

Optimizing CleanUNet Architecture Parameters for Enhancing Speech Denoising

Matheus Vieira da Silva¹[0000–0002–1673–8187], João Fernando
Mari²[0000–0001–8271–3802], and André Ricardo Backes³[0000–0002–7486–4253]

¹ School of Computer Science, Federal University of Uberlandia, Uberlandia, Brazil
`matheus.silva007@ufu.br`

² Institute of Exacts and Technological Sciences, Federal University of Viçosa, Rio
Paranaíba, Brazil `joaof.mari@ufv.br`

³ Departament of Computing, Federal University of São Carlos, São Carlos, Brazil
`arbackes@yahoo.com.br`

Abstract. Speech enhancement refers to a set of techniques aiming to recover clean speech from a corrupted signal. One way to corrupt a signal is through noise addition. Noise comes in a variety of ways. Suboptimal acoustic conditions can cause background noise and echo, hampering speech clarity and making denoising techniques necessary to enhance the speech. In this work, we optimized CleanUNet, a convolutional neural network (CNN) architecture proposed specifically for causal speech denoising tasks. We explored alternatives for the transformer bottleneck, such as Mamba architecture, capable of handling encoder outputs more efficiently with linear complexity, we also reduced the number of hidden channels in the convolutional layers. This decreases the model's parameter count and improves training and inference speed on a single GPU, offering a streamlined approach for enhanced performance. To our understanding, this is the first attempt to incorporate Mamba as a replacement for the vanilla transformer in the CleanUnet architecture.

Keywords: CleanUNet · Audio processing · Mamba architecture · Convolutional neural network.

1 Introduction

The COVID-19 pandemic and lockdowns have led to significant changes in how people work and interact with technology, exemplified by Zoom's exponential growth from 10 million daily participants in December 2019 to over 300 million by April 2020 [8]. The pandemic led to a sudden shift to remote work and virtual meetings, revealing a significant issue: poor home office acoustics negatively impact videoconferencing. Suboptimal acoustic conditions caused more background noise and echo, hampering speech clarity and increasing the effort required for video calls. Inadequate acoustic environments at home also hurt audio quality and workforce productivity. It's worth noting that the problem of insufficient acoustics isn't limited to home offices during videoconferencing.

Challenges of background noise disrupting communication have been a concern from the beginning of the digital revolution [20].

Advanced techniques like Spectral Subtraction [2] and Wiener filtering [15] have been developed to address the challenges in speech enhancement. These methods use advanced algorithms and filters to reduce background noise, enhance speaker voices, and improve audio quality. The main drawback of these techniques is that they assume stationary noise, meaning that the background noise features are expected to remain relatively constant during the entire audio recording. In real-world scenarios, like Zoom meetings, noise can often be non-stationary, fluctuating in intensity and spectral content over time. In recent years, thanks to advancements in deep learning, the efficient removal of non-stationary noise from speech has been significantly enhanced [26]. There are two primary methods for speech enhancement: the time-frequency domain and the waveform domain. Some cross-domain studies explore hybrid approaches that combine both methods [21, 25].

Recent advancements in deep learning, such as convolutional networks [10], generative adversarial networks [5], U-Net for segmentation [19], and transformer architecture [24] have facilitated the attainment of state-of-the-art performance by strategically harnessing the merits and limitations inherent to each architectural paradigm.

An architecture that attains state-of-the-art results is known as CleanUNet [9]. This model draws inspiration from [17] and combines transformers and U-Net to enhance noisy speech within the waveform domain. However, the CleanUNet architecture presents a formidable challenge: the size of the model and its vulnerability to quadratic complexity within the transformer bottleneck. These challenges lead to three main implications: 1) Constrained Computational Resources: the training and inference processes necessitate substantial computational resources due to the architecture’s computational intensity and its dependency on audio samples that vary with the sample rate; 2) Environmental Concerns: the first implication raises environmental concerns, including issues such as water and carbon footprint, which are exacerbated by the extensive computational demands when operating at scale. 3) Real-Time Performance: the model’s adaptability to resource-limited hardware is inadequate for real-time denoising applications.

In this work, we address the three primary challenges of CleanUNet by optimizing its architecture. Our contributions include introducing alternatives to the transformer bottleneck to reduce the model’s parameter count for efficiency and enhancing inference speed to facilitate real-time applications. Our revised model enhances efficiency by replacing the transformer bottleneck with alternatives that handle encoder outputs with linear complexity $O(n)$. Additionally, we empirically optimized the number of hidden channels in the convolutional layers. These modifications reduce the model’s parameter count and significantly improve both training and inference speeds on a single GPU.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 describes the material and methods used in our work.

Section 4 presents the experiments and results, while Section 5 concludes the paper.

2 Related Works

Spectral subtraction is a simple yet effective method for reducing background noise [23]. This method immediately reduces the amplitude of noise components while preserving speech signals by estimating the average noise spectrum during silent periods and subtracting it from the input signal spectrum. However, the randomness of noise may result in negative values and distortions.

Based on statistical estimates of the underlying signal and noise, Wiener filtering dynamically creates a linear filter capable of suppressing existing noise [18]. The signal and noise are modeled as stochastic processes with well-known spectral properties. Its self-adjusting and statistically-based design make the Wiener filtering very effective at removing real-world speech noise, especially in home office environments.

The statistical characteristics of stationary noises are stable and do not change significantly over time. Examples include constant motor hum, continuous background conversation, and consistent background noise. Constant background noise can be approximated by updating silent periods, techniques like spectral subtraction work best. Non-stationary noises that change rapidly are hard to estimate and suppress. Non-stationary sounds have time-varying statistics, meaning their characteristics change rapidly. Examples include abrupt sounds like breaking glass, sporadic construction noises, and sudden horns.

One of the early deep learning methods used for speech denoising was Recurrent Neural Networks (RNNs) [14]. RNNs process input sequences incrementally while preserving an internal state to capture temporal context, making them ideal for speech processing. RNN training gradually maps noisy speech inputs to clean target outputs and aims to reduce audio noise. However, RNNs have limitations, such as challenging inference and high computational cost.

As an alternative, Convolutional Neural Networks (CNNs) [12] have become popular. 1D CNNs use local patterns and work directly on raw audio for noise reduction. 2D CNNs operate using spectrogram representations. In both cases, the network captures local speech features using small convolutional filters to separate them from background noise. Pooling and downsampling aggregate information over time. Multiple convolutional layers learn higher-level features, recognizing speech components even in noisy conditions. One of the main advantages of CNNs is weight sharing, significantly reducing parameters compared to RNNs, thus resulting in more efficient training and inference. While RNNs leverage temporal context, CNNs employ localized filters to effectively isolate speech from noise in the time-frequency domain for effective noise reduction. Ongoing research is improving the performance of CNN-based speech denoising [26]. Generative Adversarial Networks (GANs) [5] employ a generator to enhance noisy speech and a discriminator that distinguishes between enhanced and clean

speech. Adversarial training improves the quality of the output. CNNs are commonly used in the generator and discriminator models in GAN architectures.

Recent advancements in deep learning have incorporated hybrid solutions involving U-Net [19] and Transformer [24] architectures. Transformers can capture long-term speech contexts and precise interactions between noisy inputs for denoising. Self-attention layers prioritize important speech components over background noise. Although the use of Transformers for speech processing is still being investigated, early results are promising [11].

3 Materials and Methods

3.1 Dataset

The database employed in this work was obtained from [22] and is composed of two separate datasets from 84 speakers with 28.4 hours of clean and noisy speech pairs, designed for training and testing 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 subsequently 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.

3.2 CleanUNet

CleanUNet [9] is a CNN architecture proposed specifically for causal speech denoising tasks. It follows the principles of the U-Net architecture, a popular model in image segmentation tasks, and it is similar to the model presented in [3]. As in U-Net, CleanUNet consists of an encoder-decoder structure with skip connections, enabling the network to preserve fine-grained details during upsampling. Each convolutional block usually contains multiple convolutional layers followed by max-pooling layers and ReLU activation. CleanUNet adapts this architecture to the domain of audio processing on the raw waveform. It also uses self-attention blocks in the bottleneck to improve its ability to separate noise from the clean signal. Each self-attention block is composed of a multi-head self-attention layer and a position-wise fully connected layer [24].

One notable aspect of CleanUNet is its ability to handle varying levels and types of noise present in audio signals. It outperforms state-of-the-art models in terms of speech quality, according to several objective and subjective evaluation metrics.

3.3 Evaluation Metrics

We employed two objective evaluation metrics to assess the performance of our trained model: the Perceptual Evaluation of Speech Quality (PESQ) and the Short-Time Objective Intelligibility (STOI) [13].

PESQ This metric focuses on the perceived quality of the speech, and it involves a psychoacoustic algorithm to simulate human hearing and compare the clean speech (original) and the enhanced speech (model output) to predict perceived quality. The calculation includes steps such as level alignment, filtering, and a detailed comparison process that considers various distortions and artifacts introduced by processing. The output score ranges from -0.5 to 4.5, where higher scores indicate better quality.

STOI Designed to predict the intelligibility of speech signals, the method involves calculating the short-time temporal correlation between the clean (original) and enhanced speech (model output) signals across various frequency bands. The process includes segmenting the signals into short frames, applying a Fourier transform to analyze the frequency content, and then computing the correlation between the corresponding frames of the clean and enhanced signals. The STOI score ranges from 0 to 1, where 0 indicates poor intelligibility and 1 indicates excellent intelligibility.

3.4 Transformer Architecture

First introduced by [24], the transformer architecture uses self-attention mechanisms [1] to capture the dependencies between different words or tokens in a sequence. It allows 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 below:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

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.

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

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^O \quad (2)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

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 . Researchers have proposed optimizations and approximations to address this challenge to reduce computational overhead [7, 16].

3.5 Longnet

The Longnet architecture was introduced by [4]. This new Transformer variant achieves linear computational complexity and logarithmic token dependency, 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 achieves this by dividing the input (Q, K, V) into equally sized segments with a predetermined segment length W . Each segment is then made sparse along the sequence dimension by selecting rows at specified intervals r . The segments are processed in parallel, after which they are scattered and concatenated to produce the output. Incorporating the concept of multi-head attention, the innovative 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 with an introduction of segment sizes r and dilation rates d hyperparameters.

3.6 Mamba Architecture

Mamba [6] is an innovative architecture designed to mitigate the quadratic complexity challenges faced by conventional attention models. This model addresses the scalability constraints of traditional Transformer models in processing long sequences and significantly enhances efficiency and flexibility with linear-time complexity for sequence modeling. It introduces two significant State Space Models (SSM) innovations: 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 (\bar{A}, \bar{B}) . This procedure translates the model from a continuous domain to a discrete one, enabling computation within a digital framework. The discretization process can be done through fixed formulas $\bar{A} = F_a(\Delta, A)$ and $\bar{B} = F_b(\Delta, A, B)$, where (F_a, F_b) pair is a discretization rule such as zero-order hold (ZOH).

One essential property of SSMs is Linear Time Invariance (LTI). In this case, 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 a gating mechanism of RNNs that can focus or filter out different parts of a context allowing SSMs to adapt over time and being input-dependent. 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 to take advantage of the memory hierarchy of modern 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 are applied to 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.

4 Experiments and Results

Experiments were performed on a PC with an AMD Ryzen 7 7800X3D processor, 32GB of RAM, and an RTX 3090 24GB GPU running Linux Ubuntu 20.04 OS. We modified the original CleanUNet⁴ code to run on Pytorch 2.1.1⁵, CUDA 12.3, and Python 3.10⁶, along with the modifications needed to implement Mamba architecture using mamba-ssm⁷.

We proposed and evaluated two variations of CleanUNet, both designed to reduce the number of learnable parameters, reducing complexing, training, and prediction time, and looking to maintain the quality. The first variation consists of applying dilated attention mechanisms, while the second involves replacing the transformer architecture with the Mamba architecture. Both models were compared with CleanUNet described in [9], which we considered the Baseline.

The baseline CleanUNet [9] consists of an encoder and a decoder, both 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 (Conv1x1) 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 is composed of 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 decoder layer by a skip connection in the reverse order. The bottleneck is composed of K 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 [9]. For this study we considered a CleanUNet model with $D = 8$, $K = 4$, and $N = 5$ [9]

The proposed dilated attention base transformers follow the UNet structure from CleanUNet, which is considered a baseline with a bottleneck composed of dilated attention layers. For this configuration, we considered a dilated attention-based CleanUNet model with $D = 4$, $K = 3$, and $N = 1$ for training and prediction in GPU. The self-attention block comprises a dilated multi-head self-attention layer (32 heads) and a fully connected layer (input/output size of 1024 and inner-layer size of 2048). The list of segment sizes $\{10, 20, 30, 60\}$ was provided with its respective dilation rates $\{2, 4, 8, 16\}$.

⁴ <https://github.com/NVIDIA/CleanUNet>

⁵ <https://pytorch.org/>

⁶ <https://www.python.org/>

⁷ <https://pypi.org/project/mamba-ssm/1.1.0/>

The proposed CleanUNet with a bottleneck composed of Mamba architecture also follows the configuration of the baseline CleanUNet but with changes in the number of encoder/decoder layers $D = 10$, and hidden channels starting from 32 from the first layers to max hidden channels 256. A Mamba architecture replaced the transformer bottleneck with $N = 1$. The dimensionality of the input vector in this bottleneck was set to 512 (input/output) with 16 layers.

We used a training configuration similar to [9]. 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%. In this work, we employ the random shift data augmentation technique within a range of 0 to S seconds and BandMask augmentation [3].

The CleanUNet loss function described in [9] combines a multi-resolution STFT (Short-Time Fourier Transform) loss with a waveform-based loss to maximize speech-denoising performance. It is composed of two parts: the first is the $l1$ loss applied directly to the waveform, which encourages the denoised output to resemble the time domain clean speech waveform closely, and the second part is the multi-resolution STFT loss, which modifies the spectrogram’s magnitude at various resolutions.

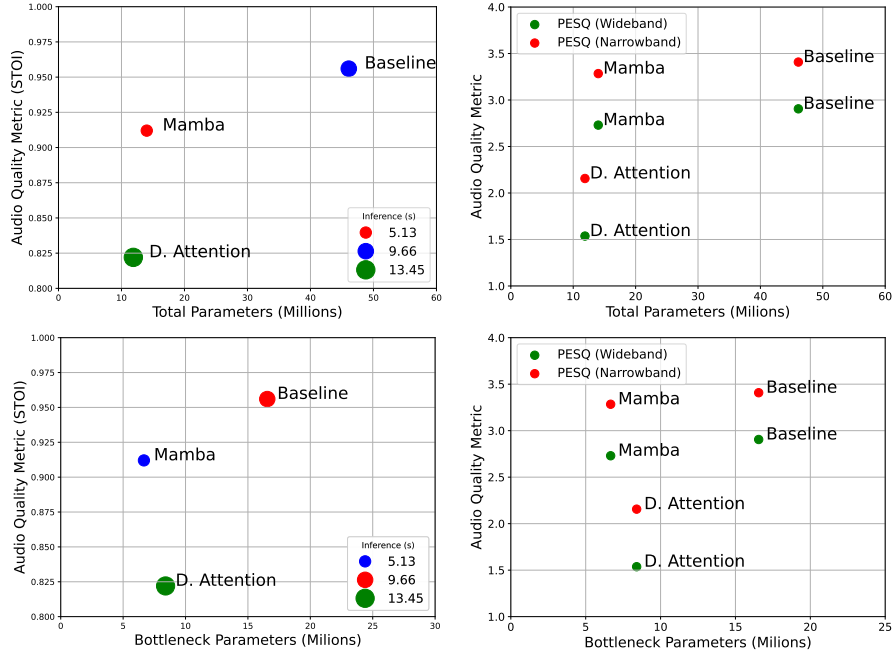
Table 1 presents the optimal outcomes and configurations for the speech enhancement model, highlighting the efficiency of implementing the Mamba architecture as a bottleneck. This approach significantly reduces the model’s parameters, enabling increased data throughput and larger batch sizes. Consequently, it facilitates training the model on a single GPU. Given the linear complexity of Mamba’s input handling, the model performs computations more quickly in the bottleneck. This design also supports larger vector sequences. Similarly, LongNet claims linear complexity, theoretically supporting longer sequences as well. According to the table results, LongNet uses fewer convolutional encoders to fit within memory constraints. This efficiency allows the preservation and utilization of features that might otherwise be discarded during the convolution process, thereby enhancing the model’s performance.

The results presented in Table 1 are visually detailed by the scatter graphs in Figure 1, in which we show the number of parameters on the x-axis versus the audio quality metrics on the y-axis. In the first row, we consider the total number of model parameters, including encoder, decoder, and bottleneck, and in the last row, we consider only the number of parameters in the bottleneck.

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 is reduced by 69.59%, the reduction in the STOI, PESQ-wideband, and PESQ-narrowband is only 4.60%, 3.63%, and 6.02%, respectively. It is important to note that, besides replacing the bottleneck with a Mamba architecture, the number of layers in the encoders/decoders was increased, however, the number of channels was reduced to enable the model to perform faster operations while saving memory. During

Table 1. CleanUNet model results and settings

Audio Quality	Baseline	D. Attention	Mamba
PESQ (Wideband)	2.905	1.537	2.730
PESQ (Narrowband)	3.408	2.156	3.284
STOI	0.956	0.822	0.912
Training Performance			
Total Params.	46.082177M	11.882497M	14.012545M
Bottleneck Params.	16.550656M	8.401920M	6.666240M
Network Configurations			
Conv. Encoders (D)	8	4	10
Kernel Size (K)	(4x4)	(3x3)	(5x5)
Stride (S)	2	2	2
Bottleneck Layers (N)	5	1	1
Training Configurations			
Learning Rate	0.0002	0.00025	0.00025
Batch Size	64	24	32
Crop Length	1.33s and 1.5s	3s	3s
Split	-	Train/Test	Train/Test

**Fig. 1.** Comparing the models considering the number of parameters Vs audio quality metrics. In the first row, we show the total number of parameters. In the last row, we show the number of parameters in the bottleneck. The first column also includes the inference time in seconds (s).

inference using the entire test set, the proposed model performed 46.9% faster than the baseline model.

5 Conclusion

This paper addressed some challenges related to using CleanUNet for speech denoising. Its good performance comes from a deep convolutional network that combines transformers and U-Net to enhance noisy speech within the waveform domain. Training requires a huge amount of resources, and its architecture limits the model’s adaptability to resource-limited hardware, making it inadequate for real-time denoising applications. We explored alternatives for the transformer bottleneck, such as Mamba architecture, capable of handling encoder outputs more efficiently with linear complexity, reducing the need for multiple convolutional layers. As a result, we drastically reduced the number of trainable parameters while preserving its performance, thus improving training and inference speed on a single GPU.

Acknowledgments. André R. Backes gratefully acknowledges the financial support of CNPq (National Council for Scientific and Technological Development, Brazil) (Grant #307100/2021-9). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Boll, S.: Suppression of acoustic noise in speech using spectral subtraction. IEEE Transactions on acoustics, speech, and signal processing **27**(2), 113–120 (1979)
3. Defossez, A., Synnaeve, G., Adi, Y.: Real time speech enhancement in the waveform domain (2020)
4. Ding, J., Ma, S., Dong, L., Zhang, X., Huang, S., Wang, W., Zheng, N., Wei, F.: Longnet: Scaling transformers to 1,000,000,000 tokens. arXiv preprint arXiv:2307.02486 (2023)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
6. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
7. Hassani, A., Shi, H.: Dilated neighborhood attention transformer. arXiv preprint arXiv:2209.15001 (2022)
8. Karl, K.A., Peluchette, J.V., Aghakhani, N.: Virtual work meetings during the covid-19 pandemic: The good, bad, and ugly. Small Group Research **53**(3), 343–365 (2022)
9. Kong, Z., Ping, W., Dantrey, A., Catanzaro, B.: Speech denoising in the waveform domain with self-attention. In: ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 7867–7871. IEEE (2022)

10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25** (2012)
11. Latif, S., Zaidi, A., Cuayahuitl, H., Shamshad, F., Shoukat, M., Qadir, J.: Transformers in speech processing: A survey. *arXiv preprint arXiv:2303.11607* (2023)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
13. Loizou, P.C.: *Speech enhancement: theory and practice*. CRC press (2013)
14. Maas, A., Le, Q.V., O'neil, T.M., Vinyals, O., Nguyen, P., Ng, A.Y.: *Recurrent neural networks for noise reduction in robust ASR* (2012)
15. McAulay, R., Malpass, M.: Speech enhancement using a soft-decision noise suppression filter. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **28**(2), 137–145 (1980)
16. Pagliardini, M., Paliotta, D., Jaggi, M., Fleuret, F.: Faster causal attention over large sequences through sparse flash attention. *arXiv preprint arXiv:2306.01160* (2023)
17. Petit, O., Thome, N., Rambour, C., Themyr, L., Collins, T., Soler, L.: U-net transformer: Self and cross attention for medical image segmentation. In: *MLMI – International Conference on Machine Learning in Medical Imaging*. pp. 267–276. Springer (2021)
18. Robinson, E.A., Treitel, S.: Principles of digital wiener filtering. *Geophysical Prospecting* **15**(3), 311–332 (1967)
19. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: *MICCAI – Medical Image Computing and Computer-Assisted Intervention*. pp. 234–241 (2015)
20. Shannon, C.E.: A mathematical theory of communication. *The Bell system technical journal* **27**(3), 379–423 (1948)
21. Tang, C., Luo, C., Zhao, Z., Xie, W., Zeng, W.: Joint time-frequency and time domain learning for speech enhancement. In: *29th International Conference on International Joint Conferences on Artificial Intelligence*. pp. 3816–3822 (2021)
22. 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 (2017)
23. Vaseghi, S.V., Vaseghi, S.V.: Spectral subtraction. *Advanced Signal Processing and Digital Noise Reduction* pp. 242–260 (1996)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
25. Wang, H., Wang, D.: Cross-domain speech enhancement with a neural cascade architecture. In: *ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing*. pp. 7862–7866. IEEE (2022)
26. Yuliani, A.R., Amri, M.F., Suryawati, E., Ramdan, A., Pardede, H.F.: Speech enhancement using deep learning methods: A review. *Jurnal Elektronika dan Telekomunikasi* **21**(1), 19–26 (2021)