REMOVING LAVALIER MICROPHONE RUSTLE WITH RECURRENT NEURAL NETWORKS

Gordon Wichern and Alexey Lukin

iZotope, Inc.
Cambridge, MA, USA
alex@izotope.com

ABSTRACT

The noise that lavalier microphones produce when rubbing against clothing (typically referred to as *rustle*) can be extremely difficult to automatically remove because it is highly non-stationary and overlaps with speech in both time and frequency. Recent breakthroughs in deep neural networks have led to novel techniques for separating speech from non-stationary background noise. In this paper, we apply neural network speech separation techniques to remove rustle noise, and quantitatively compare multiple deep network architectures and input spectral resolutions. We find the best performance using bidirectional recurrent networks and spectral resolution of around 20 Hz. Furthermore, we propose an ambience preservation post-processing step to minimize potential gating artifacts during pauses in speech.

1. INTRODUCTION

The lavalier microphone (lav mic) is an invaluable tool for the audio engineer. By inconspicuously attaching near the mouth it allows the person wearing the microphone to move freely, minimizes visual distractions, and also helps to reduce reverberation and noise from the recording environment. Because lav mics are typically attached to a subjects wardrobe, they can sometimes rub against clothing creating an auditory disturbance often described as rustle. Lav mic rustle can overlap with speech in both time and frequency and vary in unpredictable ways based on how the person wearing the microphone moves their body. This makes developing an algorithm to automatically detect and remove rustle extremely challenging.

Traditional techniques for single-channel speech enhancement, e.g., spectral subtraction [1], work well for stationary background noise (e.g., air conditioner hum), but struggle in the presence of non-stationary disturbances, such as lav mic rustle. Recently, source separation approaches have achieved success in separating speech from complex non-stationary background noise, such as music, weather, or even other speech [2]. These techniques typically operate on a time-frequency representation of the signal, e.g., the spectrogram, and often take a supervised learning approach where a collection of clean speech and isolated noise samples are used to learn a model. Once trained, this model can obtain separated speech and noise signals when given noisy speech as input.

Time-frequency masking is one approach to single-channel speech separation which estimates the amount of speech and noise present in each spectrogram bin (i.e., the mask). This mask is then used as a time-varying filter to separate speech from noise. Recent advances in deep neural networks have drastically improved the ability to learn the nonlinear mapping function necessary to estimate time-frequency masks from noisy speech. The approaches of [3, 4, 5, 6] use feedforward network architectures where the mask for a frame of audio is predicted using input features from several surrounding frames. In this architecture, increasing the amount of temporal context requires increasing the dimension of the network input, which extends the size of the entire network. This increases the risk of overfitting and the resources necessary to train and deploy the network.

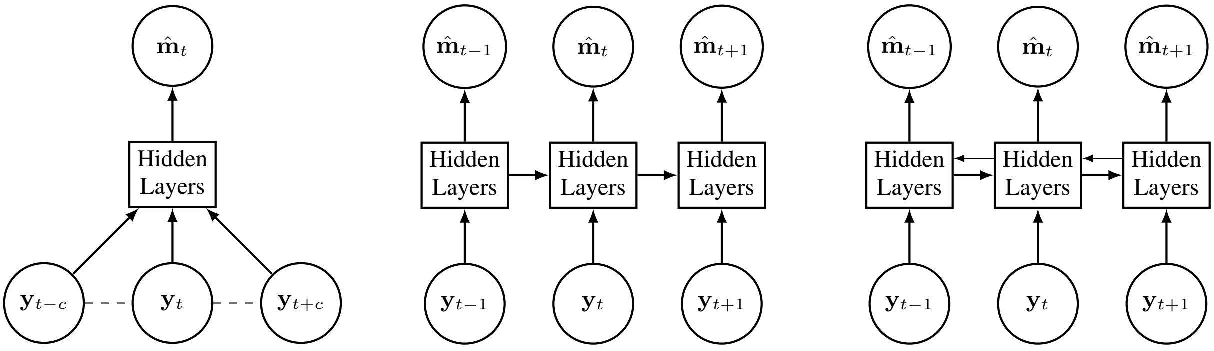
For this reason, recurrent architectures have demonstrated success on several sequential prediction tasks [7] like language translation, video captioning, speech recognition, and speech/noise separation. Recurrent architectures save an internal hidden state between time steps, and the appropriate context for a problem at hand can be learned from data. However, to avoid the vanishing/exploding gradient problem, gated architectures, such as the long short-term memory (LSTM) [8], must be used. Additionally, [9] showed improved performance on a speech noise separation task using a bidirectional LSTM (BLSTM) [10], which performs both a forward and backward pass over the data, thus incorporating future context at the cost of offline operation.

In this paper, we explore deep feedforward, recurrent LSTM, and BLSTM network architectures for removing lav-mic rustle from speech, a specific problem for audio engineers that, to the best of our knowledge, has not previously been explored in the literature. We begin by reviewing mask estimation approaches to single-channel source separation and different deep network architectures in Section 2. We benchmark the performance of our recurrent architectures against feedforward networks in terms of noise reduction and speech intelligibility and explore trade-offs in the spectral features used as network inputs in Section 3. Techniques for removing lav-mic rustle while maintaining a certain amount of background ambience to maintain the natural quality of the recording are explored in Section 4. Finally, conclusions and discussions of future work are provided in Section 5.

2. SPEECH SEPARATION NETWORKS

Our algorithm works on a mono mixture $y(t) = s(t) + n(t)$ of speech $s(t)$ corrupted by lav mic rustle $n(t)$. Given a training set with examples of isolated speech and rustle signals, we create mixtures with known ground truth to learn a mapping that estimates the clean speech signal $\hat{s}(t)$ from noisy mixture $y(t)$. Rather than operating on the time-domain waveform, our neural networks take as input the short-time Fourier transform (STFT) magnitude spectrogram of $y(t)$, denoted as $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T] \in \mathbb{R}^{d \times T}$, where d is the number of frequency bins.

We use the magnitude ratio mask as the time-varying filter for



(a) Feedforward

(b) Recurrent

(c) Bidirectional

Figure 1: Incorporating temporal context via multiple frame inputs (a) or hidden layer state propagation (b) and (c).

separating speech and rustle, which is defined as

$$\mathbf{m}_t = \frac{\mathbf{s}_t}{\mathbf{s}_t + \mathbf{n}_t}, \tag{1}$$

where t is the STFT frame (time) index, $\mathbf{s}_t$ are the spectral magnitude coefficients of clean speech, and $\mathbf{n}_t$ is the rustle magnitude spectrum. The division operation in (1) is performed element-wise. Because magnitude spectra $\mathbf{s}_t$ and $\mathbf{n}_t$ are nonnegative, the mask elements $\mathbf{m}_t$ from (1) are in the interval $[0, 1]$. The output of our neural network is $\hat{\mathbf{m}}_t$, which we use to obtain estimated magnitude spectra for separated speech and rustle, i.e.,

$$\hat{\mathbf{s}}_t = \hat{\mathbf{m}}_t \odot \mathbf{y}_t, \tag{2}$$

$$\hat{\mathbf{n}}_t = (\mathbf{1} - \hat{\mathbf{m}}_t) \odot \mathbf{y}_t, \tag{3}$$

where $\odot$ represents an element-wise product. We use (2) and (3) to obtain the estimated time-domain waveforms $\hat{s}(t)$ and $\hat{y}(t)$ through the inverse STFT, with phase information taken from the noisy mixture $y(t)$.

2.1. Network architectures

We estimate $\hat{\mathbf{m}}_t$ using a feedforward neural network architecture as follows:

$$\hat{\mathbf{m}}_t = \sigma(\mathbf{W}^L \mathbf{h}_t^{L-1} + \mathbf{b}^L), \tag{4}$$

$$\mathbf{h}_t^\ell = ReLU(\mathbf{W}^\ell \mathbf{h}_t^{\ell-1} + \mathbf{b}^\ell),\ \ell = 2, ..., L-1, \tag{5}$$

$$\mathbf{h}_t^1 = ReLU(\mathbf{W}^1 \mathbf{y}_t^c + \mathbf{b}^1), \tag{6}$$

where L represents the number of layers, $\sigma(\cdot)$ the sigmoid nonlinearity, and $ReLU(\cdot)$ the rectified linear unit activation function. The weight and bias parameters of layer ℓ, whose values are learned during training, are denoted by $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$. The input to the feedforward architecture is $\mathbf{y}_t^c = [\mathbf{y}_{t-c}, ..., \mathbf{y}_t, ..., \mathbf{y}_{t+c}]^T \in \mathbb{R}^{d(2c+1)}$, which incorporates temporal context by stacking a small number of frames to use as the network input.

We can alternatively incorporate temporal context using a recurrent network architecture for estimating $\hat{\mathbf{m}}_t$ as

$$\hat{\mathbf{m}}_t = \sigma(\mathbf{W}^L \mathbf{h}_t^{L-1} + \mathbf{b}^L), \tag{7}$$

$$\mathbf{h}_t^\ell = f(\mathbf{h}_t^{\ell-1},\ \mathbf{h}_{t-1}^\ell),\ \ell = 2, ..., L-1, \tag{8}$$

$$\mathbf{h}_t^1 = f(\mathbf{y}_t,\ \mathbf{h}_{t-1}^1), \tag{9}$$

where $f(\cdot)$ represents the nonlinear mapping function of a recurrent layer, and the state of each recurrent hidden layer, i.e., $\mathbf{h}_t^\ell$ for layer ℓ is stored and used as an additional input at the next time step. We use LSTM-style [8] recurrent layers for $f(\cdot)$, which were successfully used for speech denoising in [9, 11]. The input to the network, $\mathbf{y}_t$ in (9), is only a single spectrogram frame.

We can further incorporate temporal context into a source separation architecture by using bidirectional LSTM (BLSTM) archictures [10]. BLSTM networks require offline operation, and we can define a BLSTM layer as

$$\tilde{\mathbf{h}}_t^\ell = f(\mathbf{h}_t^{\ell-1}, \vec{\mathbf{h}}_{t-1}^\ell) + f(\mathbf{h}_t^{\ell-1}, \overleftarrow{\mathbf{h}}_{t+1}^\ell), \tag{10}$$

where $\vec{\mathbf{h}}_{t-1}^\ell$ and $\overleftarrow{\mathbf{h}}_{t-1}^\ell$ are outputs of the forward and backward recurrent layers, respectively. The backward layer $\overleftarrow{\mathbf{h}}_{t-1}^\ell$ consumes the input spectrogram in time-reversed order. Figure 1 illustrates how temporal context is incorporated in feedforward, recurrent (LSTM), and bidirectional recurrent (BLSTM) layers.

2.2. Training objective

Given a training set of isolated speech and isolated rustle noise spectrograms, we can create mixtures with known ground truth for learning the nonlinear mapping between noisy speech spectra $\mathbf{y}_t$ and estimated ratio mask $\hat{\mathbf{m}}_t$. Several studies on neural network based speech separation [2, 3, 11] have shown the utility of using the error in the estimated spectrum $\hat{\mathbf{s}}_t$ (as opposed to the error in the estimated mask $\hat{\mathbf{m}}_t$) as the network training objective. This leads to the so-called signal approximation mean squared error objective function

$$J_{MSE} = \frac{1}{T} \sum_{t=1}^{T} ||\hat{\mathbf{s}}_t - \mathbf{s}_t||_2^2. \tag{11}$$

But using this objective can sometimes cause the network to be too conservative in situations where noise is quieter than speech, yet still perceptible. An alternative objective proposed in [2] is

$$J_{DIS} = \frac{1}{T} \sum_{t=1}^{T} \Big(||\hat{\mathbf{s}}_t - \mathbf{s}_t||_2^2 + ||\hat{\mathbf{n}}_t - \mathbf{n}_t||_2^2 - \gamma ||\mathbf{s}_t - \hat{\mathbf{n}}_t||_2^2 - \gamma ||\mathbf{n}_t - \hat{\mathbf{s}}_t||_2^2 \Big), \tag{12}$$

where the parameter γ provides a trade-off between interference (i.e., rustle remaining in the separated speech) and artifacts caused by the source separation process. We found setting $\gamma = 10^{-3}$ to work well for removing rustle from speech, and we use that value and the objective function from (12) in all experiments described in Section 3. The objective function in (12) is minimized using backpropagation and stochastic gradient descent.

3. EXPERIMENTS

3.1. Dataset description

To create a training set for rustle noise removal, we needed to collect a large amount of clean speech and isolated rustle data. While several publicly available datasets for speech research offer only low sampling rate data (less than 40 kHz), we have used the pitch tracking corpus from [12], the reverberant speech from the Chime challenge [13], the processed speech from the DAPS experiment [14], and the TSP speech dataset [15]. All of these datasets provide audio at sampling rates of 44.1 or 48 kHz. We have also supplemented our clean speech training data with several hours of audio recorded for iZotope tutorial videos. While no publicly available datasets of isolated rustle exist, we were able to use sound effects from *www.prosoundeffects.com* that shared sonic qualities with lav mic rustle. However, these sound effects alone were insufficient to cover the wide range of lav mic rustle disturbances we wanted our algorithm to remove. We thus collected approximately one hour of isolated lav mic rustle noise, varying microphone type, clothing, movement, and recording environment. All of the audio processed in these experiments had a sampling rate of 48 kHz.

While most rustle disturbances are rather quiet relative to speech (i.e., $\mathrm{SNR} \gg 0$), we also wanted our algorithm to be robust to low-SNR situations, such as lav mics mounted on athletes during competition or while outdoors in extreme weather events. Thus, using these isolated speech and rustle noise datasets, we created mixtures with SNR ranging from -6 to $+9$ dB (SNR has been measured over periods with active rustle). To limit the computational resources necessary for training, all mixtures were limited to 10 seconds in length, and these mixtures sometimes consisted of multiple speech utterances and/or rustle noise concatenated together prior to forming the mixture.

While we can qualitatively test the performance of our algorithm using actual rustle-corrupted speech, to quantitatively evaluate performance using the metrics described in Section 3.2 requires mixtures with ground truth (i.e., the isolated speech and rustle used to create the mixture) available. This testing dataset is composed of speech from speakers not used to train the algorithm, as well as held out rustle noises that were distinct from those used during training. All testing set mixtures were 12 seconds in length and the SNR varied over the same -6 to $+9$ dB range.

3.2. Performance metrics

We quantitatively evaluate the performance of our algorithm in terms of separation performance and intelligibility. For separation performance, we use the SNR of the separated speech, typically referred to as source to distortion ratio (SDR) in the source separation literature [16]. For speech quality and intelligibility, we use the short-term objective intelligibility (STOI) metric proposed in [17]. The STOI algorithm returns a value in $[0, 1]$ range, with 1 representing the highest quality. However, we evaluate our rustle removal in terms of $\Delta\mathrm{STOI}$, which we define as the difference between the STOI score of the separated speech and that of the original noisy mixture, converted to a percentage.

3.3. Analysis of results

In this section we compare performance of our rustle removal algorithm for different network structures and their associated temporal context, as shown in Figure 1. Additionally, we investigate the impact of spectral resolution (i.e., the FFT size) used to create the spectrograms input and output by the network. All experiments use the Adadelta [18] optimizer and are trained for 20,000 mini-batches of 16 sequences each. Each sequence consists of 10 seconds of clean speech randomly mixed with segments of mic rustle.

Besides comparing objective measures, we have also performed informal listening tests with real-world speech signals having diverse SNR. They have shown a significant reduction in audibility of rustle. Some audio examples are available for download at *http://www.izotope.com/tech/aes_rustle*

3.3.1. FFT size

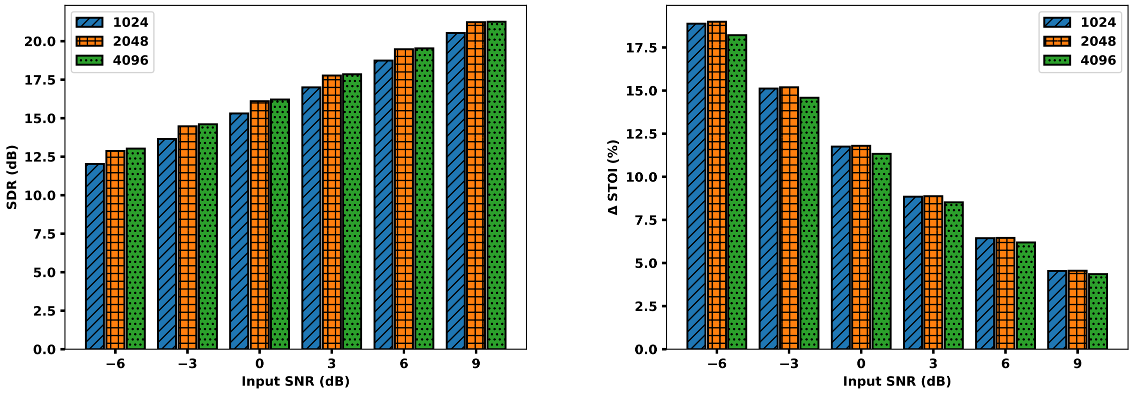
To determine an upper bound on source separation performance, we can use the so-called “oracle mask” which is the magnitude ratio mask computed using the ground truth isolated speech and rustle noise spectrograms. Figures 2(a) and (b) display the SDR and Δ STOI for FFT sizes of 1024, 2048, and 4096 (at 48 kHz sampling rate) as a function of input SNR. For all FFT sizes we used $4\times$ overlap and Hann windows. From Figure 2 we see that the FFT size of 4096 performs best in terms of SDR, but worst in terms of STOI.

Figure 3 repeats the same FFT size comparison, but this time evaluates testing set performance of a trained two-layer BLSTM network with 256 hidden units per layer. The SDR from Figure 3(a) exhibits the opposite trend with respect to increasing FFT size when compared to the oracle results from Figure 2(a), with 4096-point FFT leading to the lowest level performance. This discrepancy might be caused by the curse of dimensionality, as larger FFT sizes require more network parameters in the input and output layers. In terms of STOI performance for the trained BLSTM network shown in Figure 3(b), an FFT size of 2048 exhibits the best performance, while FFT sizes of 1024 and 4096 perform similarly, although the larger FTT size (4096) does show improvements at SNR of -6 dB. In terms of both the SDR and STOI results from Figures 2 and 3, the FFT size of 2048 appears to consistently demonstrate strong performance for both the oracle and trained BLSTM network.

3.3.2. Network structure

In this section we evaluate the feedforward, recurrent (LSTM), and bidirectional (BLSTM) architectures shown in Figure 1. All three architectures were designed to have a nearly equivalent number of parameters as shown in Table 1. For the feedforward architecture we used a context size of $c = 2$ frames, meaning that our network input is the concatenation of five frames. Because a single BLSTM layer has independent forward and backward layers, its complexity is comparable to a forward-only LSTM with four hidden layers.

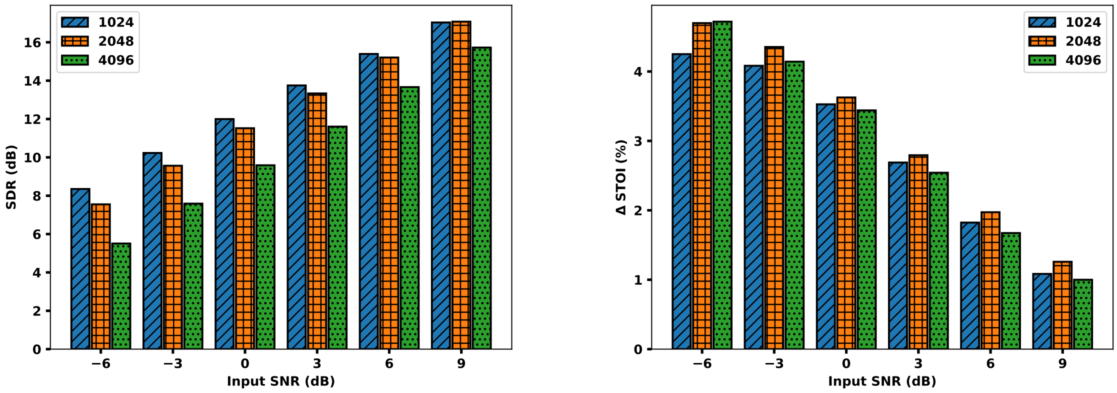
Figures 4(a) and (b) display the SDR and STOI for the three different network architectures. The LSTM and BLSTM perform similarly in terms of SDR and much better than the feedforward



(a) SDR

(b) ΔSTOI

Figure 2: Metrics of separated speech using the oracle (ground truth) ratio mask for different FFT sizes at different input SNR levels.



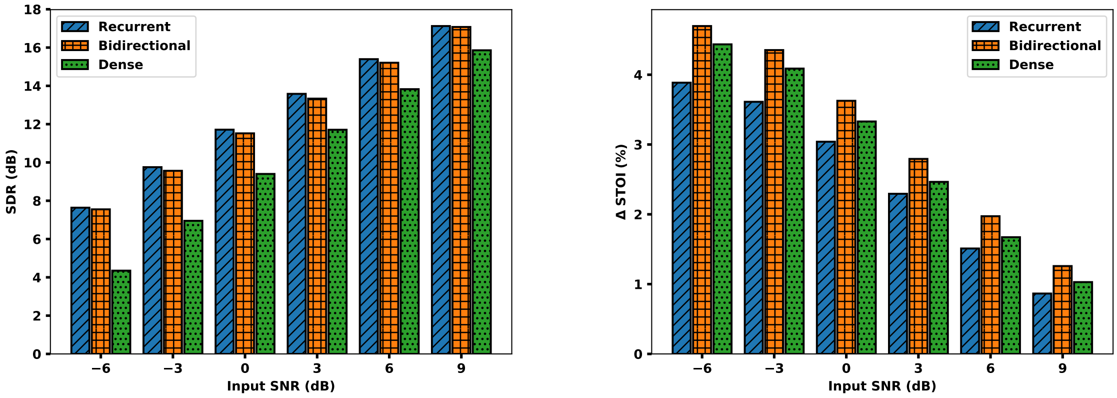
(a) SDR

(b) ΔSTOI

Figure 3: Metrics of speech separated using BLSTM network for different FFT sizes at different input SNR levels.

	Input	Hid. 1	Hid. 2	Hid. 3	Hid. 4	Output
Feedforward	5125	512	512	512	512	1025
LSTM	1025	256	256	256	256	1025
BLSTM	1025	256	256	N/A	N/A	1025

Table 1: Layer sizes for different network configurations using 2048-point FFT. Sizes were chosen such that all architectures had approximately the same number of parameters.



(a) SDR

(b) ΔSTOI

Figure 4: Metrics of separated speech comparing different network structures with FFT size of 2048 at different input SNR levels.

("dense") architecture, as shown in Figure 4(a). We can also interpret this result in terms of the amount of temporal context the network has available. Since the LSTM and BLSTM perform similarly, this could mean that future context is less important in terms of SDR. The recurrent architectures, however, exploit significantly more context than the feedforward architecture has available. In terms of the STOI shown in Figure 4(b), the BLSTM architecture performs best and the forward-only LSTM performs worst, demonstrating the importance of future context for intelligibility. Although the BLSTM architecture exhibits strong performance, it requires offline access for the backward pass over the data. A low-latency implementation becomes possible if BLSTM works on blocks of the audio signal or if a lookahead layer [19] is added to the forward-only LSTM architecture.

4. AMBIENCE PRESERVATION

The speech separation network trained on clean speech mixed with mic rustle seeks to optimally recover clean speech. In many real-life scenarios, input speech is corrupted with both mic rustle and some stationary (or quasi-stationary) noise (Figure 5(a)). In such cases our net trained for speech isolation produces excessive gating, i.e., attenuates stationary noise between sentences (Figure 5(b)). This can cause the separated speech to sound unnatural or overly processed, which was confirmed by our informal listening tests. The algorithm proposed in this section mitigates the problem by estimating the stationary noise floor and limiting the amount of spectral attenuation $\hat{\mathbf{m}}_t$ to ensure that the resulting signal $\hat{\mathbf{s}}_t$ does not have excessive gating (Figure 5(c)).

Because the algorithm adapts to the noise floor, it can be used for signals with low or high SNR. Its application is optional and often makes sense in the context of post-production, where preservation of the stationary noise floor ("room tone") is desirable.

4.1. Noise estimation

A simple adaptation algorithm is used to detect the quasi-stationary noise floor in speech. It operates on a magnitude spectrogram $\mathbf{y}_t$ of the input signal and computes magnitude estimates of the noise floor $\hat{\mathbf{n}}_t$ by applying a series of three filters: a Hann filter H, a sliding minimum filter M, and an asymmetric 1st order attack/decay filter E [20].

$$\hat{\mathbf{n}}_t = E(M(H(\mathbf{y}_t))) \tag{13}$$

The filters are independently applied to each frequency bin of the spectrogram along the time axis. The purpose of filter E is to quickly react to decays in the signal energy and slowly react to onsets of the signal energy. Its upward integration time (attack time) is set to 10000 ms, while its downward integration time (decay time) is set to 100 ms. The purpose of filter M is to keep noise floor estimates steady during speech utterances. Its window size is set to 2000 ms. The purpose of filter H is to prevent filter M from becoming trapped in spectrogram zeros. Its radius is set to 10 ms.

4.2. Limiting of attenuation

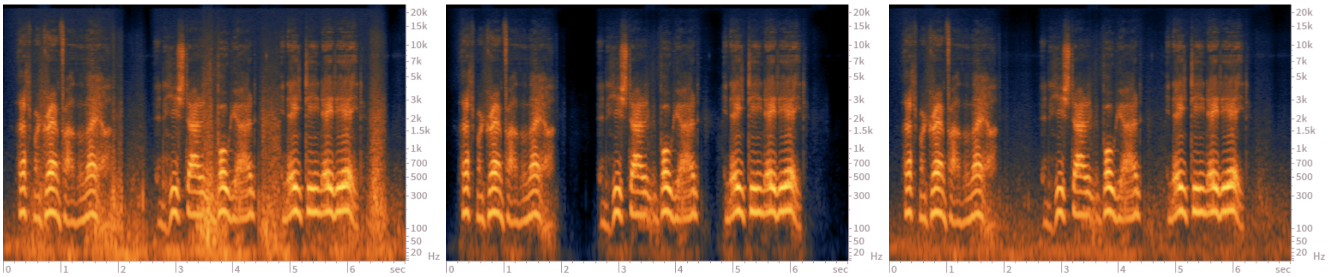
Gating is created when the resulting signal energy $\hat{\mathbf{s}}_t = \hat{\mathbf{m}}_t \odot \mathbf{y}_t$ falls below the noise floor $\hat{\mathbf{n}}_t$. To prevent this, we are limiting the spectral mask $\hat{\mathbf{m}}_t$ as follows:

$$\hat{\mathbf{m}}_t^{+} = \min\left\{1,\ \max\left\{\hat{\mathbf{m}}_t,\ \frac{\hat{\mathbf{n}}_t}{\mathbf{y}_t}\right\}\right\}. \tag{14}$$

Our noise floor estimate $\hat{\mathbf{n}}_t$ is quasi-stationary (slowly changing in time), so its distribution does not match the distribution of a typical noise power spectrum, which is random. When a quasi-stationary constraint (14) is applied to the mask and then to the signal (2), parts of the output signal obtain this unnatural distribution too. To improve naturalness of the distribution, we are applying a time-frequency smoothing to the mask $\hat{\mathbf{m}}_t^{+}$ using a "DFT thresholding" algorithm from [21]. This edge-adaptive smoothing also reduces "musical noise" artifacts resulting from processing the STFT spectrum. The updated processing formulas with smoothing of the mask are as follows:

$$\hat{\mathbf{m}}_t^{++} = \mathrm{Smooth}\left(\hat{\mathbf{m}}_t^{+}\right), \tag{15}$$

$$\hat{\mathbf{s}}_t = \hat{\mathbf{m}}_t^{++} \odot \mathbf{y}_t. \tag{16}$$



(a) Speech with rustle

(b) De-rustle, formula (2)

(c) De-rustle, formula (16)

Figure 5: Comparison of rustle attenuation without (b) and with (c) ambience preservation. Additional audio examples are available at *http://www.izotope.com/tech/aes_rustle*

5. CONCLUSIONS AND FUTURE WORK

In this paper we have described an approach for lavalier microphone rustle removal using deep neural networks, while maintaining natural sounding audio quality by supplementing the network output with spectral smoothing and stationary noise floor estimation. We also found a spectral resolution of around 20 Hz (FFT size of 2048 at 48 kHz) and bidirectional recurrent network architectures to provide the best performance for this specific speech separation application.

Bidirectional recurrent architectures (e.g., BLSTM) exhibited the overall best performance, but investigating low-latency bidirectional approximations for rustle removal is an important area for additional study. Exploring complex ratio masks [6] or time-domain Wavenet architectures [22] are other potentially interesting areas of future work.

6. REFERENCES

[1] P.C. Loizou, *Speech Enhancement: Theory and Practice*, CRC Press, Boca Raton, FL, 2013.

[2] P.S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, “Joint optimization of masks and deep recurrent neural networks for monaural source separation,” *IEEE/ACM Transactions on Speech and Language Processing*, vol. 23, pp. 2136–2147, Dec. 2015.

[3] Y. Wang, A. Narayanan, and D.L. Wang, “On training targets for supervised speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1849–1858, 2014.

[4] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “A regression approach to speech enhancement based on deep neural networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 7–19, 2015.

[5] J. Chen, Y. Wang, S.E. Yoho, D.L. Wang, and E.W. Healy, “Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises,” *Journal of the Acoustical Society of America*, vol. 139, pp. 2604–2612, 2016.

[6] D.S. Williamson, Y.. Wang, and D.L. Wang, “Complex ratio masking for monaural speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 483–492, 2016.

[7] K. Cho, A. Courville, and Y. Bengio, “Describing multimedia content using attention-based encoder-decoder networks,” *IEEE Transactions on Multimedia*, vol. 17, pp. 1875–1886, 2015.

[8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 23, no. 8, pp. 1735–1780, 1997.

[9] H. Erdogan, J.R Hershey, S. Watanabe, and J. Le Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712.

[10] M. Schuster and K.K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997.

[11] F. Weninger, J. Le Roux, J.R. Hershey, and B. Schuller, “Discriminatively trained recurrent neural networks for single-channel speech separation,” in *IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing*, 2014, pp. 577–581.

[12] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, “A pitch tracking corpus with evaluation on multipitch tracking scenario,” in *Interspeech*, 2011, pp. 1509–1512.

[13] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, “The PASCAL CHiME speech separation and recognition challenge,” *Computer Speech and Language*, vol. 27, pp. 621–633, 2013.

[14] G.J. Mysore, “Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech? A dataset, insights, and challenges,” *IEEE Signal Processing Letters*, vol. 22, pp. 1006–1010, 2015.

[15] P. Kabal, “TSP speech database,” Tech. Rep., Department of Electrical & Computer Engineering, McGill University, Montreal, Quebec, Canada, 2002.

[16] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1462–1469, 2006.

[17] C. Taal, R. Hendriks, R. Heusdens, and J. Jensen, “An algorithm for intelligibility prediction of time-frequency weighted noisy speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 2125–2136, 2011.

[18] M. D. Zeiler, “ADADELTA: an adaptive learning rate method,” *Computing Research Repository*, 2012.

[19] C. Wang, D. Yogatama, A. Coates, T. Han, A. Hannun, and B. Xiao, “Lookahead convolution layer for unidirectional recurrent neural networks,” in *Workshop Extended Abstracts of the 4th International Conference on Learning Representations*, 2016.

[20] A. Lukin, “Tips & tricks: fast image filtering algorithms,” in *Proceedings of Graphicon’2007, Moscow, Russia*, 2007, pp. 186–189.

[21] A. Lukin and J. Todd, “Suppression of musical noise artifacts in audio noise reduction by adaptive 2D filtering,” in *Audio Engineering Society Convention 123*, Oct 2007.

[22] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and Kavukcuoglu, “Wavenet: A generative model for raw audio,” *CoRR abs/1609.03499*, 2016.

[23] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, and E. Elsen, “Deep speech 2: End-to-end speech recognition in English and Mandarin,” in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 173–182.

[24] K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Conference on Empirical Methods in Natural Language Processing*, 2014.

[25] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, “Non-negative matrix factorization based compensation of music for automatic speech recognition,” in *Interspeech*, 2010, pp. 717–720.

[26] D. Wang, “Time-frequency masking for speech separation and its potential for hearing aid design,” *Trends in Amplification*, vol. 12, pp. 332–353, 2008.

[27] C. Raffel, B. McFee, E.J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D.P.W. Ellis, “mir_eval: A transparent implementation of common MIR metrics,” in *Proceedings of the 15th International Conference on Music Information Retrieval*, 2014.

[28] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, “Detection and classification of audio scenes and events,” *IEEE Transactions on Multimedia*, vol. 17, pp. 1733–1746, 2015.

[29] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in *24th European Signal Processing Conference (EUSIPCO 2016)*, 2016.