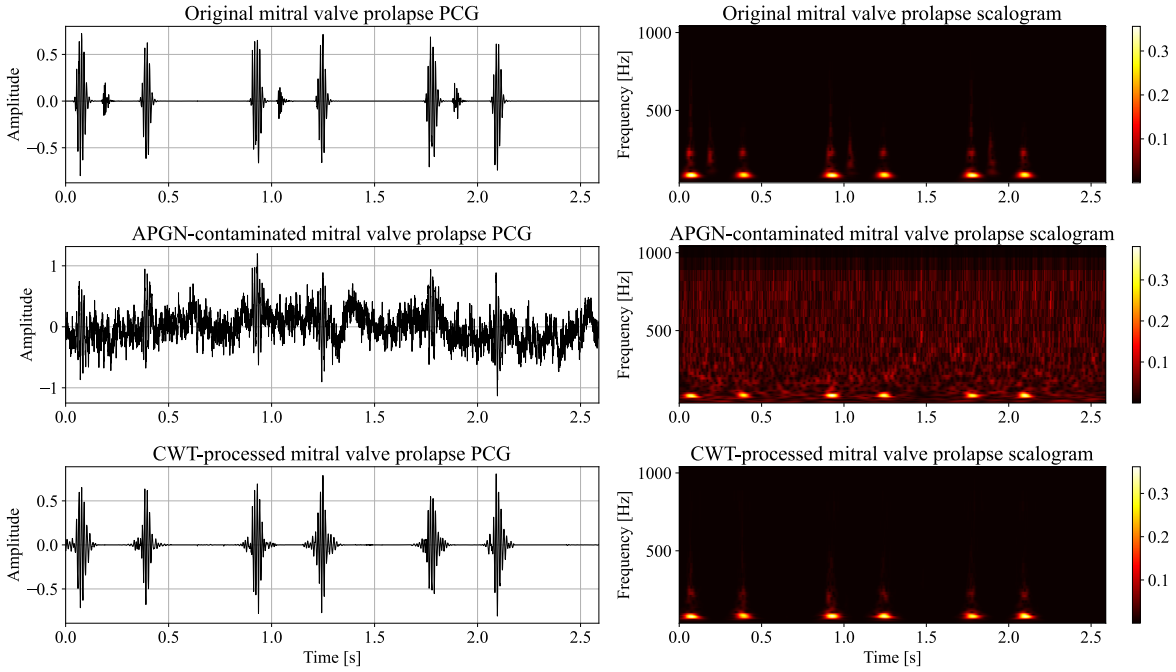


Figure A.14 MVP PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with APGN.

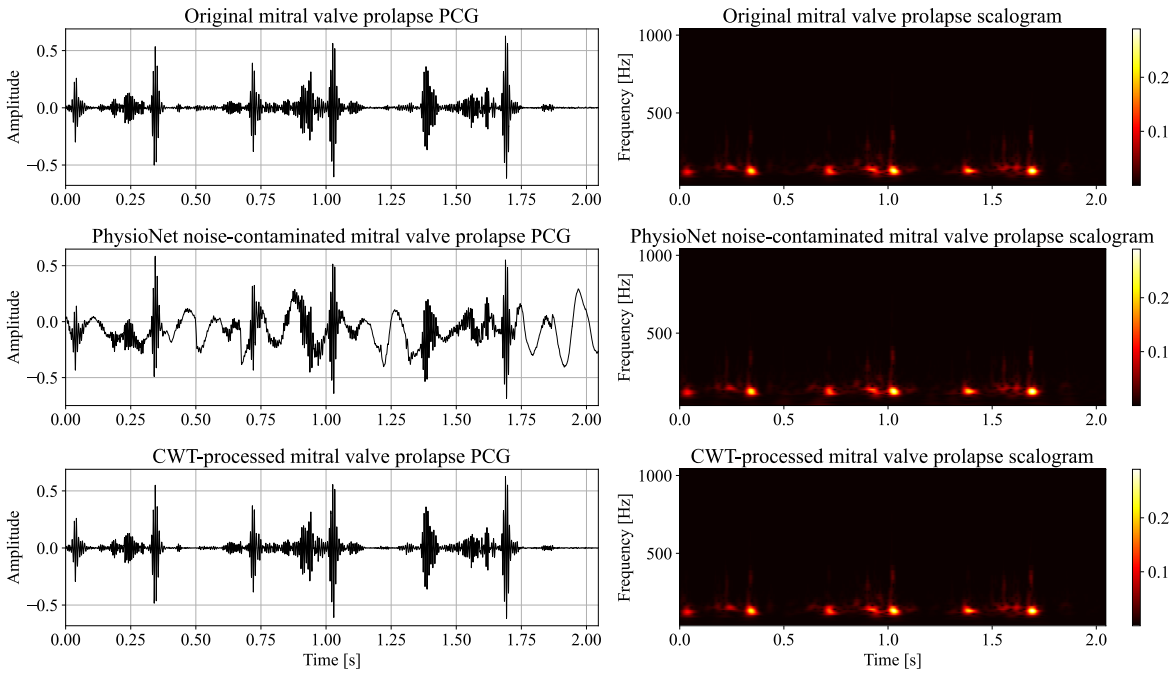


Figure A.15 MVP PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with PhysioNet noise.

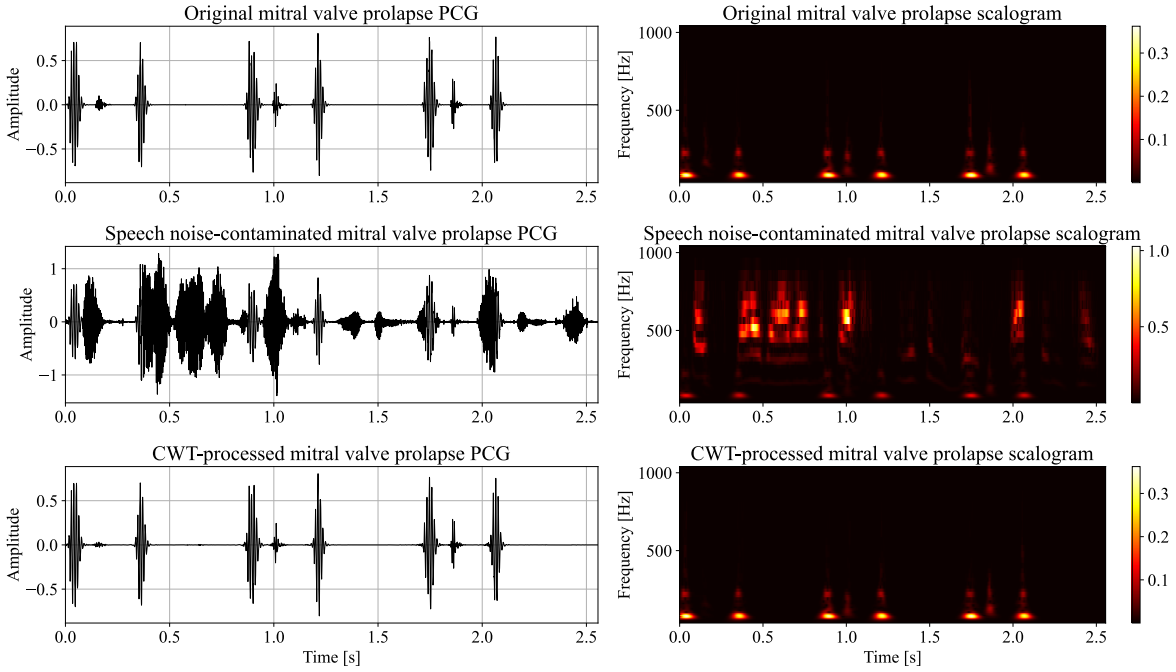


Figure A.16 MVP PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with speech noise.

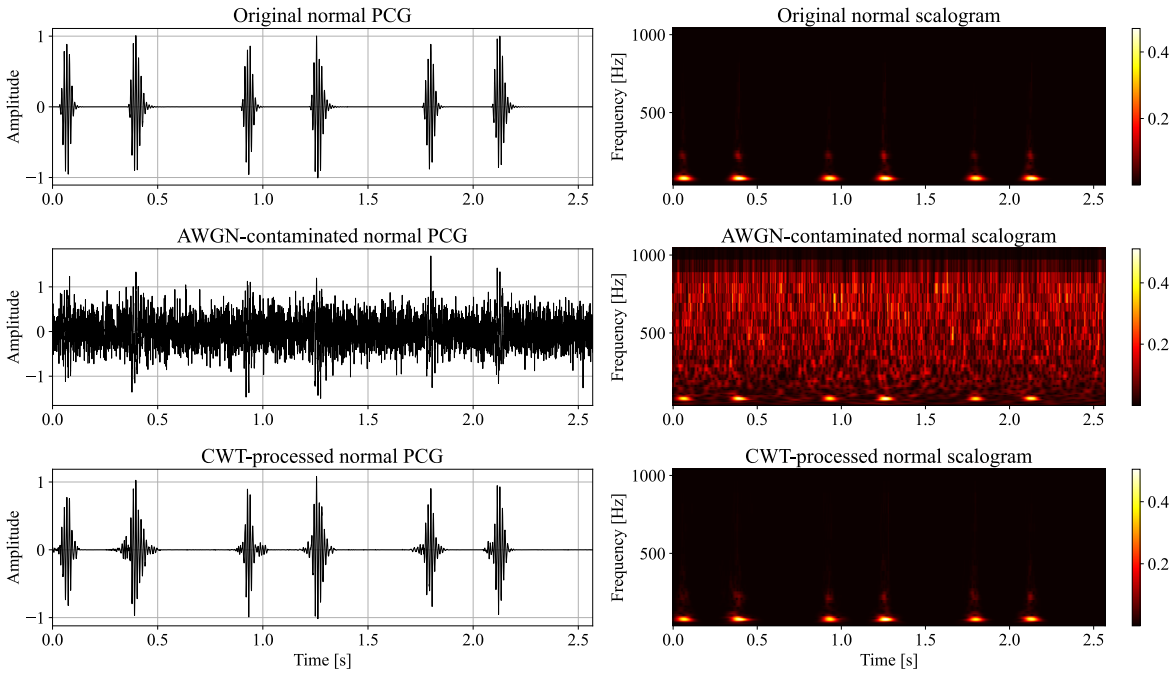


Figure A.17 Normal PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with AWGN.

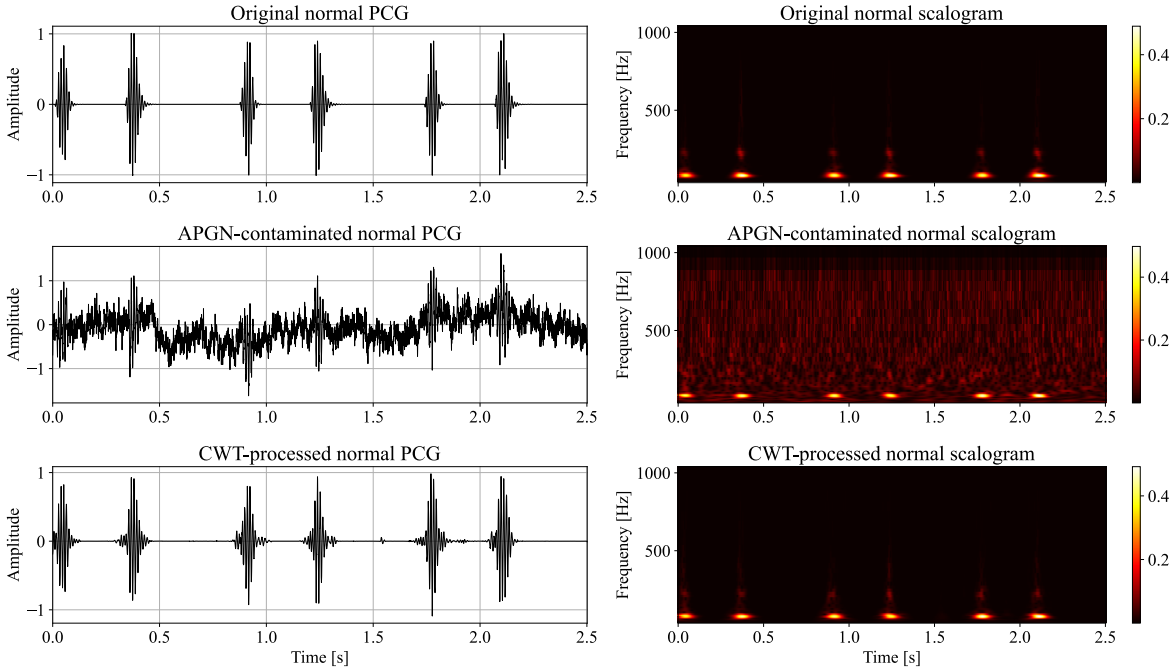


Figure A.18 Normal PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with APGN.

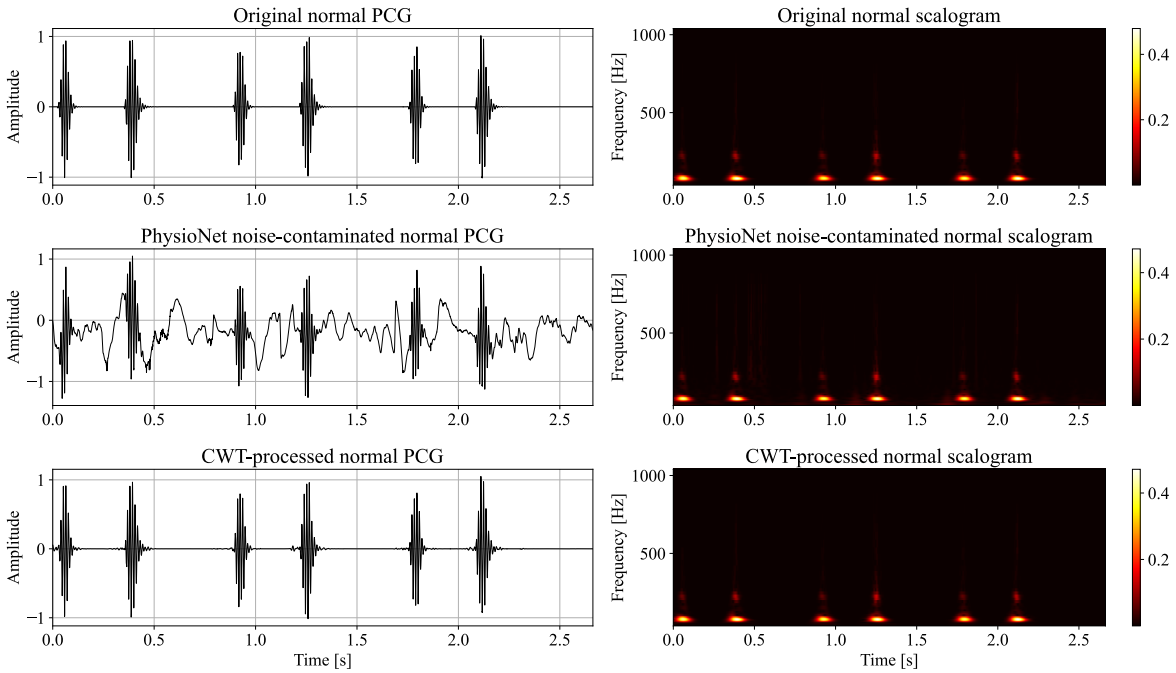


Figure A.19 Normal PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with PhysioNet noise.

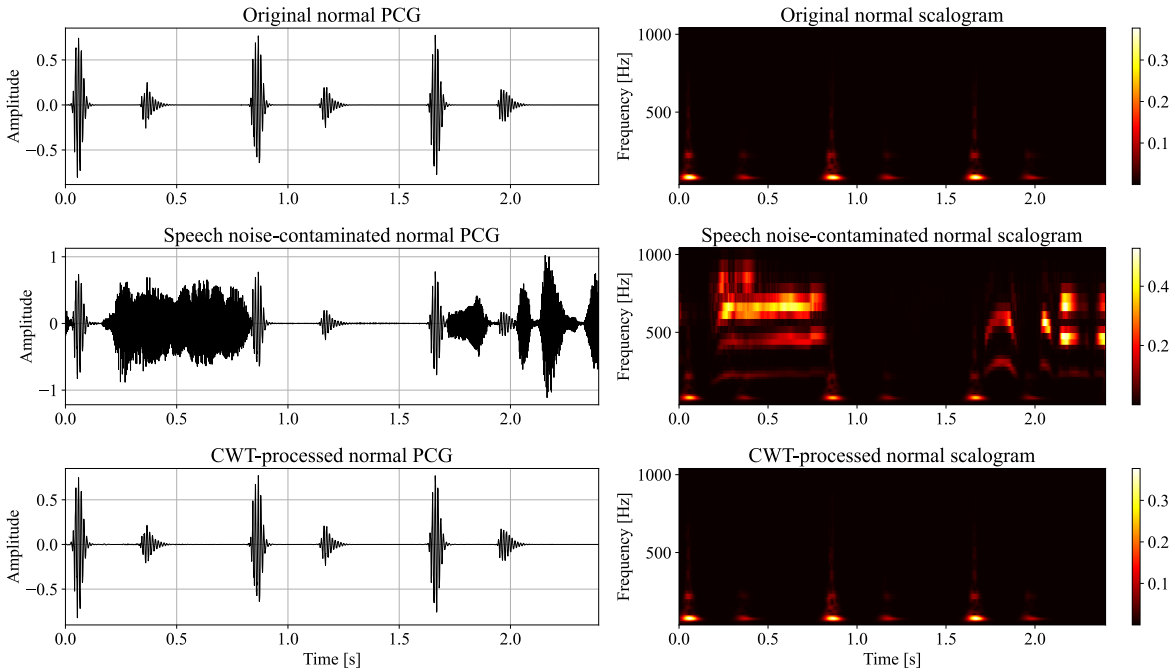


Figure A.20 Normal PCG example in which the CWT-trained model is employed to denoise a signal contaminated at -5 dB of SNR with speech noise.