Optimizing CleanUNet Architecture for Speech Denoising

Matheus Vieira da Silva



Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2024

Matheus Vieira da Silva

Optimizing CleanUNet Architecture for Speech Denoising

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: André Ricardo Backes
Coorientador: João Fernando Mari

Uberlândia
2024

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

S586o
2024

Silva, Matheus Vieira da, 1998-
Optimizing cleanUNet architecture for speech denoising [recurso eletrônico] / Matheus Vieira da Silva. - 2024.

Orientador: André Ricardo Backes.
Coorientador: João FernandoMari.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Programa de Programa de Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2024.5523
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Backes, André Ricardo, 1981-, (Orient.). II. Mari, João Fernando, 1981-, (Coorient.). III. Universidade Federal de Uberlândia. Programa de Programa de Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Rejâne Maria da Silva
Bibliotecária Documentalista – CRB6/1925

To my mother and my aunt.

Agradecimentos

To my aunt, Sandra Rodrigues dos Santos, who raised me with love and support and always believed in my dreams. To my mother, Ivany Vieira dos Santos, for her constant love and care. To my partner, Izabela Resende Costa, for her support and love throughout this project. To my advisors, Professor Dr. André Ricardo Backes and Professor João Fernando Mari, for their friendship, dedication, and guidance on this journey. To Professors Dr. Bruno Augusto Nassif Travençolo and Dr. Alexandre Luís Magalhães Levada, who kindly agreed to join the examination board for this work. I am grateful to the Federal University of Uberlândia and the Federal University of Viçosa, institutions that have supported my academic and professional growth.

"Our virtues and our failings are inseparable, like force and matter. When they separate, man is no more."

(Nikola Tesla)

Resumo

As técnicas de melhoramento de fala são cruciais para recuperar uma fala limpa a partir de sinais degradados por ruído e condições acústicas subótimas, como ruído de fundo e eco. Esses desafios exigem métodos eficazes de redução de ruído para melhorar a clareza da fala. Este trabalho apresenta uma versão otimizada da arquitetura CleanUNet, uma rede neural convolucional baseada na arquitetura U-Net, projetada explicitamente para tarefas de redução de ruído em fala causal. Nossa abordagem introduz a arquitetura Mamba como uma nova alternativa ao gargalo do modelo que utiliza *Transformer*, permitindo um processamento mais eficiente das saídas do codificador com complexidade linear. Além disso, integramos a normalização por lote (*batch normalization*) nas camadas convolucionais, estabilizando e acelerando o processo de treinamento. Também experimentamos várias funções de ativação para identificar a configuração mais eficaz para o nosso modelo. Ao reduzir o número de canais ocultos nas camadas convolucionais, reduzimos significativamente a quantidade de parâmetros do modelo, aumentando assim a velocidade de treinamento e inferência em uma única GPU, com uma ligeira degradação no desempenho. Essas melhorias tornam o modelo particularmente adequado para aplicações em tempo real. Nosso melhor modelo, 52.53% menor que o modelo base, alcançou 2,745, 3,288 e 0,911 pelas métricas PESQ (WB), PESQ (NB) e STOI, respectivamente. Também otimizamos um menor modelo usando apenas 1,36% dos parâmetros originais, atingindo resultados competitivos. Até onde sabemos, este trabalho é o primeiro a integrar a arquitetura Mamba como substituta do transformer padrão na CleanUNet e, em combinação com otimizações arquiteturais, oferece uma solução simplificada e mais eficiente computacionalmente para o melhoramento de fala.

Palavras-chave: Melhoramento de Fala. Aprendizado Profundo. Transformer. Redes Neurais Convolucionais..

Matheus Vieira da Silva



Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2024

UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU
FACULDADE DE COMPUTAÇÃO – FACOM
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO – PPGCO

The undersigned hereby certify they have read and recommend to the PPGCO for acceptance the dissertation entitled **"title"** submitted by **"student name"** as part of the requirements for obtaining the **Master's degree in Computer Science**.

Uberlândia, ___ de _____ de _____

Supervisor: _____
Prof. Dr. André Ricardo Backes
Universidade Federal de São Carlos

Cosupervisor: _____
Prof. Dr. João Fernando Mari
Universidade Federal de Viçosa

Examining Committee Members:

_____
Prof. Dr. Bruno Augusto Nassif Travençolo
Universidade Federal de Uberlândia

_____
Prof. Dr. Alexandre Luís Magalhães Levada
Universidade Federal de São Carlos

Abstract

Speech enhancement techniques are crucial for recovering clean speech from signals degraded by noise and suboptimal acoustic conditions, such as background noise and echo. These challenges demand effective denoising methods to improve speech clarity. This work presents an optimized version of CleanUNet, a Convolutional Neural Network based on the U-Net architecture designed explicitly for causal speech-denoising tasks. Our approach introduces the Mamba architecture as a novel alternative to the traditional transformer bottleneck, enabling more efficient handling of encoder outputs with linear complexity. Additionally, we integrated batch normalization across the convolutional layers, stabilizing and accelerating the training process. We also experimented with various activation functions to identify the most effective configuration for our model. By reducing the number of hidden channels in the convolutional layers, we significantly reduced the model's parameter count, thereby enhancing training and inference speed on a single GPU with slight degradation in performance. These improvements make the model particularly suitable for real-time applications. Our best model, 52.53% smaller than the baseline, achieves 2.745, 3.288, and 0.911 of PESQ (WB), PESQ (NB), and STOI, respectively. We also optimized the smallest model using only 1.36% of the original parameters, and it achieved competitive results. To the best of our knowledge, this work is the first to integrate the Mamba architecture as a replacement for the vanilla transformer in CleanUNet and, in combination with architectural optimizations, offers a streamlined, computationally efficient solution for speech enhancement.

Keywords: Speech Enhancement. Deep Learning. Transformer. Convolutional Neural Networks.

List of Figures

Figure 1 – Human ear. Source: (HAWKINS, 2024) 18
Figure 2 – STFT algorithm step on waveform signal. 19
Figure 3 – Short-Time Fourier Transform (STFT) spectrogram. 20
Figure 4 – Audio sample lengths distributed across the dataset. 22
Figure 5 – LeNet CNN architecture. Source: (LECUN et al., 1998). 26
Figure 6 – U-Net architecture. Source: (RONNEBERGER; FISCHER; BROX, 2015). 29
Figure 7 – Transformer Architecture. Source: (VASWANI et al., 2017). 32
Figure 8 – CleanUNet architecture. Source: (KONG et al., 2022). 33
Figure 9 – ReLU activation. 34
Figure 10 – Selection Mechanism with hardware-aware State Expansion. Source: (GU; DAO, 2023). 37
Figure 11 – Dilated attention mechanism patterns. Source: (DING et al., 2023). . . 38
Figure 12 – Proposed model architecture. 43
Figure 13 – Leaky ReLU activation. 45
Figure 14 – Audio signal comparison between clean and enhanced signals. The red and blue areas represent the amplitude difference in decibels (dB) between these two signals across different frequencies and over time. . 49
Figure 15 – Correlation matrix of the metrics against the number of blocks and the sum of the filter sizes. 52
Figure 16 – Trade-off between computational efficiency and performance for Perceptual Evaluation of Speech Quality (PESQ) (NB). 53
Figure 17 – Trade-off between computational efficiency and performance for PESQ (WB). 54
Figure 18 – Trade-off between computational efficiency and performance for Short-Time Objective Intelligibility (STOI). 55
Figure 19 – Audio signal comparison between clean and enhanced signals for mamba-based architecture (Id 8). 56

Figure 20 – Audio signal comparison between clean and enhanced signals for mamba-based architecture (Id 18). 57

Acronyms list

ASR Automatic Speech Recognition

BN Batch Normalization

BAK Background Noise Intrusiveness

CNN Convolutional Neural Network

CNNs Convolutional Neural Networks

DNS Deep Noise Suppression

dB decibels

FT Fourier Transform

GANs Generative Adversarial Networks

GRUs Gated Recurrent Units

GPU Graphical Processing Unit

GPUs Graphical Processing Units

GLU Gated Linear Unit

GLUs Gated Linear Units

LSTM Long Short-Term Memory

LTI Linear Time Invariance

MAE Mean Absolute Error

MANNER Multi-view Attention Network for Noise ERasure

NB Narrowband

OVRL Overall Quality

PESQ Perceptual Evaluation of Speech Quality

PReLU Parametric ReLU

RNN Recurrent Neural Network

RNNs Recurrent Neural Networks

ReLU Rectified Linear Unit

SDR Signal-to-Distortion Ratio

SIG Speech Signal Distortion

SSM State Space Model

SSMs State Space Models

STFT Short-Time Fourier Transform

STOI Short-Time Objective Intelligibility

WB Wideband

Contents

1 INTRODUCTION 13
1.1 Motivation 14
1.2 Goals 14
1.3 Hypothesis 15
1.4 Contributions 15
1.5 Thesis Organization 15

2 FUNDAMENTALS 17
2.1 Sound Representation and Hearing 17
2.2 Short-Time Fourier Transform 17
2.3 Signal Processing in Speech Enhancement 19
2.4 Dataset 21
2.5 Evaluation Metrics 21
2.5.1 Perceptual Evaluation of Speech Quality 21
2.5.2 Short-Time Objective Intelligibility 23

3 DEEP LEARNING IN SPEECH ENHANCEMENT 25
3.1 Recurrent Neural Networks 25
3.2 Convolutional Neural Networks 26
3.3 Generative Neural Networks 27
3.4 U-Net 28
3.5 Transformers 29
3.6 CleanUNet 31
3.6.1 Causal Convolution 33
3.6.2 Rectified Linear Unit 34
3.6.3 Gated Linear Units 34
3.6.4 Softmax 35
3.6.5 Transformer Bottleneck 35

3.7 Mamba Architecture . . . 36
3.8 LongNet . . . 36
3.9 Cost Function . . . 37
3.9.1 Waveform-Based L1 Loss . . . 37
3.9.2 Multi-Resolution STFT Loss . . . 38

4 RELATED WORKS . . . 39

5 PROPOSAL . . . 43
5.1 Leaky Rectified Linear Unit . . . 44
5.2 Bottleneck . . . 44
5.3 Batch Normalization . . . 45
5.4 Training Configuration . . . 46

6 EXPERIMENTAL RESULTS . . . 47
6.1 Dilated Attention-based CleanUNet Model . . . 47
6.2 Mamba-based CleanUNet Model . . . 48

7 CONCLUSION . . . 59
7.1 Main Contributions . . . 59
7.2 Limitations . . . 60
7.3 Future Works . . . 60

BIBLIOGRAPHY . . . 61

Chapter 1

Introduction

The COVID-19 pandemic and social isolation measures have brought about significant changes in the way people work. One of the changes was the rise in the use of calls and videoconferences to communicate and conduct work meetings. An illustration is the exponential growth of Zoom, which saw a significant increase in daily participants from 10 million in December 2019 to 300 million in April 2020, according to a study conducted by (KARL; PELUCHETTE; AGHAKHANI, 2022). With the pandemic forcing a rapid transition to remote work and virtual meetings, many employees faced a notable challenge: dealing with the adverse effects of poor acoustics in their home offices during video conferences. The rapid shift to remote work during the pandemic led to changes in residential spaces. However, the acoustic conditions in these spaces were not ideal, resulting in increased background noise and echo, making it more challenging to listen to video calls. The absence of a suitable setting in domestic areas can negatively impact the auditory experience and work efficiency.

The issue of poor acoustics is not exclusive to home offices during video conferences, specifically audio conferencing—the challenge of background noise disrupting communication dates back to the inception of the digital revolution. In 1948, Claude Shannon published a paper titled "A Mathematical Theory of Communication" (SHANNON, 1948), which laid the groundwork for the digital information era. Shannon demonstrated how to effectively transmit messages over channels prone to noise. Today's audio transmission operates under the same principles as before, combining a clear original transmission with various distractions such as barking dogs, crying babies, honking horns, and echoing rooms. Shannon illustrated the concept of breaking down analog information into separate packets, which enables the correction of errors, reduction of noise, and precision of the digital medium compared to the analog. However, the issue persists in remote meetings due to the transmission of analog sound waves through digital microphones. Although Shannon made significant advancements in information scanning and noise reduction, achieving the ideal teleconference remains uncertain. The solution is constructing a framework to convert noisy analog inputs into pure digitized outputs, aligning with Shan-

non's vision. Recent developments in machine learning and signal processing have paved the way for significant advancements in real-time speech enhancement. Advanced digital signal processing can effectively remove background noise from audio, making it suitable for applications in podcasts and video conferences where clear communication is crucial. The clarity and comprehensibility of the audio significantly impact the overall listening experience. Speech enhancement techniques reduce the adverse effects of background noise, reverb, and other distortions. These techniques leverage advanced algorithms and filters to suppress unwanted noise, enhance speaker voices, and improve recorded or transmitted audio quality. One notable example is CleanUNet (KONG et al., 2022), a cutting-edge deep learning approach that employs a U-Net architecture designed explicitly for speech denoising. The aim is to ensure that the voice is clear, understandable, and accessible to interference, providing a pleasant and effective hearing experience for all involved in communication. Therefore, enhancing speech is crucial in transmitting and recording audio, particularly in scenarios where the quality of the voice may be affected by external factors. Audio processing techniques can enhance sound quality significantly and improve clarity, comprehension, and communication effectiveness across various professional and entertainment settings (LOIZOU, 2013).

1.1 Motivation

Research in speech enhancement is driven by the necessity of enhancing voice communication quality in many scenarios. Noise in the background, like a dog barking, a baby crying, or a traffic whistle, can significantly affect speech clarity, making it difficult to understand the intended messages. Speech clarity is essential in telephone calls, video conferencing, and podcasts, where clear speech is crucial for effective communication. The pursuit of noise reduction solutions seeks to enhance the auditory experience by ensuring transparent and easily intelligible speech, irrespective of the noisy surroundings in which communication occurs.

1.2 Goals

The main goal is to optimize the number of parameters in CleanUNet architecture and reduce inference time for real-time applications while preserving performance. More specifically:

- ☐ Preprocess the audio dataset by cropping it into segments of n seconds to facilitate training on Graphical Processing Unit (GPU).
- ☐ Propose modifications to the model's architecture, including replacing the bottleneck, reducing the number of out channels in the encoder-decoder, incorporating

Batch Normalization (BN), and experimenting with different activation functions.

- ☐ Compare the baseline CleanUNet model with different versions that include the proposed modifications.

- ☐ Evaluate the models' performance using STOI and PESQ metrics and analyze which scenarios each model best suits by considering the number of parameters, inference time, and metric outcomes.

1.3 Hypothesis

This work is based on two main hypotheses:

1. It is possible to reduce the number of parameters in the architecture and reduce inference time without degrading performance or with only minimal performance degradation.

2. Incorporating architectural modifications such as replacing the bottleneck and reducing the number of blocks will affect speech intelligibility and perceptual quality differently.

1.4 Contributions

The contributions of this work are:

1. Development of an optimized architecture with a bottleneck replacement to efficiently reduce the number of parameters and inference time, thus consuming less energy and making it more suitable for real-time applications.

2. Evaluating and documenting the effects of specific architectural modifications, such as replacing bottlenecks, reducing out channels, adding batch normalization, and experimenting with different activation functions, could contribute valuable knowledge to deep learning-based speech processing.

1.5 Thesis Organization

The remainder of this thesis is organized as follows:

- ☐ In Chapter 2, we present the fundamentals of this work, covering essential concepts such as Sound Representation, the STFT, Signal Processing in Speech Enhancement, Datasets, and Evaluation Metrics.

- In Chapter 3, we describe the state-of-the-art techniques in speech enhancement and deep learning, including detailed discussions on architectures like RNNs, CNNs, U-Net, Transformers, CleanUNet, and LongNet.
- In Chapter 4, we review related works that provide a foundation and context for the current research, highlighting previous approaches and their outcomes in speech enhancement.
- In Chapter 5, we present the proposed modifications to the model architecture, including the introduction of Leaky ReLU, Bottleneck structures, and Batch Normalization. We also detail the training configurations employed to optimize performance.
- In Chapter 6, we analyze the experimental results obtained from testing various models, such as the Dilated Attention-based CleanUNet and the Mamba-based CleanUNet, and discuss their impact on evaluation metrics like STOI and PESQ.
- Finally, in Chapter 7, we draw conclusions from the research, discuss the study's limitations, suggest directions for future work, and summarize the contributions made to the literature.

Chapter 2

Fundamentals

Understanding how sounds are captured and processed is crucial in speech enhancement. Mathematical techniques enable the analysis of audio frequencies over time, which is essential for filtering out noise and improving speech clarity. Practical evaluation relies on specially designed benchmarks to assess how enhancements affect sound quality and intelligibility, guiding technological advancements.

2.1 Sound Representation and Hearing

The propagation of mechanical waves through a medium, such as air, produces sound. When an object vibrates, the surrounding air molecules undergo compression and rarefaction, propagating pressure variations as a sound wave. A microphone captures the pressure differences and converts them into electrical signals, and the microphones contain a membrane within their capsules that vibrates in sync with the sound waves. Consequently, a voltage corresponds to the sound waveform in the time domain. The sound signal can be represented in the frequency domain using the Fourier Transform (FT) to display the spectrum of frequencies (OLSON, 1967).

Hearing involves the transformation of sound vibrations into nerve impulses. The sound waves are captured by the ear and converted to nerve signals sent to the brain. The ear discerns aspects like loudness and pitch, with pitch related to the frequency of sound waves measured in hertz, and humans hear frequencies from 20 to 20,000 Hz. The sound wave intensity is measured in decibels, and the ear's response to these waves involves complex mechanical and neural processes, including the tympanic membrane and cochlea (HAWKINS, 2024). See Figure 1 for an illustration of the human hearing mechanism.

2.2 Short-Time Fourier Transform

The STFT is a fundamental technique in signal processing used to analyze the frequency content of a signal as it varies over time. Unlike the traditional Fourier transform,

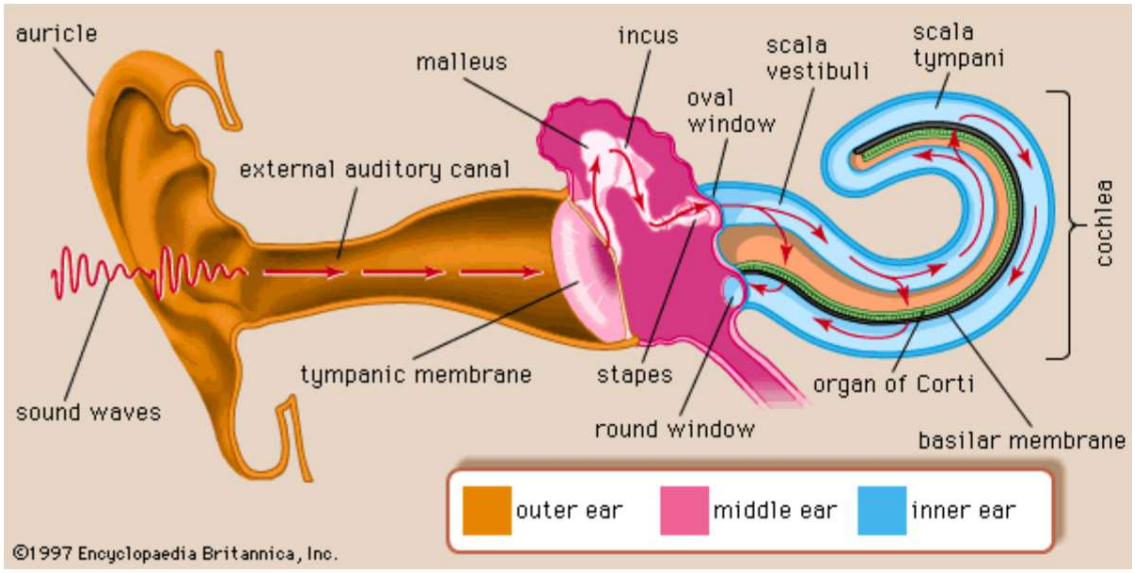


Figure 1 – Human ear. Source: (HAWKINS, 2024)

which provides only a frequency-domain representation of the entire signal, the STFT enables time-frequency representation, making it possible to observe how the spectral content evolves over time (DURAK; ARIKAN, 2003). The STFT of a signal $x(t)$ is defined as Equation 1:

$$\text{STFT}\{x(t)\}(t,\omega) = X(t,\omega) = \int_{-\infty}^{\infty} x(\tau) \cdot w(t-\tau) \cdot e^{-j\omega\tau}\, d\tau \tag{1}$$

where $x(t)$ is the input signal representing the time-domain data to be analyzed, $w(t)$ is a window function used to isolate a finite segment of the signal for each analysis step, τ is the integration variable representing time within the segment, ω is the angular frequency indicating the frequency component being analyzed, and j is the imaginary unit used to handle the complex exponentials that represent the signal's frequency components. The window function $w(t)$ plays a crucial role in determining the trade-off between time and frequency resolution, with common choices being rectangular, Hamming, and Hann windows.

Figure 2 presents a visual approach to the STFT algorithm. A window function is applied to a signal segment at each step to extract a finite portion for analysis, and the segment is multiplied by the window function, which tapers the segment at the edges to reduce discontinuities and spectral leakage. After analyzing one segment, the window is shifted by a specified hop length. The hop length determines how much overlap there is between consecutive windows; a smaller hop length results in more significant overlap and finer time resolution, while a more considerable hop length results in less overlap and coarser time resolution. For each windowed segment, the Fourier transform is computed to obtain the frequency components of that segment, and this process is repeated for each shifted window, resulting in a time-frequency representation of the entire signal.

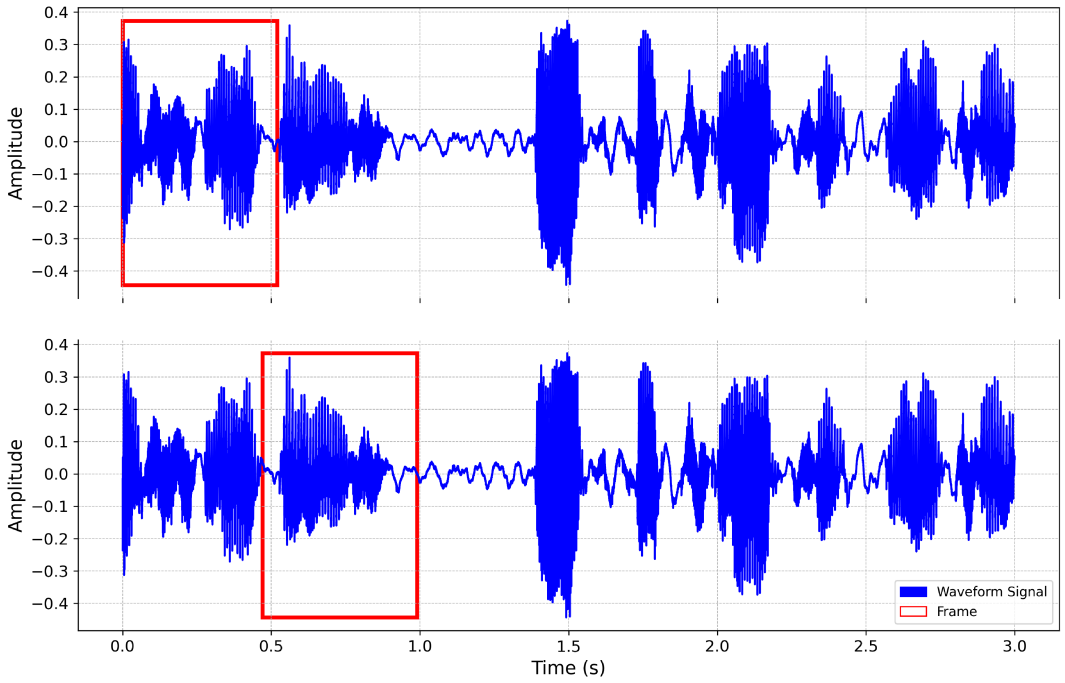


Figure 2 – STFT algorithm step on waveform signal.

Figure 3 presents the result of the STFT algorithm. The top panel of Figure 3 displays the waveform of the analyzed signal in the time domain. The horizontal axis represents time in seconds, while the vertical axis represents amplitude. This plot provides an overview of the signal's amplitude variations over time, highlighting the temporal structure and identifying regions of interest where significant changes in amplitude occur. The bottom panel of Figure 3 illustrates the STFT spectrogram of the same signal. The spectrogram offers a detailed time-frequency analysis, enabling examination of the signal's spectral content as it changes over time. The horizontal axis represents time, the vertical axis represents frequency (in Hz), and the color intensity represents the magnitude of the frequency components in decibels decibels (dB). Warmer colors, such as yellow and orange, indicate higher amplitude levels, while cooler colors, such as purple and black, indicate lower amplitude levels.

2.3 Signal Processing in Speech Enhancement

An efficient technique for minimizing background noise is spectral subtraction. This technique promptly decreases the amplitude of the noise components in the speech signal by estimating the average noise spectrum during silent periods and subtracting it from the input signal spectrum, and in comparison to other methods of noise reduction, this approach is computationally efficient. Nevertheless, the noise's unpredictability can lead

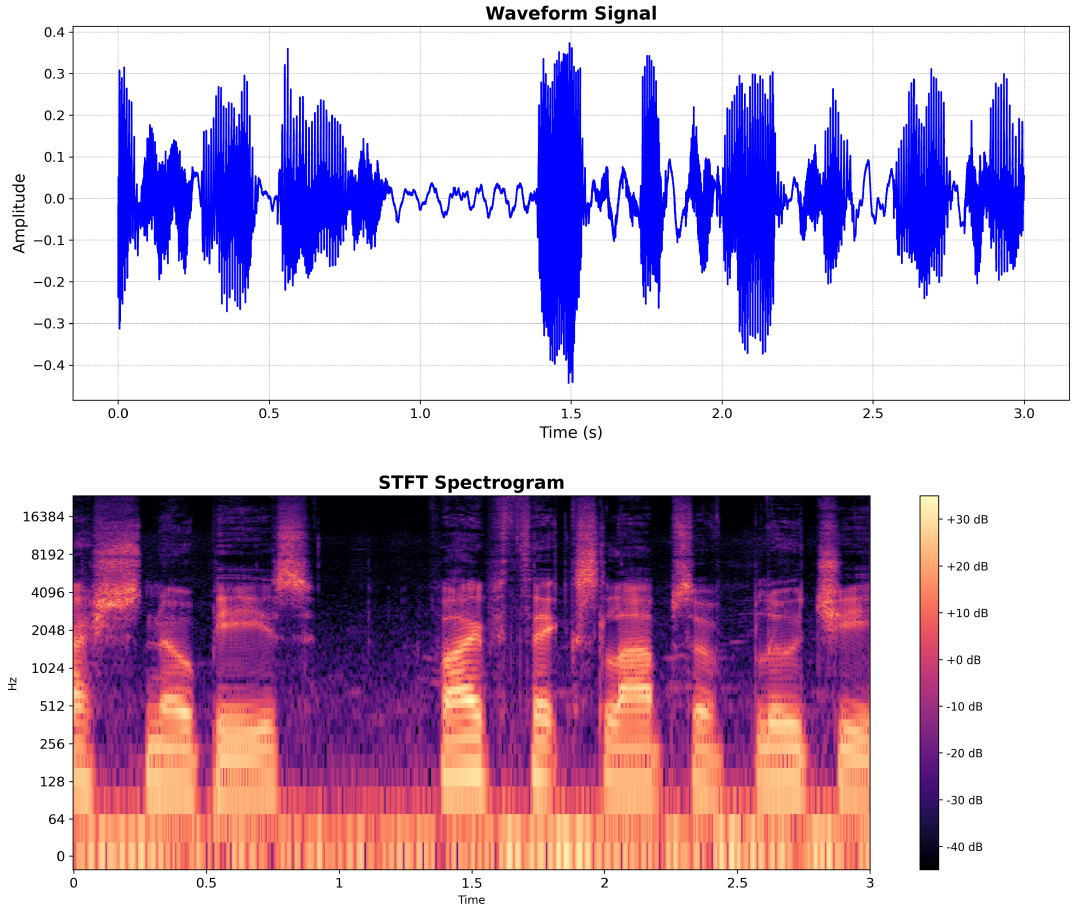


Figure 3 – STFT spectrogram.

to negative values when subtracting, necessitating rectifying such estimates to ensure they are non-negative. In situations with a low signal-to-noise ratio, distortions can arise due to this nonlinearity, with extensive research being conducted to mitigate these artifacts. Improving performance can be achieved by optimizing the subtraction factor, employing multiband processing, and utilizing a combination of subtraction and statistical estimation methods (VASEGHI; VASEGHI, 1996).

Another technique used to reduce noise in speech is Wiener filtering, a technique that is based on statistical estimates of the signal and noise; this approach dynamically creates a linear filter that can effectively suppress the noise in the speech signal (ROBINSON; TREITEL, 1967). Signal and noise can be represented as stochastic processes that possess established spectrum properties. The Wiener filtering technique generates a filter that diminishes the noise spectrum while maintaining the signal spectrum, thereby minimizing the average square error between the estimated and true signal. The filter can continuously adapt to changing noise conditions by updating its signal and noise spectral models over time. The filter coefficients are adjusted for optimal noise reduction in dynamic acoustic environments. Wiener filtering eliminates noise from real-world speech, particularly in

home office environments. Its self-adjustable and statistic-based design makes it a reliable choice for this purpose.

The statistical properties of stationary noise remain consistent and do not change significantly over time. Some examples are the constant whistling of the engine, the ongoing conversation in the background, and the consistent background noise, the power spectra and noise remain largely unaltered. The techniques mentioned earlier can decrease noise through stationarity, for optimal performance, techniques like spectral subtraction are most effective when dealing with continuous background noise that can be estimated by updating during periods of silence. Noises that are unstable and change rapidly pose a more significant challenge regarding estimation and suppression, unstable sounds exhibit fluctuating statistics over time, resulting in rapid changes in their characteristics. Some examples of abrupt sounds are broken glass, sporadic building noises, and sudden whispers. The signal's spectral composition and noise power exhibit irregular fluctuations over time.

2.4 Dataset

The database employed in this work was obtained from (VALENTINI-BOTINHAO et al., 2017) and comprises two separate datasets from 84 speakers with 28.4 hours of clean and noisy speech pairs designed to train and test speech enhancement models that operate at 48 kHz. The training set consists of 34,647 pairs of audio samples, while the test set comprises 824 pairs. Due to variations in the lengths of the audio samples within each pair, a preprocessing step was undertaken to normalize their sizes. This involved concatenating all the files and cropping a new file to a specified length, denoted as L. If the last file is smaller than L, it will be padded with the last valid value. Figure 4 presents a histogram showing the distribution of audio sample lengths in seconds.

2.5 Evaluation Metrics

In this work, we employed two objective evaluation metrics: the perceptual evaluation of speech quality PESQ and the short-time objective intelligibility STOI (LOIZOU, 2013). PESQ assesses the perceived audio quality by comparing the enhanced speech signal to the original clean signal. STOI evaluates speech intelligibility, indicating how well the enhanced signal retains the original speech's clarity.

2.5.1 Perceptual Evaluation of Speech Quality

The Perceptual Evaluation of Speech Quality (PESQ) model, defined by the ITU-T P.862 standard, predicts speech quality by simulating human auditory perception through

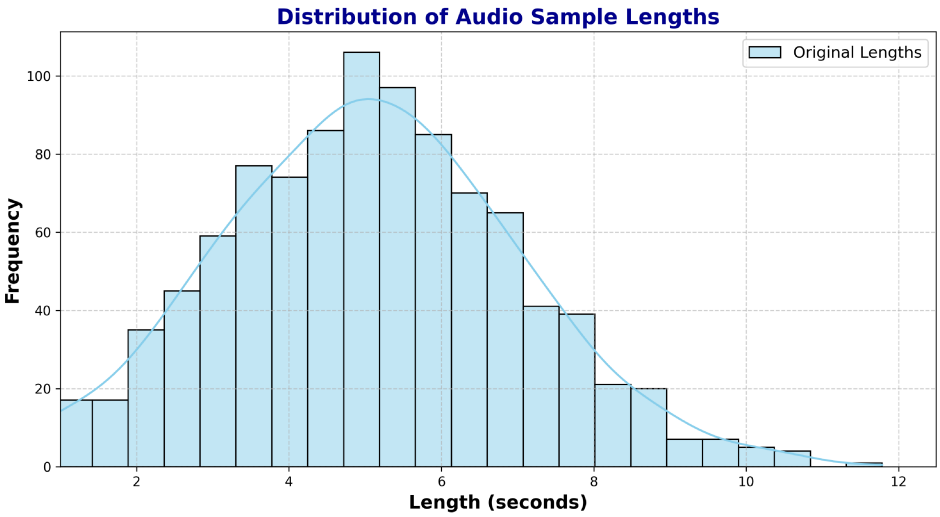


Figure 4 – Audio sample lengths distributed across the dataset.

a series of structured processes. These processes are carefully designed to capture the nuances of how humans perceive audio quality.

The first step involves the auditory transform, where the model computes the Bark spectrum by applying a fast Fourier transform (FFT) to overlapping signal frames, each with a duration of 32 milliseconds. This transformation is mathematically represented in the following equation:

$$S_b(f) = \mathrm{FFT}\{s(t) \cdot w(t)\} \tag{2}$$

Next, frequency equalization is performed to balance the reference and degraded signals. The frequency equalization is achieved by estimating the transfer function between these two signals, as described by the equation:

$$H(f) = \frac{\overline{S_{b,\mathrm{ref}}(f)}}{S_{b,\mathrm{deg}}(f)} \tag{3}$$

Following this, the model maps the signal to perceived loudness, which reflects how loud each time-frequency cell appears to the human ear. This mapping is computed using a loudness model, as shown in the equation:

$$L(t, f) = \mathrm{Loudness}(S_b(t, f)) \tag{4}$$

Disturbance processing then comes into play, where the model accounts for the audible differences between the reference and degraded signals. This process includes a masking function, which reduces the impact of smaller differences by considering them inaudible under certain conditions. The equation for this masking process is:

$$D_m(t,f) = \max(0, |L_{\text{deg}}(t,f) - L_{\text{ref}}(t,f)| - \frac{L_{\text{ref}}(t,f)}{4}) \tag{5}$$

An asymmetry factor is then applied to address additive distortions that may affect the degraded signal more than the reference signal. This factor is crucial in situations where certain types of distortions are more prominent, and it is calculated using the following formula:

$$A(t,f) = \left(\frac{S_{b,\text{deg}}(t,f)}{S_{b,\text{ref}}(t,f)} \right)^{1.2} \tag{6}$$

The model then aggregates the disturbance values using an Lp norm, which comprehensively measures the total disturbance across the entire signal. The equation for this aggregation is:

$$L_p = \left(\frac{1}{N} \sum_{m=1}^{N} |D(m)|^p \right)^{\frac{1}{p}} \tag{7}$$

Finally, it is important to note that the following equation is often used in literature to approximate the PESQ score as a linear combination of symmetric and asymmetric disturbance parameters:

$$\text{PESQ} = 4.5 - 0.1 \cdot d_{\text{SYM}} - 0.0309 \cdot d_{\text{ASYM}} \tag{8}$$

However, this equation does not directly represent the ITU-T P.862 standard. Instead, it provides a simplified interpretation of how the symmetric and asymmetric disturbances contribute to the overall PESQ score. The PESQ score is computed using more complex perceptual modeling steps, including time alignment, auditory transformation, loudness mapping, and disturbance processing (BEERENDS et al., 2002).

2.5.2 Short-Time Objective Intelligibility

The short-time objective intelligibility (STOI) measure is a robust metric developed to predict the intelligibility of speech signals by assessing the short-time temporal correlation between the original (clean) and processed (enhanced) speech signals across multiple frequency bands (TAAL et al., 2010). The method begins with resampling both signals to 10 kHz and ensuring precise time alignment.

Subsequently, the signals are divided into overlapping frames with a 50% overlap, windowed using a Hanning window, zero-padded to 512 samples, and then transformed via the Fourier transform. The resulting spectrograms are analyzed within one-third octave bands, where the energy for the j-th band in the m-th frame for the clean signal $X_j(m)$ and processed signal $Y_j(m)$ is computed as follows:

$$X_j(m) = \sqrt{\sum_{k=k1(j)}^{k2(j)-1} |\hat{x}(k,m)|^2} \tag{9}$$

$$Y_j(m) = \sqrt{\sum_{k=k1(j)}^{k2(j)-1} |\hat{y}(k,m)|^2} \tag{10}$$

Next, a local normalization procedure is applied to the processed speech by scaling it with a factor α, ensuring that its energy aligns with the clean speech energy within a short-time region M:

$$\alpha = \sqrt{\frac{\sum_{n\in M} X_j(n)^2}{\sum_{n\in M} Y_j(n)^2}} \tag{11}$$

The Signal-to-Distortion Ratio (SDR) for each time frame and frequency band is then computed using:

$$\mathrm{SDR}_j(n) = 10\log_{10}\left(\frac{X_j(n)^2}{(\alpha Y_j(n) - X_j(n))^2}\right) \tag{12}$$

To mitigate the effects of low SDR values, the processed signal undergoes a clipping process, defined as:

$$Y'_j(n) = \max\left(\min\left(\alpha Y_j(n), X_j(n) + 10^{-\beta/20} X_j(n)\right), X_j(n) - 10^{-\beta/20} X_j(n)\right) \tag{13}$$

The intermediate intelligibility measure, $d_j(m)$, is subsequently derived as the linear correlation coefficient between the clean and normalized processed signals within each band and frame:

$$d_j(m) = \frac{\sum_{n\in M}\left(X_j(n) - \frac{1}{N}\sum_{l\in M} X_j(l)\right)\left(Y'_j(n) - \frac{1}{N}\sum_{l\in M} Y'_j(l)\right)}{\sqrt{\sum_{n\in M}\left(X_j(n) - \frac{1}{N}\sum_{l\in M} X_j(l)\right)^2 \sum_{n\in M}\left(Y'_j(n) - \frac{1}{N}\sum_{l\in M} Y'_j(l)\right)^2}} \tag{14}$$

Finally, the STOI score, representing the overall predicted speech intelligibility, is obtained by averaging the intermediate intelligibility measures across all bands and frames:

$$d = \frac{1}{JM}\sum_{j=1}^{J}\sum_{m=1}^{M} d_j(m) \tag{15}$$

This score ranges from 0 to 1, with higher values indicating better-predicted intelligibility (TAAL et al., 2010).

Chapter 3

Deep Learning in Speech Enhancement

Deep learning involves computational models with multiple processing layers that learn data representations at various abstraction levels. These methods have significantly advanced fields like speech and visual object recognition, object detection, drug discovery, and genomics. Deep learning models, including deep convolutional and recurrent networks, excel at processing images, video, speech, audio, and sequential data. The backpropagation algorithm, combined with the computational power of GPUs, is crucial for adjusting internal parameters, facilitating the learning process, and achieving breakthroughs in many scientific and practical applications (LECUN; BENGIO; HINTON, 2015).

3.1 Recurrent Neural Networks

One of the first techniques employed to minimize speech noise was the adoption of Recurrent Neural Networks (RNNs). RNNs can sequentially process input sequences while maintaining an internal state that captures the context of time. Considering the presence of substantial time dependencies, this makes it well-suited for speech. The Recurrent Neural Network (RNN) is intended to systematically transform noisy speech inputs into clean destination outputs to minimize audio noise. Recurring connections can effectively reduce noise by simulating the contextual aspects of speech. The following equations can mathematically describe a RNN:

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h) \tag{16}$$

$$\hat{y}_t = \sigma(W_y h_t + b_y) \tag{17}$$

where h_t and x_t are the hidden states and input respectively, at time step t,W_h and, U_h are weight matrices for input and hidden state, and b_h is the bias vector for the hidden state. $\hat{y}_t$ represents the predicted output at time step t,W_y and b_y are the weight matrix

and the bias vector for the output. σ represents an activation function, typically a sigmoid or hyperbolic tangent (tanh).

Despite their advantages, RNNs have limitations, including difficulties in making accurate predictions and the requirement for significant computational resources. These challenges are primarily due to the complex nature of sequential data and the vanishing gradient problem, which can hinder the training of deep networks. Addressing these limitations often involves using more advanced architectures, such as Long Short-Term Memory (LSTM) networks (HOCHREITER; SCHMIDHUBER, 1997) or Gated Recurrent Units (GRUs) (CHO et al., 2014), designed to capture long-range dependencies better and mitigate gradient-related issues.

3.2 Convolutional Neural Networks

Convolutional Neural Networks (Convolutional Neural Networks (CNNs)) are a specialized artificial neural network architecture designed initially to process images. The main distinctive feature of a Convolutional Neural Network (CNN) is a convolutional layer, which applies a convolution operation to the input to produce maps of valuable features. Figure 5 illustrates an early version of CNN.

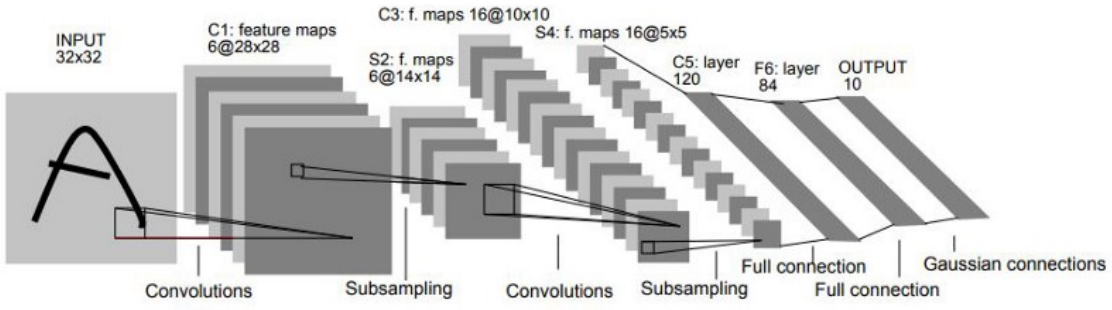


Figure 5 – LeNet CNN architecture. Source: (LECUN et al., 1998).

Let X be the input tensor with dimensions $C \times H_{in} \times W_{in}$, where C is the number of input channels, H_{in} and W_{in} are height and width, respectively. Let K the kernel with size $C \times H_k \times W_k$, where H_k and W_k are height and width. The convolution operation is performed by shifting the kernel with its respective weights over the input tensor and calculating the dot product of each position. The output tensor Y is computed as follows:

$$Y_{c,h,w} = \sum_{i=1}^{C} \sum_{j=1}^{H_k} \sum_{k=1}^{W_k} X_{i,h+j-1,w+k-1} \cdot K_{i,j,k}, \tag{18}$$

where $C \times H_{out} \times W_{out}$ are the dimensions of Y, $H_{out} = H_{in} - H_k + 1$ and $W_{out} = W_{in} - W_k + 1$ height and width of the output tensor.

The network can hierarchically extract top-level features from pixel data by stacking convolutional layers. By gradually reducing the spatial dimensions, the introduction of pooling layers helps to summarize the features and regulate the incidence of overfitting. Later layers in a traditional CNN architecture typically consist of fully connected layers with the softmax function to the output layer that generates a prediction based on the representation of the learned feature. The following equation describes the softmax function that encodes a real vector into a probability distribution, where e is Euler's natural constant, and x is the input vector.

$$\sigma(z) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{19}$$

CNNs have gained popularity as an alternative, initially discussed in the paper (LECUN et al., 1998). In the context of speech enhancement, it is common to use 1D CNNs that use local patterns to process raw audio directly to reduce noise, while 2D CNNs operate using spectrogram representations. Both scenarios involve the utilization of small convolutional filters to isolate the distinct features of local speech from any interfering background noise. Pooling and sampling reduction are methods used often to gather and condense information over a period of time. Several convolutional layers can acquire more advanced features, enabling the identification of speech elements even in environments with background noise.

A key benefit of CNNs is their utilization of weight sharing, which leads to a notable reduction in parameters compared to RNNs, which is more effective during training and inference. RNNs use the temporal context to their advantage, and in contrast, convolutional neural networks use localized filters to separate speech from noise in the time-frequency domain, effectively reducing the noise.

3.3 Generative Neural Networks

Generative Adversarial Networks (GANs), introduced by (GOODFELLOW et al., 2014), represent a revolutionary approach to generative modeling in machine learning. This architecture comprises a pair of neural networks, namely the generator G and the discriminator D, that participate in a dynamic and adversarial procedure to enhance their performances, and this interaction enables the generation of realistic data samples. The generator network G receives a random noise vector z from a prior distribution, such as a Gaussian distribution, and uses it to create a synthetic data sample $G(z)$. The goal of G is to generate data indistinguishable from the actual data distribution p data. In contrast, the discriminator network D functions as a binary classifier that calculates the likelihood that a specific sample x is real (derived from the training data) or fake (generated by the generator). The discriminator is trained to maximize the probability of correctly classifying real and fake data samples (GUI et al., 2021). The training of GANs involves

a minimax game where G and D are optimized simultaneously with opposing goals. The objective function can be formulated as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{20}$$

where D aims to maximize $\log D(X)$ for real samples and $\cdot \log 1 - D(G(z))$ for generated samples, while G minimizes $\log(1-D(G(z)))$. In each interaction, the discriminator is first trained on a batch of real samples x and generated samples $G(z)$, maximizing the ability to distinguish between real and generated data. Subsequently, the generator is trained by backpropagating the gradients through D to update G, producing more realistic data that deceives D. The adversarial training continues until the generator produces data that the discriminator cannot reliably distinguish from real data, achieving a Nash equilibrium (KREPS, 1989).

GANs have demonstrated application in various speech applications, including speech synthesis and enhancement. In speech synthesis, GANs can generate high-quality and natural-sounding speech, similar to human speech, in speech enhancement to improve the quality of speech signals by reducing noise and other artifacts.

3.4 U-Net

U-Net is a convolutional network architecture specifically designed for biomedical image segmentation. Traditional deep networks require extensive annotated data, often unavailable in biomedical applications. U-Net addresses this issue by extensively utilizing data augmentation to overcome the lack of annotated samples. This architecture, characterized by its contracting path for capturing context and a symmetric expanding path for precise localization, is designed to be trained end-to-end from a limited dataset, making it highly efficient in scenarios with scarce annotated images. The U-Net outperformed the previous best method, a sliding-window convolutional network, in the ISBI challenge to segment neuronal structures in electron microscopic stacks. Furthermore, it won the ISBI Cell Tracking Challenge 2015 (SOLÓRZANO et al., 2015) in multiple categories. Figure 6 presents the U-Net scheme.

The architecture consists of a contracting path followed by an expansive path, forming a U-shaped design. The contracting path includes repeated applications of 3×3 convolutions, each followed by a Rectified Linear Unit (ReLU) and a 2×2 max pooling operation for downsampling. The expansive path involves upsampling the feature map, followed by 2×2 convolutions that halve the number of feature channels, concatenating with the corresponding feature map from the contracting path, and applying two 3×3 convolutions with ReLU. This process results in a segmentation map that contains only the valid part of each convolution, enabling seamless segmentation of arbitrarily large images using an

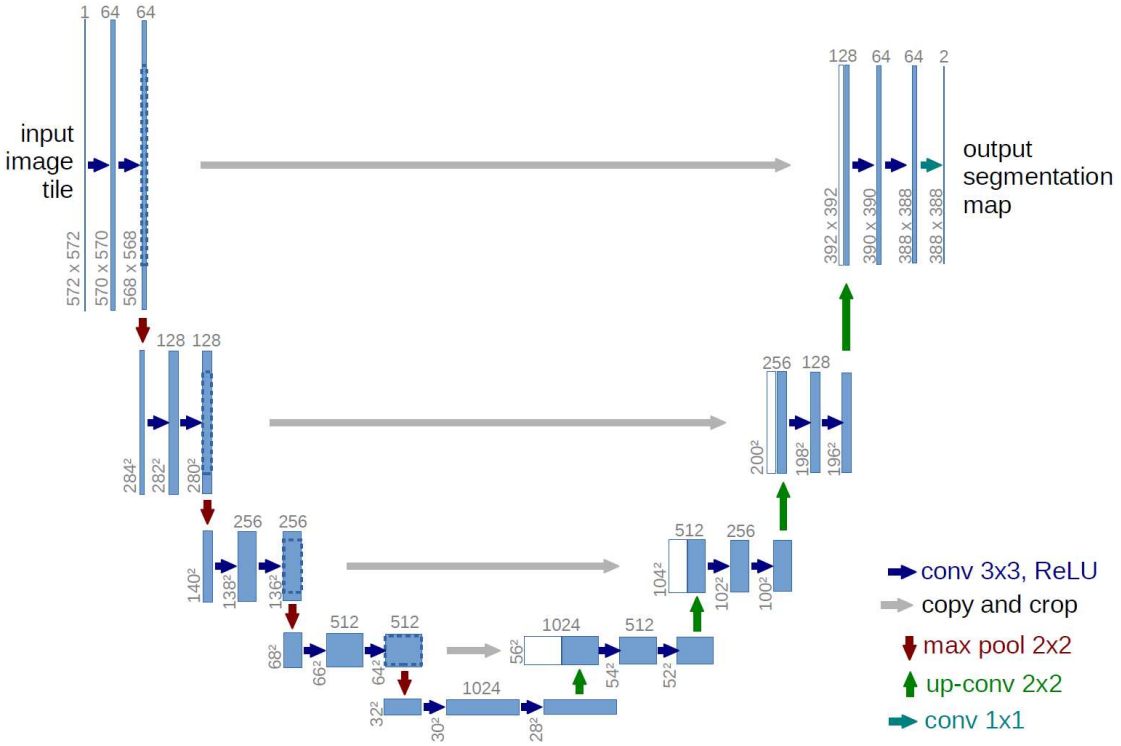


Figure 6 – U-Net architecture. Source: (RONNEBERGER; FISCHER; BROX, 2015).

overlap-tile strategy. The pixel-wise soft-max function used in the final layer is defined as:

$$p_k(x) = \frac{\exp(a_k(x))}{\sum_{k'} \exp(a_{k'}(x))} \tag{21}$$

where $a_k(x)$ denotes the activation in feature channel k at pixel position x. The cross-entropy loss function penalizes the deviation of the predicted output from the true label:

$$E = \sum_x w(x) \log(p_{\ell(x)}(x)) \tag{22}$$

where $l(x)$ is the true label of each pixel $w(x)$ is a weight map that compensates for class frequency imbalance and enforces the learning of border pixels between touching cells.

3.5 Transformers

The Transformer is an architectural design for neural networks specifically engineered to process sequential data and incorporate attention mechanisms to represent long-range dependencies. Currently, it has emerged as the benchmark for natural language processing tasks.

First introduced by (VASWANI et al., 2017), the transformer architecture uses self-attention mechanisms (BAHDANAU; CHO; BENGIO, 2014) to capture the dependencies

between different words or tokens in a sequence. It enables the model to weigh the importance of each token in the sequence concerning every other token. The key component of self-attention is the calculation of attention scores based on queries (Q), keys (K), and values (V). The mathematical representation of self-attention is described in the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{23}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V \tag{24}$$

where Q, K, V represent the queries, keys, and values matrices derived from the input matrix. $\sqrt{d_k}$ is a scaling factor, where d_k has the dimensionality of K and V. The softmax function normalizes the attention scores, and the result is used to compute a weighted sum of the V vectors. Equation 3 also includes the term M, a mask matrix with values $-\infty$ for future positions and 0 for positions up to and including the current token. This masked multi-headed attention mechanism ensures that predictions at each position depend solely on the known outputs preceding that position. This mechanism is vital for maintaining the autoregressive property of the model, which facilitates sequence generation one token at a time without accessing future tokens.

Inspired by CNNs, the architecture employs a variant of the self-attention technique called multi-headed attention to process input sequences simultaneously. The following expression illustrates the process:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{25}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{26}$$

where $head_i$ is the attention score, and W_i^Q, W_i^K, W_i^V and W_i^O are parameter matrices. During training, the computational complexity of computing attention scores for each token pair of results has a quadratic growth of $O(n^2)$ with an input sequence of length n.

The transformer architecture eliminates recurrence, removing the constraints that recurrence imposes on sequential tasks. Self-attention is the only attention mechanism considered when establishing global dependencies between input and output in the architecture. This configuration enables the representation of relationships among distant tokens in the input sequence.

Another addition to the architecture is positional encoding, a technique used to inject position-specific information into the input embeddings since Transformers process sequences in a non-sequential manner. This encoding is added to the input embeddings, enabling the model to differentiate between the positions of tokens in the sequence. The encoding is generated using sine and cosine functions of varying frequencies, ensuring

unique and smooth position-specific representations. The positional encoding for a position pos and dimension i is given by:

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad \mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right), \tag{27}$$

where d is the dimensionality of the embeddings. The Add & Norm step is essential for enhancing model stability and learning efficiency. This step consists of two fundamental operations: residual connections (Add) and layer normalization (Norm). Residual connections involve directly passing the input of a sub-layer, such as multi-headed attention or feed-forward networks, to its output, which can be mathematically represented as:

$$\mathrm{Output} = \mathrm{Layer}(X) + X, \tag{28}$$

where X is the input to the sub-layer, and $\mathrm{Layer}(X)$ is the output of the sub-layer. This bypass connection helps mitigate the vanishing gradient problem, facilitating better gradient flow during backpropagation and enabling the model to learn more effectively. Following the residual connection, layer normalization is applied to the resulting sum, which the equation can describe:

$$\mathrm{Norm}(X) = \frac{X - \mu}{\sigma + \epsilon} \cdot \gamma + \beta, \tag{29}$$

where μ and σ are the mean and standard deviation of the input tensor, respectively, ϵ is a small constant for numerical stability, and γ and β are learnable parameters. The combined Add & Norm operation is applied after the multi-headed attention and feed-forward sub-layers within each encoder and decoder block. The feed-forward sub-layer consists of two linear transformations with a ReLU activation in between, applied independently to each position. This mechanism ensures smooth gradient flow and prevents gradient vanishing or exploding issues. Figure 7 details the transformer architecture.

Advancements in deep learning have integrated hybrid solutions that combine the U-Net and Transformer architectures. Transformers can capture and analyze extended speech contexts, enabling them to effectively reduce noise by accurately processing noisy inputs. Self-attention layers give priority to significant speech components over background noise. Research is underway to explore the application of Transformers in speech processing, and its early results are promising.

3.6 CleanUNet

CleanUNet (KONG et al., 2022) is a CNN architecture specifically designed for causal speech-denoising tasks. It builds upon the principles of the U-Net architecture, a well-known model in image segmentation, and bears similarity to the model presented in

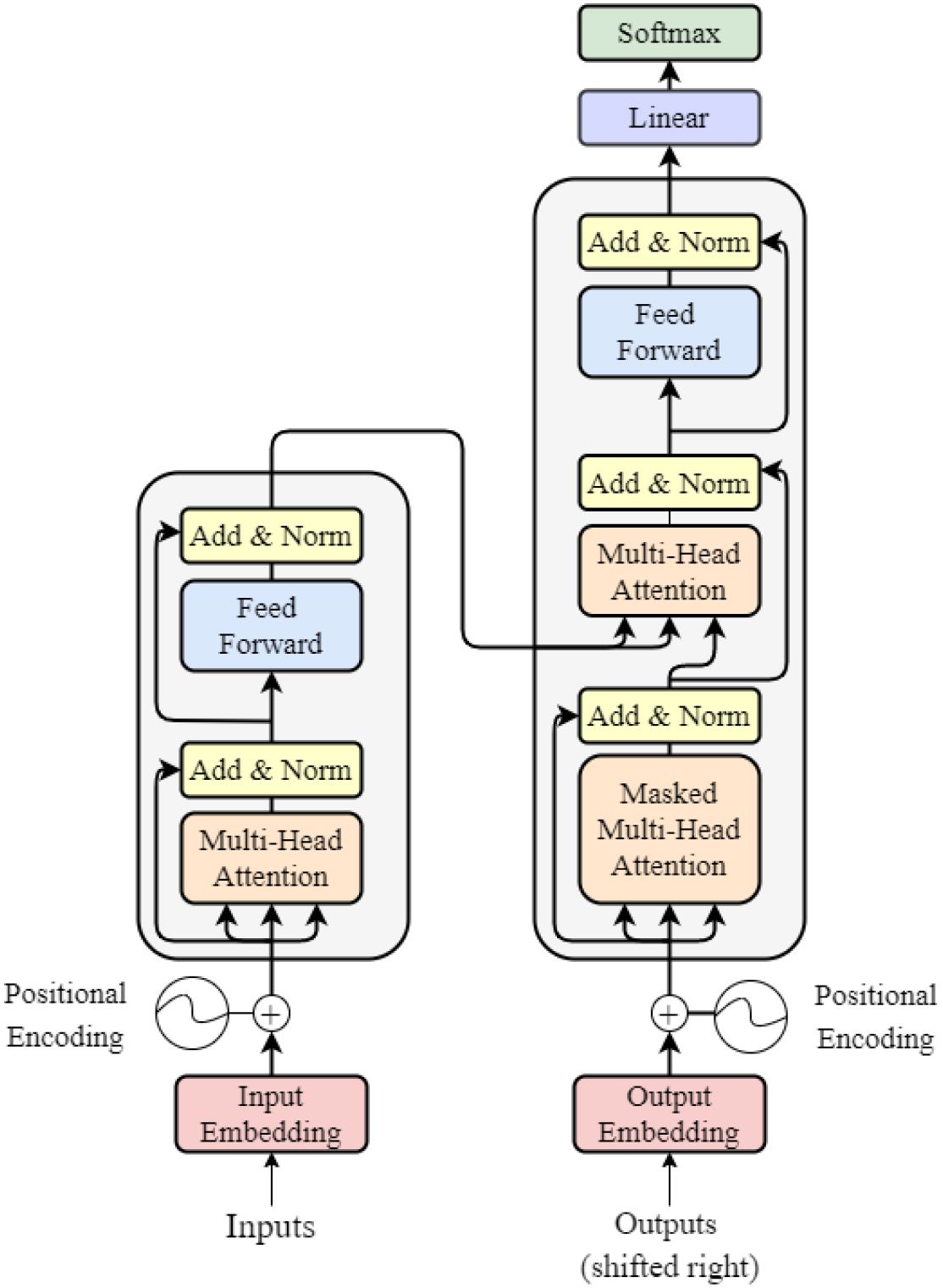


Figure 7 – Transformer Architecture. Source: (VASWANI et al., 2017).

(DEFOSSEZ; SYNNAEVE; ADI, 2020). CleanUNet follows the encoder-decoder structure with skip connections, a characteristic feature of U-Net, which allows the network to retain fine-grained details during the upsampling process.

The architecture of CleanUNet includes multiple convolutional blocks, each consisting of several convolutional layers followed by Gated Linear Unit (GLU) and ReLU activation

functions. This configuration enables the network to capture intricate features of the input speech signal. CleanUNet adapts the U-Net architecture to process audio signals directly on the raw waveform, enhancing its suitability for speech denoising.

A key innovation in CleanUNet is the integration of self-attention blocks in the network bottleneck. These self-attention blocks improve the model's ability to distinguish between noise and clean speech signals. Each self-attention block comprises a multi-head self-attention layer and a position-wise fully connected layer, following the Transformer architecture principles introduced in (VASWANI et al., 2017). The model employs 3 to 5 bottleneck layers and 8 convolutional layers, depending on the dataset, with the largest model consisting of approximately 46 million parameters. See Figure 8.

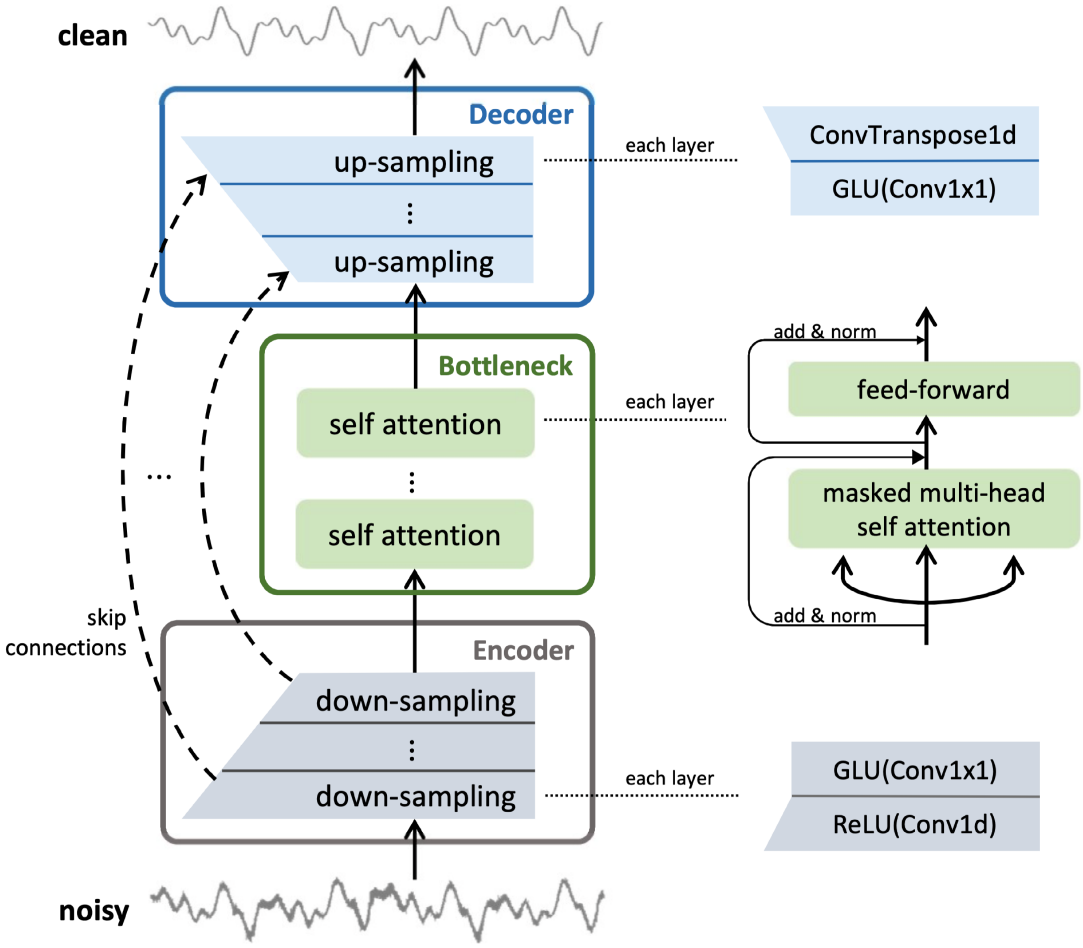


Figure 8 – CleanUNet architecture. Source: (KONG et al., 2022).

3.6.1 Causal Convolution

In a 1D causal convolution, the output $y(t)$ at time step t is computed using the input values $x(t), x(t-1), \ldots, x(t-k)$, where k is the filter size minus one. This can be represented mathematically as $y(t) = \sum_{i=0}^{k} w(I) \cdot x(t-i)$ where $w(I)$ are the weights of

the convolutional filter. The convolution is designed so that the output at time t does not incorporate any input from future time steps $t+1, t+2, \ldots$. By only using current and past inputs, causal convolutions preserve the temporal order of data, which is crucial for processing the speech sequence (OORD et al., 2016).

3.6.2 Rectified Linear Unit

The ReLU activation function (GLOROT; BORDES; BENGIO, 2011) is fundamental in neural networks, particularly within deep learning architectures. Mathematically defined by the equation

$$\text{ReLU}(x) = \max(0, x), \tag{30}$$

it yields the input value directly if it is positive; otherwise, it returns zero. ReLU introduces non-linearity into the model, thereby enabling the learning of complex patterns while maintaining computational efficiency due to its simple comparison operation. This activation function also promotes sparse activations, enhancing computational efficiency and mitigating the risk of overfitting. Furthermore, ReLU plays a critical role in alleviating the vanishing gradient problem, which is essential for effectively training deep neural networks. See Figure 9 for more details.

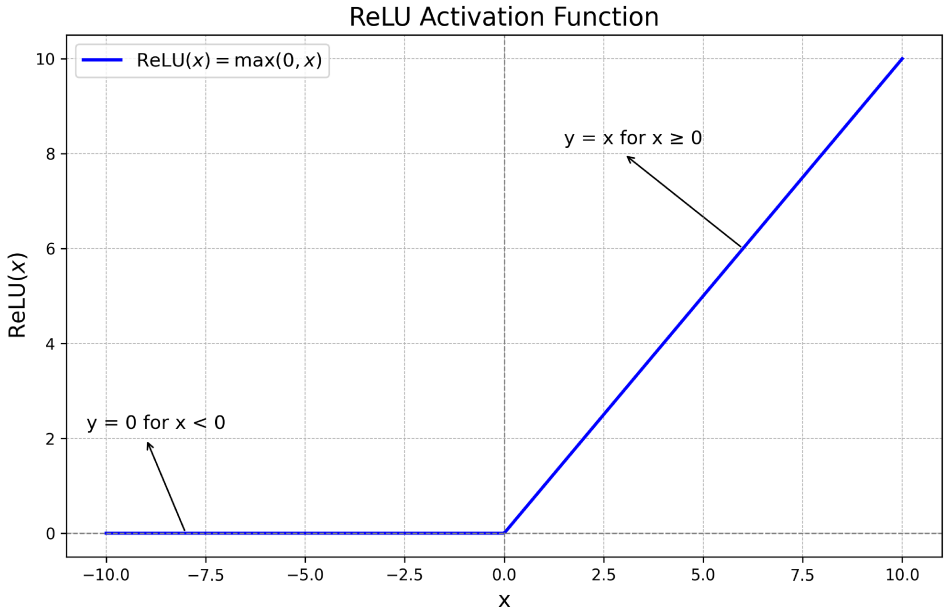


Figure 9 – ReLU activation.

3.6.3 Gated Linear Units

Gated Linear Units (GLUs) introduce a gating mechanism in neural networks to control the flow of information, enhancing the model's capacity to capture complex patterns.

A GLU is defined as:

$$\text{GLU}(a, b) = a \times \sigma(b) \tag{31}$$

where a and b are the outputs of linear transformations applied to the input, and σ denotes the sigmoid activation function. The sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{32}$$

The linear transformations partition the input into two parts, a and b. The gating mechanism $\sigma(b)$ produces values between 0 and 1, effectively acting as a filter that modulates the signal a. When $\sigma(b)$ is close to 0, it blocks the information, while values close to 1 allow the information to pass through. This mechanism aids in better gradient flow during backpropagation, mitigating issues like the vanishing gradient problem, and introduces a form of non-linearity that can capture intricate dependencies in the data.

3.6.4 Softmax

The softmax function (BRIDLE, 1990) is a crucial component in machine learning, particularly in classification problems. It transforms the raw output scores of a neural network into a probability distribution over possible classes, making it essential for tasks such as multi-class classification. In the CleanUNet architecture, the softmax function is essential in the final processing stage to precisely classify the audio signal components. CleanUNet utilizes the U-Net's encoder-decoder structure to map noisy speech waveform inputs to their clean counterparts. During this process, the network outputs a raw score for each time-frequency bin of the transformed spectrogram. The softmax function is then applied to these raw scores to convert them into probabilities, effectively distinguishing between noise and clean speech for each bin. Formally, this transformation is defined as

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for} \quad i = 1, \ldots, K, \tag{33}$$

where $\mathbf{z}$ represents the output vector from the CleanUNet model.

3.6.5 Transformer Bottleneck

The Transformer bottleneck processes speech by utilizing self-attention mechanisms across multiple hidden layers, typically 6 to 8 in the baseline architecture, capturing dependencies within the audio sequence while also employing positional encoding to retain the order of the speech signal, allowing the model to understand the temporal structure; additionally, it uses masked self-attention and positional encoding for audio tokens to enhance the representation and understanding of temporal relationships within the data.

3.7 Mamba Architecture

Mamba (GU; DAO, 2023) is an innovative architecture designed to address the quadratic complexity challenges of conventional attention models. This model tackles the scalability constraints of traditional Transformer models in processing long sequences, enhancing efficiency and flexibility with linear-time complexity for sequence modeling. It introduces two significant innovations in State Space Model (SSM): a selection mechanism and a hardware-aware algorithm.

A SSM can be simplified by the parameters (Δ, A, B, C), where each parameter represents a diagonal matrix. The discretization of (Δ, A, B) yields $(\overline{A}, \overline{B})$, translating the model from a continuous domain to a discrete one, thus enabling computation within a digital framework. This discretization process can be done through fixed formulas $\overline{A} = F_a(\Delta, A)$ and $\overline{B} = F_b(\Delta, A, B)$, with (F_a, F_b) being a discretization rule such as zero-order hold (ZOH).

One essential property of State Space Models (SSMs) is Linear Time Invariance (LTI). In this context, all time steps in the dynamics represented by the matrices (Δ, A, B, C) are the same. This characteristic enables parallelizing training methods, using convolutional or recursive algorithms to efficiently compute the model's output.

The Mamba architecture introduces a selection mechanism similar to the gating mechanism of RNNs, enabling SSMs to adapt over time and be input-dependent by focusing or filtering out different parts of a context. However, this modification compromises the LTI property, resulting in the inability to apply convolution methods as previously done. To overcome this, the authors implement a hardware-aware method that takes advantage of the memory hierarchy of modern Graphical Processing Units (GPUs). The hardware-aware algorithm incorporates three key techniques: kernel fusion, parallel scan, and recomputation. Kernel fusion reduces memory I/O by combining multiple operations into a single GPU operation. Parallel scan algorithms enable parallel computation despite the non-linear nature of the SSMs. Recomputation is used during backpropagation to minimize memory usage by recalculating intermediate states instead of storing them. Figure 10 illustrates the mechanism.

3.8 LongNet

The LongNet architecture (DING et al., 2023) is a new Transformer variant that achieves linear computational complexity and logarithmic token dependency by adding dilated attention to its core, optimizing efficiency for extensive sequence processing. The core innovation of the dilated attention mechanism lies in its ability to expand the attentive field exponentially with increasing distance. This mechanism divides the input (Q, K, V) into equally sized segments with a predetermined segment length W. Each

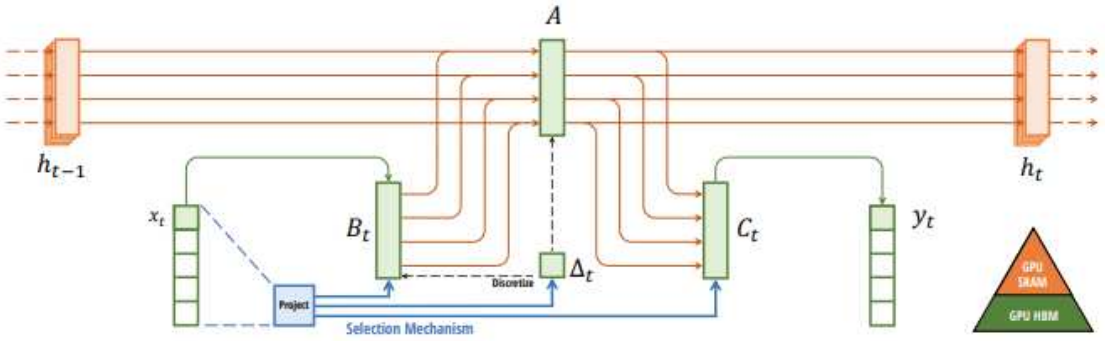


Figure 10 – Selection Mechanism with hardware-aware State Expansion. Source: (GU; DAO, 2023).

segment is then made sparse along the sequence dimension by selecting rows at specified intervals r. The segments are processed in parallel, then scattered and concatenated to produce the output. By incorporating the concept of multi-head attention of the Transformer, the method further enhances the dilated attention mechanism by diversifying the computation across different heads. Specifically, the computation varies among heads by sparsifying distinct segments of the (Q, K, V) pairs. This architecture is fully compatible with Transformers by introducing segment sizes r and dilation rates d hyperparameters. Figure 11 illustrates the attention patterns for short-range and long-range dependency.

3.9 Cost Function

The CleanUNet cost function, as detailed in (KONG et al., 2022), integrates a multi-resolution STFT loss with a waveform-based loss to enhance speech-denoising performance. This cost function has two main components, and combining these enables CleanUNet to reduce noise effectively while maintaining the integrity of the speech signal in both time and frequency domains.

3.9.1 Waveform-Based L1 Loss

This component applies the L1 loss directly to the waveform, encouraging the denoised output to closely resemble the clean speech waveform in the time domain. It ensures that the temporal structure of the speech signal is preserved during denoising. The $L1$ loss, also known as Mean Absolute Error (MAE), is a common loss function used in regression tasks. It measures the average absolute difference between the predicted values and the actual target values (KONG et al., 2023). Mathematically, the $L1$ loss for a set of predictions $\hat{y}$ and true values y is defined as:

$$L1 \text{ loss} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{34}$$

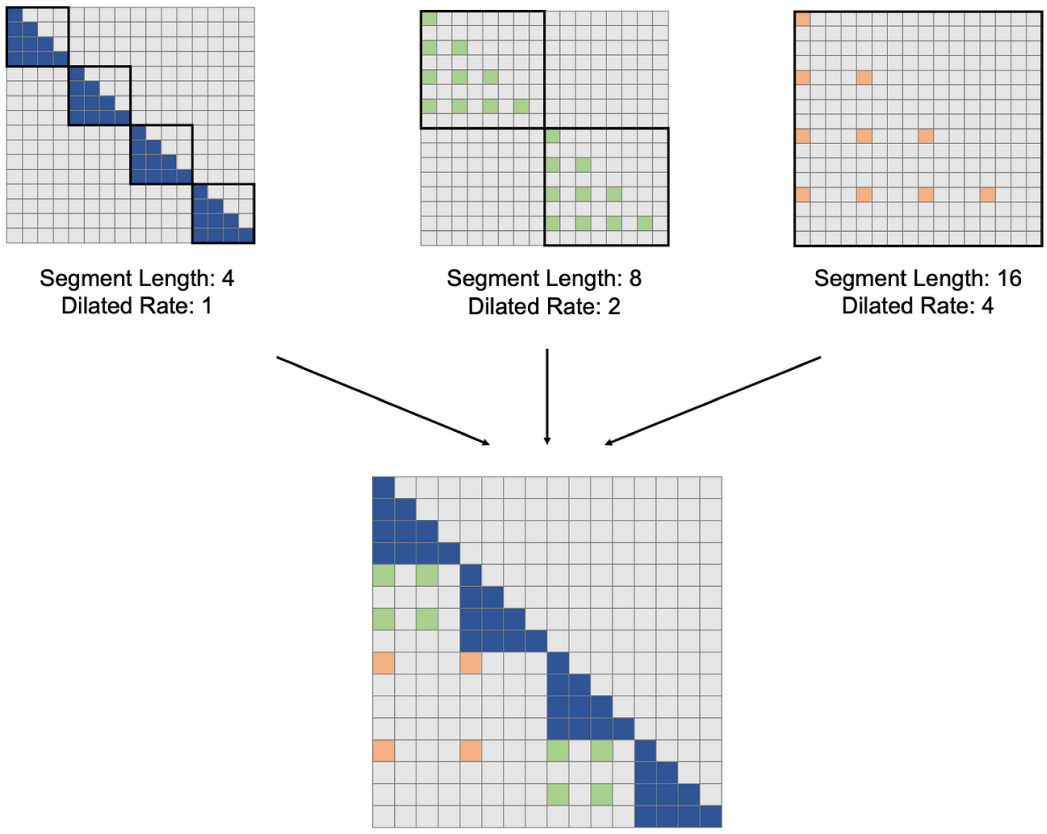


Figure 11 – Dilated attention mechanism patterns. Source: (DING et al., 2023).

where n is the number of samples, y_i is the true value of the i-th sample, and $\hat{y}_i$ is the predicted value of the i-th sample.

3.9.2 Multi-Resolution STFT Loss

This component involves a multi-resolution STFT loss, which focuses on the magnitude spectrograms of the speech signal at various resolutions. Modifying the spectrogram's magnitude at multiple resolutions ensures that the frequency content of the denoised speech matches the clean speech across different time-frequency scales. This helps capture and preserve the fine details in the speech signal's frequency domain, leading to higher-quality denoising (KONG et al., 2023).

Chapter 4

Related Works

Rethage, Pons e Serra (2018) presents a novel approach to speech denoising using a modified Wavenet model. Unlike traditional methods that discard phase information by relying on magnitude spectrograms, this method processes raw audio directly, retaining valuable phase data. The proposed model eliminates Wavenet's autoregressive nature, incorporating non-causal, dilated convolutions and predicting target fields to enhance parallelizability and reduce time complexity. Evaluations show this method outperforms Wiener filtering in computational and perceptual quality metrics. Key innovations include real-valued predictions, energy-conserving loss, speaker identity conditioning, and noise-only data augmentation, collectively improving performance and robustness. This discriminative adaptation allows for efficient, high-quality speech denoising, making it a promising alternative to conventional techniques.

The work presented by (DEFOSSEZ; SYNNAEVE; ADI, 2020) introduces a real-time speech enhancement model based on the DEMUCS architecture optimized to run efficiently on a laptop CPU. The model processes raw waveforms using an encoder-decoder structure with skip connections and is trained with both time and frequency domain loss functions. New data augmentation techniques, such as frequency band masking and signal reverberation, are utilized to improve performance, and standard benchmarks evaluate the model's effectiveness in objective and subjective metrics. An ablation study confirms the significant contributions of specific model components and augmentation methods. Additionally, the model enhances Automatic Speech Recognition (ASR) accuracy in noisy environments, demonstrating practical applicability.

The paper (KONG et al., 2022) presents CleanUNet, a causal speech denoising model that operates directly on the raw waveform. The model uses an encoder-decoder architecture with self-attention blocks to refine bottleneck representations, optimizing through waveform and multi-resolution spectrogram losses. CleanUNet demonstrates superior denoising quality in objective and subjective evaluations compared to existing models. Evaluations on datasets like Deep Noise Suppression (DNS) and Valentini show its effectiveness, with ablation studies highlighting the contributions of various model components

and loss functions.

The CleanUNet 2 (KONG et al., 2023) brings an improvement that leverages both waveform and spectrogram-based denoisers to enhance performance. CleanUNet 2 utilizes a two-stage framework where a spectrogram-based denoiser, CleanSpecNet, first predicts clean spectrograms from noisy inputs. These spectrograms then condition a waveform-based denoiser, CleanUNet, to produce the final clean waveform. This approach aims to combine the strengths of both methods, effectively removing noise while maintaining high speech quality. Evaluations on the DNS dataset demonstrate that CleanUNet 2 outperforms existing models in objective and subjective metrics, achieving superior noise reduction and speech clarity even under high noise levels.

In (PASCUAL; BONAFONTE; SERRA, 2017), the authors introduce an approach to speech enhancement using GANs. Traditional methods often operate in the spectral domain and struggle with diverse noise conditions. SEGAN operates directly on raw audio waveforms, using an encoder-decoder architecture with fully convolutional layers and skip connections to maintain speech detail. Trained on a dataset with various noise conditions and speakers, SEGAN performs in most objective metrics and is preferred in subjective evaluations over traditional Wiener filtering. The study demonstrates SEGAN's viability and effectiveness, suggesting future improvements in convolutional structures and perceptual weightings to enhance speech quality further.

Oliveira, Peer e Gerkmann (2022) presents a speech enhancement approach leveraging the SepFormer architecture. Traditionally, transformer-based models for speech processing use short frames due to their performance benefits despite the increased computational complexity this entails. The authors propose substituting the learned-encoder features with magnitude STFT representations, enabling the use of longer frames. This substitution maintains the perceptual enhancement performance while significantly reducing computational demands—by approximately eight times for a 10-second utterance. The experimental results on the DNS-Challenge and WSJ0 datasets demonstrate that the proposed method achieves similar quality and intelligibility scores compared to the conventional learned-domain methods. Additionally, the new approach substantially lowers memory usage, making it more feasible for practical applications, particularly embedded devices. The study highlights the potential of integrating longer frame time-frequency representations in transformer-based speech enhancement systems to achieve efficient, high-quality performance.

The paper by (PARK et al., 2022) introduces a novel approach for speech enhancement in the time domain, overcoming the limitations of existing dual-path models. The proposed Multi-view Attention Network for Noise ERasure (MANNER) utilizes a convolutional encoder-decoder structure integrated with a multi-view attention block. This network effectively extracts three distinct representations from noisy speech and synthesizes high-quality, clean speech. Evaluated on the VoiceBank-DEMAND dataset, MANNER

demonstrates state-of-the-art performance across five objective speech quality metrics, highlighting its efficiency in memory usage and inference speed. Key contributions include a robust architecture leveraging multi-view attention to enhance channel and sequential features and an optimized loss function combining L1 and multi-resolution STFT losses to balance clean and noise reduction. The study affirms the model's superior performance and efficiency, positioning MANNER as a significant advancement in speech enhancement technology.

In their paper, (JANNU; VANAMBATHINA, 2023) proposes a novel two-stage convolutional Transformer model for speech enhancement in the time domain. The model combines the strengths of CNNs and Transformers to effectively address short-term and long-term dependencies in noisy speech signals. Like a few aforementioned papers, this method operates directly with raw waveforms, avoiding the complexities of phase reconstruction. The proposed model uses separate Transformer structures for intra and inter-transformer stages to extract local and global features. Additionally, CNN layers are integrated to enhance the model's ability to handle short-term noise. Experimental results demonstrate that this convolutional transformer significantly outperforms existing models, achieving higher scores in short-time STOI and PESQ on the Valentini and Librispeech datasets under various noise conditions. This work advances the field of speech enhancement by providing a robust, end-to-end solution that leverages the parallel processing and attention mechanisms of transformers combined with the local feature extraction capabilities of CNNs.

Chapter 5

Proposal

Acoustic issues during home office meetings can significantly hinder communication. A straightforward solution to this problem is the use of noise suppression algorithms. This section will discuss the CleanUNet components and how to optimize them for real-time applications while minimizing resource consumption.

Our proposed model incorporates several key components: 1D convolutions, transposed 1D convolutions, GLU, batch normalization BN, Leaky ReLU, and Parametric ReLU activation functions. Additionally, we have completely replaced the bottleneck architecture to enhance performance and efficiency. Figure 5 illustrates an example of architecture configuration in the experiments.

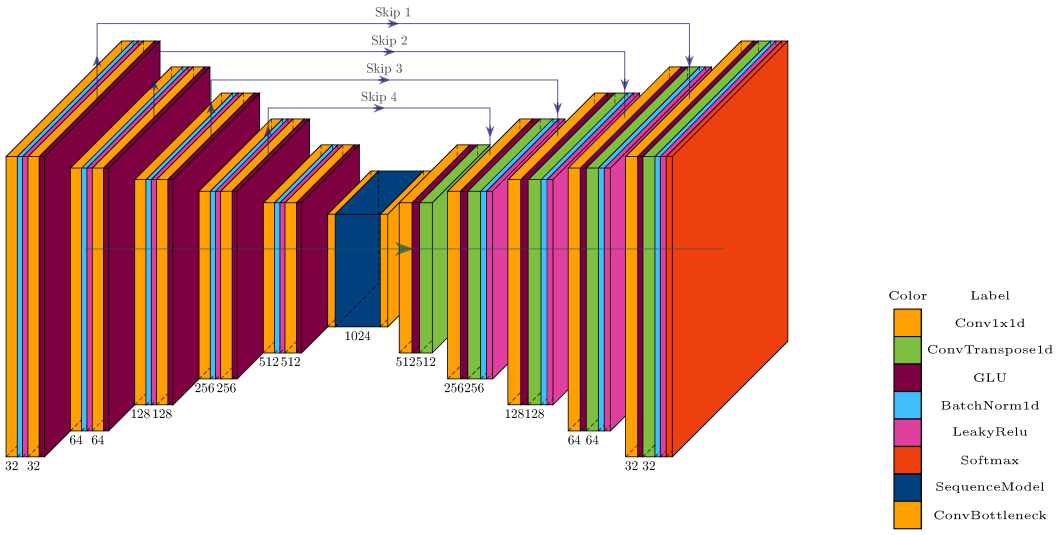


Figure 12 – Proposed model architecture.

5.1 Leaky Rectified Linear Unit

ReLU can suffer from the "dying ReLU" problem, where neurons can become inactive and only output zero, effectively preventing them from learning during training. This issue arises because the gradient of ReLU is zero for negative inputs, which can lead to neurons that never activate and thus contribute nothing to the learning process.

To address this issue, the Leaky ReLU (MAAS et al., 2013) was introduced as a variation of ReLU. Leaky ReLU allows a small, non-zero gradient for negative inputs, typically a small constant like 0.01. This slight slope for negative values helps to keep the neurons active even when they would otherwise be deactivated by standard ReLU, thereby mitigating the dying ReLU problem. However, the slope for negative inputs in Leaky ReLU is fixed across all neurons and layers, which may not always be optimal for different network parts.

To introduce more flexibility and adaptivity in handling negative inputs, the Parametric ReLU (PReLU) was proposed by (HE et al., 2015). In PReLU, the slope of the negative part is not fixed; instead, it is a learnable parameter that is updated during the training process. This enables the model to adapt the negative slope for each neuron individually, providing a more dynamic and tailored response to negative inputs. By allowing each neuron to learn its optimal negative slope, PReLU can lead to better performance, especially in deep networks where the variety of activation patterns can be crucial for capturing complex patterns in data. The following equation and Figure 13 illustrate Leaky ReLU.

$$\mathrm{ReLU}(x) = \max(\alpha x, x), \tag{35}$$

5.2 Bottleneck

Integrating convolutional layers for local feature extraction with sequence models for global context understanding results in a more robust and efficient model architecture. Incorporating Mamba or dilated attention in the bottleneck enables the model to effectively capture relationships and dependencies across the entire input sequence, which is crucial for tasks like speech denoising, where context plays a significant role. The bottleneck compresses the data to focus on the most salient features and also plays a pivotal role in enhancing the model's generalization capabilities by preventing overfitting. While Mamba and dilated attention were applied in a single layer, reducing the parameter count and computational complexity, CleanUNet utilized 3-5 layers of Transformers, achieving better representation learning but at the expense of increased computational resources. Additionally, a key consideration for the dilated attention model is ensuring that the segment sizes are divisible by the output channels of the last convolutional layer, adding a

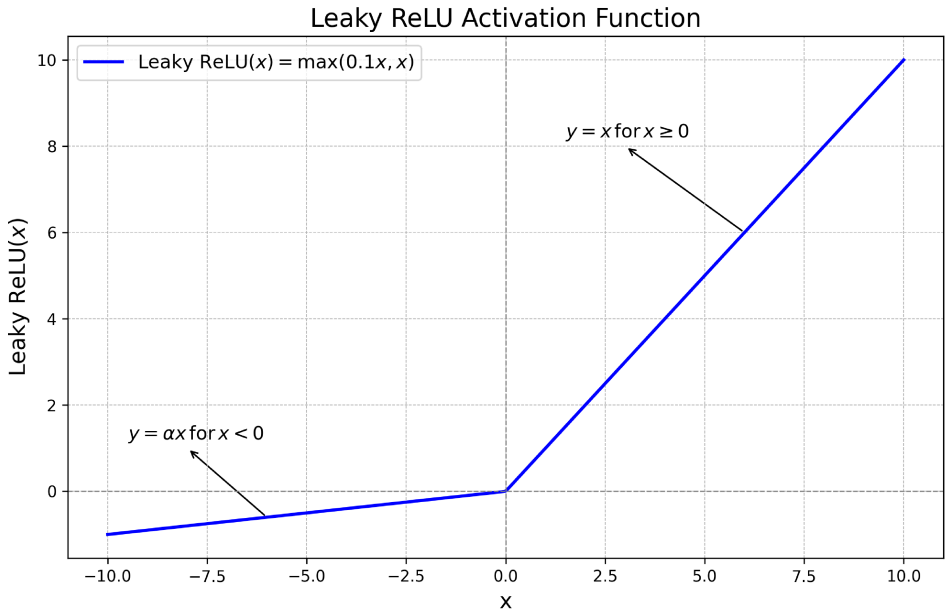


Figure 13 – Leaky ReLU activation.

layer of complexity to the architecture. This balance between depth and efficiency highlights the trade-offs inherent in designing models for complex tasks like speech denoising.

5.3 Batch Normalization

Batch Normalization is a widely used technique in training deep neural networks to mitigate issues such as covariate shift and improve convergence speed (SANTURKAR et al., 2018). In convolutional layers, batch normalization is applied to the activations of the feature maps (or features directly), which helps stabilize and accelerate the training process. The method involves two primary steps: normalization and scaling. For a given mini-batch B with activations $\{x_1, x_2, ..., x_m\}$, the Equations 36, 37, and 38 describe the batch normalization process:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{and} \quad \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{36}$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{37}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{38}$$

Equation 36 computes the mean μ_B and variance σ_B^2 of the activations, and then normalization is applied in Equation 37, where ϵ is a small constant added for numerical stability. After normalization, a linear transformation is applied to scale and shift the

normalized activations. Finally, γ and β are added as learnable parameters that allow the model to adjust the normalized values, preserving the representation capacity of the network.

Covariate shift refers to changes in input data distribution to a model during training, which can adversely affect the learning process. The convolutional layers in the architecture are affected by the weights and biases of the preceding layers, causing shifts in the data distribution as learning progresses. This shift can slow down training and lead to poor convergence. Batch normalization addresses this issue by normalizing the input to each layer to have a consistent distribution, thus mitigating the effects of covariate shift. This method enables the network to learn more stable and effective representations, leading to faster and more reliable convergence.

5.4 Training Configuration

We used a training configuration similar to (KONG et al., 2022). The models were trained using Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate decreases along the training using a cosine annealing rate scheduler with a maximum learning rate of 0.00025 and a warmup ratio of 5%. This work employs the random shift data augmentation technique within 0 to 3 seconds. For some experiments, we also included BandMask augmentation (DEFOSSEZ; SYNNAEVE; ADI, 2020), and a variant selects the bands randomly.

Experiments were performed on a PC with an AMD Ryzen 7 7800X3D processor, 32GB of RAM, and RTX 3090 24GB GPU running Linux Ubuntu 20.04 OS. We modified the original CleanUNet[1] code to run on Pytorch 2.1.1[2], CUDA 12.3, and Python 3.10[3], along with the modifications needed to implement Mamba architecture using mamba-ssm[4] and dilated attention.[5]

[1] https://github.com/NVIDIA/CleanUNet
[2] https://pytorch.org/
[3] https://www.python.org/
[4] https://pypi.org/project/mamba-ssm/1.1.0/
[5] https://pypi.org/project/dilated-attention-pytorch/

Chapter 6

Experimental Results

We proposed and evaluated several variations of CleanUNet, aiming to reduce the number of learnable parameters, thereby decreasing complexity, training, and prediction time while maintaining quality. The variations involve replacing the bottleneck with alternatives to self-attention. Specifically, we explored the dilated attention mechanism from LongNet and Mamba, which are sequence models capable of capturing both long- and short-range complex patterns. Additionally, we conducted experiments with minor architectural modifications, including applying batch normalization, using different activation functions, and reducing the number of parameters in the encoder-decoder layers. The trained models in this work were compared with CleanUnet described in (KONG et al., 2022), which we considered the Baseline.

The baseline CleanUnet (KONG et al., 2022) trained on the (VALENTINI-BOTINHAO et al., 2017) consists of an encoder and a decoder, with D layers and a bottleneck composed of N self-attention blocks (bottleneck layers). Each encoder layer is composed of a 1D convolution layer (Conv1D) followed by a ReLU layer and a 1×1 convolution (Conv1×1) followed by a gated linear unit (GLU). Each Conv1D has kernel size K and stride $S = K/2$. The first Conv1D has H kernels, and the other layers double the number of channels. The Conv1x1 doubles the number of channels while the GLU reduces it by half. Each decoder comprises a 1×1 convolution followed by a GLU and a transposed 1D convolution layer (ConvTranspose1D). Following the U-Net architecture, each encoder layer is connected to one encoder layer by a skip connection in the reverse order. The bottleneck is composed of N self-attention blocks. Each self-attention block comprises a multi-head self-attention layer (8 heads) and a fully connected layer (input/output size of 512 and inner-layer size of 2048) as described in (KONG et al., 2022).

6.1 Dilated Attention-based CleanUNet Model

For this CleanUNet model, we considered the following configurations: $D = 6$, $K = 5$, and $N = 1$ for training and inference in GPU. The self-attention block comprises a dilated

multi-head self-attention layer (16 heads) and a fully connected layer (input/output size of 1024 and inner-layer size of 2048). The list of segment sizes {7, 14} was provided with its respective dilation rates {2, 4}. The model evaluation results are categorized into 3 main metrics: PESQ Wideband (WB) at 1.537, PESQ Narrowband (NB) at 2.156, and STOI at 0.822. This model's total parameter count was 11.8 million, with 8.4 million bottleneck parameters. The training configuration set the learning rate to 0.00025, batch size 24, and crop length of 3 seconds. Despite reducing the number of parameters and achieving moderate results, the replacement in the bottleneck was a step in the right direction. This adjustment successfully reduced the complexity of processing the sequence.

Figure 14 shows 3d surface plots for noisy, enhanced, and clean signals. The plot differences illustrate where the difference amplitude of the signal is positive (blue), which might represent successful noisy reduction; the difference illustrated in red potentially represents audio degradation where the difference is negative. The clean speech plot is used as a reference or ground truth. The plots were created following the Nyquist Theorem (NYQUIST, 1928), which states that to capture a signal without aliasing accurately, the sampling rate must be at least twice the highest frequency in the signal (sr/2). Therefore, in the STFT, only frequencies up to sr/2 are represented, as higher frequencies are redundant. We use min-max normalization to ensure consistency across different signals, scaling amplitude values to a range of [0, 1]. This approach enhances the comparability and uniform analysis of signal features.

6.2 Mamba-based CleanUNet Model

The Mamba-based model was optimized to utilize fewer hidden channels, and several experiments were conducted using this configuration, including changes in the bottleneck parameters. Table 1 presents various model configurations, each differing in the number of parameters, blocks, and out channels. The models range from 46,082,177 to 626,753 parameters, with the largest model serving as the baseline for comparison at 100%. The number of blocks and out channels varies, indicating different levels of complexity. The percentage column shows the relative size of each model compared to the baseline. Additionally, the Mamba column includes specific values for the hidden projections. Generally, higher parameter counts and more blocks suggest a more complex model that may yield better performance but at increased computational costs.

Table 1 details the model training settings, such as the number of iterations, batch size, and various configurations: Batch Normalization (BN), BandMask (BM), ReLU (R), Leaky Relu (LR), Parametric ReLU (PR) and Random Band Mask (RBM). The performance metrics PESQ (WB), PESQ (NB), and STOI are also provided, evaluating the models' audio quality and intelligibility. Inference times are also listed, highlighting differences in computational efficiency. Models with 1000k and 500k iterations, especially

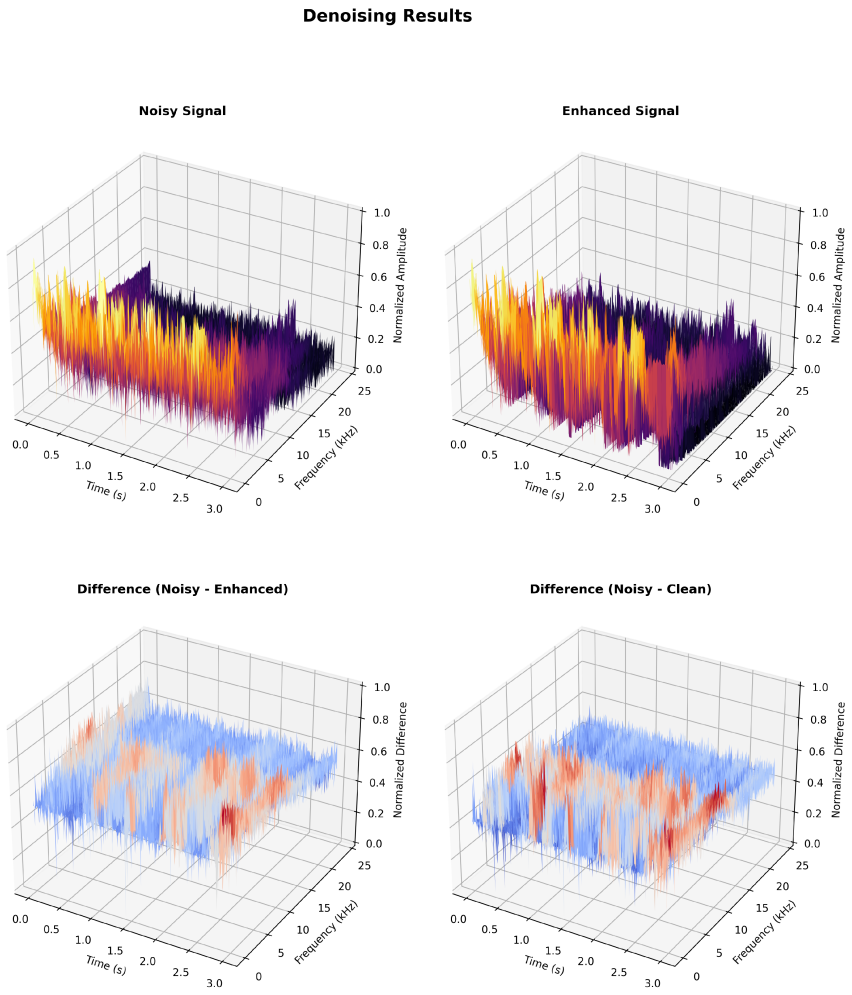


Figure 14 – Audio signal comparison between clean and enhanced signals. The red and blue areas represent the amplitude difference in decibels (dB) between these two signals across different frequencies and over time.

those employing configurations like [R, BM], tend to have higher PESQ and STOI scores, indicating better performance. For instance, Id 8, with 21,878,913 parameters and 1000k iterations, achieves higher scores and a short inference time of 5.7 seconds. Smaller models, such as model 18, exhibit shorter inference times but lower performance metrics, illustrating the trade-offs between model complexity, performance, and computational efficiency. All the experiments used a learning rate of 0.00025 except the IDs with (*), which uses 0.0002.

Table 1 – CleanUNet model settings and metrics.

Id	Configs	Batch size	# of iter	K	# of blocks and out channels	Mamba	# of Params	Perc.	PESQ (WB)	PESQ (NB)	STOI	Infer.
*1	[R]	8	250k	3	[64, 128]	768	4,100,865	8.89%	2.449	3.064	0.900	42.9s
2	[R]	20	250k	3	[64, 128]	768	4,100,865	8.89%	2.211	2.878	0.881	43.1s
3	[R]	24	250k	3	[64, 128, 256, 512, 512]	1024	15,394,305	33.40%	2.667	3.209	0.909	10.0s
4	[R]	28	250k	3	[64, 128, 256, 512]	512	4,650,497	10.09%	2.201	2.844	0.880	8.7s
*5	[R]	32	250k	3	[64, 128, 256, 512]	512	4,650,497	10.09%	2.397	3.017	0.894	8.8s
6	[R]	32	500k	3	[64, 128, 256, 512, 512, 512]	1024	16,325,633	46.21%	2.432	3.066	0.901	9.4s
*	[R, BM]	64	1000k	4	[48, 96, 192, 384, 768, 768, 768, 768]	-	46,082,177	100.00%	2.905	3.408	0.956	9.6s
7	[R, BM]	16	500k	3	[48, 96, 192, 384]	1500	16,751,077	36.35%	2.680	3.244	0.910	16.1s
8	**[R, BM]**	**32**	**500k**	**5**	**[32, 64, 128, 256, 512, 512, 512, 512]**	**1024**	**21,878,913**	**47.47%**	**2.745**	**3.288**	**0.911**	**5.7s**
9	[R, BM]	32	1000k	5	[32, 64, 128, 256, 256, 256, 256, 256, 256, 256]	2028	33,785,985	73.31%	2.730	3.284	0.912	4.8s
10	[R, BM]	32	1000k	7	[48, 96, 192, 384, 384, 384, 384, 384]	1024	27,828,097	60.38%	2.726	3.270	0.910	7.6s
11	[R, BM]	32	1000k	5	[48, 96, 192, 384, 384, 384, 384]	512	14,339,265	31.11%	2.609	3.117	0.907	6.5s
12	[LR, BN, BM]	32	500k	5	[32, 64, 128, 256, 256, 256, 256, 256]	512	6,414,977	13.92%	2.722	3.255	0.912	4.9s
13	[LR, BN, BM]	32	500k	5	[128, 128, 128, 128, 128, 128, 128, 128]	512	3,504,897	7.60%	2.686	3.219	0.911	5.3s
14	[LR, BN, BM]	48	500k	5	[24, 48, 96, 192, 384, 768]	512	9,564,257	20.75%	2.564	3.139	0.905	5.1s
15	[R, BN, BM]	32	500k	5	[48, 96, 192, 384, 384, 384, 384]	1024	24,355,521	52.85%	2.661	3.205	0.911	8.0s
16	[R, BN, BM]	32	500k	5	[32, 64, 128, 256, 256, 256]	512	4,576,897	9.93%	2.580	3.147	0.906	4.7s
17	[PR, BN, RBN]	32	500k	4	[64, 64, 64, 64, 64, 64, 64]	512	5,784,193	12.55%	2.545	3.113	0.903	5.8s
18	**[LR, BN, RBM]**	**64**	**500k**	**7**	**[32, 32, 32, 32, 32, 32, 32, 32, 32, 32]**	**256**	**626,753**	**1.36%**	**2.544**	**3.099**	**0.903**	**3.4s**

- ❏ BN: Batch Normalization
- ❏ BM: BandMask
- ❏ LR: Leaky ReLU
- ❏ PR: Parametric ReLU - PReLU
- ❏ R: ReLU
- ❏ RBM: Random Band Mask

The configurations highlight that models with fewer blocks and out channels generally exhibit longer inference times. For example, configurations 1 and 2 use fewer blocks and have inference times of 42.9s and 43.1s, respectively. This longer inference time is attributed to the bottleneck layer in these less deep architectures, which produces more features to be processed by the sequence model in the bottleneck, increasing computational time. In contrast, deeper models with more blocks and smaller out channels, such as Id 18, show a shorter inference time of 3.4 seconds. With additional blocks, these configurations discard more features as they go deeper; the sequence model handles these features more efficiently.

It is worth noting that if experiment Id 2, which uses the Mamba architecture, were replaced with a Transformer-based architecture, inference times would likely increase further due to the quadratic complexity of Transformers. It took 43.1 seconds during inference, which, according to the paper (GU; DAO, 2023), is five times faster than a traditional Transformer. While the Mamba architecture efficiently processes longer sequences, it shows only marginal benefits for shorter sequences. This implies that experimenting with fewer blocks might affect metrics such as PESQ and STOI, as the model would process more features that could potentially be discarded during training. Balancing model complexity with inference efficiency remains crucial for optimizing performance.

We can observe that the number of blocks in the model architecture plays a crucial role in enhancing speech quality, as measured by PESQ (both WB and NB). More blocks generally lead to better performance in these metrics, indicating that increased model complexity contributes significantly to perceived speech quality. However, despite this positive correlation (Figure 15), speech intelligibility, as measured by STOI, does not show the same level of sensitivity to the number of blocks. This suggests that while more blocks improve overall speech quality, they do not proportionally enhance speech intelligibility to the same extent. The differential sensitivity of PESQ and STOI to the number of blocks could be attributed to the nature of these metrics. PESQ is designed to assess perceptual quality, which can benefit significantly from the additional detail and nuance captured by more blocks in a complex model architecture. In contrast, STOI measures intelligibility, which may rely more on specific aspects of the speech signal that are adequately captured by fewer blocks. Thus, beyond a certain point, adding more blocks may yield diminishing returns for intelligibility, as the key features necessary for understanding speech are already effectively modeled.

One significant advantage of using batch normalization is that it stabilizes training, which can potentially lead to better results. In the CleanUNet architecture, the Transformer already includes layer normalization, contributing to more stable training. However, when replacing the bottleneck with Mamba, the training process became more chaotic, with the loss fluctuating excessively. Incorporating batch normalization helped mitigate these issues, enabling the model to achieve the same results in 500k iterations

that previously required 1000k iterations without BN. In the results table, experiment IDs 11 and 15 illustrate the results achieved by BN.

This study aimed to optimize the number of model parameters, and standardizing experimental parameters was not crucial for this task. For a more accurate investigation of the correlation matrix, we need to maintain consistent batch size, iterations, number of blocks, out channels, and learning rate across different model configurations; we can ensure that the effects on PESQ and STOI were attributable to changes in the number of blocks rather than other variables. This controlled approach enables a precise understanding of the impact of model architecture on speech quality and intelligibility.

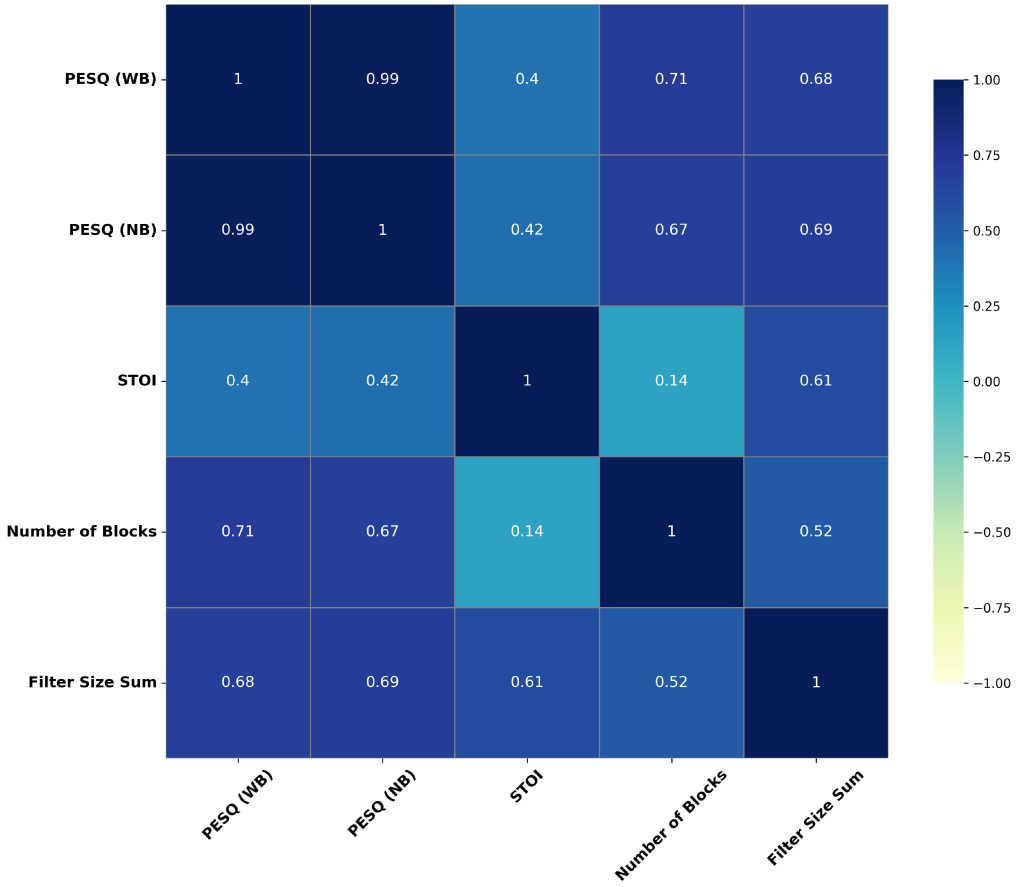


Figure 15 – Correlation matrix of the metrics against the number of blocks and the sum of the filter sizes.

Figures 16, 17, and 18 present scatter plots that illustrate the distribution of the experiments, focusing on key aspects such as the number of parameters and inference time. The baseline model is also included in these plots, highlighting the advantages of certain models in terms of speed and parameter count.

The proposed CleanUNet with a Mamba-based bottleneck significantly reduces the number of trainable parameters with a slight reduction in the audio quality indexes. While the total number of parameters for the best model (Id 8) is reduced by 52.53%, the reduction in PESQ (WB), PESQ (NB), and STOI was 5.50%, 3.52%, and 4.70%,

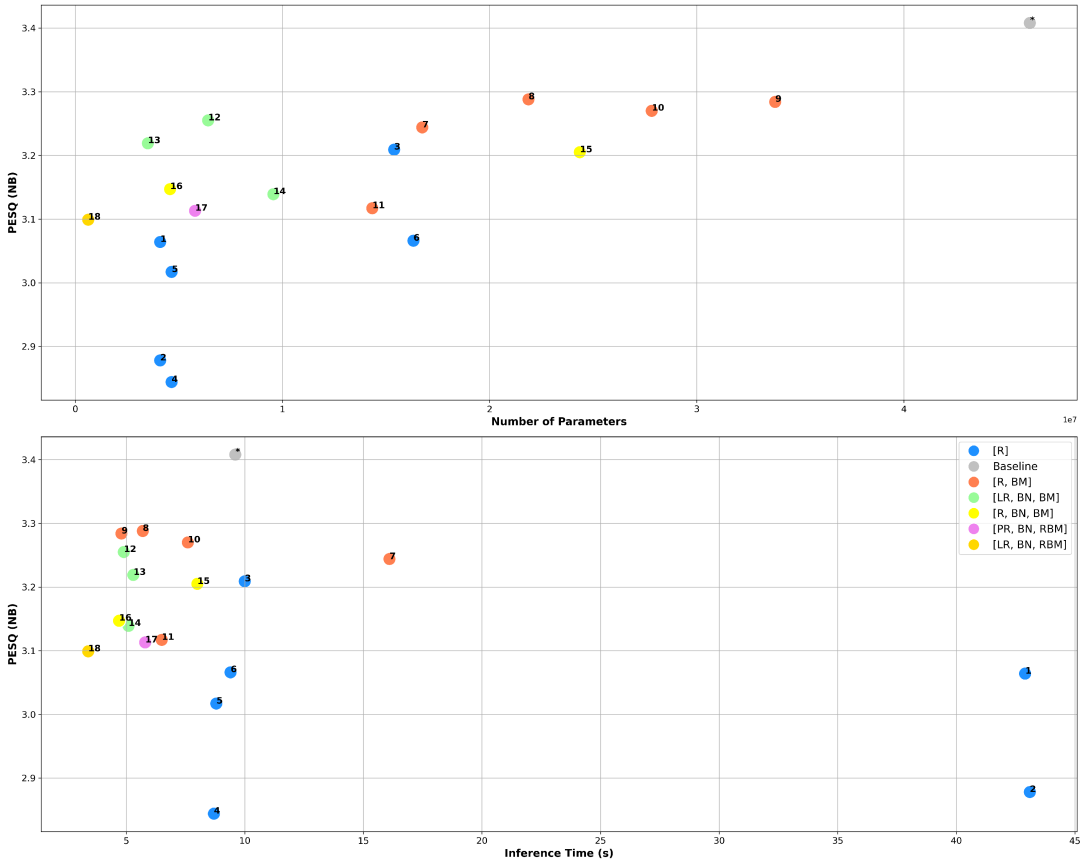


Figure 16 – Trade-off between computational efficiency and performance for PESQ (NB).

respectively. It is important to note that the Mamba hidden projection was 1024. Figure 19 illustrates surface plots for noisy, enhanced, and clean signals, similar to Figure 14.

The fastest model (Id 18) showed notable performance, even reducing 98.64% of the baseline parameters. The reduction of PESQ (WB), PESQ (NB), and STOI was 12.42%, 9.06%, and 5.54%, respectively. This model's inference time was 3.4 seconds on the test set, which makes it well-suited for real-time denoising applications and environments where computation resources are limited. Figure 20 shows the difference and potential degradations of the enhanced speech compared to the clean speech.

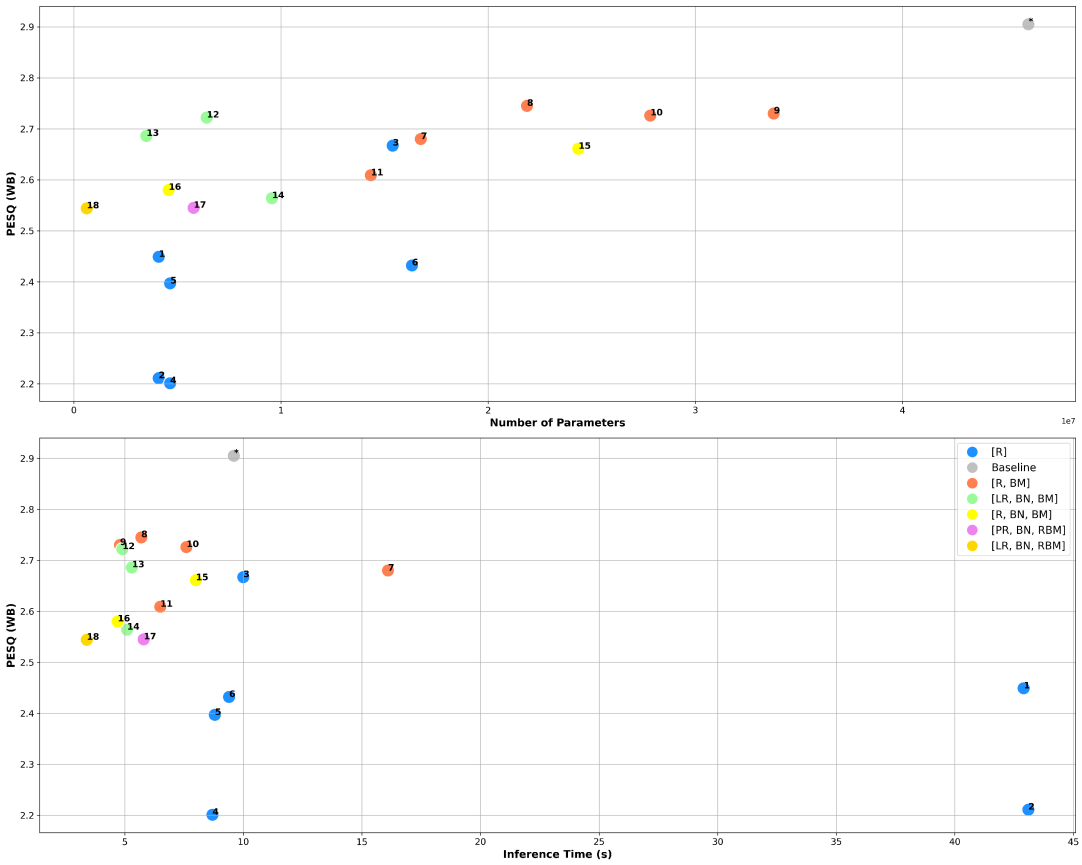


Figure 17 – Trade-off between computational efficiency and performance for PESQ (WB).

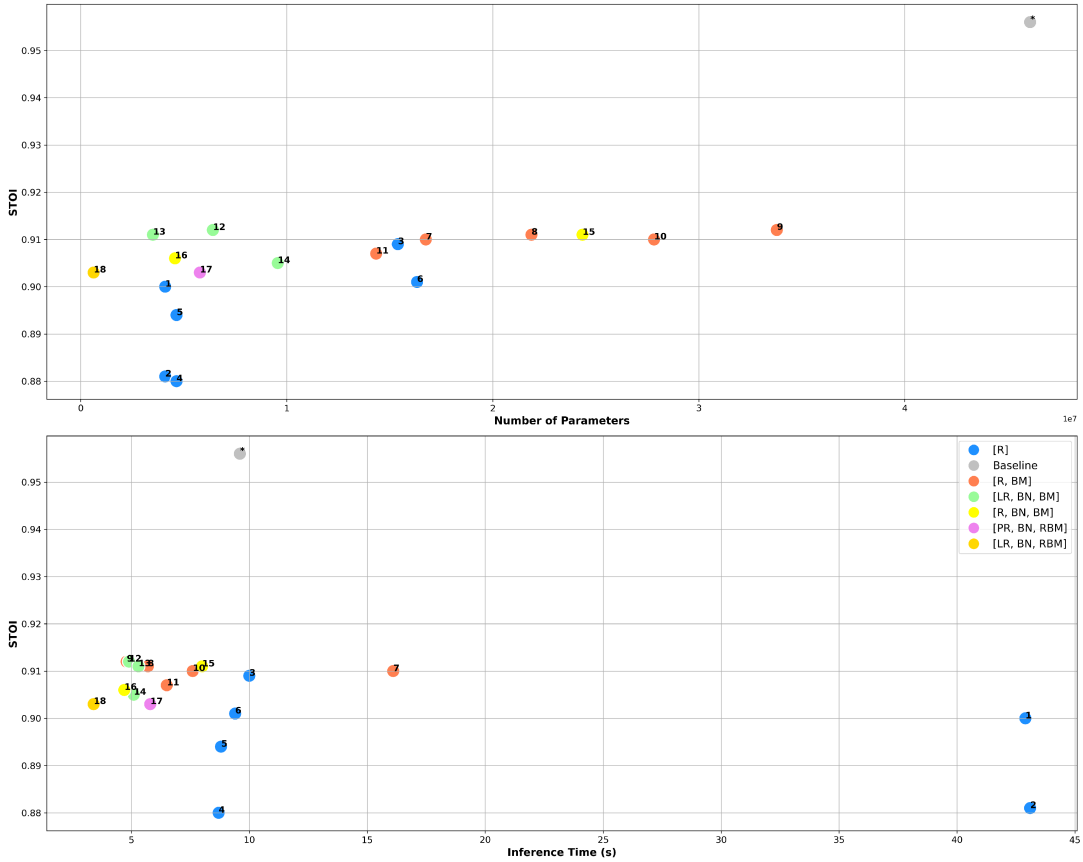


Figure 18 – Trade-off between computational efficiency and performance for STOI.

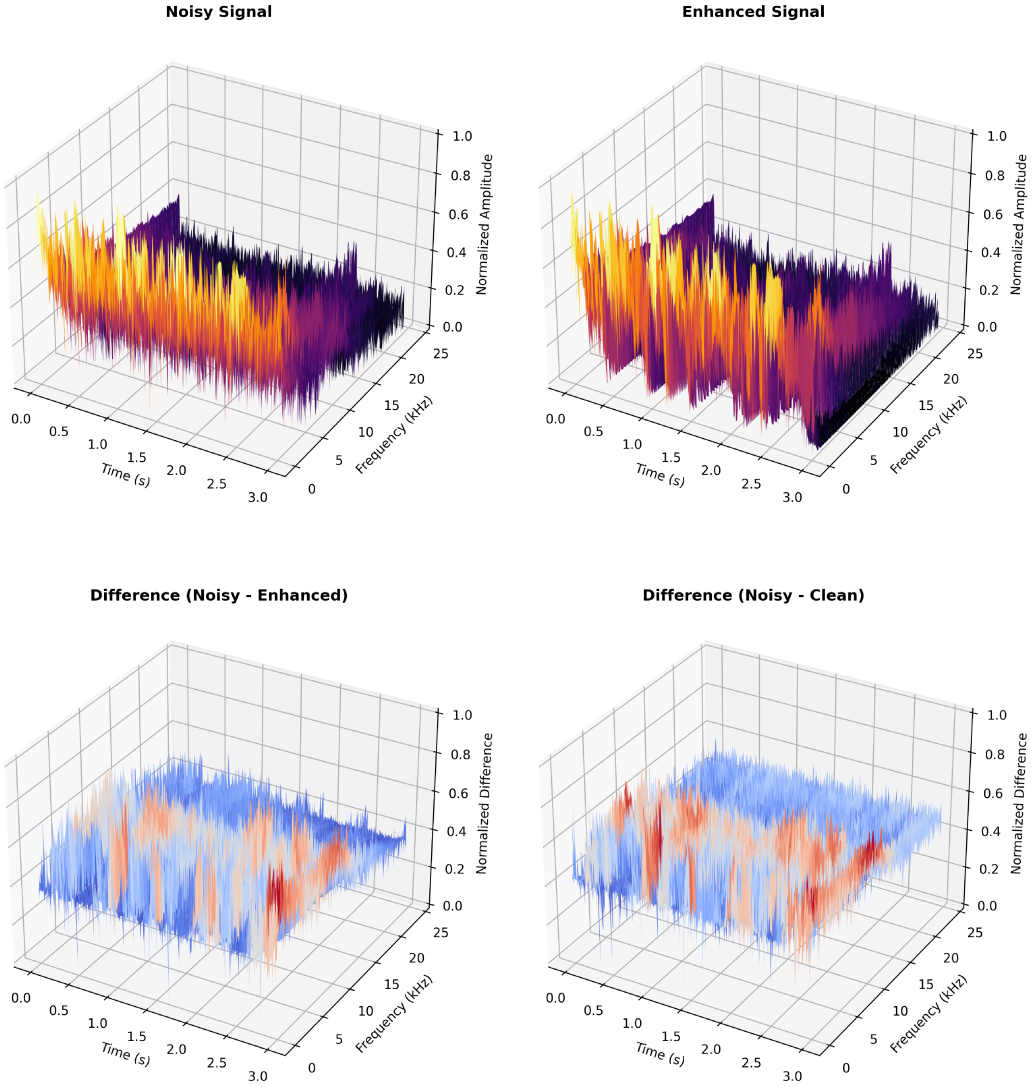


Figure 19 – Audio signal comparison between clean and enhanced signals for mamba-based architecture (Id 8).

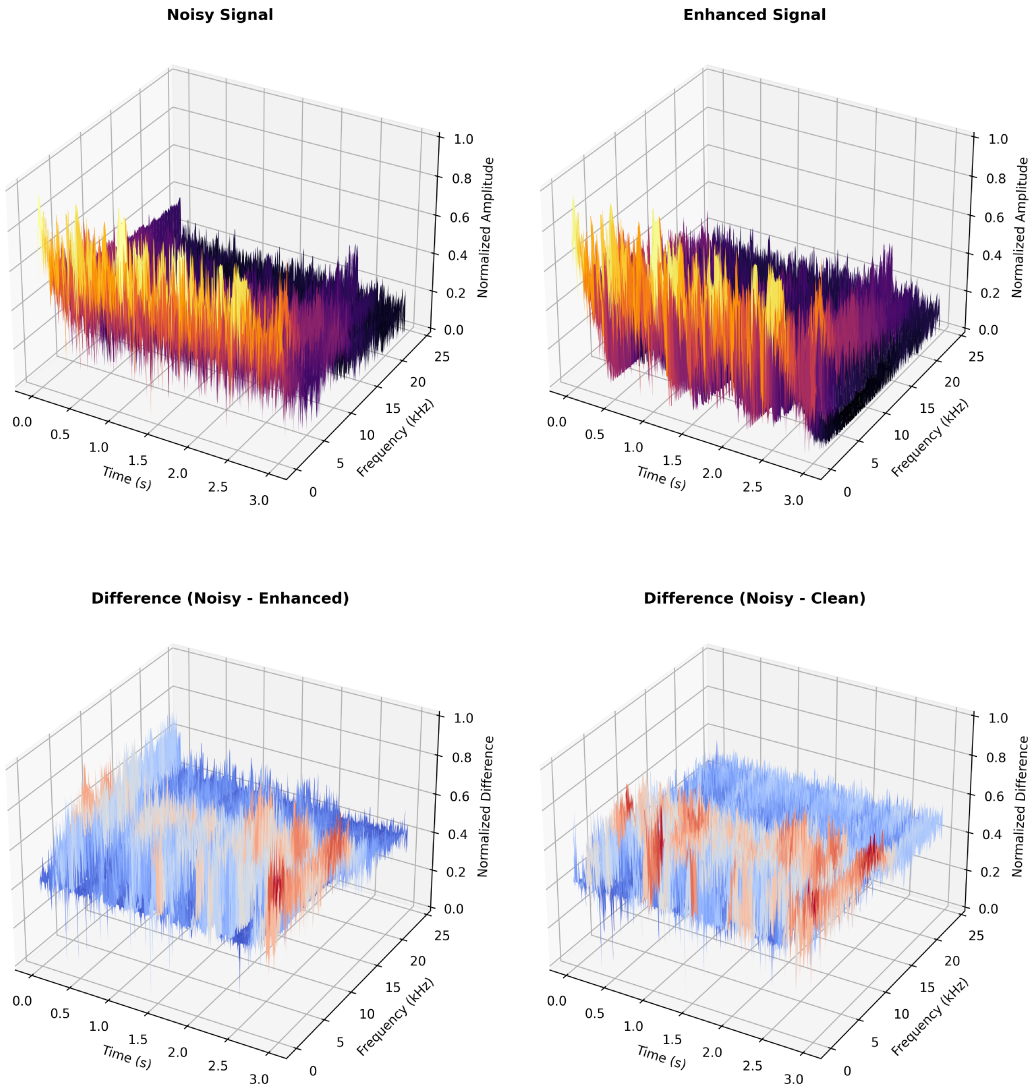


Figure 20 – Audio signal comparison between clean and enhanced signals for mamba-based architecture (Id 18).

Chapter 7

Conclusion

The experiments in this work highlight a critical balance between model complexity, performance, and computational efficiency. Larger models with more parameters generally achieve better performance metrics but have longer inference times and higher computational demands. For instance, the model with 21,878,913 parameters delivers high-performance scores while maintaining a relatively short inference time of 5.7 seconds. In contrast, the model with 626,753 parameters shows more degradation, but it is the fastest, with an inference time of just 3.4 seconds.

Choosing between these models depends on the specific application goals. The smaller model is advantageous for scenarios requiring reduced parameter counts and inference times to lower hardware requirements. It consumes only 1.36% of the original model's parameters, thus being more efficient and demanding less energy. The smaller model makes it more suitable for scaling to millions of users or environments with moderate background noise. In contrast, the larger model is better suited for environments with significant background noise due to its superior performance in challenging conditions.

7.1 Main Contributions

1. We developed and tested the new sequence models as bottlenecks and efficiently reduced the parameters of the entire model while maintaining performance, making it more suitable for real-time applications on single GPU.

2. We demonstrated that deeper models with more blocks and out channels generally provide shorter inference times and better performance metrics (PESQ and STOI) than models with fewer blocks despite increased computational demands.

3. We also added new activation functions and batch normalization to CleanUNet architecture, revealing their impact on model performance during training and inference, contributing to a more nuanced understanding of their effects on audio quality and computational efficiency.

4. We found that more blocks generally improve speech quality PESQ but do not linearly enhance intelligibility STOI. This highlights the need to balance model complexity with practical speech intelligibility and quality requirements.

5. The code is also available and can be found on GitHub at the following link: <https://github.com/Matheuspp/CleanUNet_MambaSSM>.

7.2 Limitations

- ☐ We couldn't test models larger than the baseline and with batch sizes bigger than 64. These configurations would require substantial computational resources that were not available. This hardware constraint limited the scope of our experiments and affected the feasibility of training these models.

- ☐ We did not perform subjective evaluations or include metrics for Speech Signal Distortion (SIG), Background Noise Intrusiveness (BAK), and Overall Quality (OVRL) as part of our analysis.

- ☐ Lack of architectural comparison standardization. We aimed to optimize the parameters with different configurations modified empirically.

7.3 Future Works

Future directions should include standardizing experimental parameters to better understand the impact of architectural changes and incorporating automated methods to optimize the parameters efficiently. This approach will help balance model complexity, performance, and computational efficiency, enhancing practical applicability and resource management. Another potential improvement for future work could involve experimenting with different crop lengths beyond the current 3-second duration. This would allow for assessing how varying the segment length impacts the model's performance, potentially uncovering optimal durations that better capture the relevant features.

Bibliography

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. **arXiv preprint arXiv:1409.0473**, 2014.

BEERENDS, J. G. et al. Perceptual evaluation of speech quality (pesq) the new itu standard for end-to-end speech quality assessment part ii: psychoacoustic model. **Journal of the Audio Engineering Society**, Audio Engineering Society, v. 50, n. 10, p. 765–778, 2002.

BRIDLE, J. S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: **Neurocomputing: Algorithms, architectures and applications**. [S.l.]: Springer, 1990. p. 227–236.

CHO, K. et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.

DEFOSSEZ, A.; SYNNAEVE, G.; ADI, Y. Real time speech enhancement in the waveform domain. **arXiv preprint arXiv:2006.12847**, 2020.

DING, J. et al. Longnet: Scaling transformers to 1,000,000,000 tokens. **arXiv preprint arXiv:2307.02486**, 2023.

DURAK, L.; ARIKAN, O. Short-time fourier transform: two fundamental properties and an optimal implementation. **IEEE Transactions on Signal Processing**, IEEE, v. 51, n. 5, p. 1231–1242, 2003.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. **Proceedings of the fourteenth international conference on artificial intelligence and statistics**. [S.l.], 2011. p. 315–323.

GOODFELLOW, I. et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, 2014.

GU, A.; DAO, T. Mamba: Linear-time sequence modeling with selective state spaces. **arXiv preprint arXiv:2312.00752**, 2023.

GUI, J. et al. A review on generative adversarial networks: Algorithms, theory, and applications. **IEEE transactions on knowledge and data engineering**, IEEE, v. 35, n. 4, p. 3313–3332, 2021.

HAWKINS, J. E. **Human Ear**. 2024. Encyclopedia Britannica, 29 Jul. 2024. Accessed: 2024-08-02. Disponível em: <https://www.britannica.com/science/ear>.

HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1026–1034.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.

JANNU, C.; VANAMBATHINA, S. D. Convolutional transformer based local and global feature learning for speech enhancement. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 14, n. 1, 2023.

KARL, K. A.; PELUCHETTE, J. V.; AGHAKHANI, N. Virtual work meetings during the covid-19 pandemic: The good, bad, and ugly. **Small group research**, SAGE Publications Sage CA: Los Angeles, CA, v. 53, n. 3, p. 343–365, 2022.

KONG, Z. et al. Speech denoising in the waveform domain with self-attention. In: IEEE. **ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2022. p. 7867–7871.

_____. Cleanunet 2: A hybrid speech denoising model on waveform and spectrogram. **arXiv preprint arXiv:2309.05975**, 2023.

KREPS, D. M. Nash equilibrium. In: **Game theory**. [S.l.]: Springer, 1989. p. 167–177.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LOIZOU, P. C. **Speech enhancement: theory and practice**. [S.l.]: CRC press, 2013.

MAAS, A. L. et al. Rectifier nonlinearities improve neural network acoustic models. In: ATLANTA, GA. **Proc. icml**. [S.l.], 2013. v. 30, n. 1, p. 3.

NYQUIST, H. Certain topics in telegraph transmission theory. **Transactions of the American Institute of Electrical Engineers**, IEEE, v. 47, n. 2, p. 617–644, 1928.

OLIVEIRA, D. de; PEER, T.; GERKMANN, T. Efficient transformer-based speech enhancement using long frames and stft magnitudes. **arXiv preprint arXiv:2206.11703**, 2022.

OLSON, H. F. **Music, physics and engineering**. [S.l.]: Courier Corporation, 1967. v. 1769.

OORD, A. V. D. et al. Wavenet: A generative model for raw audio. **arXiv preprint arXiv:1609.03499**, v. 12, 2016.

PARK, H. J. et al. Manner: Multi-view attention network for noise erasure. In: IEEE. **ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2022. p. 7842–7846.

PASCUAL, S.; BONAFONTE, A.; SERRA, J. Segan: Speech enhancement generative adversarial network. **arXiv preprint arXiv:1703.09452**, 2017.

RETHAGE, D.; PONS, J.; SERRA, X. A wavenet for speech denoising. In: IEEE. **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2018. p. 5069–5073.

ROBINSON, E. A.; TREITEL, S. Principles of digital wiener filtering. **Geophysical Prospecting**, European Association of Geoscientists & Engineers, v. 15, n. 3, p. 311–332, 1967.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18**. [S.l.], 2015. p. 234–241.

SANTURKAR, S. et al. How does batch normalization help optimization? **Advances in neural information processing systems**, v. 31, 2018.

SHANNON, C. E. A mathematical theory of communication. **The Bell system technical journal**, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948.

SOLÓRZANO, C. de et al. **ISBI cell tracking challenge**. 2015.

TAAL, C. H. et al. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In: IEEE. **2010 IEEE international conference on acoustics, speech and signal processing**. [S.l.], 2010. p. 4214–4217.

VALENTINI-BOTINHAO, C. et al. Noisy speech database for training speech enhancement algorithms and tts models. **University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR)**, 2017.

VASEGHI, S. V.; VASEGHI, S. V. Spectral subtraction. **Advanced Signal Processing and Digital Noise Reduction**, Springer, p. 242–260, 1996.

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.