Physical Sensors and Systems for Environmental Signals

-

A Comparative Study of Denoising Techniques for Speech Audio Signals

Mirko Morello
920601
m.morello11@campus.unimib.it

Abstract

Environmental acoustic recordings provide valuable data for ecological research, urban planning, and environmental monitoring. However, these recordings are often corrupted by noise from various sources, hindering their analysis. This study investigates and compares a range of denoising techniques applied to environmental audio, spanning classical signal processing methods and modern deep learning approaches. We evaluate techniques including spectral subtraction, Wiener filtering, and various autoencoder architectures (Residual, U-Net, Hybrid, and Transformer) on a synthetically generated dataset comprising clean speech from LibriSpeech and environmental noise from UrbanSound8K. Performance is assessed using quantitative metrics (PESQ, STOI, SI-SDR, MOS) and qualitative analysis via spectrograms and aural inspection. Our results demonstrate the relative strengths and weaknesses of each technique, providing insights into the selection of appropriate denoising methods for specific environmental audio applications. Deep learning models, particularly those utilizing a hybrid time-frequency loss, consistently outperform traditional methods, achieving significant improvements in signal quality and intelligibility. The U-Net architecture, operating in the frequency domain, and the Hybrid model, leveraging both time and frequency information, stand out as particularly effective.

1 Introduction

Environmental acoustic monitoring is a rapidly growing field with applications ranging from biodiversity assessment and wildlife conservation to urban noise pollution analysis and smart city development [15, 11]. The ability to automatically analyze large volumes of audio data from diverse environments holds immense potential for understanding ecological processes, monitoring environmental changes, and improving urban living conditions. However, a major challenge in analyzing environmental acoustic recordings is the presence of unwanted noise. Noise can originate from a variety of sources, including human activities (traffic, construction, industrial processes), weather events (wind, rain, thunder), and even the recording equipment itself. This noise can significantly degrade the quality of the audio data, obscuring important signals of interest and making it difficult to extract meaningful information.

Effective denoising techniques are therefore essential for preprocessing environmental audio data before further analysis. A wide range of denoising methods have been developed, from classical signal processing techniques like spectral subtraction and Wiener filtering to more recent deep learning-based approaches that leverage the power of neural networks to learn complex noise patterns. However, the relative performance of these different methods in the context of *environmental* audio, with its unique characteristics and diverse noise sources, remains an open question.

This study aims to address this gap by conducting a comprehensive comparative evaluation of several prominent denoising techniques applied to environmental acoustic recordings. We investigate both classical signal processing methods (spectral subtraction, Wiener filtering) and a range of deep learning architectures, including Residual Autoencoders, U-Net, Hybrid models, and Transformer-based autoencoders. Our evaluation is performed on a synthetically generated dataset combining clean speech signals from the LibriSpeech corpus [9] with environmental noise recordings from the UrbanSound8K dataset [14]. We assess the performance of each method using a combination of quantitative metrics, including Perceptual Evaluation of Speech Quality (PESQ), Short-Time Objective Intelligibility (STOI), Scale-Invariant Signal-to-Distortion Ratio (SI-SDR), and Mean Opinion Score (MOS), as well as qualitative evaluation through visual inspection of spectrograms and aural assessment.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of related work in audio denoising. Section 3 describes the datasets used in this study. Section 4 details the denoising methods investigated. Section 5 outlines the experimental setup, including training procedures and evaluation metrics. Section 6 presents the quantitative and qualitative results. Section 8 discusses the findings and their implications, concludes the paper and suggests directions for future research.

2 Related Work

Audio denoising has been a long-standing research topic with numerous proposed solutions. Traditional signal processing techniques, such as spectral subtraction [1], Wiener filtering [19], and wavelet denoising [2], have been widely used for noise reduction. These methods rely on statistical properties of the signal and noise and often make assumptions about their stationarity, which may not hold true for the complex and dynamic nature of environmental sounds. These initial methods often struggle with non-stationary noise and can introduce artifacts, such as ”musical noise,” which can be detrimental to the perceptual quality of the audio.

In recent years, deep learning has emerged as a powerful approach for audio denoising. Convolutional Neural Networks (CNNs) have been successfully applied to speech enhancement [3], demonstrating their ability to learn complex noise patterns. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have also been used for denoising sequential data like audio [18]. More recently, Generative Adversarial Networks (GANs) have shown promising results in speech enhancement by learning to generate clean speech from noisy input [10].

The U-Net architecture, originally developed for biomedical image segmentation [13], has been adapted for audio denoising and source separation [5]. Its encoder-decoder structure with skip connections allows it to effectively capture both local and global features of the audio signal.

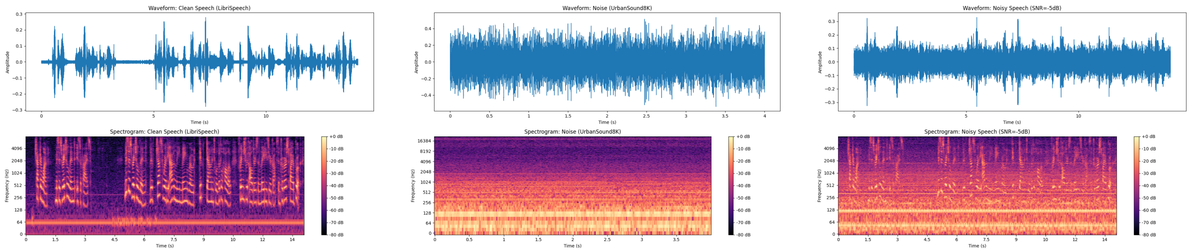
While much of the existing research has focused on speech denoising, the application of these techniques to environmental audio is less explored. Environmental audio presents unique challenges due to the wide variety of noise sources and their non-stationary characteristics. This study builds upon previous works by evaluating a focused selection of architectures – Residual Autoencoders, U-Net, Hybrid models, and Transformers – to establish their relative efficacy on environmental noise. We use a controlled synthetic dataset and a uniform evaluation methodology to ensure a fair comparison.

3 Dataset

For this study, we utilized two publicly available datasets: LibriSpeech [9] for clean speech and UrbanSound8K [14] for environmental noise. LibriSpeech is a large corpus of read English speech derived from audiobooks, containing approximately 1,000 hours of speech sampled at 16 kHz. We used the ”train.100” subset, which provides 100 hours of clean speech data. UrbanSound8K is a dataset of urban sound recordings, categorized into 10 classes (e.g., air conditioner, car horn, children playing, etc.). The recordings vary in length and sampling rate.

To create a controlled experimental setup, we generated a synthetic noisy dataset by mixing clean speech from LibriSpeech with environmental noise from UrbanSound8K. We first filtered the UrbanSound8K dataset to select noise samples that were at least 50% of the length of a randomly selected clean speech sample. This ensured that the noise was sufficiently long to cover a significant portion of the speech. We then randomly selected noise samples from this filtered set. Both the clean speech and the selected noise sample were resampled to 16 kHz, if necessary, using the ‘librosa‘ library [7].

The mixing process was performed at three different Signal-to-Noise Ratio (SNR) levels: -5 dB, 0 dB, and 5 dB. These levels were chosen to represent a range of noise conditions, from high noise (-5 dB) to moderate noise (5 dB). The SNR was controlled by scaling the noise signal



(a) Clean audio (b) Noise (c) Noisy audio

Figure 1: A sample of a clean audio and noise, which combined with a SNR of -5dB gives a noisy audio track.

relative to the clean speech signal. The mixing process can be mathematically represented as:

$$y(t) = x(t) + \alpha n(t) \tag{1}$$

where $y(t)$ is the noisy signal, $x(t)$ is the clean speech signal, $n(t)$ is the noise signal, and α is a scaling factor determined by the desired SNR:

$$\alpha = \sqrt{\frac{P_x}{10^{(SNR/10)} P_n}} \tag{2}$$

where P_x is the power of clean speech signal, P_n is the power of noise signal. A total of 28539 (equal to the number of the clean signals) noisy speech samples were generated, covering all combinations of clean speech samples, noise samples, and SNR levels.

The generated dataset was split into training and validation sets with an 80/20 ratio. A fixed random seed was used to ensure reproducibility of the split. This ensures consistent training and evaluation across different runs and experiments.

4 Methods

This section details the denoising techniques employed in our study, encompassing both classical signal processing approaches and deep learning models.

4.1 Classical Methods

4.1.1 Spectral Subtraction

Spectral subtraction is a classic noise reduction technique that estimates the noise spectrum and subtracts it from the noisy signal's spectrum. The underlying assumption is that the noise is additive and relatively stationary. This method is computationally inexpensive, but can lead to the ”musical noise” artifact, a common problem with spectral subtraction due to the inaccuracies in noise estimation.

The process involves the following steps:

1. **Short-Time Fourier Transform (STFT)**: The noisy audio signal, $y(t)$, is transformed into the frequency domain using the STFT, resulting in $Y(f,\tau)$, where f is the frequency bin and τ is the time frame.

2. **Noise Estimation**: The magnitude spectrum of the noise, $|N(f,\tau)|$, is estimated. In our implementation, we estimated the noise profile from the first 10 frames of the noisy audio, assuming these frames primarily contained noise. This assumes a short period of ”silence” or background noise at the beginning of each recording.

3. **Subtraction**: The estimated noise magnitude spectrum is subtracted from the magnitude spectrum of the noisy signal:

$$|\hat{X}(f,\tau)| = \max(|Y(f,\tau)| - |N(f,\tau)|, 0) \tag{3}$$

The max operation ensures that the resulting magnitude is non-negative. This can lead to isolated peaks in the spectrum, contributing to the ”musical noise” artifact.

4. **Inverse STFT (iSTFT)**: The denoised magnitude spectrum, $|\hat{X}(f,\tau)|$, is combined with the original phase of the noisy signal, and the iSTFT is applied to reconstruct the time-domain denoised signal, $\hat{x}(t)$.

We implemented spectral subtraction using the Librosa library [7]. The STFT parameters were set as follows: `n_fft=1024`, `win_length=512`, `hop_length=256`.

4.1.2 Wiener Filtering

Wiener filtering is a statistical approach to noise reduction that aims to minimize the mean squared error (MSE) between the estimated clean signal and the true clean signal. It requires knowledge (or estimation) of the power spectral densities of the signal and the noise. It generally performs better than basic spectral subtraction but still struggles with non-stationary noise.

We implemented the Wiener filter using the ‘scipy.signal.wiener‘ function [17]. This function applies a time-domain Wiener filter. The ‘mysize‘ parameter, which controls the filter size, was set to 512. This parameter determines the length of the impulse response of the Wiener filter, impacting its frequency selectivity.

4.2 Deep Learning Models

We investigated four deep learning architectures: Residual Autoencoder, U-Net, Hybrid model, and Transformer. Each model was trained in two variants: one using a ”simple” time-domain loss (v1) and the other using a ”hybrid” time-frequency loss (v2).

4.2.1 Residual Autoencoder (ResAutoencoder)

The Residual Autoencoder is a time-domain model that directly denoises the raw waveform. It is built using a convolutional encoder-decoder architecture with residual learning.

1. **Encoding Stage:** The encoder comprises a series of 1D convolutional layers that progressively extract higher-level temporal features from the noisy waveform. Each convolution is followed by batch normalization and a ReLU activation.

2. **Decoding Stage:** The decoder mirrors the encoder structure using transposed convolutions (ConvTranspose1d) to upsample the feature maps back to the original signal length.

3. **Residual Learning:** The network is designed to predict the residual noise. The predicted noise is subtracted from the input waveform, thereby preserving the original signal structure while effectively removing noise.

4.2.2 UNet for Spectrogram Enhancement (UNetSpec)

UNetSpec is a U-Net style architecture designed to enhance the magnitude spectrogram of an audio signal. Operating in the frequency domain, it takes the magnitude spectrogram (obtained via STFT) as input and processes it through a symmetric encoder-decoder structure with skip connections.

1. **Input Processing:** The model receives the magnitude spectrogram (with an added channel dimension) as input. This spectrogram is typically derived from the STFT of a time-domain waveform.

2. **Downsampling Path:** A series of convolutional blocks progressively extract features while reducing the spatial resolution. Each block consists of two convolutional layers with batch normalization and ReLU activations. The outputs of these blocks are saved for skip connections.

3. **Bottleneck:** After the downsampling stages, a bottleneck block further refines the features by increasing the channel dimensions through additional convolutions, effectively capturing higher-level representations.

4. **Upsampling Path:** The decoder mirrors the encoder using transposed convolutions to upsample the feature maps. At each upsampling step, the corresponding skip connection from the encoder is concatenated with the upsampled features, helping to preserve fine details.

5. **Output:** A final convolution reduces the number of channels back to one, yielding the enhanced magnitude spectrogram.

4.2.3 Hybrid Denoiser

The Hybrid Denoiser leverages both time-domain and frequency-domain processing to improve denoising performance by fusing complementary information from each domain.

1. **Time-Domain Branch:** A Residual Autoencoder (ResAutoencoder) operates directly on the raw waveform. This branch extracts temporal features using 1D convolutions in the encoder and reconstructs the signal via transposed convolutions in the decoder. A residual connection is employed, where the network predicts the noise component that is subtracted from the input.

2. **Frequency-Domain Branch:** In parallel, the input waveform is transformed into the frequency domain via STFT. The resulting magnitude spectrogram is processed using UNetSpec to enhance its quality. The enhanced magnitude is then recombined with the original phase information, and the inverse STFT (iSTFT) is applied to reconstruct a time-domain signal.

3. **Fusion:** The outputs from both the time-domain and frequency-domain branches are fused along the channel dimension. A final convolutional layer with

a Tanh activation function combines these outputs into a single denoised waveform.

4.2.4 Transformer Autoencoder Frequency based (TransformerAutoencoderFreq)

Although the model is named `TransformerAutoencoderFreq`, it does not implement a full transformer architecture with multi-head self-attention and position-wise feed-forward networks. Instead, it incorporates a `AttentionBlock2D` in the bottleneck that applies channel-wise attention based on global statistics. This is a much simpler attention mechanism compared to typical transformer layers but still provides a way to emphasize important features.

This model operates in the **frequency domain**. Other transformer-based audio models may work directly in the time domain (processing raw waveforms) or may apply transformer blocks along the time or frequency axes of the spectrogram. Here, the heavy lifting is done by convolutional operations that exploit local correlations, while the attention block provides a global weighting over channels.

By combining CNNs with a simple attention mechanism, the model reduces complexity and computational cost compared to full transformer models.

1. **STFT Conversion:** The model starts by converting the raw time-domain waveform (shape: $[B, 1, L]$) into its frequency-domain representation using the Short-Time Fourier Transform (STFT). This yields a complex-valued spectrogram from which the magnitude and phase are separated:

   $$\text{mag} = |STFT(x)|, \quad \text{phase} = \angle STFT(x).$$

   The magnitude spectrogram (with an added channel dimension) becomes the input to the CNN-based autoencoder.

2. **Encoding Stage:** The encoder is implemented with several 2D convolutional blocks (`EncoderBlock2D`). Each block performs two convolutions (with batch normalization and a LeakyReLU activation) followed by max pooling. This process progressively reduces the spatial resolution (frequency and time) while increasing the number of feature channels. The encoder also stores intermediate outputs for later use in skip connections.

3. **Bottleneck with Attention:** After the encoder, the network enters a bottleneck stage where features are further processed by a convolutional layer. Here, a custom `AttentionBlock2D` is applied. This block computes a channel attention map by performing global average pooling across the spatial dimensions (frequency × time) and then passing the pooled features through a small network (two 1×1 convolutions with a ReLU and a Sigmoid activation). This attention mechanism is similar in spirit to the self-attention found in transformers, although it is applied only at the channel level rather than over temporal or sequential positions.

4. **Decoding Stage:** The decoder mirrors the encoder but uses upsampling layers (via bilinear interpolation) instead of pooling. Skip connections are used to concatenate corresponding encoder outputs with the upsampled features. This U-Net-like structure helps to recover fine-grained details that may be lost during downsampling. The final convolution reduces the number of channels back to one, producing the enhanced magnitude spectrogram.

5. **Reconstruction:** Finally, the enhanced magnitude is recombined with the original phase to form a complex spectrogram:

   $$\text{enhanced\_spec} = \text{enhanced\_mag} \cdot e^{j\,\text{phase}},$$

   and the inverse STFT (iSTFT) is used to reconstruct the denoised time-domain waveform.

4.2.5 Simple Loss (v1)

The ”simple” loss is defined as a combination of the L1 loss (mean absolute error) and the mean squared error (MSE) loss computed in the time domain:

$$L_{\text{simple}} = L_{\text{L1}} + L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i| + \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2, \tag{4}$$

where x_i is the i-th sample of the clean waveform, $\hat{x}_i$ is the i-th sample of the denoised waveform, and N is the total number of samples.

The rationale behind combining these two losses is to leverage the strengths of each: the L1 loss is robust to outliers and helps in preserving sharp transitions by penalizing absolute differences, while the MSE loss is more sensitive to larger errors and thus encourages overall fidelity in the reconstruction. By summing them, the loss function balances robustness with sensitivity, helping the model to effectively reduce errors across the entire signal while maintaining fine-grained details.

4.2.6 Hybrid Loss (v2)

The ”hybrid” loss combines a time-domain loss (L1 loss between waveforms), a frequency-domain loss (L1 loss between magnitude spectrograms), and the negative of the Scale-Invariant Signal-to-Distortion Ratio (SI-SDR). This encourages the model to learn both temporal and spectral features of the clean signal, while also directly optimizing for a metric related to signal distortion, potentially leading to better perceptual quality and preserving perceptually important features.

$$L_{hybrid} = w_{time} \cdot L_{time} + w_{freq} \cdot L_{freq} + w_{sisdr} \cdot (-\text{SI-SDR}) \tag{5}$$

where:

- L_{time}:
$$L_{time} = \frac{1}{N} \sum_{i=1}^{N} |x_i - \hat{x}_i|$$
(L1 loss in the time domain)

- L_{freq}:
$$L_{freq} = \frac{1}{FT} \sum_{f=1}^{F} \sum_{t=1}^{T} \big| |X(f,t)| - |\hat{X}(f,t)| \big|$$
(L1 loss in the frequency domain)

- **-SI-SDR:** Negative Scale-Invariant Signal-to-Distortion Ratio. A higher SI-SDR indicates better signal quality, so minimizing the negative SI-SDR maximizes the SI-SDR.

- **Waveforms:** x and $\hat{x}$ are the clean and denoised waveforms, respectively.

- **Spectrograms:** $|X|$ and $|\hat{X}|$ are the magnitude spectrograms of the clean and denoised signals, respectively.

- **Loss Weights:** w_{time}, w_{freq}, and w_{sisdr} are weights that control the relative importance of the time-domain, frequency-domain, and SI-SDR losses. In our experiments, we used $w_{time} = w_{freq} = w_{sisdr} = 1/3$.

- **Dimensions:** F is the number of frequency bins and T is the number of time frames.

The frequency-domain loss encourages the model to preserve spectral details and the harmonic structure of speech. The inclusion of SI-SDR allows for the optimization of a metric that considers the overall signal distortion, leading to an improvement in perceptual features. The hybrid loss, by combining these three components, is designed to capture a more comprehensive representation of signal quality than any individual component alone. The use of a **negative** SI-SDR in the loss is a standard practice; while the loss value itself might become negative, the optimization process still works correctly by following the gradient.

5 Experimental Setup

We trained our deep learning models using the AdamW optimizer [6] with a learning rate of 3e-4, weight decay of 1e-5, and the `fused=True` option for improved performance. We used a ReduceLROnPlateau learning rate scheduler with a mode of “min”, a factor of 0.5, and a patience of 3 epochs. This scheduler reduces the learning rate when the validation loss plateaus. The models were trained with a batch size of 24 for a total of 6 epochs (for demonstration - longer training is recommended in practice). Training was performed on an NVIDIA 3090 GPU.

Checkpoints were saved every 100 training steps, storing the model state, optimizer state, scheduler state, training losses, validation losses, and STOI scores. We also implemented a mechanism to resume training from a saved checkpoint, allowing us to continue training if it was interrupted. The best-performing model based on the validation loss was saved separately.

Model performance was evaluated using a combination of objective and subjective metrics:

- **Perceptual Evaluation of Speech Quality (PESQ):** Measures the perceived quality of the denoised signal, with higher scores indicating better quality. We used the `pesq` library [12] with the wideband mode (“wb”) for 16 kHz signals. PESQ scores typically range from -0.5 to 4.5.

- **Short-Time Objective Intelligibility (STOI):** Measures the intelligibility of the denoised signal, with scores ranging from 0 to 1. We used the `pystoi` library [16] with the standard wideband setting (`extended=False`).

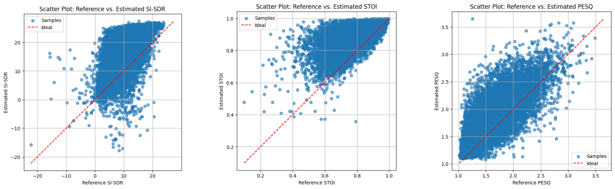
- **Scale-Invariant Signal-to-Distortion Ratio (SI-SDR):** Measures the overall distortion introduced by the denoising process, with higher values indicating better fidelity. We used our implementation of the SI-SDR calculation.

- **Mean Opinion Score (MOS):** We obtain MOS predictions using the pre-trained subjective model from the `torchaudio-squim` package [8]. It is important to remark that we use the model for non-matching references.

For models operating in the frequency domain (U-Net), the denoised waveform was reconstructed using the Griffin-Lim algorithm [4] to estimate the phase from the denoised magnitude spectrogram. We used 32 iterations for the Griffin-Lim algorithm.

It's relevant to add that STOI and PESQ are not differentiable (they are computed with fixed algorithms and cannot be used directly in gradient-based training). SI-SDR, on the other hand, can be computed in a differentiable manner.

We initially explored using neural network approximations of STOI, PESQ, and SI-SDR from the `torchaudio-squim` package to potentially accelerate the evaluation process. However, a comparison between these approximations and the reference implementations (using `pesq`, `pystoi`, and our SI-SDR implementation) revealed that the correlation, while present, was not strictly linear. Scatter plots of the approximated metrics versus the reference metrics showed considerable spread, indicating that the approximations, while faster, did not accurately reflect the reference values. Therefore, we opted for the more accurate, albeit slower, reference implementations to ensure the reliability of our evaluation (see Figure 2).



(a) SI-SDR

(b) STOI

(c) PESQ

Figure 2: Scatterplot of reference vs estimated

The evaluation was performed on the validation set. We employed functions to automate the evaluation process, including loading the trained models, computing the metrics, and generating visualizations.

As a baseline, we compared the performance of the denoising methods to the case where no denoising is applied (i.e., comparing the noisy signal directly to the clean signal).

6 Results

Table 1 shows the mean and standard deviation of the evaluation metrics (PESQ, STOI, SI-SDR, MOS) for each denoising method and the baseline (no denoising) on the validation set.

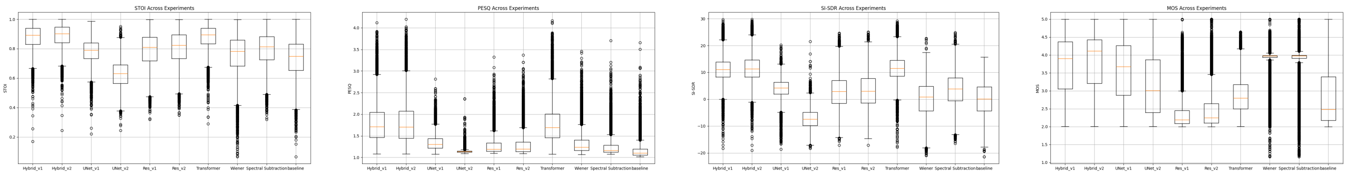
Table 1: Evaluation Metrics (Mean ± Standard Deviation)

Method	PESQ	STOI	SI-SDR (dB)	MOS
Baseline (No Denoising)	1.16 ± 0.19	0.74 ± 0.12	0.10 ± 4.22	2.84 ± 0.82
Spectral Subtraction	1.25 ± 0.21	0.80 ± 0.11	3.76 ± 5.72	3.83 ± 0.54
Wiener Filtering	1.33 ± 0.26	0.76 ± 0.14	-0.10 ± 6.51	3.82 ± 0.47
ResAutoencoder (v1)	1.28 ± 0.21	0.79 ± 0.11	2.73 ± 5.88	2.39 ± 0.51
ResAutoencoder (v2)	1.29 ± 0.24	0.81 ± 0.11	3.18 ± 5.69	2.52 ± 0.63
U-Net (v1)	1.35 ± 0.18	0.78 ± 0.08	3.95 ± 3.63	3.54 ± 0.78
U-Net (v2)	1.14 ± 0.04	0.62 ± 0.09	-7.27 ± 3.66	3.12 ± 0.80
Hybrid (v1)	1.81 ± 0.46	0.88 ± 0.08	11.29 ± 4.81	3.68 ± 0.74
Hybrid (v2)	1.81 ± 0.50	0.89 ± 0.08	11.69 ± 5.27	3.79 ± 0.74
Transformer	1.78 ± 0.45	0.88 ± 0.08	11.65 ± 4.90	2.87 ± 0.50

Figure 6 shows a representative training curve (Hybrid Autoencoder v2). Figure 4 shows example spectrograms. Figure 5 shows the performances of the most performing models. Most of the plots are omitted for brevity but are generated by the provided code.

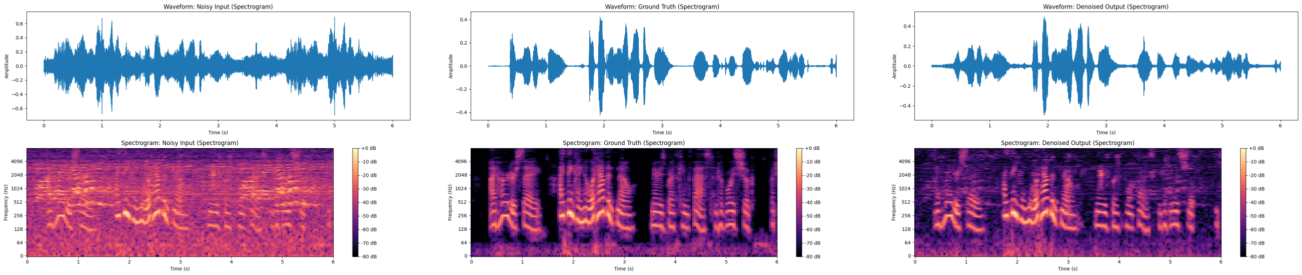
As shown in Table 1, every denoising method evaluated in this study significantly improves over the unprocessed baseline, although the magnitude of improvement varies considerably across methods. Classical signal processing techniques such as spectral subtraction and Wiener filtering yield modest gains. For instance, Wiener filtering achieves a mean PESQ of 1.33 compared to 1.25 for spectral subtraction, while the latter obtains a slightly higher STOI (0.80 versus 0.76). Although these classical methods are computationally efficient and remain useful in resource-constrained settings, their performance is generally limited relative to more sophisticated deep learning approaches.

In contrast, the deep learning models exhibit a pronounced advantage. The Hybrid Autoencoder and Transformer models, in particular, achieve superior performance, with SI-SDR values ranging from approximately 11.3 to 11.7 dB and robust STOI scores in the vicinity of 0.88–0.89. These results underscore the efficacy of leveraging frequency-domain information and incorporating a loss function that simultaneously considers both time-domain and frequency-domain discrepancies. The integration of these dual aspects appears to be instrumen-



(a) STOI (b) PESQ (c) SI-SDR (d) MOS

Figure 3: Metrics across all models in the folloqing order:
Hybrid v1 and v2, Unet v1 and v2, ResAutoencoder v1 and v2, Transformer v2, Wiener, Spectral Subtraction, Baseline



(a) Noise (b) Clean (ground truth) (c) Denoised

Figure 4: Hybrid Model v2 results over a validation's sample, with noise, ground truth and denoised result.

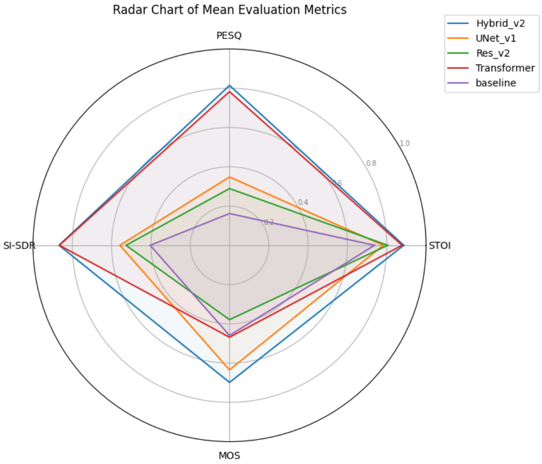


Figure 5: Training and validation loss for the Hybrid Autoencoder (v2) model.

tal in reducing distortion while preserving speech intelligibility.

Notably, the U-Net models, which employ an encoder-decoder architecture with skip connections designed to preserve fine spectral details, also improve upon the baseline. However, they fall short compared to the Hybrid and Transformer models. In our experiments, U-Net (v1), which utilizes a simple time-domain loss, outperforms U-Net (v2) that employs a combined time-frequency loss. This finding suggests that the optimal loss function may be architecture-dependent; in the case of U-Net, a loss focused solely on the time-domain may better exploit its frequency-domain processing capabilities, or alternatively, a different weighting scheme for the hybrid loss may be necessary.

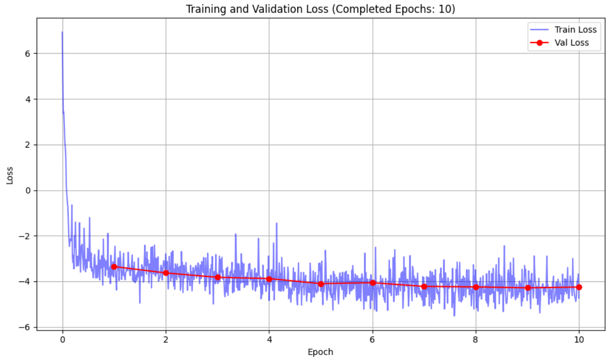


Figure 6: Training and validation loss for the Transformer Autoencoder (v2) model.

The Residual Autoencoder models deliver moderate performance improvements, yet their SI-SDR scores remain substantially lower than those achieved by the top-performing deep learning models. This indicates that while the residual learning framework can be beneficial, its effectiveness is highly contingent upon the architectural design and the loss formulation employed.

Figure 6 presents a representative training curve for the Transformer Autoencoder (v2), demonstrating steady convergence with validation losses closely tracking the training losses—an indication of minimal overfitting. Additionally, Figure 4 (not shown here for brevity) offers qualitative evidence in the form of spectrograms that highlight the ability of models such as the Hybrid (v2) to effectively suppress noise while retaining key spectral features of the clean speech signal. Figure 5 further summarizes the performance across the top-performing models.

It is important to note that the evaluation metrics employed—PESQ, STOI, SI-SDR, and MOS—may not fully capture real-world performance. The noise used in our experiments is derived from the UrbanSound8K dataset, which includes environmental sounds such as background chatter that can mimic intelligible speech. Consequently, metrics like STOI and PESQ, which are designed to assess speech intelligibility and quality, might erroneously attribute portions of the background noise to the target speech signal. Similarly, SI-SDR and MOS could be affected by the spectral and perceptual similarities between background chatter and actual speech. These limitations suggest that while the reported metrics provide a useful comparative benchmark, they may not encompass the full range of challenges encountered in real-world acoustic environments. Future work should thus explore additional objective and subjective metrics that can better differentiate between true speech content and speech-like noise.

In summary, although advanced deep learning methods—particularly the Hybrid Autoencoder and Transformer architectures—demonstrate superior denoising performance, classical methods and U-Net models still yield significant improvements over the baseline. The choice of denoising method should therefore be guided not only by performance metrics but also by the specific application requirements, computational constraints, and the complexity of the acoustic environment. Further research is warranted to refine evaluation metrics and to optimize loss formulations tailored to individual model architectures.

7 Future Work

Future research directions include:

- **Evaluation on Real-World Data:** Testing the trained models on real-world environmental recordings to assess their generalization capabilities.

- **Exploration of Other Architectures:** Investigating other deep learning architectures, such as more advanced Transformer variants or Generative Adversarial Networks (GANs).

- **Adaptive Denoising:** Developing methods that can adapt to different noise conditions and SNR levels without requiring explicit SNR information.

- **Semi-Supervised or Unsupervised Learning:** Exploring approaches that can leverage unlabeled data, as obtaining large amounts of clean environmental audio can be challenging.

- **Integration with Downstream Tasks:** Examining how denoising impacts the performance of downstream tasks such as sound event detection or species identification.

- **Exploration of Alternative Evaluation Metrics:** Given that some noise samples in UrbanSound8K mimic intelligible speech, traditional metrics (e.g., STOI, PESQ, SI-SDR, MOS) may not fully capture the perceptual quality of denoised audio. Future work should explore additional objective and subjective metrics that better differentiate between true speech and speech-like noise, thereby providing a more comprehensive evaluation.

8 Conclusion

This study presented a comparative evaluation of various denoising techniques for environmental acoustic recordings, encompassing both classical signal processing methods and deep learning models. Our findings demonstrate the clear advantage of deep learning approaches, with the Transformer and Hybrid models achieving superior performance in terms of signal quality and intelligibility. In particular, leveraging frequency-domain information through self-attention mechanisms (in the Transformer model) and combining time- and frequency-domain losses (in the Hybrid model) proved highly effective. Although the U-Net architecture showed promise, its overall performance was lower than that of the Transformer and Hybrid models. Moreover, for U-Net, a simpler time-domain loss (v1) yielded better results than a combined time-frequency loss (v2), indicating that the benefits of hybrid loss formulations depend on the specific model design. Future work should focus on validating these findings on real-world datasets and exploring more sophisticated architectures and training strategies.

References

[1] Steven F Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2):113–120, 1979.

[2] David L Donoho. Denoising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[3] Szu-Wei Fu, Cheng Yu, Tao-Wei Hsieh, Peter Plantinga, Shahab Mirsamadi, Mirco Ravanelli, and Yu Tsao. End-to-end waveform utterance enhancement for robust speech recognition. *arXiv preprint arXiv:1803.07070*, 2018.

[4] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[5] Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. In *18th International Society for Music Information Retrieval Conference*, 2017.

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[7] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. 8:18–25, 2015.

[8] A. Bhimanpally P. Seitz C. Weng D. Ditter Y. Tsao Y. Tsao P. Babu, J. Huang. torchaudio's SQUIM: Reference-less Speech Quality and Intelligibility Measures.

[9] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[10] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[11] Bryan C Pijanowski, Luis J Villanueva-Rivera, Sarah L Dumyahn, Almo Farina, Bernie L Krause, Brian M Napoletano, Stuart H Gage, and Nadia Pieretti. Soundscape ecology: the science of sound in the landscape. *BioScience*, 61(3):203–216, 2011.

[12] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2:749–752, 2001.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241, 2015.

[14] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. Dataset and taxonomy for urban sound research. *22nd ACM international conference on Multimedia*, pages 1041–1044, 2014.

[15] Jérôme Sueur, Sandrine Pavoine, Olivier Hamerlynck, and Stéphanie Duvail. Rapid acoustic survey for biodiversity appraisal. In *PLoS One*, volume 3, page e4065. Public Library of Science, 2008.

[16] Christian H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2125–2136, 2011.

[17] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[18] Felix Weninger, John R Hershey, Jonathan Le Roux, and Björn Schuller. Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. pages 91–99, 2015.

[19] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*. Wiley, 1949.