Audio Denoising with Deep Network Priors

Michael Michelashvili[1], *Lior Wolf*[1,2]

[1]Tel Aviv University
[2]Facebook AI Research

mosheman5@gmail.com, wolf@cs.tau.ac.il

Abstract

We present a method for audio denoising that combines processing done in both the time domain and the time-frequency domain. Given a noisy audio clip, the method trains a deep neural network to fit this signal. Since the fitting is only partly successful and is able to better capture the underlying clean signal than the noise, the output of the network helps to disentangle the clean audio from the rest of the signal. The method is completely unsupervised and only trains on the specific audio clip that is being denoised. Our experiments demonstrate favorable performance in comparison to the literature methods, and our code and audio samples are available at `https://github.com/mosheman5/DNP`.

Index Terms: Audio denoising; Unsupervised learning

1. Introduction

Many unsupervised signal denosing methods work in a similar way. First, a spectral mask is estimated, which predicts for every frequency, whether it is relevant to the clean signal or mostly influenced by the noise. Then, one of a few classical methods, such as the Wiener filter [1] or MMSE-LSA [2] are used to clean the audio.

The denoising methods differ in the way in which the mask is first estimated. Each method is based on a different set of underlying assumptions on the properties of the signal, the noise, or both. For example, some algorithms assume that the change in the power spectrum of the noise is slower than the change in that of the clean signal and, therefore, in order to estimate the noise statistics, averaging of the power signal over multiple time points is performed.

In this work, we investigate the use of deep network priors for the task of unsupervised audio denoising. These priors are based on the assumption that the clean signal, in the time domain, is well-captured by a deep convolutional neural network. The method, therefore, trains a network to fit the input signal, and observes the part of the signal that has the largest amount of uncertainty, i.e., which was modeled most poorly. This part is then masked out and one of the classical speech-enhancement methods is applied.

Similar priors have been recently used in computer vision, in order to reconstruct noise free images [3]. However, we note that the cleaning of audio signals is much more involved. We observe three major differences between the usage of deep network priors in images vs. its use in audio:

1. In computer vision, the clean image emerges from the learned network simply as its output. While averaging over multiple training iterations does improve the accuracy to some degree, its contribution is minor. In contrast, when a similar method is applied to audio, the network produces an output that is unacceptable in quality.

2. In computer vision, if early stopping is not applied, the network fits the noisy input image. In audio, this fitting does not occur nearly as quickly (if at all) and instead of converging to a solution with a very small loss, the network displays relatively large fluctuations.

3. In vision, the networks train much faster on a clean image than on a mixed signal that contains both image and noise. In audio, there is a difference in the training progress between a clean and an extremely noisy signal, but moderate amounts of noise do not significantly change the convergence speed.

Due to these differences, we cannot assume, as is done when applying deep image priors in computer vision, that the output of the network can be used directly. Instead, our method tracks the sequence of network outputs during training and observes its behavior. A robust spectral mask is obtained, by considering the relative stability of every point in the spectrogram of the signal.

Our method achieves results that surpass all of the unsupervised literature methods and approach those of the supervised methods. Note that similar to most unsupervised methods in the literature, the method observes only the input signal and does not benefit from observing (even in an unsupervised way) other signals in the dataset.

2. Related work

Unsupervised Noise estimation algorithms Noise spectrum estimation is a crucial part of speech enhancement systems. Traditional noise estimation algorithms are based on similar assumptions, namely that the speech signal contains pauses and low-energy segments where statistics of the noise can be measured, and that the noise is more stationary than the speech signal. Noise estimation algorithms can be divided into three main categories: minimal-tracking algorithms [4, 5], which find the minimum for each frequency bin using a short time window; time-recursive averaging algorithms [6, 7, 8, 9], which average over time in order to provide the noise estimation; and histogram-based algorithms [10], in which the amplitude histogram is calculated for different frequency bands and the noise power is assumed to be the most occurring value.

The estimated a-priori SNR of the noise signal is then used as an input for one of a few classical speech enhancement algorithms. These algorithms multiply the original signal with a gain calculated from the a-priori SNR, either by a direct element-wise application, as in the Wiener filter, or using a regularization over the time-freqeuncy domain.

Supervised Noise estimation algorithms Supervised speech denoising algorithms observe, during training, both the noisy sample and the underlying clean samples and learn to map from noisy samples to clean samples. The SEGAN method [11] employs an encoder-decoder architecture, which is trained with an additional GAN loss [12].

The current state-of-the-art method, called Deep Feature Loss [13] uses a context aggregation network, and instead of using the MSE loss between the output and the target, employs a perceptual loss function. The perceptual loss is derived from the deep layer activations of a network that is pre-trained for audio classification tasks.

Deep Image Priors The DIP method [3] can be viewed a regularized inverse-problem method, in which the regularization is given implicitly, by training a deep CNN to fit the data. Specifically, a CNN of a given architecture is trained to produce the input image as its output, given a random tensor as the network's input. Assuming that the learning algorithm can fit a clean image much faster than it fits a noisy signal, the algorithm is stopped after it starts to fit the given image, but before it fits all of its details. In other words, there is a "noise impedance" effect that arises from the challenge of fitting the noisy signal, which is unpredictable in nature.

3. Method

The same phenomenon of noise impedance can be observed in audio, however, in a markedly different way that necessitates a different algorithmic approach. In our experiments, we employ the CNN architecture known as the WaveUnet [14], which consists of an encoder-decoder architecture with skip-connections between the two subnetworks. We create a random input signal z of the same dimension as the noisy signal $y = x + n$ (we assume an additive noise model, and the clean signal x and the noise n are unknown) and train the network $f = f_\theta$ to fit the noise, i.e., we solve the following minimization problem:

$$\min_\theta \|f_\theta(z) - y\|, \tag{1}$$

where θ is the parameter vector of the function f, i.e., the weights and biases of this network.

As can be seen in the example given in Fig. 1, the network fits clean speech or music signals much faster than it fits noise signals. However, unlike the situation in computer vision, there is little difference in the coarse behavior between the clean signal and the signal with the added noise.

While in images, the signal recovered by the network $f_\theta(z)$ during training becomes very similar to x, before it starts to resemble y, in audio the situation is different. As can be seen in Fig. 2, at every iteration i of training, the current network, which we denote as f_i, produces a signal $f_i(z)$ that is only partly denoised. In addition, while in images the method converges to a stable solution, i.e., $f_i(z) \sim f_{i+1}(z)$ after a few training iterations [3], this is not the case in audio. In the case of audio, the network output rapidly changes between iterations.

Moreover, the noise-free signal x was never reached in our experiments, even after extremely long training sessions. This can also be seen in the baseline experiment we perform (Sec. 4), in which we report the minimal error obtained when training the network ($\min_i \|f_i(z) - x\|$). This hindsight experiment, which would have produced a good result in computer vision, produces poor outputs in audio.

The discussion above does not mean that f does not evolve during training. As can be seen in Fig. 3, as the iterations progress, the output of f becomes more expressive, and the network models additional frequencies in the signal.

Based on these observations, we propose the method depicted in Alg. 1 for estimating the a-priori SNR of the clean signal. The input to the method is the signal y. Its output is a mask of the dimensions of the signal's STFT, with values in the range [0,1].

Algorithm 1 The denoising with network priors method

Input: n: noisy input, t: number of iterations

1: $\theta_0 \leftarrow \texttt{XavierInit}()$ ▷ Initialize the weights of f_0
2: $z \sim N(0,1)$ ▷ Initialize the random vector z
3: $Y_0 \leftarrow STFT(f_0(z))$
4: $C = 0$
5: **for** $i \leftarrow 1 : t$ **do**
6: $\theta_i \leftarrow \texttt{semi}\arg\min_\theta \|f_\theta(z) - y\|$ ▷ One training iteration on f_{i-1}, starting with $\theta = \theta_{i-1}$ obtaining f_i
7: $Y_i \leftarrow \texttt{STFT}(f_i(z))$
8: $H_i \leftarrow (\,||Y_i| - |Y_{i-1}|\,|)/|Y_i|$ ▷ Absolute differences
9: $p_1 \leftarrow \texttt{percentile}(H_i, 10)$
10: $p_2 \leftarrow \texttt{percentile}(H_i, 90)$
11: $H_i \leftarrow \max(\min(H_i, P_2), P_1)$ ▷ Clip values
12: $C \leftarrow C + H_i$ ▷ Accumulate the differences
13: **end for**
14: $M = (\max(C) - C)/(\max(C) - \min(C))$ ▷ Normalize
15: **return** M ▷ Estimated a-priori SNR of the signal

After computing a random z vector in line 2, the method undergoes an iterative process for t iterations. Unlike the situation in computer vision, in speech, and other audio signals that we tried, the network f cannot easily fit y. Early stopping is, therefore, not a major concern, and we can choose any number of iterations t that is large enough.

Each iteration consists of the following steps, where i is the iteration index. First, in line 6 of the algorithm, the network f_{i-1} is trained for one iteration, obtaining f_i. Then, in line 7, one computes $f_i(z)$ and its STFT Y_i. We next compute H_i, which is the absolute difference between $|Y_{i-1}|$ and $|Y_i|$ normalized by the latter.

In order to avoid extreme values, every value in H_i that is above the 90th percentile or below the 10th percentile is clipped. An accumulator C sums the resulting matrices (line 12). The accumulator would have high values in the coordinates of the time-frequency domain, in which there is the least stability in the reconstruction of y by the network f.

Once the t iterations are over, C is normalized to be in the range of $[0, 1]$ (line 14). High accumulated variability implies noise and we, therefore, flip the values, before returning the mask M.

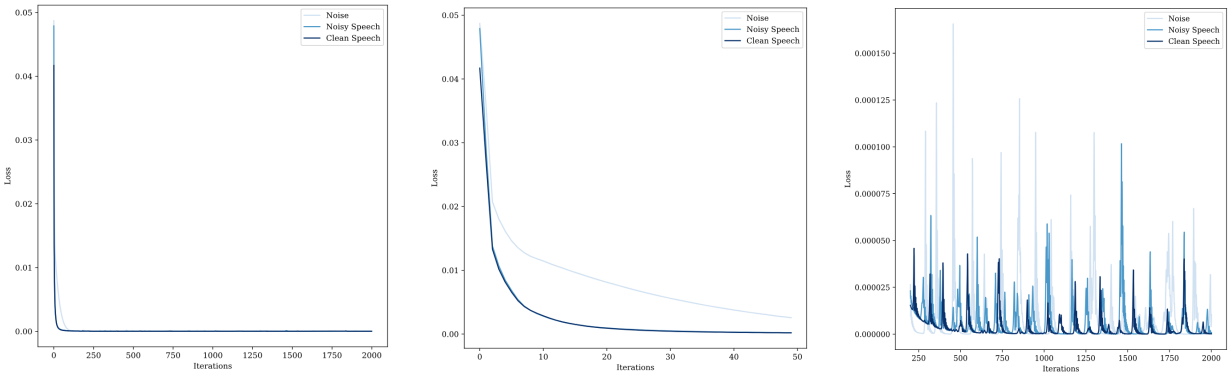
With this estimation of the a-pirori SNR, a classical denoising method, such as LSA [2] or the Weiner filter can be used to perform denoising. In our experiments, we employ the former.

4. Experiments

When applying our method, we employ a WaveUnet with six layers and 60 filters per layer. Each mask filter was produced after $t = 5000$ iterations using the Adam optimizer with a learning rate of 0.0005. The method seems insensitive to either of these parameters.

Noisy speech samples were provided by the authors of [16]. For the purpose of our work, only the test set has been used. This test set is composed by mixing multiple speakers with 5 different noise types and 4 different SNR setting (2.5, 7.5, 12.5 and 17.5 dB). The original 48 kHz files were downsampled to 16 kHz, the same as other baseline methods [13, 11]. The spectrograms are obtained by using 32 ms Hann window and 8 ms hop length.

For the purpose of computing the results of the baseline unsupervised denoising algorithms, the open source metrics evalu-



(a) (b) (c)

Figure 1: *Typical loss profiles obtained during training for a signal that is clean, noisy, or entirely noise. (a) first 2000 iterations. (b) zoom-in to the first 50 iterations. (c) zoom in to iteration 250 onward.*

Table 1: *Quantitative evaluation denoising. A higher score means better performance.*

Approach	Supervised	CSIG	CBAK	COVL	PESQ	SSNR
SEGAN [11]	yes	3.48	2.94	2.80	2.16	7.73
Deep Feature Loss [13]	yes	3.86	3.33	3.22	-	-
MCRA [6]	no	2.23	2.36	1.91	1.80	5.17
IMCRA [7]	no	2.49	2.53	2.13	1.92	5.89
MCRA2 [9]	no	2.39	2.50	2.08	1.97	5.92
Martin [5]	no	2.48	2.61	2.21	2.12	6.37
Doblinger [4]	no	2.55	2.66	2.29	2.21	6.48
Hirsch [10]	no	2.67	2.66	2.35	2.21	6.26
Connected Frequencies [8]	no	2.73	2.66	2.38	2.21	6.18
Wiener [15]	no	**3.23**	2.68	**2.67**	2.22	5.07
Ours	no	3.08	**2.84**	**2.67**	**2.39**	**7.27**
Best $f_i(z)$	hindsight	3.04	2.36	2.39	1.81	1.59
Averaged $f_i(z)$	no	3.13	2.39	2.47	1.87	1.66
Noisy Samples	no	3.35	2.44	2.63	1.97	1.68

ation and noise estimation toolbox [17] was used. The statistics of the supervised baselines were taken from the respective papers [11, 13].

Multiple quality scores are used to measure the success of the methods, including (1) CSIG: Mean opinion score (MOS) predictor of signal distortion, (2) CBAK: MOS predictor of background-noise intrusiveness, (3) COVL: MOS predictor of overall signal quality, (4) PESQ: Perceptual evaluation of speech quality, and (5) SSNR: Segmental SNR.

All unsupervised methods were post-processed by a high-pass filter with a cutoff frequency of 60 Hz, to remove noise below the human speech base frequency. The noise cleaning method used is MMSE-LSA [2], and the various measures were computed by using the open source toolbox mentioned above.

A sample result is give in Fig. 4, and the statistics are reported in Tab. 1. As can be seen from the table, our method outperforms all unsupervised literature methods in all metrics, with the exception of the CSIG metric, in which it is the second highest method. The method is also largely comparable to the SEGAN method [11], despite not being trained on any sample outside the single sample y, while SEGAN is fully supervised. Our unsupervised method is outperformed by the Deep Feature Loss method [13], which enjoys both a large fully supervised training set and a strong pretrained perceptual loss.

We also present the results of a baseline, in which the best network output obtained during training $f_i(z)$, $i = 1..t$ is returned. This happens in hindsight, by comparing it to the clean signal x. As can be seen, the statistics for this result are very similar to those of the original noisy signal. One can also observe that averaging multiple network reconstructions $f_i(z)$, $i = 250, 500, 750, \ldots, 5000$, does not lead to an acceptable result. This is in contrast to the application of deep network priors in computer vision [3], where the network produces clean outputs during its training.

The samples, up to the conference file size limit, are attached as supplementary. More samples can be found next to our code at `https://github.com/mosheman5/DNP`.

5. Conclusions

The advent of deep learning has led to effective supervised denoising algorithms. However, as far as we know, little progress has been made in the unsupervised domain. In this work, we explore the usage of deep network priors for this task and observe that the approach used to employ these priors in images is not suitable for audio signals. We, therefore, develop a new method, which is shown to outperform the unsupervised methods and even approach the quality of the supervised methods.

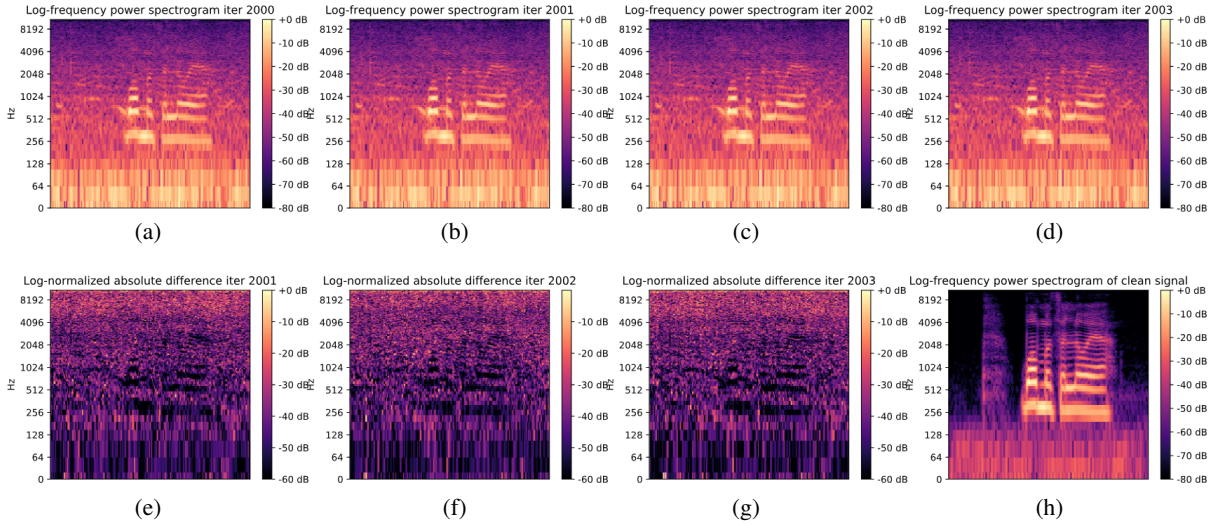


Figure 2: *Instability during training. (a-d) the network output $f_i(z)$ for four consecutive iterations. (e-g) the difference between pairs of consecutive iterations. (h) the spectrogram of the clean signal*

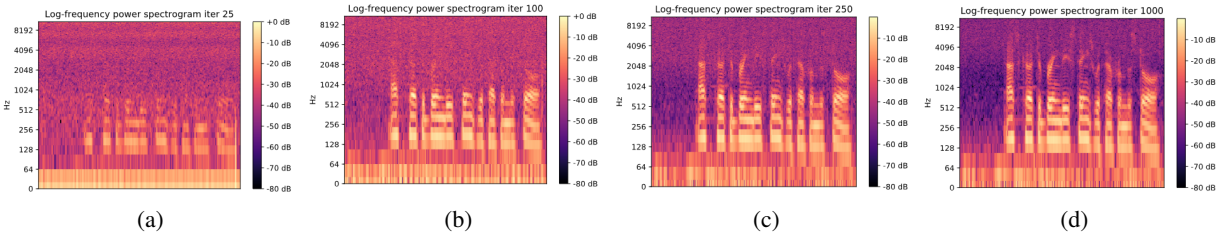


Figure 3: *Progress during training. (a) iteration 25. (b) iteration 100. (c) iteration 250. (d) iteration 1000. The low frequencies are learned first, and the higher frequencies follow.*

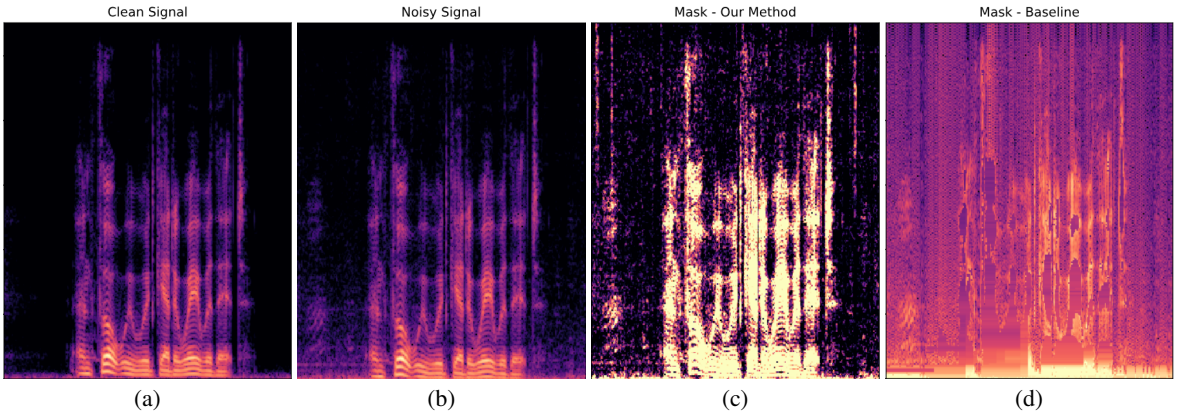


Figure 4: *Sample results. (a) The spectrogram of the clean signal x. (b) The spectrogram of the noisy signal y. (c) The a-priori mask of the signal M returned by our method. (d) The mask obtained by the Connected Frequencies [8] method.*

6. References

[1] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, Dec 1979.

[2] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, April 1985.

[3] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[4] G. Doblinger, "Computationally efficient speech enhancement by spectral minima tracking in subbands," in *in Proc. Eurospeech*, 1995, pp. 1513–1516.

[5] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, July 2001.

[6] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 12–15, Jan 2002.

[7] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 466–475, Sep. 2003.

[8] K. V. Sørensen and S. V. Andersen, "Speech enhancement with natural sounding residual noise based on connected time-frequency speech presence regions," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 18, p. 305909, Nov 2005. [Online]. Available: https://doi.org/10.1155/ASP.2005.2954

[9] S. Rangachari and P. C. Loizou, "A noise-estimation algorithm for highly non-stationary environments," *Speech Communication*, vol. 48, no. 2, pp. 220 – 231, 2006.

[10] H. G. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, May 1995, pp. 153–156 vol.1.

[11] S. Pascual, A. Bonafonte, and J. Serrà, "Segan: Speech enhancement generative adversarial network," in *Interspeech*, 2017, pp. 3642–3646.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[13] F. G. Germain, Q. Chen, and V. Koltun, "Speech denoising with deep feature losses," *arXiv preprint arXiv:1806.10522*, 2018.

[14] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018.

[15] P. Scalart and J. V. Filho, "Speech enhancement based on a priori signal to noise estimation," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2, May 1996, pp. 629–632 vol. 2.

[16] C. Valentini-Botinhao, "Noisy speech database for training speech enhancement algorithms and tts models," 2017.

[17] P. C Loizou, *Speech Enhancement: Theory and Practice 2nd Edition*. CRC Press, 2013.