POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Enhancing a Stationary Noise Suppressor with Artificial Neural Networks for Non-Stationary Noise Removal

Tesi di Laurea Magistrale in
Music and Acoustic Engineering

Michele Perrone, 10472602

Advisor:
Prof. Fabio Antonacci

Co-advisors:
Christof Faller, Ph.D.

Academic year:
2025-2026

Abstract: Despite artificial neural networks (ANNs) making rapid progress in the field of noise removal for audio signals, the problems of computational complexity and behavior on unseen noise types remain relevant to the present day. Noise suppression systems often need to be adopted in low-resource communications systems such as mobile phones, webcams, and micro-controller units (MCUs), that cannot meet the requirements of most deep learning models and that require both real-time and causal processing. Additionally, these systems have to be used in a wide variety of noise situations, which can be a problematic aspect in the case of purely data-driven deep learning methods. To overcome these limitations, we propose a system for noise removal in speech signals that combines the robustness of a traditional stationary noise suppressor with the generalization capabilities of artificial neural networks. Unlike most end-to-end models, which employ ANNs for a direct enhancement of the noisy spectrogram or raw waveform, the scope of the ANN in our system is limited to enhancing gain filters that are computed by the stationary noise suppressor, with the goal of removing residual non-stationary noise, which is notoriously difficult to eliminate without complex heuristics. This gives us greater control on the denoising process and limits artifacts that can arise from direct short-time spectral manipulation. Our evaluation shows that the proposed system is able to perform effective real-time denoising on unseen noise types, while retaining a lower complexity than the vast majority of state of the art deep learning techniques.

Key-words: Speech Enhancement, Noise Suppression, Deep Learning, Digital Signal Processing, Convolutional Neural Networks, Recurrent Neural Networks

1. Introduction

1.1. Problem statement

Most communication devices have to deal with the presence of background noise, which can hinder the effectiveness of such systems and impair the perceived quality of audio signals. With the rapid spread of high quality broadband telephony, voice over internet protocol (VoIP) devices, and multi-platform communication software, the problem of noise in speech signals has become even more pronounced [1] [2]. Whereas traditional telephony has always limited bandwidth severely, the adoption of higher sampling frequencies has allowed for higher fidelity audio communication thanks to the full spectrum of speech signals being transmitted [3]. This has introduced the need for noise reduction systems that could address wideband noise [4] [5].
Fortunately, addressing noise reduction in a specific domain such as speech has its advantages, and many techniques have been developed specifically for this task over the years. For instance, model-based speech enhancement [6] uses speech models in order to separate more reliably speech from noise, and even reconstruct missing parts of the voice spectrum with pitch estimation synthesis of speech harmonics. On the other hand, noise removal in domains that are different from speech can present many additional challenges.
Another aspect that is common to most of today's communication devices is that speech signals are usually acquired and transmitted with a single audio channel. While it is possible to achieve a substantial noise suppression for monaural speech signals, it is debatable whether this can lead to improved intelligibility [7].
When dealing with communication devices such as mobile phones, webcams, or portable microphones, the computational complexity of speech enhancement solutions is also a key aspect. The noise removal often takes place in low-resource and low-power microcontroller units (MCUs) that are embedded in the device. These MCUs may also run other signal processing related tasks, which further limits the available resources. Therefore, designing efficient noise removal techniques is a fundamental aspect for most real-world applications.

1.2. Approach

In this manuscript, we present our work for the task of noise suppression in speech signals. We approach the problem by designing a system that seamlessly integrates a traditional stationary noise suppressor with an artificial neural network (ANN). The goal of the ANN is to improve the performance of the noise suppressor by providing a further non-stationary noise reduction, removal of musical noise, and securing a more consistent performance of the system across different noise types and signal-to-noise ratios. Compared to state of the art deep learning techniques, we want to build a system that is more suitable for applications with constrained memory and computational resources. In order to achieve this, we build a pre-processing pipeline that implements a priori knowledge about the audio signal domain and about the nature of the noise. This enables us to drastically reduce the requirements of the ANN while achieving consistent results.

1.3. Contents

This manuscript is structured as follows. Section 2 presents the current state of art, while Section 3 describes our proposed method. This includes the dataset, the pre-processing stage and the neural network architectures. Section 4 shows the experimental results of our proposed system with several objective metrics. Section 5 draws the conclusions of our work a presents possibile future developments.

2. State of the art

The state of the art presents us with techniques that are based on different approaches. Many systems are not designed for real-time usage, and some of them require heavy computational resources or have a significant memory footprint. For the problem statement of this manuscript, the following sections split up the existing methods into three main categories: stationary noise suppressors, deep learning noise suppressors, and combinations of both.

2.1. Stationary noise suppressors

Traditional noise suppression algorithms are usually based on two steps: noise estimation and noise removal.
The noise estimation can be achieved in different ways. In the case of a double microphone setup, one microphone can be aimed at the speaker, while the second one can be used for capturing the noise. In those setups where only

one microphone is available and both speech and noise are mixed together, the noise estimate can be obtained by carefully identifying those portions of audio where the speech is absent. This task is usually performed with the help of a voice activity detector (VAD) [8]. Alternatively, it is possible to identify the noise with statistical approaches without resorting to a VAD, assuming that the speech signal presents frequent pauses where only the background noise is present [9] [10]. In most applications, the noise can change over time, which presents an additional challenge: it is therefore important to decide how quickly should the system react to these changes. Once the noise has been estimated, it can be removed from the signal. This can be achieved by means of spectral subtraction or short-time Wiener filtering [11], which prove to be particularly effective under the assumption that the noise is additive. This hypothesis holds true for most types of environmental noise. These approaches can be further generalized [12] in order to compute gain filters, which can be multiplied with the noisy magnitude spectrogram or power spectrogram to obtain a denoised signal. Frequently, a time and frequency smoothing operation is performed on the aforementioned gain filters in order to reduce noise removal artifacts [13].
Figure 1 shows the structure of a typical time-frequency stationary noise suppressor. First, the noisy waveform is transformed into a magnitude spectrogram or a power spectrogram, that is used to compute a stationary noise estimate. For this step, the computational complexity can be reduced by downsizing the frequency resolution of the noisy spectrogram, which is usually done by grouping several frequency components of the spectrogram into perceptual bands, such as mel-bands [14], ERB bands [15], or the Bark scale [16]. Based on the noise estimate and the noisy spectrogram, a gain filter is computed. This filter is then multiplied with the noisy spectrogram in an element-wise fashion in order to obtain a denoised version of the spectrogram. If the dimensionality of the gain filter is lower than that of the noisy spectrogram, the gain filter has to be applied in a way that properly maps the attenuation of each band onto the frequency components of the noisy spectrogram. Once the denoised spectrogram is obtained, the denoised version of the audio waveform can be synthesized. Since the gain filter is applied only to the magnitude or the power of the noisy spectrogram, the original phase of the signal is used for the synthesis stage. This procedure is common to most noise suppressors, because phase manipulation in additive noise removal brings no clear benefits [17].

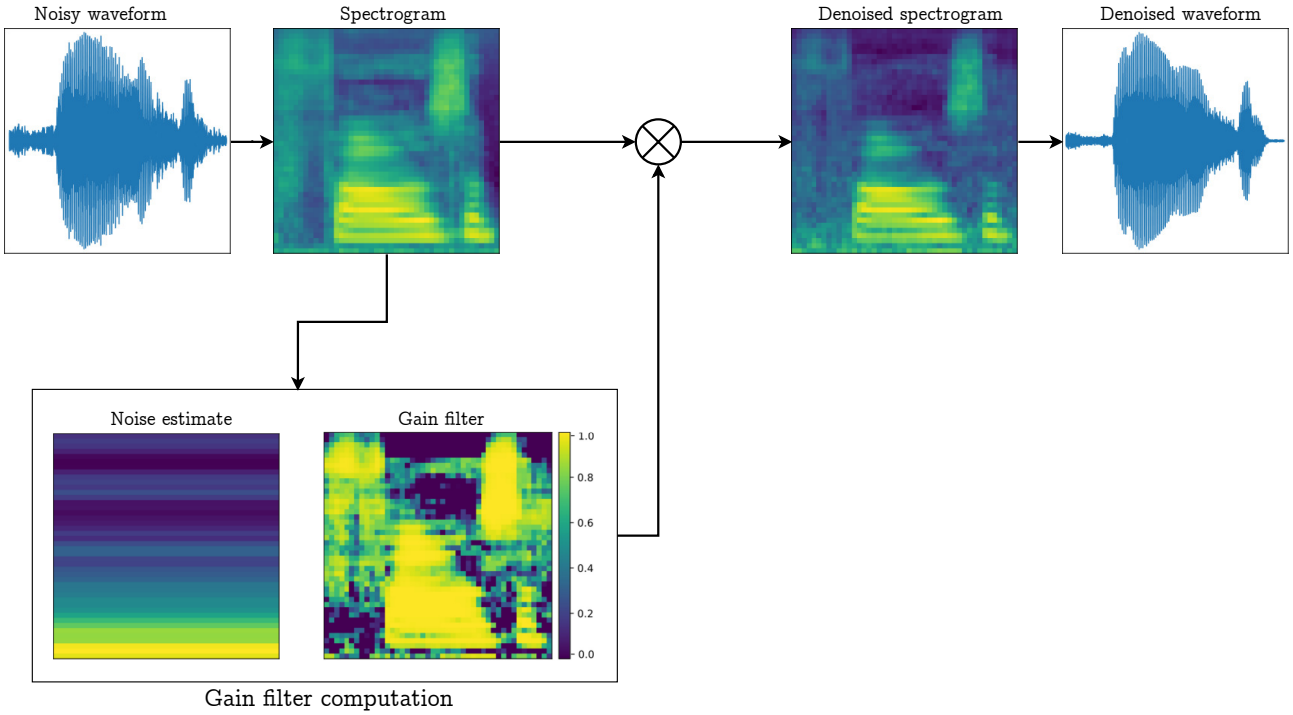


Figure 1: Schematic representation of a typical stationary noise suppressor.

2.2. Deep learning noise suppressors

In recent years, data-driven and deep learning methods have made significant advancements in the field of speech enhancement. Unlike traditional stationary noise suppressors, deep learning models are able to find a suitable mapping between a representation of the noisy signal and that of a denoised signal with little or no domain-specific knowledge. This representation can be based on the raw waveform [18], on a time-frequency spectrogram [19], or on a combination of both time and frequency domain features [20]. Deep learning models usually need a large collection of examples in order to find such a mapping, and this aspect can present additional challenges. First, the dataset has to be representative enough so that a model is able to give consistent

performance in real-world scenarios. To achieve this, datasets are often augmented by duplicating some or all of their data and performing specific transformations onto them; for example, when dealing with raw waveforms, it is useful to train the network with portions of signals that present different amplitudes. Second, the complexity of the deep learning model has to be chosen carefully: a model too simple may not be able to approximate a suitable mapping function, while a model too complex may learn patterns that are specific to the training dataset only and give poor performance in real-world applications. The problem of generalization in audio denoising tasks has been recently explored in [21], where the authors propose to train a multitude of models based on properties like speaker gender, noise type, and signal-to-noise ratios.
State of the art deep learning models take frequently advantage of the fact that time-frequency signal representations present a certain similarity with regular two-dimensional images. Recent research has mainly presented models based on the combination of convolutional layers [22], residual learning, autoencoder architectures [23] [24] [25], and recurrent neural networks [26] [27]. Thanks to their ability of efficiently identify patterns in images, convolutional layers have been shown to outperform traditional algorithms in tasks like hand-written character recognition [28], automatic document recognition [29], and medical image segmentation [30]. These innovations are also frequently applied in the audio domain. For example, the authors of [31] present U-Net, a highly-performing image segmentation ANN which is based on a fully convolutional autoencoder with skip connections. Architectures based on U-Net have been first employed in the audio field for effective source separation [32], and more recently also for denoising of audio signals [33] [18].
As previously mentioned, recent deep learning research is mostly focused on finding new models that approximate a mapping between noisy and clean power spectrograms, or between noisy and clean audio waveforms. Unfortunately, even those approaches that are designed as causal and real-time are generally unsuitable for low-resource computing, or they require a GPU for their execution. Moreover, only a minority of approaches tries to combine data-driven methods with digital signal processing expertise; however, presenting an artificial neural network with features that are engineered with domain-specific knowledge can lead to substantially leaner models without compromising their performance and with potentially higher generalization capabilities. In [34], the authors train an ANN for the estimation of Wiener filters for noise suppression, and they show how such a network is able to outperform an end-to-end model when dealing with unseen noise types. In [20], the author proposes a real-time "hybrid DSP/deep learning approach" for the denoising of speech signals. This approach computes the features based on a perceptually motivated sub-band decomposition, pitch extraction and spectral non-stationarity. The output of the ANN, which is based on gated recurrent layers (GRUs), consists of a voice activity detector and of a gain filter.

3. Proposed method

The primary aspect of the proposed approach is the combination of a traditional stationary noise suppressor with an artificial neural network (ANN). Figure 2 provides a schematic representation of our approach. Given a noisy spectrogram, the stationary noise suppressor provides an estimate of the stationary noise. Based on this estimate, we compute a gain filter that could be multiplied element-wise with the noisy spectrogram in order to remove the estimated stationary noise. However, since the goal is to eliminate also non-stationary noise, an ANN is used to enhance the aforementioned gain filter, resulting in the modified gain filter. This filter is then used to obtain the denoised spectrogram. In order to re-synthesize the denoised waveform, the denoised power spectrogram is combined with the phase spectrogram of the noisy waveform.
By integrating the ANN into a system that is designed with domain-specific knowledge, we are able to reduce the complexity of network, simplify its training procedure, and reduce the risk of poor performance in situations that differ from the training dataset. There are key advantages in training the ANN to enhance a gain filter instead of the noisy spectrogram directly. First, gain filters contain values that are fixed in a well-defined numerical range that does not vary over time. Spectrograms, on the other hand, have to be normalized, which is not easy task to achieve if the sound intensity varies over time, and would involve the design of an automatic gain control (AGC). Second, we take advantage of the fact that we are able to remove a substantial amount of stationary noise with a computationally inexpensive noise suppressor. This way, the attention of the ANN is focused on the non-stationary noise components that are difficult to estimate and treat with a traditional digital signal processing approach, and it is able to reduce artifacts that are typical of stationary noise suppressors, such as musical noise. Third, by working with gain filters the feature becomes space smaller and sparser in comparison to power spectrograms. We can achieve a smaller feature space because the number of frequency bands needed for an effective gain filter is lower than the amount of bands needed for re-synthesizing a time-domain signal from a power spectrogram. Low-resolution power spectrograms as those shown in Figure 4 can not be used for obtaining a waveform because of the loss of details in the frequency axis, as is visible in comparison to Figure 3. In terms of data sparsity, we can observe from the two pairs of Figures 4a – 4b and 5a – 5c that it is easier to find a mapping between the stationary noise suppressor gain filter and the target gain filter rather than from the noisy spectrogram to the clean spectrogram.

A further consideration to be made is that the aim of our system is not to reduce noise at the cost obtaining an artificially-sounding denoised signals. We target a reasonable balance between noise reduction, residual artifacts, and rejection of sound that does resemble the time-frequency structure of speech. In a typical use-case such as a video conference, an overlap of two speakers may occur; in this case, the noise removal should focus on other background noise, and not on suppressing one of the two speakers.
In the following sections, our proposed method is presented in further detail. Section 3.1 provides a description of the dataset, while Section 3.2 details our pre-processing strategy. Finally, Section 3.3 discusses the different architectures of artificial neural networks that are employed in our proposed noise suppressor.

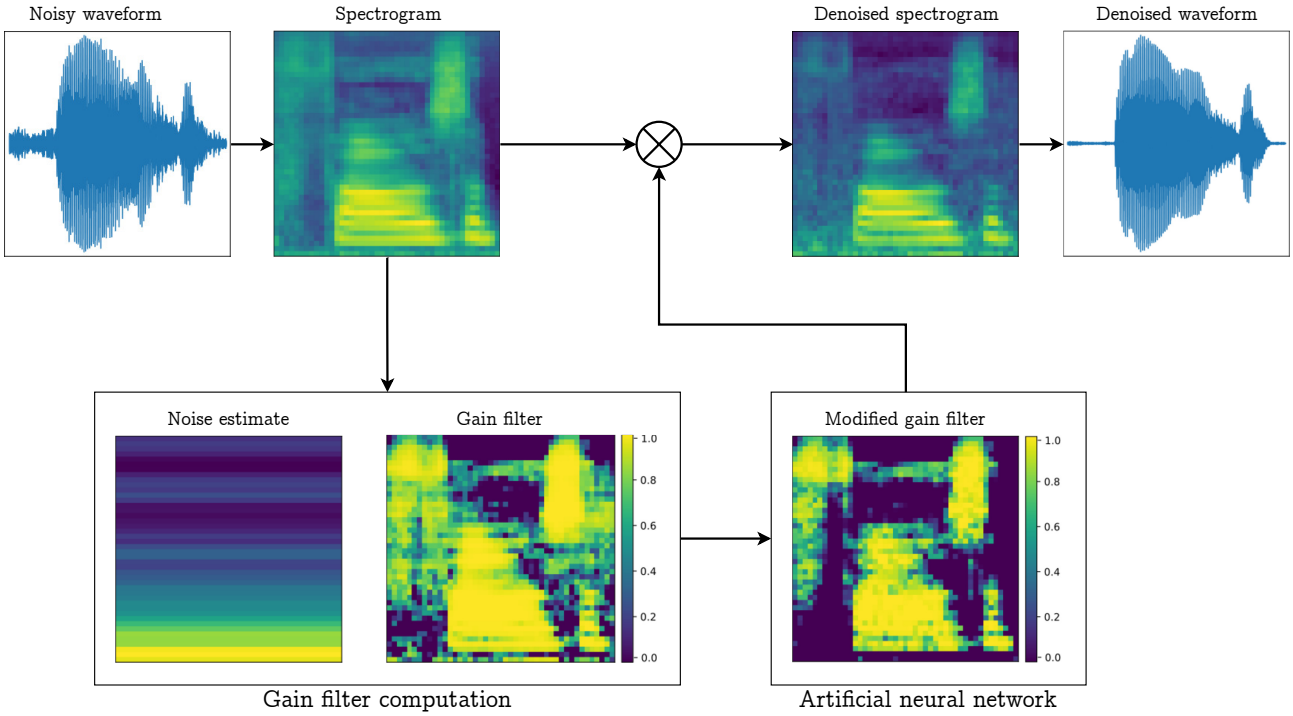


Figure 2: Schematic representation of the proposed noise suppressor.

3.1. Dataset

We use a publicly available dataset [35] in order to train, validate and test the proposed ANNs. The dataset is presented in detail by the authors of [36] and its specifications are presented in Tables 1 – 2. The clean audio clips consist of sentences recorded by different speakers in a noise-free environment, while their noisy counterparts can be expressed by the additive noise signal model:

$$y[n] = x[n] + v[n], \tag{1}$$

where n is the current time index, $x[n]$ is the clean speech, $v[n]$ is the background noise, and $y[n]$ is the resulting noisy speech signal.

Number of noisy/clean audio pairs	23'075
Sampling rate (kHz)	48
Number of speakers	56
Noise types	Babble noise Cafeteria Car Kitchen Office meeting Metro Restaurant Speech-shaped noise (SSN) Station Traffic
Signal to noise ratios (dB)	15 10 5 0

Table 1: Training dataset specifications.

Number of noisy/clean audio pairs	824
Sampling rate (kHz)	48
Number of speakers	7
Noise types	Bus Cafeteria Living room Public square
Signal to noise ratios (dB)	7.5 2.5 1.75 1.25

Table 2: Testing dataset specifications.

3.2. Pre-processing

The goal of the pre-processing is to build a representative feature space for the ANNs. For each noisy and clean audio clip, we compute the stationary noise suppressor gain filter and the target gain filter. These gain filters are then post-processed in order to reduce their dynamic range and achieve better data sparsity.

3.2.1. Signal preparation

All the audio clips are cut to a length of two seconds. Since the sampling frequency of the dataset is $F_s = 48kHz$, the target length of each clip is $N = F_s \cdot 2 = 96000$ samples. In case a clip is shorter than N, it is zero-padded at the end.
To compute the short-time discrete Fourier transform (STFT), each signal is split into overlapping segments, with length $W = 1024$ and an overlap of $M = 512$ samples. Each segment is then multiplied with a sine window, defined as

$$w(n) = \sin\left[\frac{n + 0.5}{W} \cdot \pi\right], \tag{2}$$

where $0 \leq n < W$ is the range of definition for the window. A discrete Fourier transform (DFT) is applied to the windowed segment. We then compute the magnitude spectrogram. Under the assumption of the signal model defined in (1), the STFT of $y[n]$ can be written as

$$Y[k, m] = X[k, m] + V[k, m], \tag{3}$$

where $1 \leq k \leq 1024$ is the index for the current frame of the windowed signal, and $1 \leq m \leq 1024$ is the frequency index of the component of the STFT.
After obtaining the STFT, for each frame $Y[k, m]$ we compute the magnitude power $|Y[k, m]|^2$, resulting in the power spectrogram of the signal. Under the assumption of $x[n]$ and $y[n]$ being uncorrelated and zero-mean, starting from (3) we can write the magnitude power spectrogram as

$$|Y[k, m]|^2 \approx |X[k, m]|^2 + |V[k, m]|^2. \tag{4}$$

This means that the power spectrogram of the clean speech signal can be obtained as

$$|X[k, m]|^2 = |Y[k, m]|^2 - |V[k, m]|^2. \tag{5}$$

Finally, the power spectrogram $|Y[k, m]|^2$ is mapped to a 44-band mel-spectrogram [14] denoted as $P_Y[j, m]$, where $1 \leq j \leq 44$ is the mel-band index. Triangular frequency bands are used. In the mel-spectrogram domain, (5) therefore becomes

$$P_X[j, m] = P_Y[j, m] - P_V[j, m]. \tag{6}$$

3.2.2. Gain filters

The next step is the computation of the gain filters. (5) can be equivalently expressed in terms of a per-band gain filter, defined as

$$G[j, m] = \frac{P_X[j, m]}{P_X[j, m] + P_V[j, m]}. \tag{7}$$

Since our dataset is composed of noisy/clean pairs of speech signals, and $P_Y[j, m] = P_X[j, m] + P_V[j, m]$, we can rewrite (7) as simply

$$G[j, m] = \frac{P_X[j, m]}{P_Y[j, m]}. \tag{8}$$

Following this, an approximation of the clean mel-spectrogram can be obtained as

$$P_X[j, m] = G[j, m] \cdot P_Y[j, m]. \tag{9}$$

For each noisy/clean pair in the dataset, two sets of gain filters are computed: the stationary noise suppressor gains and the target gains. The dynamic range of these two is then reduced, resulting in the noise suppressor data and the target data, that are shown in Figure 5. Our neural network models are then trained to find a mapping between the noise suppressor data and the target data.

Stationary noise suppressor gains Based on the mel-spectrogram $P_Y[j, m]$ of the noisy speech signal, we start by computing an estimate of the stationary background noise mel-spectrogram $P_{V_s}[j, m]$, based on the following assumptions:

- in the noisy signal, there are frames when the speech is absent and only the noise is present;
- the spectral content of the noise does not change substantially over time.

The procedure for obtaining the noise estimate is the following. We first start with a over-estimation of the stationary noise, in order to obtain an initial noise estimate. Then, for each $k - th$ sub-band, we compute the minimum between the current stationary noise estimate and the mean of the temporal analysis frame of each sub-band, which is comprised of $N = 6$ time-frames. The longer is the audio segment used, the higher is the accuracy of the stationary noise estimate. The advantage of this method is that we do not need a voice activity detector in order to separate the speech from the background noise; the disadvantage is that if we encounter several analysis frames with very low noise floor, the noise could be under-estimated, resulting in a less effective noise removal.
The last step consists in obtaining the gain filters based on the noise estimation. Since we have just computed the estimate $P_{V_s}[j, m]$ for the mel-spectrogram of the background noise, based on (7) we can write

$$G_{ns}[j, m] = \frac{P_Y[j, m] - P_{V_s}[j, m]}{P_Y[j, m]}, \tag{10}$$

where G_{ns} denote the noise suppressor gains. In order to better control the strength of the noise removal, we can introduce a parameter β into (10), which then becomes

$$G_{ns}[j, m] = \max\left\{\frac{P_Y[j, m] - \beta \cdot P_{V_s}[j, m]}{P_Y[j, m] + \epsilon}, 0\right\}, \tag{11}$$

where $0 \leq \beta \leq 1$ and $\epsilon = 1e - 20$ is a very small value used for ensuring numerical stability in the eventuality that $P_Y[j, m] = 0$. The parameter β can be fine-tuned to find an optimal balance between stronger noise removal and more natural sounding audio, since gain-filtering can potentially introduce unwanted artifacts.

Target gains As a first step, we compute the ideal gains as per (8), i.e. as the ratio between the clean mel-spectrogram and the noisy mel-spectrogram. We compute the mel-spectrogram of the noise as

$$P_V[j,m] = P_Y[j,m] - P_X[j,m]. \tag{12}$$

Then, the ideal gains G_{id} are obtained as

$$G_{id}[j,m] = \frac{P_Y[j,m] - \beta \cdot P_V[j,m]}{P_Y[j,m] + \epsilon}, \tag{13}$$

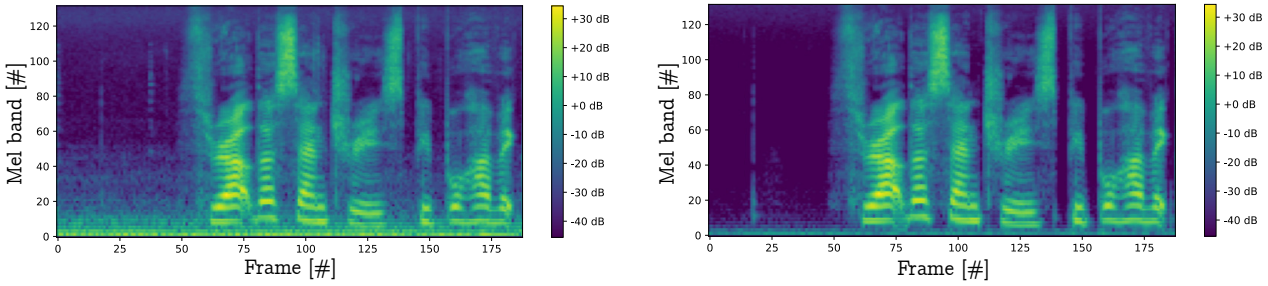
where $0 \leq \beta \leq 1$. To achieve better data sparsity, the ideal gains are post-processed by computing the element-wise minimum between the ideal gains G_{id} and the non-stationary noise removal gains G_{ns}, resulting in the target gains G_{tg}. This operation makes the target gains more sparse, especially at higher frequencies, as it can be seen from Figures 5b – 5c.

3.2.3. From gain filters to neural network data

A last step is performed before using the previously computed gains for the training, validation, and testing of the neural networks. In order to make the domain more sparse, we reduce the the dynamic range of the gain filters. The dynamic range reduction is regulated by an attenuation limit $L_{dB} \leq 0$. Given a gain filter $G[j,m]$, each j^{th} mel-frequency component with a value lower than $L_{lin} = 10^{\frac{L_{dB}}{20}}$ is clamped to L_{lin}, resulting in a range $L_{lin} \in [L_{lin}, 1]$. This range is then rescaled to $[0,1]$ with the following equation:

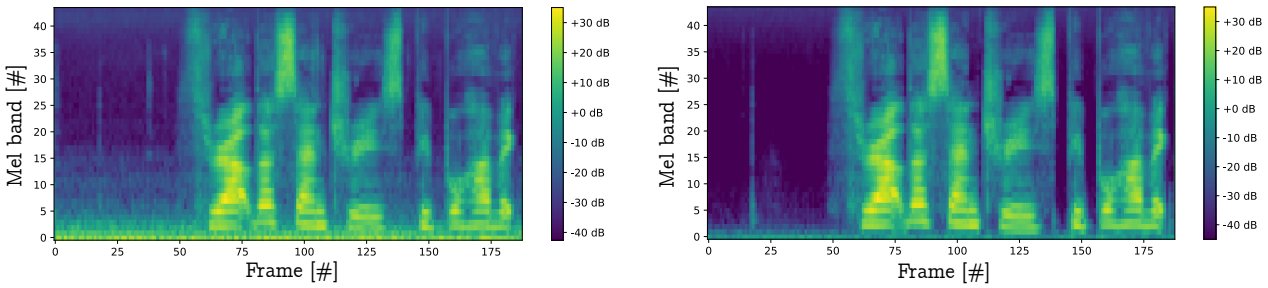
$$D[j,m] = \frac{G[j,m] - L_{lin}}{1 - L_{lin}}, \tag{14}$$

where $D[j,m]$ is the resulting data, ready to be used with the artificial neural networks.



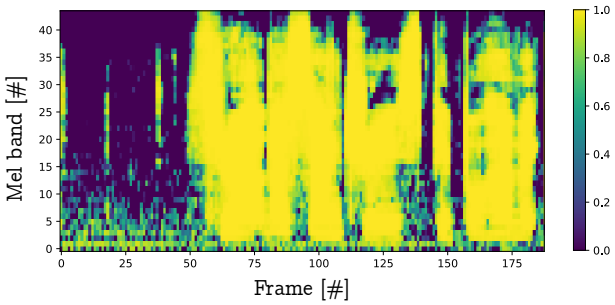
(a) Noisy power spectrogram. (b) Clean power spectrogram.

Figure 3: High resolution power spectrograms for 2 seconds of speech (132 MEL bands).

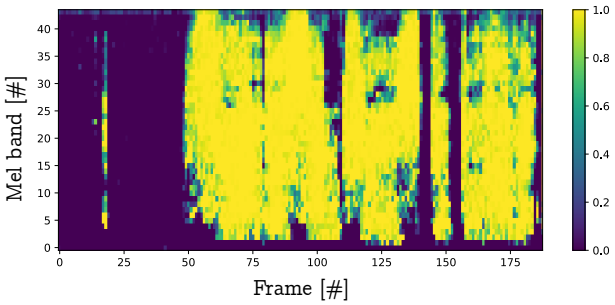


(a) Noisy power spectrogram. (b) Clean power spectrogram.

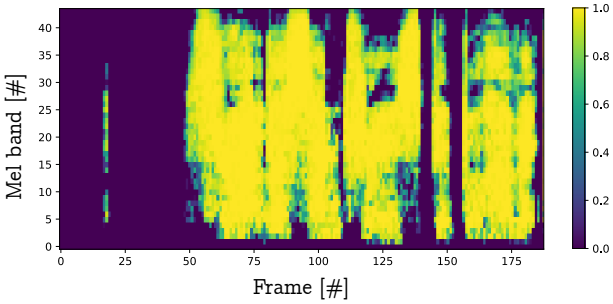
Figure 4: Low resolution power spectrograms for 2 seconds of speech (44 MEL bands).



(a) Stationary noise suppressor data.



(b) Ideal data.



(c) Target data.

Figure 5: Post-processed gain filter data that is used for training the ANNs: 5a is the gain filter outputted by the stationary noise suppressor, 5b is the ideal gain filter, and 5c is the target gain filter.

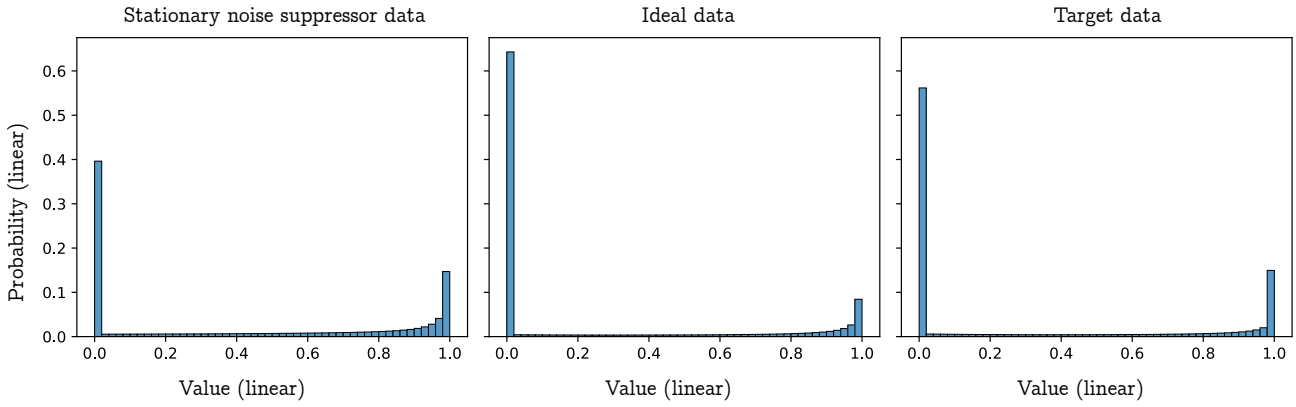


Figure 6: Linear probability distributions for neural network data.

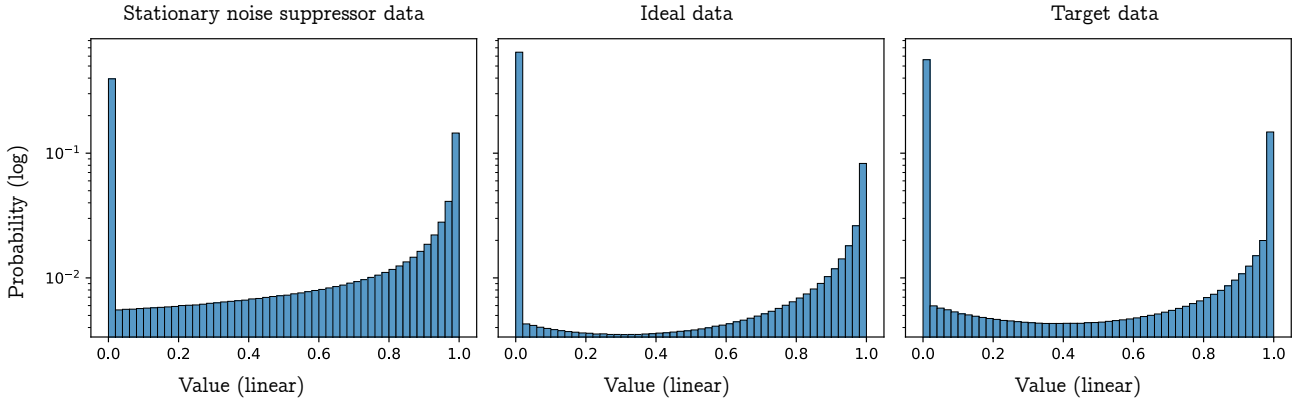


Figure 7: Log-probability distributions for neural network data.

3.3. Proposed networks

We test two main families of artificial neural networks: denoising convolutional autoencoders (DCAEs) and recurrent neural networks (RNNs). Autoencoders (AEs) are specific architectures that introduce a bottleneck, which forces the neural network to create a compressed representation of the input [37], also called as latent vector. This way, the neural network discards most non-essential information when learning a mapping between its input and the target. Originally proven useful for unsupervised learning, autoencoders can be also employed for the task of noise removal [38]. Convolutional autoencoders (CAEs) build upon traditional autoencoders by substituting several or all fully connected layers with convolutional layers. The usage of convolutional layers makes it possible to efficiently identify patterns that can be found in a small window across their input. For this reason, denoising convolutional autoencoders (DCAEs) can be used to discriminate those portions of the input that correspond to noise, and to isolate them in their latent representation [39].
Recurrent neural networks (RNNs) have the ability to model the evolution of data over time thanks to having an internal memory, which is composed of hidden units [40]. The performance of the basic RNN, however, degrades quickly when dealing with long-term dependencies because of vanishing and exploding gradient problems. This has led to the development of long short-term memory recurrent networks (LSTM) [41]; by introducing a forget gate, an LSTM is able regulate what information should be kept or discarded in modeling a sequence. This mechanism also allows for more efficient and predictable training results. Gated recurrent units (GRUs) follow the idea of LSTM, but are able to achieve a similar performance with a smaller amount of parameters [42]. For this reason, we choose to test GRUs in our proposed system.
The following paragraphs introduce in detail the ANNs that are trained to enhance the gain filters computed by our stationary noise suppressor. All four networks are implemented with the PyTorch deep learning framework[1]. We also provide a benchmark of their computational and memory requirements, which have been estimated with a PyTorch profiler[2].

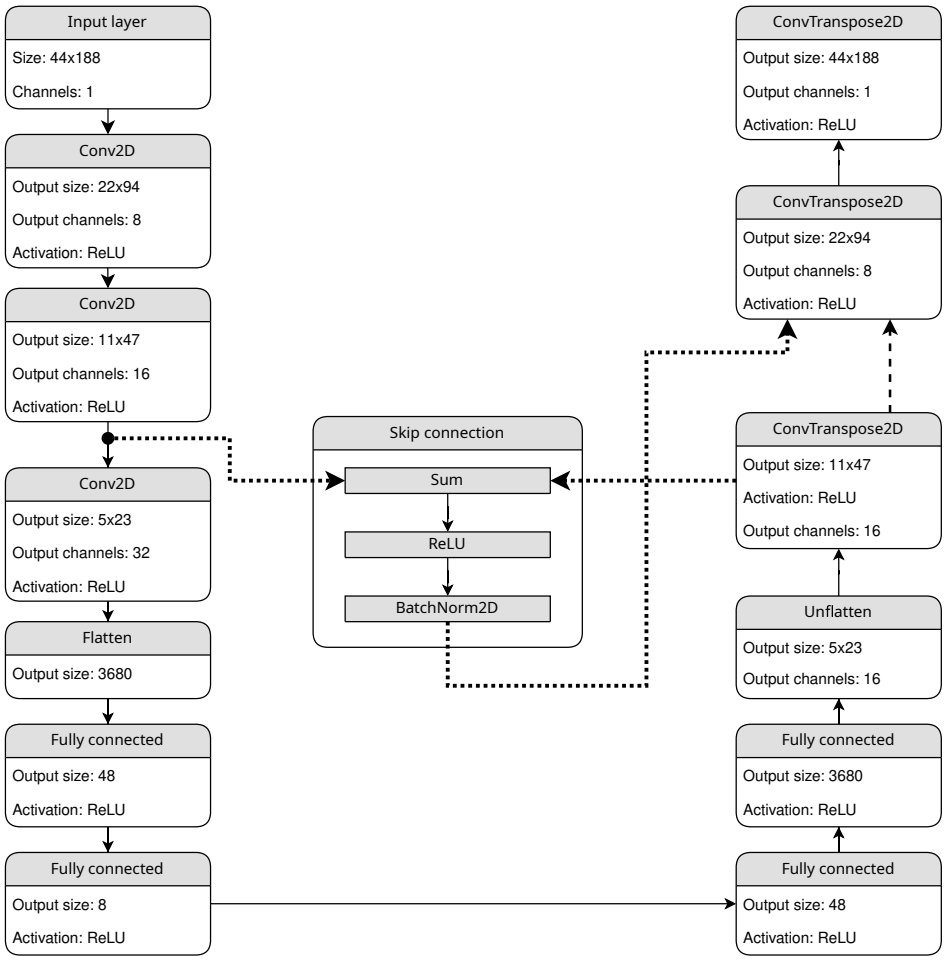


Figure 8: Schematic architecture of ConvEncoderDecoder (solid and dashed lines) and ConvEncoderDecoderSkip (solid and dotted lines).

[1]See `https://github.com/pytorch/pytorch`
[2]See `https://github.com/Lyken17/pytorch-OpCounter`

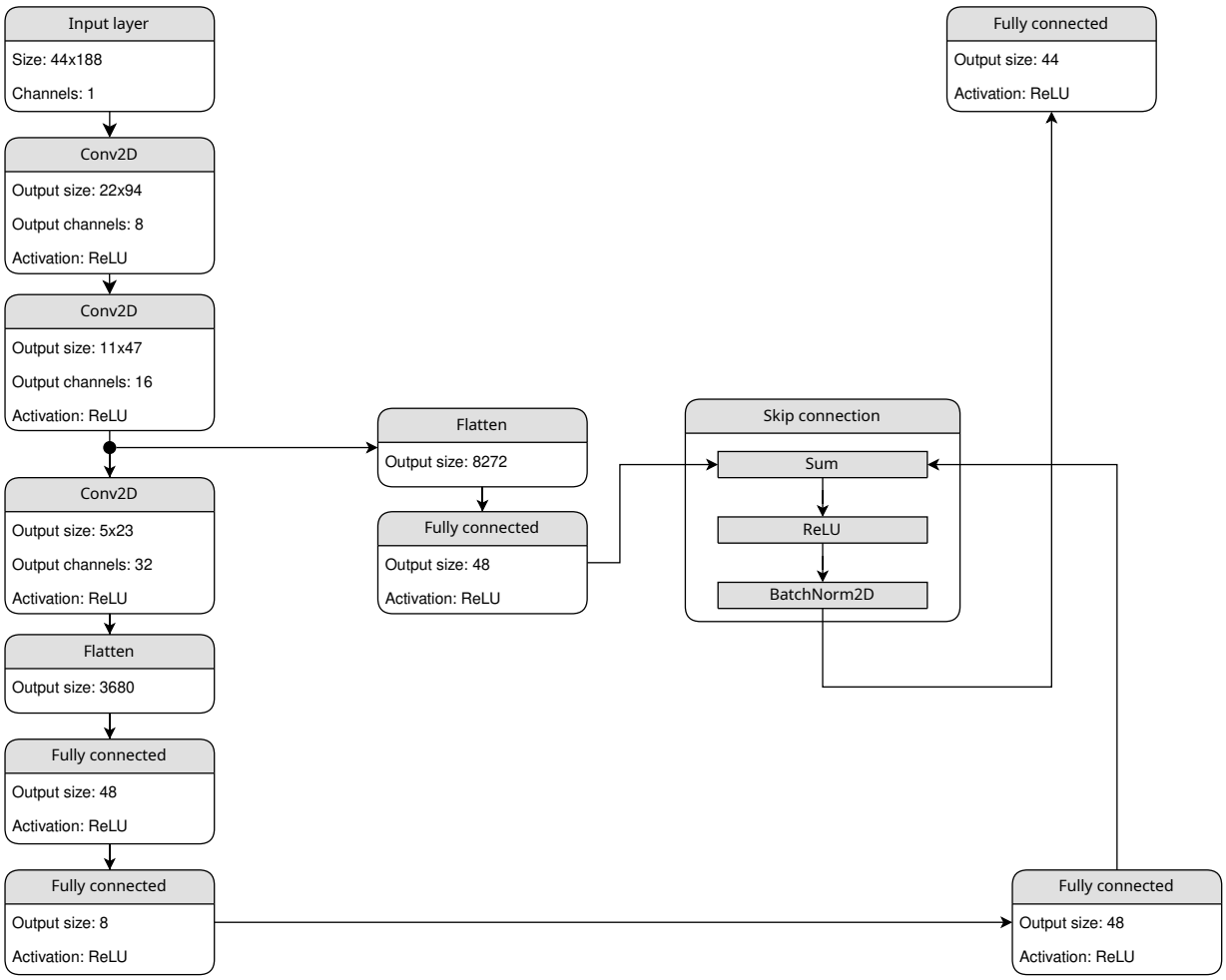


Figure 9: Schematic architecture of ConvEncoderDecoderSkipRT.

400539

ANN model	N. of parameters	MACs/inference
ConvEncoderDecoder	369'577	6'988'704
ConvEncoderDecoderSkip	400'539	6'785'760
ConvEncoderDecoderSkipRT	68'644	1'337'056
GRUNet	59'400	60'940

Table 3: Memory requirements and computational complexity comparison for the implemented neural networks.

ConvEncoderDecoder This neural network is designed as a fully symmetrical encoder-decoder. The encoder is comprised of three 2D convolutional layers with a decreasing output size and an increasing number of channels, followed by two fully connected layers of decreasing output size. Conversely, the decoder starts with two fully connected layers with increasing output size, followed by three 2D convolutional layers. The output size of the last convolutional layer of the decoder is the exact same size as the input layer of the encoder. The architecture is shown in Figure 8 (solid and dashed lines).

ConvEncoderDecoderSkip The architecture of this network is similar to ConvEncoderDecoder, but it introduces a skip connection between the second convolutional layer of the encoder and the second convolutional layer of the decoder, as shown in Figure 8 (solid and dotted lines). Skip connections are a subset of techniques know as deep residual learning, and have been proven useful in the case of vanishing gradients and degradation of high-level features [43]. The architecture of ConvEncoderDecoderSkip is loosely based on U-Net [31], a fully convolutional encoder-decoder with skip connections, which has been originally developed for image segmentation. U-Net based neural networks have been recently applied to the removal of noise in images [24] [44], audio

separation [32], and audio denoising [33]. The skip connection sums the output of the aforementioned layers, applies a rectified linear unit function (ReLU), and a 2D batch normalization.

ConvEncoderDecoderSkipRT This neural network is an asymmetrical autoencoder and is shown in Figure 9. Similarly to ConvEncoderDecoderSkip, the encoder is composed of three 2D convolutional layers, two fully connected layers, and a skip connection that taps in between the second and third convolutional layers. The decoder, however, is designed to output only the gain filter corresponding to the latest temporal frame. There are two main advantages in this design: first, it is possible to reduce the number of parameters and the computational complexity substantially; second, outputting only the current gain filter means that this model is more suitable for real-time usage when compared to ConvEncoderDecoder and ConvEncoderDecoderSkip, because we eliminate the redundancy of predicting an entire set of filters that are applicable only to past temporal frames.

GRUNet We implement a lightweight recurrent network based on five stacked gated recurrent unit (GRU) layers. Each GRU layer computes the function defined in (15) – (18):

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \tag{15}$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \tag{16}$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{(t-1)} + b_{hn})) \tag{17}$$

$$h_t = (1 - z_t) \odot n_t + z_t \odot h_{(t-1)}, \tag{18}$$

where $\odot$ is the Hadamard (element-wise) product, σ and tanh are the sigmoid activation and hyperbolic tangent activation functions respectively, r_t, z_t and n_t are the reset, update and new gates respectively, h_t is the hidden state of the layer at the time instant t, x_t is the input at the time instant t, W are the weight matrices, and b are the bias values.
Compared to 2D convolutional layers, GRUs focus on capturing the evolution of gain filters across time, instead of finding time-frequency independent noise patterns in the 2D image that span over a significant number of time frames. This makes it possible to reduce the input dimensionality and the memory requirements of our system, because the input layer of GRUNet is supplied with gain filters that correspond to a single time frame only. Additionally, the lower dimensionality of the GRU layers that compose GRUNet imply a significant reduction in computational complexity, as shown in Table 3.

3.4. Network training

In order to train our networks, we first perform a 90%/10% split of the training dataset into training/validation portions. During each training epoch, we compute the loss function between the network predictions, both on training dataset split and the validation split. This is performed in batches of 256 samples each, which are randomized at the beginning of every epoch. The loss computed for the training split is then displayed and used for backpropagation, while the validation loss split is only displayed and has no influence on the training procedure. A similar training a validation loss indicates that our models are not overfitting on the training data and have good generalization capabilities. The loss function is the mean squared error (MSE), which is computed in an element-wise fashion between the target data $D[j, m]$ and the predicted data $\hat{D}[j, m]$ with the following equation:

$$\text{MSE} = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{M}\sum_{m=1}^{M}\Big(D[j, m] - \hat{D}[j, m]\Big)^2, \tag{19}$$

where j is the band index, J is the number of bands, m is the time-frame index, and M is the number of time-frames. For each ANN model (ConvEncoderDecoder, ConvEncoderDecoderSkip, ConvEncoderDecoderSkip, and GRUNet), we choose an optimal learning rate and number of training epochs, which are shown in Table 4.

ANN model	Loss function	N. of epochs	Learning rate
ConvEncoderDecoder	MSE	20	0.0001
ConvEncoderDecoderSkip	MSE	20	0.0001
ConvEncoderDecoderSkipRT	MSE	25	0.0001
GRUNet	MSE	30	0.001

Table 4: Training parameters for the implemented ANNs: loss function, number of training epochs, and learning rate.

4. Results

4.1. Evaluation setup

To evaluate the performance of our setup, we use the testing set provided in [35], whose characteristics are reported in Table 2. The speakers, the types of noise and the SNR levels are different than those contained in the training set, which is particularly suitable for evaluating the generalization properties of the trained ANNs and the robustness of the pre-processing procedure. We use two objective metrics that perform a comparison between the clean and the different denoised version of each audio file. These metrics are computed across the entire testing set, and we analyze their distributions in order to evaluate the outputs of our stationary noise suppressor, target gain filters, and the proposed ANN models. Our comparison also includes RNNoise[3], which is presented in [20].

4.2. Metrics for evaluation

We use two metrics to compare the clean audio signal with the signals denoised by the proposed methods: perceptual evaluation of speech quality (PESQ) and short-time objective intellegibility (STOI).

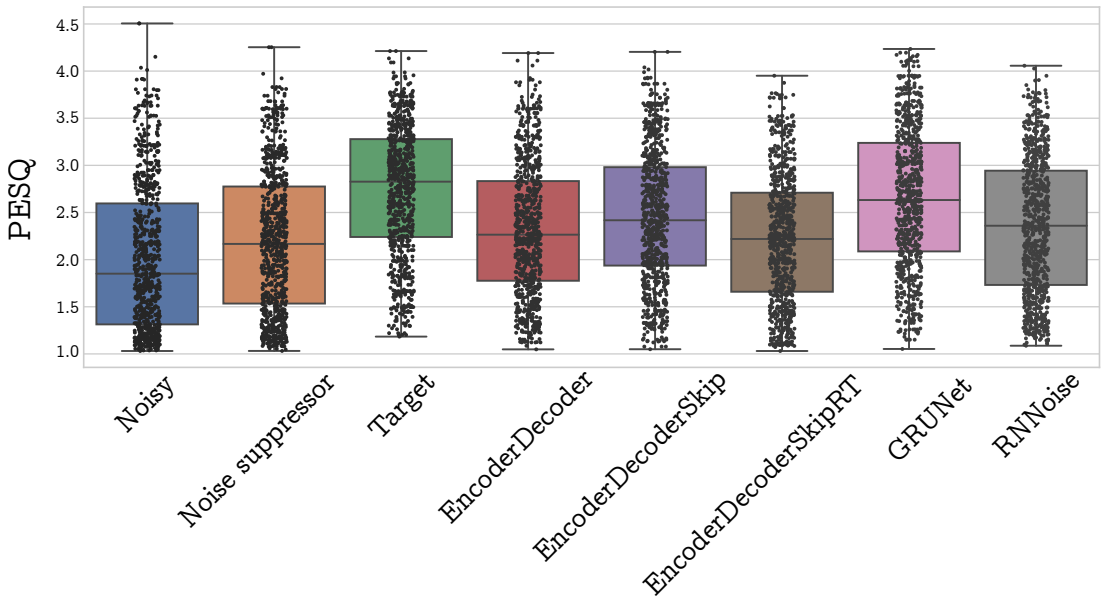
PESQ The perceptual evaluation of speech quality is a methodology standardized by the International Telecommunication Union (ITU) under recommendation ITU-T P.862[4] with the goal of testing the quality of speech signals. It has been developed by taking into account the physiology of human hearing, and it outputs values that follow the mean opinion score (MOS) scale, ranging from 1 (bad) to 5 (excellent).

STOI The short-time objective intelligibility has been first presented in [45]. This measure has been developed to evaluate the degradation of waveforms that have been processed with time-frequency filtering, and it presents a high correlation with the subjective intelligibility of speech.

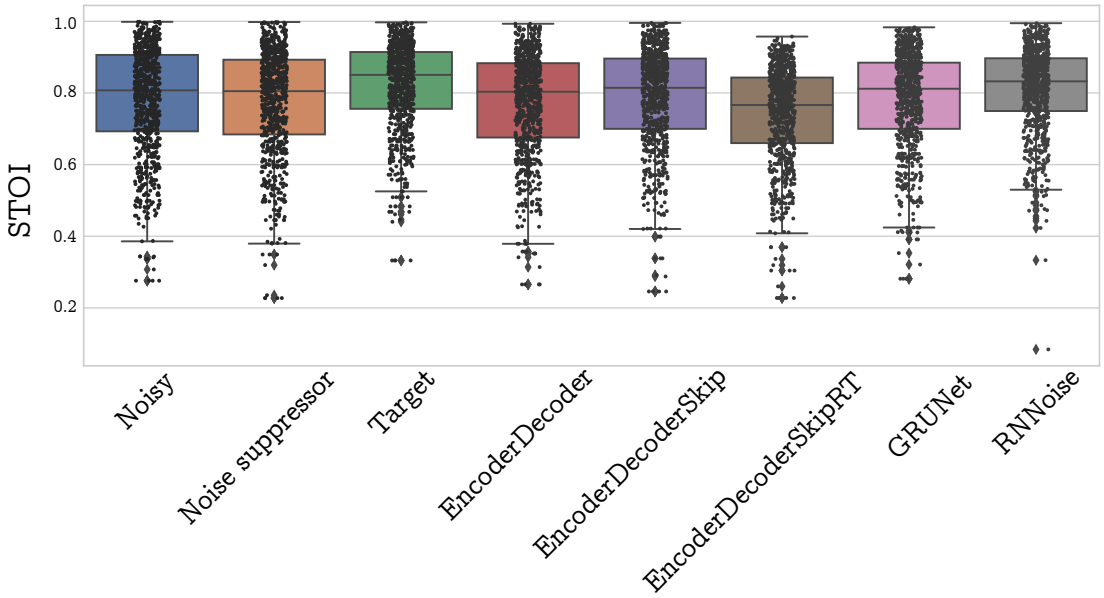
4.3. Experimental results

The results of our evaluation procedure are shown in Figure 10 and Table 5. Figure 10 shows the distribution of the objective metrics as box-and-whisker plots, which are a graphical representation based on the interquartile range (IQR)[46]. The high distribution spread is a consequence by the fact that the testing set presents different SNR values (7.5dB, 2.5dB, 1.75dB, and 1.25dB), and that each noise type has a different effect on the perceived degradation of the speech signals. In terms of PESQ, we can observe from Figure 10a that our stationary noise suppressor improves the overall quality of the testing set, and that our target gain filters provide another significant improvement step, which validates our problem formulation. Compared to the other ANNs, we can clearly see that GRUNet is able to outperform all three convolutional autoencoders (ConvEncoderDecoder, ConvEncoderDecoderSkip, and ConvEncoderDecoderSkipRT) and RNNoise, achieving a mean PESQ score of 2.657, which is significantly closer to that of the target (2.734) rather than the stationary noise suppressor (2.193). STOI, on the other hand, shows no improvement in the signals outputted the stationary noise suppressor, but still assigns the highest score to the signals that are denoised with the proposed target gain filters, which is followed by RNNoise and GRUNet.
Since GRUNet and RNNoise achieve the best score in terms of PESQ and STOI respectively, we compare their computational complexity in Table 6, which shows that GRUNet has lower requirements in terms of model weights and a significantly lower computational cost. Compared to RNNoise, the number of parameters of GRUNet is 32% lower, and the decrease of multiply-accumulate operations for one prediction amounts to 187%. In order to evaluate the suitability of GRUNet for real-time usage, we additionally implement a command-line tool (CLI) in C based on the open neural network exchange (ONNX)[5]. The purpose of ONNX is to facilitate the interoperability of different machine learning frameworks and to provide a cross-platform API for training and running ONNX models. Since PyTorch supports exporting ANN models in ONNX format, we can take advantage of the C API that comes with the Microsoft ONNX runtime[6]. Our CLI takes two positional arguments: the path of the ONNX model and the number of inferences to run. The time for performing the inferences is then measured, based on which we compute the average time for a single inference.

[3] Compiled from `https://gitlab.xiph.org/xiph/rnnoise/-/tree/7f449bf8bd3b933891d12c30112268c4090e4d59`
[4] See `https://itu.int/rec/T-REC-P.862`
[5] See `https://github.com/onnx/onnx`
[6] See `https://github.com/microsoft/onnxruntime`



(a) PESQ results.



(b) STOI results.

Figure 10: Evaluation results on the testing set across all noise types and SNR values.

	PESQ		**STOI**	
	Mean	Median	Mean	Median
Noisy audio	2.018	1.851	0.779	0.807
Stationary noise suppressor	2.193	2.167	0.775	0.805
Target gain filter	2.734	2.827	0.821	0.850
ConvEncoderDecoder	2.325	2.265	0.771	0.803
ConvEncoderDecoderSkip	2.452	2.418	0.787	0.814
ConvEncoderDecoderSkipRT	2.221	2.219	0.739	0.766
GRUNet	**2.657**	**2.632**	0.791	0.815
RNNoise	2.357	2.359	**0.810**	**0.833**

Table 5: Mean and median values of PESQ and STOI on the testing set. Bold values highlight the highest performance of the tested ANNs.

ANN model	N. of parameters	MACs/inference
GRUNet	**59'400**	**60'940**
RNNoise	87'503	175'000

Table 6: Memory requirements and computational complexity comparison between the proposed ANN (GRUNet) and the ANN presented in [20] (RNNoise). Bold values highlight the lowest requirements.

CPU model	Inference time (ms)	Real-time performance (×)
Apple Silicon M1	0.0206	≈ 516×
Intel i7-10875H	0.0313	≈ 340×
Intel i3-1000NG4	0.0429	≈ 248×
Raspberry Pi 3B+	0.4205	≈ 25×

Table 7: Single core performance of GRUNet with ONNX Runtime.

5. Conclusion

This manuscript presents a low-complexity system for real-time noise reduction of speech signals that combines a stationary noise suppressor with an artificial neural network (ANN). The task of the stationary noise suppressor is to compute gain filters based on a statistical noise estimate, while the ANN enhances the aforementioned filters in order to achieve a higher level of noise reduction. The enhanced filters can also remove residual artifacts that are potentially caused by gain filtering.

We implement several ANN architectures in order to assess the performance of image-based convolutional autoencoders, residual learning strategies, and recurrent neural networks (RNNs). Our results show that a multi-layer recurrent network based on gated recurrent units (GRUs) is able to outperform convolutional autoencoders with a small fraction of their computational complexity.

Additionally, we compare our proposed approach against RNNoise [20], a denoising system that has been designed with similar goals in mind, i.e. low complexity, causality, and real-time processing. The comparison is performed with two objective metrics, the perceptual evaluation of speech quality (PESQ) and the short-time objective intelligibility (STOI), which assess the degradation of a signal relatively to the original signal. Compared to RNNoise, our approach achieves a higher PESQ score and a slightly lower STOI score with a fraction of the computational cost and significantly lower memory requirements.

Future research will mainly focus on two goals. First, we will explore the possibility of enhancing the extracted features, with the aim of computing target filters with higher perceptual quality and stronger noise suppression. Second, we will endeavor in further reducing the overall complexity of the proposed system.

References

[1] Sibiri Tiemounou, Régine le Bouquin Jeannès, and Vincent Barriac Ewert. Perception-based automatic classification of background noise in super-wideband telephony. *Journal of the Audio Engineering Society*, 62(11):776–781, November 2014. doi: 10.17743/jaes.2014.0040.

[2] Zdenek Becvar, Lukas Novak, Jan Zelenka, Miloslav Brada, and Pavel Slepicka. Impact of additional noise on subjective and objective quality assessement in voip. In *2007 IEEE 9th Workshop on Multimedia Signal Processing*, pages 39–42, 2007. doi: 10.1109/MMSP.2007.4412813.

[3] Peter Noll. Wideband speech and audio coding. *IEEE Communications Magazine*, 31(11):34–44, 1993. doi: 10.1109/35.256878.

[4] Milan Jelinek and Redwan Salami. Noise reduction method for wideband speech coding. In *2004 12th European Signal Processing Conference*, pages 1959–1962, 2004.

[5] Malay Gupta, Chris Forrester, and Sean Simmons. Review of wideband speech noise reduction techniques. *Canadian Acoustics*, 37(3):84–85, September 2009. URL `https://jcaa.caa-aca.ca/index.php/jcaa/article/view/2145`.

[6] Mohamed Krini and Gerhard Schmidt. *Speech and Audio Processing in Adverse Environments*, chapter 4. Model-Based Speech Enhancement, pages 89–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-70602-1_4.

[7] Gaston Hilkhuysen, Nikolay Gaubitch, Mike Brookes, and Mark Huckvale. Effects of noise suppression on intelligibility: Dependency on signal-to-noise ratios. *The Journal of the Acoustical Society of America*, 131 (1):531–539, 2012. doi: 10.1121/1.3665996.

[8] L. Rabiner and M. Sambur. Voiced-unvoiced-silence detection using the itakura lpc distance measure. In *ICASSP '77. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 323–326, 1977. doi: 10.1109/ICASSP.1977.1170330.

[9] Y. Ephraim and D. Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6):1109–1121, 1984. doi: 10.1109/TASSP.1984.1164453.

[10] Rainer Martin. Noise power spectral density estimation based on optimal smoothing and minimum statistics. *IEEE Transactions on Speech and Audio Processing*, 9(5):504–512, 2001. doi: 10.1109/89.928915.

[11] 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. doi: 10.1109/TASSP.1979.1163209.

[12] Walter Etter and George S. Moschytz. Noise reduction by noise-adaptive spectral magnitude expansion. *Journal of the Audio Engineering Society*, 42(5):341–349, May 1994.

[13] Alexis Favrot and Christof Faller. Perceptually motivated gain filter smoothing for noise suppression. *Journal of the Audio Engineering Society*, October 2007.

[14] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4): 357–366, 1980. doi: 10.1109/TASSP.1980.1163420.

[15] E. Zwicker and E. Terhardt. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of the Acoustical Society of America*, 68(5):1523–1525, 1980. doi: 10.1121/1.385079.

[16] E. Zwicker. Subdivision of the audible frequency range into critical bands. *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961. doi: 10.1121/1.1908630.

[17] D. Wang and Jae Lim. The unimportance of phase in speech enhancement. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(4):679–681, 1982. doi: 10.1109/TASSP.1982.1163920.

[18] Alexandre Défossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. In *Proc. Interspeech 2020*, pages 3291–3295, 2020. doi: 10.21437/Interspeech.2020-2409.

[19] Jayakumar Ramanathan and Pankaj Topiwala. Time–frequency localization and the spectrogram. *Applied and Computational Harmonic Analysis*, 1(2):209–215, 1994. ISSN 1063-5203. doi: https://doi.org/10.1006/acha.1994.1008. URL `https://www.sciencedirect.com/science/article/pii/S1063520384710086`.

[20] Jean-Marc Valin. A hybrid dsp/deep learning approach to real-time full-band speech enhancement. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5, 2018. doi: 10.1109/MMSP.2018.8547084.

[21] Cheng Yu, Ryandhimas E. Zezario, Syu-Siang Wang, Jonathan Sherman, Yi-Yen Hsieh, Xugang Lu, Hsin-Min Wang, and Yu Tsao. Speech enhancement based on denoising autoencoder with multi-branched encoders. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2756–2769, 2020. doi: 10.1109/TASLP.2020.3025638.

[22] Se Rim Park and Jin Won Lee. A Fully Convolutional Neural Network for Speech Enhancement. In *Proc. Interspeech 2017*, pages 1993–1997, 2017. doi: 10.21437/Interspeech.2017-1465.

[23] Prashanth Gurunath Shivakumar and Panayiotis Georgiou. Perception optimized deep denoising autoencoders for speech enhancement. In *Proc. Interspeech 2016*, pages 3743–3747, 2016. doi: 10.21437/Interspeech.2016-1284.

[24] Sheyda Ghanbaralizadeh Bahncmiri, Mykola Ponomarenko, and Karen Egiazarian. Deep convolutional autoencoder for estimation of nonstationary noise in images. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 238–243, 2019. doi: 10.1109/EUVIP47703.2019.8946273.

[25] Kyung-Hyun Lim, Jin-Young Kim, and Sung-Bae Cho. Non-stationary noise cancellation using deep autoencoder based on adversarial learning. In Hujun Yin, David Camacho, Peter Tino, Antonio J. Tallón-Ballesteros, Ronaldo Menezes, and Richard Allmendinger, editors, *Intelligent Data Engineering and Automated Learning*, pages 367–374. Springer International Publishing, 2019. ISBN 978-3-030-33607-3.

[26] Andrew Maas, Quoc V. Le, Tyler M. O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y. Ng. Recurrent neural networks for noise reduction in robust asr. In *INTERSPEECH*, 2012.

[27] Cassia Valentini Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks. In *Proceedings of Interspeech 2016*, Interspeech, pages 352–356. International Speech Communication Association, September 2016. doi: 10.21437/Interspeech.2016-159.

[28] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[30] Wei Zhang, Akira Hasegawa, Kazuyoshi Itoh, and Yoshiki Ichioka. Image processing of human corneal endothelium based on a learning network. *Appl. Opt.*, 30(29):4211–4217, October 1991. doi: 10.1364/AO.30.004211.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[32] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 334–340, 2018. ISBN 978-2-9540351-2-3.

[33] Ritwik Giri, Umut Isik, and Arvindh Krishnaswamy. Attention wave-u-net for speech enhancement. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 249–253, 2019. doi: 10.1109/WASPAA.2019.8937186.

[34] Seyedmahdad Mirsamadi and Ivan Tashev. Causal speech enhancement combining data-driven learning and suppression rule estimation. In *Proc. Interspeech 2016*, pages 2870–2874, 2016. doi: 10.21437/Interspeech.2016-437.

[35] Cassia Valentini Botinhao. Noisy speech database for training speech enhancement algorithms and tts models, 2017. doi: 10.7488/ds/2117.

[36] Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. Investigating rnn-based speech enhancement methods for noise-robust text-to-speech. In *9th ISCA Speech Synthesis Workshop*, pages 146–152, 2016. doi: 10.21437/SSW.2016-24.

[37] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991. doi: 10.1002/aic.690370209.

[38] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, chapter 14. Autoencoders, pages 507–512. MIT Press, 2016. doi: 10.1007/s10710-017-9314-z.

[39] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 241–246, 2016. doi: 10.1109/ICDMW.2016.0041.

[40] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. doi: 10.1038/323533a0.

[41] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. doi: 10.1162/neco.1997.9.8.1735.

[42] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[44] Javier Gurrola-Ramos, Oscar Dalmau, and Teresa E. Alarcón. A residual dense u-net neural network for image denoising. *IEEE Access*, 9:31742–31754, 2021. doi: 10.1109/ACCESS.2021.3061062.

[45] Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4214–4217, 2010. doi: 10.1109/ICASSP.2010.5495701.

[46] F.M. Dekking, C. Kraaikamp, H.P. Lopuhaa, and L.E. Meester. *A Modern Introduction to Probability and Statistics*, chapter 16. Exploratory data analysis: numerical summaries, pages 234 – 238. Springer-Verlag London, 2005. doi: 10.1007/1-84628-168-7.

[47] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. doi: 10.48550/arXiv.1312.6114.

Abstract in lingua italiana

Nonostante i rapidi sviluppi delle reti neurali artificiali nel campo della rimozione del rumore in segnali audio, le criticità relative all'onere computazionale e al comportamento di fronte a tipologie di rumore diverse rimangono tutt'ora attuali. I sistemi di riduzione del rumore devono essere spesso utilizzati all'interno di dispositivi con scarse risorse computazionali, come telefoni cellulari, telecamere per il web, o microcontrollori, che non possiedono i requisiti richiesti dalla maggior parte dei modelli basati su reti neurali artificiali, e che necessitano un'elaborazione causale e in tempo reale del segnale in entrata. Inoltre, questi sistemi devono essere utilizzati in un'ampia varietà di situazioni, il che può essere un aspetto problematico nel caso di metodi basati puramente su reti neurali. Per ovviare a queste limitazioni, proponiamo un sistema di rimozione del rumore per segnali di parlato che combina la robustezza di un soppressore del rumore tradizione con le capacità di generalizzazione delle reti neurali artificiali. A differenza della maggior parte dei cosiddetti modelli "end-to-end", i quali impiegano una rete neurale per effettuare un restauro diretto dello spettrogramma o della forma d'onda rumorosi, l'ambito di applicazione della rete neurale nel nostro sistema è limitato al miglioramento di un filtro di guadagno, che regola l'attenuazione delle singole componenti dello spettrogramma rumoroso, con l'obiettivo di rimuovere i residui di rumore non stazionario, i quali sono notoriamente difficili da eliminare senza ricorrere a sofisticati algoritmi euristici. Ciò ci permette di ottenere un maggiore controllo sul processo di rimozione del rumore e di limitare gli artefatti che possono derivare da una manipolazione diretta dello spettro. La nostra valutazione mostra che il sistema proposto è capace di eseguire un'efficace rimozione del rumore anche nel caso di tipi di rumore non trattati durante l'allenamento, mantenendo allo stesso tempo una complessità computazionale molto più bassa della maggior parte delle tecniche di apprendimento profondo nello stato dell'arte.

Parole chiave: Restauro di registrazioni del parlato in tempo reale, cancellazione del rumore, apprendimento profondo, elaborazione digitale di segnali, reti neurali convoluzionali, reti neurali ricorrenti