Monoaural Speech Denoising: Obtaining clean speech from noisy environments via Source Separation

Héctor Martel
Tsinghua University
hkt20@mails.tsinghua.edu.cn

Samuel Pegg
Tsinghua University
peggsr10@mails.tsinghua.edu.cn

Toghrul Abbasli
Tsinghua University
abbaslit10@mails.tsinghua.edu.cn

Abstract

Real-world speech recordings are subject to degradation due to environmental noises, which can make the listening experience very unpleasant. In this report, we study different methods to obtain a clean speech signal by removing the background noise, also known as Speech Denoising. We specifically work on monoaural recordings. In contrast to traditional Speech Enhancement approaches, we model this problem as Source Separation problem because the nature of the background noise considered in this work is non-stationary and highly varied. Hence we consider our task as a separation of two sources, the clean speech and the background noise, from a single mixed signal. We evaluate the performance using objective metrics for source separation and speech intelligibility. We have found that working directly with the audio in time domain leads to better objective performance, but the results from the frequency domain were preferable to human listeners.

1. Introduction

The use of mobile communications and video conferencing technologies has recently experienced a very fast growth, with millions of users relying on them as their main communication method during the Covid-19 pandemic. Despite the advances in these technologies, the presence of background noises can make the user experience very unpleasant. Besides, there are a variety of speech processing systems that must be robust to background noises from different environments, such as Automatic Speech Recognition (ASR) [1]. We address some of these issues with the problem of Speech Denosing.

The main goal of Speech Denosing is to reduce or suppress the background noise to recover a clean speech signal, that is easier to understand for human listeners and automatic systems alike. Often times, denoising is a necessary preprocessing step to achieve other higher-level audio processing tasks. The basic assumption is that the background noise is additive, so the input mixture is the sum of speech and noise. Therefore, a good estimation of the background noise is sufficient to remove it from the mixture signal by subtraction. Alternatively, this process can be viewed as estimating a clean target signal directly.

The remaining of this paper is structured as follows: Previous research works are introduced in section 2. The intuition behind our approach, the theoretical background, and the model architectures are described in section 3. The evaluation metrics and datasets are covered in section 4. Objective evaluation results are presented in section 5. Finally, the conclusions of this work and some future directions are outlined in section 6.

2. Related work

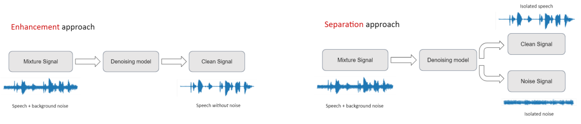
Speech Denoising has been a widely studied research problem in audio processing [1]. Before the dominance of Deep Learning, traditional signal processing methods relied on noise statistics obtained with prior information about its distribution. Therefore, only stationary noises could be suppressed effectively. Wiener filtering [10] minimizes the Mean Square Error (MSE) between the estimated noise and the signal, under the assumption of stationary and additive noise. Linear Time-Invariant (LTI) filters are employed to obtain the noise estimate. Spectral subtraction [2] refines this idea and obtains the noise profile from a segment without speech activity, to then subtract it from the mixture.

The introduction of Deep Learning techniques has provided significant performance improvements in traditional signal-processing tasks due to their ability to model non-linear relationships in the data without prior knowledge of the noise statistics. Examples of network architectures that have been applied successfully to noise suppression in audio and image processing are Denoising Autoencoders (DAE) [25, 7], Convolutional Neural Networks (CNN) [5, 6], or Recurrent Neural Networks (RNN) [8]. The encoder-decoder paradigm is typically used to recover the denoised estimate [1].

3. Methodology

We start this section by discussing two possible ways to tackle this problem, namely enhancement and source separation. In *enhancement*, the network receives the audio mixture as input and outputs only the clean signal, such that the noise has been removed from it. In contrast, *source separation* aims to isolate both the speech and the background noise. The objective of the model is to find a decomposition of the input mixture into these two audio sources. In fact, source separation can be applied not only to speech and background noise, but also to various speech signals from different speakers or musical instruments.

We apply a separation approach to this problem. The noises considered in our dataset present a high variability (see subsection 4.1) and some classes of background noise are rich enough to be considered an additional source, such as kids playing, dogs barking, or street music. This contrasts with the stationary noises typically assumed in speech enhancement applications.



(a) (b)

Figure 1. High-level comparison between Enhancement and Separation. (a) Enhancement. (b) Separation.

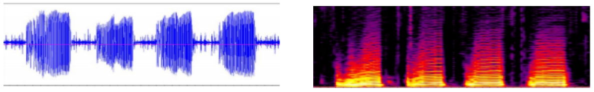
3.1. Audio Representations

Another important design consideration is the audio representation to be used. We distinguish between models that operate in time domain and frequency domain.

An audio signal is naturally represented as a value of air pressure or voltage that evolves over time, stored as a 1D array. This is called the **time domain** representation, or raw audio waveform. However, this representation is not very informative in terms of the signal contents and it constitutes a long sequence.

Another choice is the **frequency domain** representation. The frequency contents are represented along with the time dimension in a 2D array, where each value corresponds to the energy of a particular frequency bin at a particular time. Some common time-frequency transformations are the Short-Time Fourier Transform (STFT) or the MFCC (Mel Frequency Cepstral Coefficients).

In this work, we consider the time domain representation and the STFT magnitude for the frequency domain representation. Time and frequency domain representations are compared in Figure 2.



(a) (b)

Figure 2. Comparison of audio representations for the same example. (a) Time domain. (b) Frequency domain (STFT).

3.2. Masking

For frequency domain methods, it is usually hard for the model to estimate the magnitude STFT directly. Instead, the model estimates a set of multiplicative masks, one for each source, that are applied to the input mixture to obtain the source estimations. The interpretation of this masking operation is a segmentation map of the input mixture into the different sources. Each time-frequency bin is given a gain that indicates whether it should be kept or not, and in what proportion. This type of mask is called a Soft Mask, or Ratio Mask [12].

Let $\hat{M}_i \in [0.0, 1.0]^{T \times F}$ denote the mask of the ith source, and $|X| \in \mathbb{R}^{T \times F}$ denote the input magnitude spectrogram. T and F are the bins in time and frequency respectively. The estimation for the source $\hat{S}_i$ can be obtained as expressed in Equation 1, where the $\odot$ operator corresponds to the Hadamard product (element-wise multiplication).

$$\hat{S}_i = \hat{M}_i \odot |X| \tag{1}$$

To impose that the mixture can be reconstructed from the source estimates, the mask values in one particular location must sum to 1 across all sources. Therefore, the masks represent a probability distribution. Let $J \in [1.0]^{T \times F}$ be a matrix where all values are 1.0 and let N be the total number of sources (for our application $N = 2$). The reconstruction constraint is expressed in Equation 2.

$$J = \sum_{i=1}^{N} \hat{M}_i \implies |X| \approx \sum_{i=1}^{N} \hat{S}_i = \sum_{i=1}^{N} \hat{M}_i \odot |X| \tag{2}$$

We integrate the masking operation in the models by applying a Softmax activation after the last layer and then multiplying by the input magnitude $|X|$ channel-wise to obtain a reconstruction $\hat{S}_i$ of each source S_i. The optimization is performed with the Mean Squared Error (MSE) of the estimated source magnitudes directly: $MSE(S_i, \hat{S}_i)$. To recover the time domain signal for each source, the estimated source magnitude is combined with the original mixture phase. The entire pipeline for frequency domain methods is shown in Equation 3. The absolute value $|.|$ denotes magnitude and the angle symbol $\angle$ denotes phase.

$$
\begin{aligned}
X &= STFT(mixture) \\
\hat{M} &= Softmax(Network(|X|)) \\
\hat{S}_i &= \hat{M}_i \odot |X| \\
source_i &= iSTFT(\hat{S}_i \angle X)
\end{aligned}
\tag{3}
$$

In our task, learning a mask translates the problem to a (binary) classification problem, where the two classes are the clean speech class and the background noise class. Entries of each mask represent the proportion of a pixel to assign to one of the classes. Not using masks is essentially an image generation task which generates either just the clean spectrogram, or both spectrograms. Image generation is a particularly complex task, and the results are often inferior as it is prone to introducing artifacts, noise and other undesirable traits to the output spectrograms due to errors in the generation.

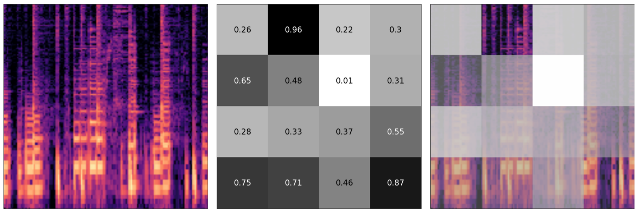


Figure 3. Example mask multiplication.

Figure 3 shows an example of mask multiplication. To make things more visual, the mask is an upscaled 4x4 mask, but in reality the dimensions of the mask would match the dimensions of the spectrogram. In this example, a hypothetical model learns a *clean mask* and a *noisy mask*; suppose without loss of generality that the mask seen in the centre of Figure 3 is the *clean mask*, denoted M_{clean}. The input mixed audio spectrogram on the left is multiplied by the mask, to produce the estimated clean audio spectrogram on the right. This hypothetical model classifies the top right corner as mostly noise, and hence the values in the top right corner of M_{clean} are closer to 0. In contrast, the model classifies the bottom row as mostly clean audio, and hence the values in this row are closer to 1. For completeness, the *noisy mask* in this case would be $M_{noisy} = J - M_{clean}$ where $J \in [1.0]^{T \times F}$ is as defined before Equation 2.

3.3. Frequency domain models

UNet [16] is a Convolutional Neural Network (CNN) architecture developed for biomedical image segmentation. Instead of a biomedical images, we re-purpose this network and pass it the STFT magnitude spectrograms. The UNet architecture uses a multi-layered design as seen in Figure 4 that contains a downsampling path and an upsampling path.

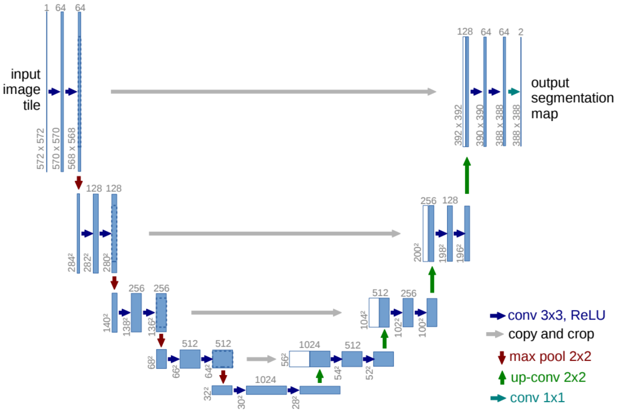


Figure 4. UNet architecture diagram. Extracted from [16].

In order to localize, the upsampling path takes the high resolution features from the downsampling path and combines them with the upscaled output. A successive convolution layer can then learn to assemble a more precise output based on this information. When upsampling, a large number of feature channels are utilized, this allows the network to propagate context information to higher resolution layers. As a consequence, the network yields the u-shaped architecture seen in the figure. However, a slightly modified version of UNet was used in our training. Inspired by ResNet [9], we sum the residual connections instead of concatenating. Since this reduces the number of filters of the convolutions, the model size also decreases. Hence we can feasibly train a deeper UNet architecture than in the original paper. The network depth is set to 6 downsampling/upsampling blocks and the base number of convolution filters is set to 16, which are doubled after each block.

TransUNet [4], as suggested by the name, proposes a transformer architecture for image segmentation. The structure of the network is essentially the same as UNet's, but the bottleneck layer of the u-shape involves a feature extraction CNN and a transformer block, rather than only convolutions. The transformer block is made of several stacked transformer layers, where each transformer layer consists of a stacked MSA and MLP layer, with a skip connection. The architecture diagram is shown in Figure 5.

Due to hardware limitations, and to make this model comparable in size to UNet, we reduce its capacity by using 6 transformer blocks instead of 12, set the multi-head attention to 4 heads, and the number of features in the MSA block to 128, in contrast to the 1024 from the original implementation.

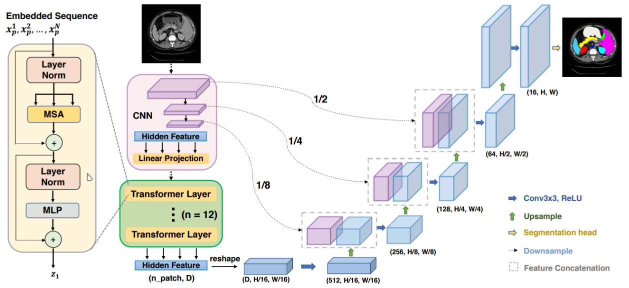


Figure 5. TransUNet architecture diagram. For simplicity, only the upsampling path is displayed. Extracted from [4].

3.4. Time domain models

UNetDNP [13] is an adaptation of WavUNet [21], which itself is an adaptation of UNet to work with the audio signal directly, rather than working with spectrograms. UNetDNP replaces the 2D convolutions for images with 1D convolutions, and utilizes batch normalization instead of double convolutions on each block. The model receives a single-channel input and returns a two-channel output. One channel is the clean estimation, and the other is the background noise estimation.

ConvTasNet [11] has a different architecture to UNet, as is clear from Figure 6. It uses an encoder-decoder structure to learn a representation of the audio signal in base signals. To put this another way, it learns its own way of converting between time and frequency domains internally using the *encoder* and *decoder* blocks. After encoding, the time domain signal passes through several sets of convolutional layers in the *separator* block. The output of each convolutional layer is fed into the next layer, and also saved separately. The saved outputs are then aggregated with the output of the last convolution, combined with the inputs and sent to the *decoder* to retrieve the separated audio sources.

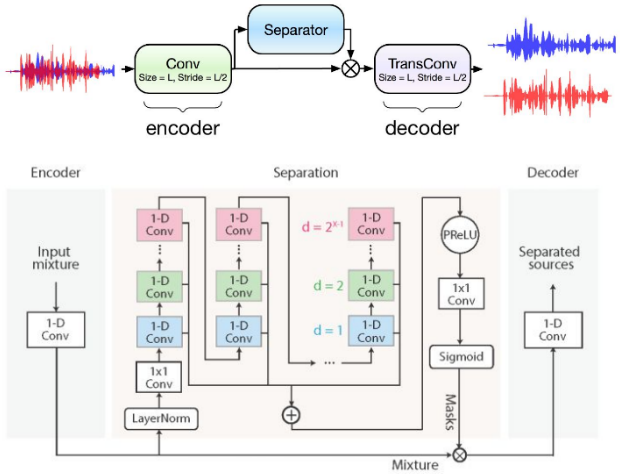


Figure 6. ConvTasNet architecture diagram. Extracted from [11].

4. Evaluation

In this section we introduce the datasets and objective evaluation metrics. In subsection 4.1 we cover two audio datasets commonly used for other tasks and how to repurpose them for our application. We do so by generating synthetic audio mixtures. The Source Separation metrics are presented in subsection 4.2 and the Speech Quality metrics in subsection 4.3.

4.1. Datasets

LibriSpeech [14] is an audio corpus originally proposed to train Automatic Speech Recognition (ASR) models, by accompanying the audios with their transcriptions. The contents are Audiobooks in English, that are balanced in terms of speaker gender and minutes per speaker. The sampling rate of all the recordings is 16kHz. For our experiments, we consider the predefined split of 100 hours of audio for training and 5.4 hours for testing. On the official website[1], the training set is called *train-clean-100* and the test set is called *test-clean*.

UrbanSound8K [18] has been proposed for environmental sound classification, focusing on sounds recorded in urban areas. The dataset contains a total of 8,732 sounds that last for approximately 4 seconds each, resulting in 8.75 hours of audio. The sounds are labeled using the following 10 classes: air conditioner, car horn, children playing, dog barking, drilling, engine idling, gun shots, jackhammering, sirens and street music. Originally, the dataset is designed for sound classification, providing 10 folds for cross-validation. We use the first 9 folds for training (from *fold1* to *fold9*), and the remaining one (*fold10*) for testing.

Data generation: The noisy examples are generated by combining the audios from LibriSpeech and UrbanSound8K using the following procedure.

- First, all the files are sliced into 4 second segments to ensure that clean speech and noise have the same length. The motivation behind this segment length is that it approximately matches the length of the noises, and that the generated examples are long enough to capture the long-term temporal structure of speech and noise.
- Then, the sampling rate in both sets is adjusted to 16kHz. This value is acceptable for speech applications while it significantly reduces the computational requirements from the original 44.1kHz or 48kHz of professional audio recordings.
- As a final step, each speech segment is mixed with one noise segment from the noise dataset to obtain the noisy mixtures. This mixing operation is performed using a specified value of Signal-to-Noise-Ratio (SNR), sampled from a uniform distribution in the range $(0, 18)$ dB, such

[1] LibriSpeech ASR corpus: `https://www.openslr.org/12`

that the level of the clean speech is the same as the noise or higher [26].

With the default train and test splits of LibriSpeech, this process leads to 103,079 examples for training and 6,102 for evaluation.

4.2. Source separation metrics

SDR, SIR and SAR [24] have been proposed as blind source separation objective evaluation metrics, that consider different aspects. The **Source-to-Distortion Ratio (SDR)** can be viewed as a global measurement of the separation quality, by assessing the amount of distortion present in the audio source estimations, with reference to their ground truth. Here, the distortion is defined as the mixture of interference, noise, and artifact error terms. Hence the relative measure between target signal and distortion is defined as:

$$SDR := 10\log_{10}\frac{||s_{target}||^2}{||e_{interf}+e_{noise}+e_{artif}||^2}$$

which is the log ratio (dB) of the target and distortion energies. In the expression, e denotes a component of the distortion. e_{interf}, e_{noise}, and e_{artif} denote the errors caused by *interference*, *noise*, and *artifacts* respectively.

The **Source-to-Interference Ratio (SIR)** is defined as:

$$SIR := 10\log_{10}\frac{||s_{target}||^2}{||e_{interf}||^2}$$

The **Source-to-Artifacts Ratio (SAR)** as:

$$SAR := 10\log_{10}\frac{||s_{target}+e_{interf}+e_{noise}||^2}{||e_{artif}||^2}$$

Scale Invariance (SI) [17] was introduced to the above source separation metrics to alleviate scaling issues. The authors proposed a method to obtain a stable and fair evaluation metric by ensuring that the residual is orthogonal to the ground truth signal. A method to achieve this is rescaling the ground truth signal to the orthogonal projection of the estimated signal on the ground truth.

After that, the rescaled signal is used to calculate SDR and since it is scale invariant, it is called Scale Invariant SDR (SI-SDR) [17]. SI-SDR is widely considered an upgrade to SDR as multiple failure cases for SDR are solved with SI-SDR.

We calculate the Scale Invariant (SI) versions of SDR, SIR and SAR using the python implementation proposed by `bsseval`[2].

4.3. Speech quality metrics

Perceptual Evaluation of Speech Quality (PESQ) [15] is often used for the evaluation, selection and optimization of codec and speech processing systems. The main purpose of PESQ is to emulate human auditory perception mechanisms to obtain an objective measurement that represents the quality of a speech signal to human listeners.

The evaluation method starts with the time alignment of clean and estimated signals. This identifies any delays and two types of filters can be used. A **narrowband** filter can be applied to both signals to emphasize perceptually important parts of the frequency range, and a **wideband** filter can be applied to emphasize the entire frequency range. We denote the metrics using these filters as PESQ-nb (narrowband) and PESQ-wb (wideband) respectively.

We calculate PESQ-wb and PESQ-nb based on the python package `python-pesq`[3].

Short-Time Objective Intelligibility (STOI) [22] is an objective measure for the intelligibility of both clean and estimated signals, which considers human hearing perception. This method gave higher performance than other reference Objective Intelligibility Measures (OIM), while being noticeably more lightweight. The model is based on an intermediate intelligibility measure for short-time Time and Frequency (TF) regions ($\approx$ 400 ms) and uses a simple Discrete Fourier Transform (DFT) based TF-decomposition. STOI score ranges from 0 to 1 and a higher value indicates better speech intelligibility [6]. The overall computational process is shown in Figure 7.

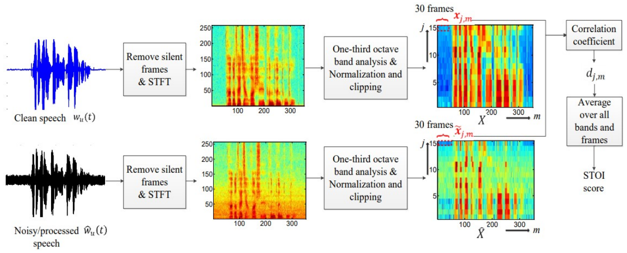

Clean speech $w_u(t)$
Remove silent frames & STFT
One-third octave band analysis & Normalization and clipping
30 frames
$x_{j,m}$
j
X
m
Correlation coefficient
$d_{j,m}$
Average over all bands and frames
STOI score
Noisy/processed speech $\hat{w}_u(t)$
Remove silent frames & STFT
One-third octave band analysis & Normalization and clipping
30 frames
$\tilde{x}_{j,m}$
$\hat{X}$
250
200
150
100
50
100
200
300
15
10
5

Figure 7. Calculation of STOI is based on the correlation coefficient between the temporal envelopes of the ground truth and the model estimation for short segments. Extracted from [6].

In our experiments, STOI is calculated based on the python package `pystoi`[4].

5. Experimental results

In this section we present experimental results on the synthetic mixtures generated by combining the LibriSpeech

[2] bsseval: Issue #3 Add SI-SDR, on GitHub: `https://github.com/sigsep/bsseval/issues/3`

[3] PESQ (Perceptual Evaluation of Speech Quality) Wrapper for Python Users: `https://github.com/ludlows/python-pesq`

[4] STOI Python package: `https://github.com/mpariente/pystoi`

Model	Domain	SI-SDR ↑	SI-SIR ↑	SI-SAR ↑	PESQ-wb ↑	PESQ-nb ↑	STOI ↑
UNet	Frequency	-26.99	18.11	-26.98	1.29	1.74	0.75
TransUNet	Frequency	-27.83	12.24	-27.81	1.42	1.90	0.75
UNetDNP	Time	7.61	16.90	8.55	1.56	2.07	0.82
ConvTasNet	Time	**10.69**	**22.19**	**11.30**	**1.83**	**2.38**	**0.85**

Table 1. Objective evaluation results for all models. Values represent the mean across all evaluation examples. For all metrics, higher values indicate better performance.

test-clean data with the *fold10* of UrbanSound8K. The same set of mixtures is used to evaluate all the models.

We perform the training for 10 epochs on the 100 hours of synthetic mixtures described in subsection 4.1 on a single GTX 1080Ti GPU. We use Adam optimizer, a learning rate of 1×10^{-4} and a batch size of 16. Gradients larger than 1.0 are clipped to avoid large weight updates in a single step. The other optimizer hyperparameters are left as default. For *frequency domain models*, we employ the MSE loss on the magnitude spectrograms of the source estimations after masking. For *time domain models*, the optimization objective is the SI-SDR metric which directly uses the raw waveform, as in the original ConvTasNet paper [11]. A checkpoint of the models is saved only when the validation loss improves.

The objective evaluation results are presented in Table 1, containing the Source Separation metrics and Speech Quality metrics presented in section 4. Discussions and in-depth analysis can be found in subsection 5.1.

5.1. Experiment discussion

Our experiments show that the highest performance is obtained by time domain models, which outperform the frequency domain models in all the objective evaluation metrics.

The performance gap is particularly noticeable in the **source separation metrics**. The SI-SDR values for ConvTasNet and UNetDNP were 10.69 dB and 7.61 dB respectively, whereas their frequency domain counterparts yield negative values. The interpretation of a negative value here is that the contribution of the distortion is greater than the energy of the sources themselves. In other words, there is a more distortion than actual correct estimates. Even though the SI-SDR is negative, the SI-SIR gives relatively high, positive values. This indicates that there is not a significant leakage between the sources, i.e. the noisy and clean sources are well separated.

The distortion mentioned above is mainly comprised of artifacts, as reflected in the SI-SAR metric. A more careful inspection of the results show that the distortion is concentrated in the noise estimation, which suffers significantly from the time-frequency transforms and reconstruction using the original mixture phase. Another consideration is that the training objectives are different for time and frequency domain models. In the first case, the SI-SDR is used directly in the loss function. In the second, the loss function is the Mean Square Error (MSE) of the magnitude spectrograms. This fundamental difference can also explain the disparity in performance of the two types of methods, since the time domain models have the same loss function as the evaluation metric.

The **speech quality metrics** are actually more representative of the models' performance. It is the quality of the *speech estimation* that is the final result, regardless of the quality of the *noise estimation*. For our task, the quality of the noise estimation is fundamentally irrelevant.

In Table 1, the values obtained by the speech quality metrics, PESQ and STOI, also support the conclusion that the time domain methods perform better in this task, at least from an objective point of view.

However, by listening to the output speech estimations of the models, we have found that these **objective evaluation metrics do not fully correlate with the listener's preference** [12]. For example, the estimated clean speech signals from ConvTasNet present a noticeable noise residual in the higher frequencies (crackling/static) that is not being reflected in the objective speech quality metrics. This noise is located in a region of the audible spectrum that does not affect the speech intelligibility, but that is unpleasant to listen to. Under the presence of oscillating noises, such as engines, the amplitude of the estimated clean speech signal is modulated to match the background noise and is not pleasant to human listeners either. To summarize, the noise suppression by ConvTasNet is numerically the best, but a human listener might disagree.

In contrast, UNet and TransUNet tend to over-attenuate some of the frequencies of the speaker to achieve a higher degree of noise suppression. This means that the fine details in the high frequencies are lost, but subjectively the audios result in a more natural listening experience with less apparent background noise.

6. Conclusions

In this report we have presented the problem of Speech Denoising and studied the effectiveness of various models using a Source Separation approach, as opposed to the Speech Enhancement approach that is usually taken. In our experiments, we optimize the models on both the estimated clean audio source **and** the estimated background noise audio source. This means we train in a much stricter environment than the traditional approach since we force the models find a more robust solution, and additionally model the noisy acoustic environment. However, when analyzing the experimental results we prioritize the original goal of Speech Denoising and give more importance to the speech quality metrics. Finally, we also inspected the outputs manually to extract further conclusions on the subjective performance.

From subsection 5.1, we concluded that ConvTasNet performed best in terms of numerical evaluation metrics, but did not sound that good to a human listener. In contrast, TransUNet gave results that performed poorly numerically, but sounded very good to a human listener.

6.1. Future work

To finalize this discussion, we comment on certain directions that would be worth considering if research were to continue on this topic.

Subjective tests: To confirm the conclusions extracted from the previous experiment discussion, it would be necessary to perform more rigorous subjective quality tests. A MUSHRA experiment can be designed [3, 19] for trained human listeners to provide subjective feedback on the denoising performance. The time and budget required to carry out such an experiment with a sufficiently large number of users is well beyond the scope of this project. Nevertheless, subjective tests should not be disregarded, as our target is to improve the experience for all human listeners.

Joint audio representation: Here we have studied models that operate in either time domain or frequency domain. However, some recent work has been done in the context of Speech Enhancement that jointly learns the two representations simultaneously, such as TFT-Net [23]. We believe that this direction could be beneficial and improve the quality of the results in our problem, adapting the proposed TFT-Net to our Source Separation setting.

Hyperparameter and architecture search: As explained in section 3, we have modified various parameters such as the depth and the number of convolutional filters, which differs from the baselines of the original papers. In particular, we reduce the capacity of the biggest models. Our motivations were the following: 1) match the capacity of the four models by constructing them with a similar number of parameters, and 2) reduce the computational costs of training and inference without significantly degrading the output audio quality. We believe, however, that further improvements could be achieved in terms of audio quality and efficiency by means of a more exhaustive architecture search. Also, training strategies like learning rate annealing [20] could be explored to converge to more optimal solutions.

7. Reproducibility

We have made our code available on GitHub[5]. It includes the environment dependencies, automatic downloads for the datasets, and example command line inputs for the training and evaluation of all the models in PyTorch. We hope that it can be useful for the open source and audio research community.

References

[1] A. Azarang and N. Kehtarnavaz. A review of multi-objective deep learning speech denoising methods. *Speech Communication*, 122:1–10, 2020.

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

[3] M. Cartwright, B. Pardo, G. J. Mysore, and M. Hoffman. Fast and easy crowdsourced perceptual audio evaluation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 619–623, 2016.

[4] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou. Transunet: Transformers make strong encoders for medical image segmentation, 2021.

[5] S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai. Raw waveform-based speech enhancement by fully convolutional networks, 2017.

[6] S.-W. Fu, T.-W. Wang, Y. Tsao, X. Lu, and H. Kawai. End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks, 2018.

[7] E. M. Grais and M. D. Plumbley. Single channel audio source separation using convolutional denoising autoencoders, 2017.

[8] A. Graves, A. rahman Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks, 2013.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[10] J. Lim and A. Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67(12):1586–1604, 1979.

[11] Y. Luo and N. Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, Aug. 2019.

[12] E. Manilow, P. Seetharman, and J. Salamon. *Open Source Tools & Data for Music Source Separation*. https://source-separation.github.io/tutorial, Oct. 2020.

[13] M. Michelashvili and L. Wolf. Speech denoising by accumulating per-frequency modeling fluctuations, 2020.

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[5] Speech Denoising, available on GitHub: `https://github.com/hmartelb/speech-denoising`

[15] A. Rix, J. Beerends, M. Hollier, and A. Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 2, pages 749–752 vol.2, 2001.

[16] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[17] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey. Sdr - half-baked or well done?, 2018.

[18] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 1041–1044, New York, NY, USA, 2014. Association for Computing Machinery.

[19] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre. webMUSHRA — A Comprehensive Framework for Web-based Listening Tests. *Journal of Open Research Software*, 6, Jan. 2018. Publisher: Ubiquity Press, Ltd.

[20] L. N. Smith. Cyclical learning rates for training neural networks, 2017.

[21] D. Stoller, S. Ewert, and S. Dixon. Wave-u-net: A multiscale neural network for end-to-end audio source separation, 2018.

[22] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. 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.

[23] C. Tang, C. Luo, Z. Zhao, W. Xie, and W. Zeng. Joint time-frequency and time domain learning for speech enhancement. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3816–3822. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[24] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, Dec. 2010.

[26] D. Wang and J. Chen. Supervised speech separation based on deep learning: An overview, 2018.