SFSRNet: Super-resolution for Single-Channel Audio Source Separation

Joel Rixen, Matthias Renz

Department of Computer Science, Kiel University, Germany
joelrixen@gmail.com, mr@informatik.uni-kiel.de

Abstract

The problem of single-channel audio source separation is to recover (separate) multiple audio sources that are mixed in a single-channel audio signal (e.g. people talking over each other). Some of the best performing single-channel source separation methods utilize downsampling to either make the separation process faster or make the neural networks bigger and increase accuracy. The problem concerning downsampling is that it usually results in information loss. In this paper, we tackle this problem by introducing SFSRNet which contains a super-resolution (SR) network. The SR network is trained to reconstruct the missing information in the upper frequencies of the audio signal by operating on the spectrograms of the output audio source estimations and the input audio mixture. Any separation method where the length of the sequence is a bottleneck in speed and memory can be made faster or more accurate by using the SR network.

Based on the WSJ0-2mix benchmark where estimations of the audio signal of two speakers need to be extracted from the mixture, in our experiments our proposed SFSRNet reaches a scale-invariant signal-to-noise-ratio improvement (SI-SNRi) of 24.0 dB outperforming the state-of-the-art solution SepFormer which reaches an SI-SNRi of 22.3 dB.

Introduction

In real-world environments, audio often contains parts where multiple speakers talk over each other. This is known as the cocktail party problem (Bronkhorst 2000; Haykin and Chen 2005). Being able to accurately separate multiple speakers from a single-channel mixture is of interest to a number of speech processing tasks (Narayanan and Wang 2014). One example of such a task is automatic speech recognition (ASR). If the input audio consists of multiple people speaking over each other, ASR methods typically perform significantly worse (Lam et al. 2019; Luo, Chen, and Yoshioka 2020). Therefore, to improve the accuracy of ASR methods, it is advisable to separate the individual speakers in mixed audio signals before applying ASR methods on each individual speaker audio signal.

Problem Definition

The problem of single-channel audio source separation is to separate (recover) the C audio sources $s_1, \ldots, s_c$ that overlay in a given audio mixture x, s.t.

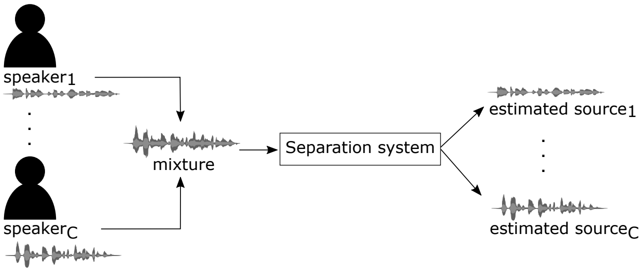

Figure 1: Single-channel audio source separation.

$$\vec{x} = \sum_{i=1}^{C} \vec{s_i} \tag{1}$$

where the mixture and sources can be expressed as vectors $\vec{x} \in \mathbb{R}^{D\times 1}$ and $\vec{s_i} \in \mathbb{R}^{D\times 1}$, respectively, with D being the sequence length. Figure 1 illustrates the problem. The audio source separation system takes the given audio mixture as its input and outputs C estimations of the audio sources. Note, in the reminder we will call the estimations of the recovered audio sources simply *estimations*.

Basic Idea

Latest solutions of audio separation systems have suggested to downsample the input before the audio source separation (Tzinis, Wang, and Smaragdis 2020; Subakan et al. 2021). Downsampling has the advantages of speeding up and lowering the memory usage of the separation process. However, the issue with downsampling the input signal is that the later required upsampling process to get the original audio signal frequencies is unable to fully restore the information that gets lost during the downsampling process. To solve this problem, we propose a separate super-resolution (SR) network to achieve better upsampling results. The proposed SR network is different from common upsampling tasks as follows. Since the goal of the upsampling in this case is to return to the original audio signal frequency, it is not necessary to generate any new information. Instead, aside from the downsampled estimations, the input audio mixture in its original sampling rate is used as an additional input to improve the upsampling process.

The state-of-the-art approach proposed in (Subakan et al. 2021) applies downsampling by a factor of 8 suggesting that the separation can be sped up while staying highly accurate even without the full resolution audio. In combination with an SR network, the separation can be made even more accurate, which makes SR a natural fit for source separation. Let us note that our SR network can be added to most existing separation methods to speed them up and make them more accurate.

Increasing the downsampling factor will speed up the separation, while adding the SR network will compensate the loss in accuracy that would occur without SR. Since the proposed SR network only consists of a few convolutional layers, it is highly parallelized and its computational cost is fairly minor compared to the overall cost of the separation. In addition, we can increase accuracy by adding more layers to the separation network to improve the separation since the downsampling process does not only save speed, but also memory.

Main Contributions

In this paper we propose the SFSRNet approach containing a super-resolution (SR) network to address the single-channel audio source separation problem. Our approach adopts the downsampling usage of the existing SepFormer architecture (Subakan et al. 2021). Our main contributions are

1. Improving the existing SepFormer architecture by calculating intermediate estimations after each block and varying the resolution of the sources these estimations are compared to.
2. Introducing the super-resolution (SR) network which can be used to improve most existing separation architectures.
3. Experimental evaluation of our SFSRNet showing significant improvement over state-of-the-art based on the WSJ0-2mix benchmark with 24.0 dB SI-SNRi, on the clean Libri2Mix with 21.7 dB SI-SNRi and on the noisy Libri2Mix with 16.4 dB SI-SNRi.

Outline

The remainder of the paper is organized as follows: First we will briefly summarize *related work* followed by the *dual-path model* section introducing the basic architecture, which most recent source separation models, including the SFSRNet, are based on. Section *SFSRNet Model* goes more in-depth on our proposed approach introducing the new separation architecture in detail and our super resolution network. In the *experiments* section we empirically evaluate the performance of SFSRNet in detail and show how the SR network can also be used in combination with other architectures to speed them up. Finally, we conclude the paper by summarizing the advances and limitations of our SFSRNet.

Related Work

The task of single-channel source separation has seen a lot of progress, recently, using deep learning techniques. Early neural network based source separation systems (Wang and Chen 2018; Hershey et al. 2016; Bahmaninezhad et al. 2019) use the short-time Fourier transform (STFT) of the mixture, take the magnitude as the input of the neural network and calculate a mask for each source. This mask is then multiplied with the magnitude of the mixture and the resulting magnitude combined with the phase of the mixture is brought back into the time-domain using the inverse STFT (iSTFT).

Later, it is shown that better results can be achieved when staying in the time-domain by replacing the STFT and iSTFT steps with a convolutional encoder and decoder (Bahmaninezhad et al. 2019; Luo and Mesgarani 2018). The neural network is able to operate on the waveform directly which means that magnitude and phase information are no longer decoupled.

One of the main challenges of audio source separation is that the sequences the neural network needs to process are very long. The dual-path recurrent neural network (DPRNN) as proposed in (Luo, Chen, and Yoshioka 2020) turns the sequence into overlapping chunks and treats both the neighbouring samples inside the chunks and the neighbouring chunks themselves, as two sequences. Since both these sequences are much shorter than the original sequence, it allows for a more extensive neural network for the separation task.

More recently it has been shown that Transformers (Vaswani et al. 2017) instead of recurrent neural networks (RNN) or temporal convolutional networks (TCN) (Lea et al. 2016) achieve the best separation results (Subakan et al. 2021). Unlike RNNs, Transformers consume considerably more memory when sequence length is increased. This is why most of these approaches either try to limit the number of Transformers by also using RNNs (Chen, Mao, and Liu 2020; Lam et al. 2021), or they downsample the input (Subakan et al. 2021; Lam et al. 2021). In our approach, we build on the recently introduced downsampling solution while addressing its problem of information loss.

Dual-Path Model

The general concept of the dual-path model shown in Figure 2 is based on the TasNet (Luo and Mesgarani 2018). Note that we only show one chunking and one overlapping step while some dual-path models use two chunking and overlap steps (Luo, Chen, and Yoshioka 2020; Lam et al. 2021). The entire separation process remains in the time-domain, in contrast to former STFT solutions. A convolutional layer is used to encode the mixture and later decode the estimations. The TasNet as well as the dual-path model is based on mask estimation.

$$\vec{est}_i = Decoder(\vec{m}_i \odot \vec{enc}) \tag{2}$$

For each source s_i a mask $\vec{m}_i \in \mathbb{R}^{D\times N}$ is calculated and multiplied with the encoded mixture $\vec{enc} \in \mathbb{R}^{D\times N}$ with N being the channel size of the encoder. The Decoder processes the result to return the waveform $\vec{est}_i \in \mathbb{R}^{D\times 1}$.

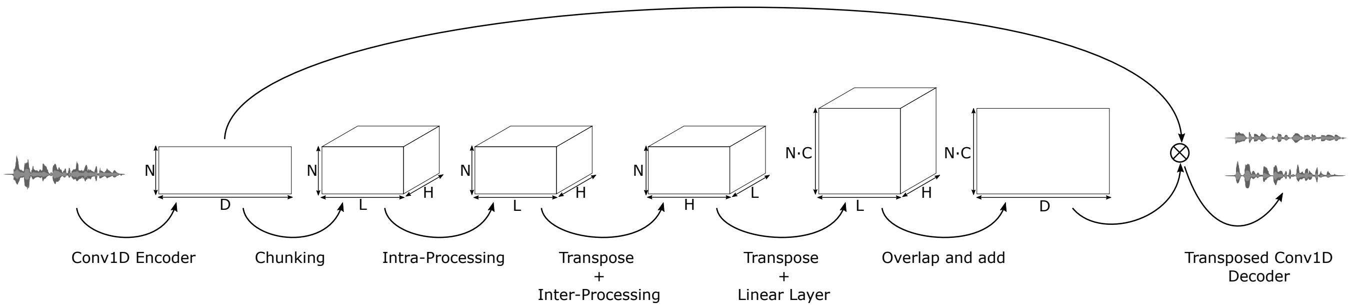

Figure 2: Basic dual-path model. With sequence length D, encoder filter size N, chunk size L, number of chunks H and number of sources C.

The idea is that the encoder expands the mixture in a way which allows the masks to better estimate, how involved each source is at each timestep.

The Dual-Path separation approach was first proposed in (Luo, Chen, and Yoshioka 2020). The main idea of this approach is to split the sequence into overlapping chunks, thus effectively turning one long sequence into two shorter sequences and then to operate on the two shorter sequences. This is useful as operating on long sequences is usually very resource intensive, both in time and memory.

Encoder

The encoder takes the mixture of multiple speakers as its input and is usually a convolutional layer. The encoder adds a second dimension to the one dimensional sequence. The idea behind this step, initially introduced in TasNet (Luo and Mesgarani 2018), is to mimic a similar function as the STFT.

Chunking

The chunking is one of the core contributions of the DPRNN (Luo, Chen, and Yoshioka 2020) in its application for source separation. The basic idea is to split up one long sequence into a number of shorter sequences and then stack them on top of each other. It should be noted, that these shorter sequences, or chunks, overlap each other. If the chunks would not overlap, then contextual information of the sequence between each chunk would get lost. Since the chunks overlap, the total size of the tensor actually increases due to the overlap.

Mask Estimation

The mask estimation process includes all the steps after the chunking and before the decoder as illustrated in Figure 2. The idea of the chunking step is to create two sequences. First, there is the sequence of neighboring samples inside each chunk, which were also neighboring samples in the original sequence. Working on this sequence is called intra-processing. Second, there is the sequence of the chunks themselves. In reference to the original sequence, neighboring values here have a space of the chunk size minus the overlapping bits between them. Working on this sequence is called inter-processing. The basic concept is that the intra-processing step captures local patterns, while the inter-processing step is able to capture long term patterns.

In the DPRNN paper (Luo, Chen and Yoshioka 2019) bidirectional RNNs were used for the intra- and inter-processing. However, as the dual-path approach was used in many other papers, other methods than the bidirectional RNNs have been tried. RNNs (Luo, Chen, and Yoshioka 2020), Transformers (Subakan et al. 2021), or both (Chen, Mao, and Liu 2020; Lam et al. 2021) have all been used in dual-path models for the intra- and inter-processing. Basically, any method suitable for capturing sequential patterns could be used for intra- and inter-processing.

The intra- and inter-processing steps usually repeat themselves in an alternating pattern.

After all the intra- and inter-processing blocks are done, the encoder dimension size is increased by C using a linear layer. To reverse the chunking step, the chunks are sequentially assembled with the overlapping parts being added to each other. With this, the original sequence length is reconstructed. Next, the tensor is split among the encoder dimension into C parts. These parts make up the mask estimation tensors for each source. They are then multiplied with the encoded mixture from the encoder step.

Decoder

The decoder fulfills the opposite function of the encoder. The dimension added through the encoder is removed for each source in order to return to a waveform which is usually done through a transposed convolutional layer.

SFSRNet Model

Our SFSRNet architecture is based on the dual-path model SepFormer (Subakan et al. 2021) using an encoder - mask estimator - decoder pipeline as introduced in the previous section. Instead of using RNNs, the SepFormer model uses Transformers based on Multi-Head Attention (MHA) (Vaswani et al. 2017) for the intra- and interprocessing. In the SepFormer model, downsampling is used during the encoding and upsampling during the decoding.

Figure 3 shows the differences between the SepFormer and SFSRNet architectures. The first difference is calculating intermediate estimations after each block of intra- and interprocessing and including these estimations for the loss calculation. The second difference is the additional step of SR.

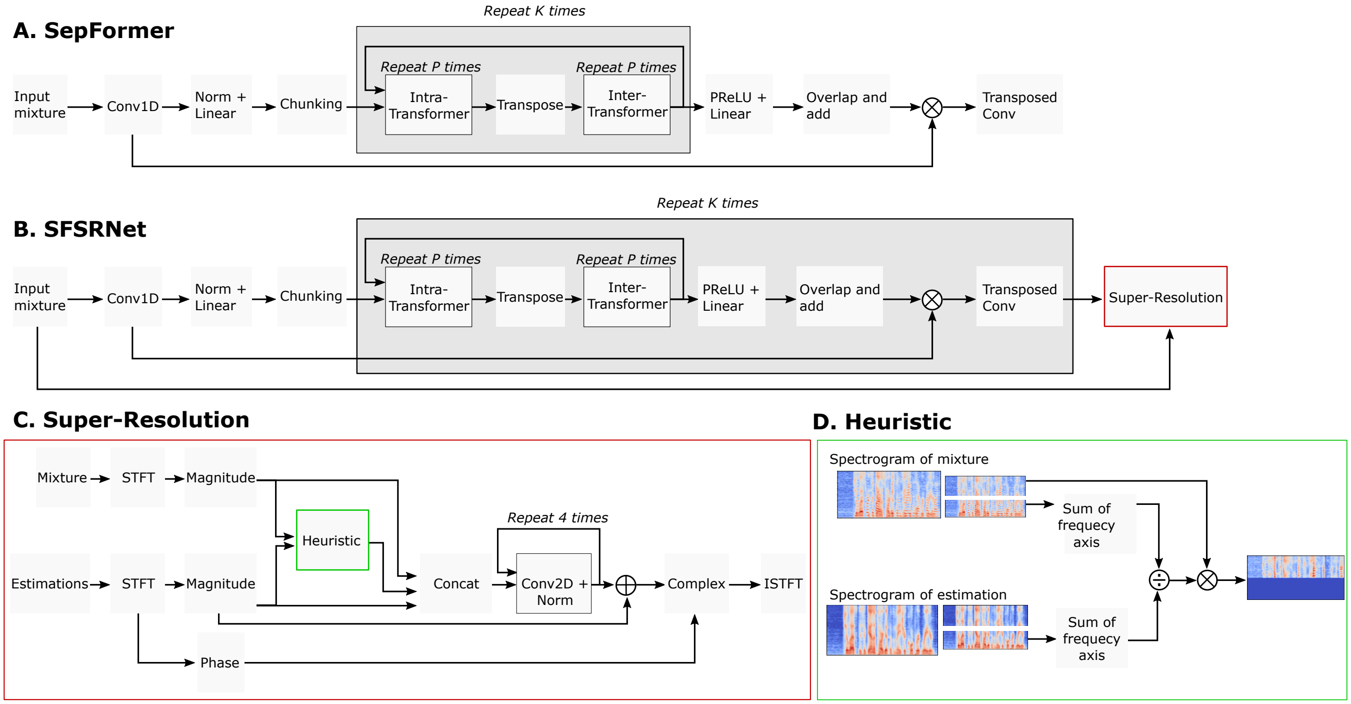

Figure 3: Comparison with previous SOTA (A) and the SFSRNet architecture (B).

Separation Blocks and Multi-Loss

After the encoding and the chunking, the separation process begins. Both Transformer blocks are repeated P times. The entire SepFormer block is repeated K times with the output of the last InterTransformer being the input for the next SepFormer block.

The IntraTransformer operates on the sequence of neighboring samples inside each chunk, while the InterTransformer operates on the sequence of neighboring chunks. The IntraTransformer and InterTransformer both have the same architecture. First, positional encoding is added to the input. The result is fed through a layer normalization and MHA with a residual connection at the end of it. Next is another layer normalization, a linear layer, and a two dimensional convolutional layer with another residual connection. Using a two dimensional convolutional layer instead of a linear layer is the first difference to the original SepFormer block. The second and more significant difference is the use of a multi-loss system, similar to what was used in (Nachmani, Adi, and Wolf 2020).

After each SepFormer block, intermittent results are calculated. This is done by increasing the channel size to match the encoded channel size multiplied by C by using a linear layer with a rectified linear activation, overlapping and adding the chunks, multiplying the resulting masks with the encoded representation and feeding it into the decoder. This is shown in Figure 3, as the process of calculating estimations only happens once in the original SepFormer, while it happens K times in the SFSRNet. Additionally, the sources these intermittent results are compared to are increasing in resolution, meaning the first output is compared to the original source at a low sampling rate, while the last output is compared to the original source at the full sampling rate.

Super-Resolution

The SR step is added after the decoder. Unlike most SR problems, there is no need to generate any new information. All the necessary information is contained within the mixture. Some of this information gets lost in the separation process since downsampling is used, however, this can be reversed by taking the mixture as an input for the SR process.

Unlike the separation, the SR process operates in the frequency-domain. Using the estimations for each source and the original mixture as the input, STFTs are calculated for each of them.

The phase information of the estimations are set aside for later, while the magnitude is operated on in order to restore the detail which was lost during the separation.

First, heuristics are used in an attempt to correct the magnitude of the higher frequencies. To calculate the corrected magnitude for each source, the magnitude spectrograms of the mixture and estimations are split into two parts, resulting in a matrix holding the low frequencies and a matrix holding the high frequencies. For both the low and high frequency matrices, all the frequency bins at each timestep are added together. This results in a sequence for the low and high frequencies. By dividing the low frequency sequence mixture by the low frequency sequence of each estimation, it can be estimated, which estimation is contributing to the mixture at each timestep. After dividing the two sequences, the next step is to take the resulting sequences and multiply them with the higher frequency matrix of the mixture. This is how the higher frequency of the corrected magnitudes of each estimation is calculated. The corrected magnitudes of the higher frequencies for each estimation consist of the combination of the lower frequency matrix of the estimation and the multiplication of the high frequency matrix of the mixture with the sequence previously calculated.

The idea is to compare the lower frequencies of the estimations with the lower frequencies of the mixture and figure out, how much each estimation is contributing. This information is then extrapolated to the high frequencies of the mixture, since it is likely that the amount a source is contributing to a mixture is similar in the low and high frequencies and we assume that the estimations are more accurate in the low frequencies due to downsampling.

The input to the SR network is four dimensional. The magnitude spectrograms of the mixture, the estimations and the corrected estimations through the heuristic are concatenated in the channel dimension. This serves as the input to the network. The network itself consists of four two dimensional convolutional layers with rectified linear activations and group normalizations between each of them. The last layer has a filter size equal to C. Next, this output is split for each estimation and added to the original magnitude spectrogram of the estimations. Using the new magnitude and the original phase information of the estimations, the STFTs of the estimations are recalculated and with the iSTFT, each estimation is returned to the time-domain.

Experiments

Datasets

We evaluated our system on the two-speaker speech separation problem using the WSJ0-2mix dataset (Hershey et al. 2016) which is based on the WSJ0 corpus (Garofolo, John S. et al. 1993). This dataset contains 30 hours of training, 10 hours of validation data and 5 hours of evaluation data. The speech mixtures are generated by selecting utterances from the 49 male and 51 female speakers in the Wall Street Journal (WSJ0) training set si_tr_s, and mixing them at random signal-to-noise ratios (SNR) between 0 dB and 5 dB. The 5 hour long evaluation set is generated in the same way, using utterances from 16 unseen speakers from si_dt_05 and si_et_05 in the WSJ0 dataset.

Aside from the WSJ02-Mix, the clean and noisy Libri2Mix datasets (Cosentino et al. 2020) are also used to evaluate the SFSRNet. The Libri2Mix datasets are based on the LibriSpeech ASR corpus (Panayotov et al. 2015). Similarly to to the WSJ02-Mix, the Libri2Mix datasets take two utterances and mix them together. In the noisy Libri2Mix dataset, background noise is added on top of the two utterances. The Libri2Mix datasets consist of 212 hours of training, 11 hours of validation and 11 hours of evaluation data. For all datasets we use the 8 kHz versions.

Data Augmentation

For data augmentation, we use dynamic mixing (DM) which was introduced in (Zeghidour and Grangier 2020). This method keeps randomly selecting utterances of the training set at runtime and mixes them at random SNRs between 0 dB and 5 dB, which is how the mixtures in the WSJ0-2Mix dataset were created. Like in (Subakan et al. 2021), we also change the speed of the utterances randomly between 95% and 105%. This helps with generalization. As was proposed in (Lam et al. 2021), we include utterances of the same speaker for dynamic mixing in order to improve separation accuracy of mixtures with similar voices.

Experiment Configurations

The encoder has a channel size of 256, a kernel size of 16 and a stride factor of 8. The chunk size is 50 with an overlap of 50%. We use $K = 8$ SepFormer blocks with $P = 2$ Intra- and InterTransformers each. The kernel size of the two dimensional convolutional layers of the Transformers is 3. The linear layer preceding the two dimensional convolutional layers has a 512 hidden units. The MHAs use 8 parallel heads.

For the SR, we use a frame length of 256 and a frame step of 64 for the STFTs. The convolutional layers have channel sizes of 128, 256, 128 and C, respectively. Their kernel sizes are 5, 9, 11 and 11, respectively. The group size of the group normalization is set to 1. The SR itself is trained separately from the separation process. The utterances are downsampled by the same amount as the encoder downsamples them for the separation and then the utterances are upsampled back to the original sampling rate using bilinear interpolation. Just like in the separation process, the utterances are added together to simulate a mixture as this mixture is used as an input for the SR process. The other utterance is slightly mixed into each utterance to simulate the outcome of the separation. To summarize, we use slightly noisy utterances and a mixture of the overlapping utterances as the input of the SR network training.

The training objective for both the separation and SR is scale-invariant signal-to-noise-ratio (SI-SNR) (Roux et al. 2018), which is defined as

$$\vec{g} := \frac{\langle \vec{est}, \vec{s} \rangle \vec{s}}{\|\vec{s}\|^2} \tag{3}$$

$$\text{SI-SNR} := 10\, log_{10} \frac{\|\vec{g}\|^2}{\left\|(\vec{est} - \vec{g})\right\|^2} \tag{4}$$

where $\vec{est} \in \mathbb{R}^{D \times 1}$ is the estimation of the source and $\vec{s} \in \mathbb{R}^{D \times 1}$ is the clean source. Both $\vec{est}$ and $\vec{s}$ are normalized to zero-mean in order to ensure scale-invariance. For the separation, utterance-level permutation invariant training (uPIT) (Yu et al. 2017; Kolbaek et al. 2017) is used to maximise the SI-SNR. For the SR, we also maximise SI-SNR, however uPIT is not necessary.

Since 8 SepFormer blocks are used, there are 8 estimations for which the losses are calculated for. The sources with which the estimations are compared to are downsampled for the first 6 blocks to 500 Hz, 1 kHz, 2 kHz, 3 kHz, 4 kHz and 5 kHz, respectively. The remaining estimations use the original 8 kHz sources. After these 8 uPIT SI-SNRs are calculated, the average of the 8 losses is calculated and added to the SR loss.

Method	**Model size**	**SI-SNRi (db)**	**SDRi (db)**
Deep Clustering (Hershey et al. 2016)	13.6M	10.8	–
Conv-TasNet (Luo and Mesgarani 2019)	5.1M	15.3	15.6
FurcaNeXt (Zhang et al. 2020)	51.4M	18.4	–
DPRNN (Luo, Chen, and Yoshioka 2020)	2.6M	18.8	19.0
Sandglass (Lam et al. 2021)	**2.3M**	21.0	21.2
Wavesplit (Zeghidour and Grangier 2020)	-	21.0	21.2
Wavesplit + DM (Zeghidour and Grangier 2020)	-	22.2	22.3
SepFormer (Subakan et al. 2021)	26M	20.4	20.5
SepFormer + DM (Subakan et al. 2021)	26M	22.3	22.4
SFSRNet	59M	22.0	22.1
SFSRNet + DM	59M	**24.0**	**24.1**

Table 1: Model size, SI-SDR and SDR improvements (dB) on WSJ0-2Mix dataset.

For the optimization, the Adam optimizer (Kingma and Ba 2017) is utilized with a learning rate of $15e^{-5}$. After the first 100 epochs, the learning rate is halved, once the performance on the validation dataset does not improve for 3 epochs. Gradient clipping is used with a maximum L_2-norm of 5. The network is trained for 200 epochs in total.

WSJ0-2mix Results

Table 1 compares the performance of different source separation systems on the WSJ0-2mix task. As shown, our SFSRNet outperforms the SOTA baseline.

Furthermore, Figure 4 shows the improvement in the higher frequencies when compared to the original SepFormer and the clean source. Even though the SepFormer is using a stride factor of 8, it manages to reconstruct the higher frequencies quite well. This is probably due to having a kernel size double that of the stride factor, effectively preserving the information of the higher frequencies in the channel dimension.

Method	Libri2mix			
	clean (db)		noisy (db)	
	SI-SNRi	SDRi	SI-SNRi	SDRi
Conv-TasNet (Cosentino et al. 2020)	14.7	–	12.0	–
IRM (oracle) (Cosentino et al. 2020)	12.9	–	12.0	–
IBM (oracle) (Cosentino et al. 2020)	13.7	–	12.6	–
Wavesplit (Zeghidour and Grangier 2020)	19.5	20.0	15.1	15.8
Wavesplit + DM (Zeghidour and Grangier 2020)	20.5	20.9	15.2	15.9
SFSRNet	20.4	20.7	15.6	16.1
SFSRNet + DM	**21.7**	**22.0**	**16.4**	**16.9**

Table 2: SI-SDR and SDR improvements (dB) on clean and noisy Libri2Mix.

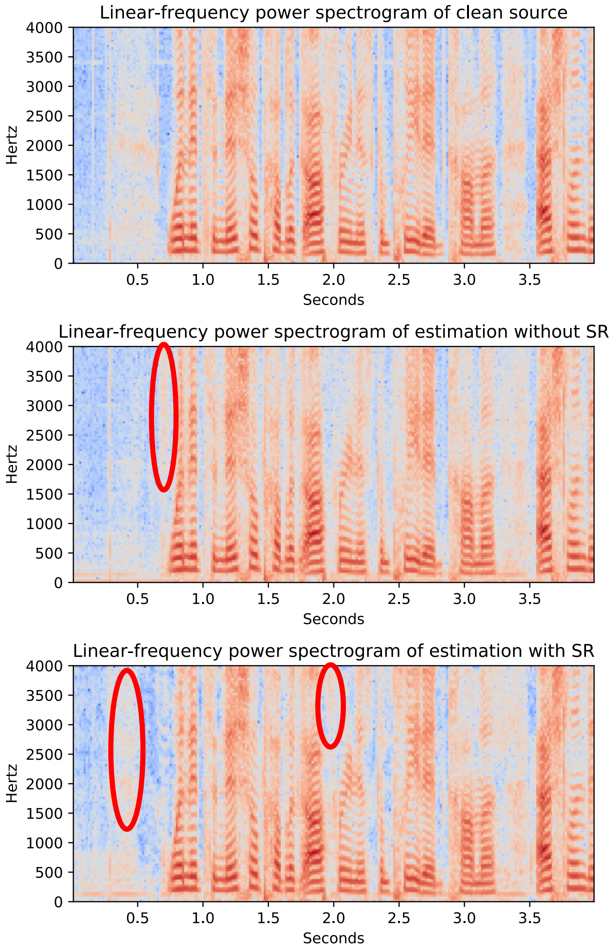

Figure 4: Top: Original, clean source spectrogram. Middle: Spectrogram of SepFormer output without SR with red ellipsis showing incorrect information in higher frequencies. Bottom: Spectrogram of SepFormer output with SR with red ellipses showing information that was added by SR.

That being said, adding the SR network to the SepFormer is a clear improvement in the frequencies above 1 kHz. Since the SR network utilizes a residual connection and the final output has a rectified linear activation, it can only add information to the magnitudes of the estimations. However, as the separation and SR networks are trained at the same time, it is possible that the separation network learned to rely more on the SR network to reconstruct higher frequencies rather than the transposed convolutional layer it would normally rely on without the SR network.

This would explain why the addition of the SR network also seemingly removes incorrect information in the high frequencies as shown in Figure 4.

Libri2Mix Results

Our approach also outperforms SOTA results on the clean and noisy Libri2Mix (Table 2). While one could think that the SR network could be trained for denoising and super-

Method	SI-SNRi (db)
SepFormer	22.3
+ multi-loss, same sampling rates	23.0
+ multi-loss, changing sampling rates	23.3
+ SR w/o heuristics	22.9
+ SR w/ heuristics	23.3
+ multi-loss and SR w/ heuristics	24.0

Table 3: Ablation on WSJ02-Mix. Dynamic Mixing is used with all methods.

resolution simultaneously, thus making it particularly good for the noisy Libri2Mix, in our testing it did not work out this way. The performance with the noisy Libri2Mix is not improved when trying to additionally optimize the SR network for denoising. We have provided some audio samples for the noisy Libri2Mix [1].

Ablation Study

Table 3 shows the results of the base model and how the results change depending on which part is added. The results of Table 3 show the importance of the heuristics as well as changing the sampling rate.

Using Super-Resolution to Speed up Other Architectures

Although SR has been mostly discussed in combination with the SepFormer architecture, it can be used with almost any other audio separation method.

In Table 4 it is shown how the same SR network can be used to achieve more than double the speed of the DPRNN without significantly lowering its accuracy. The results suggest, that it is not necessary to work with the full resolution for the source separation. Instead, it makes more sense to work with a downsampled representation for the separation and upsample the separated estimations using SR.

Since the separation is the most resource intensive part of the network and this resource intensiveness is linked with the sequence length, our approach also allows us to improve accuracy instead of speed by extending the separation network since downsampling reduces the sequence length and thus frees up resources.

Method	Stride	SI-SNRi (dB)	Speed (ms)	Chunk size
DPRNN	1	19.1	65.5	200
DPRNN	8	18.0	25.8	200
DPRNN + SR	8	18.4	31.9	200
DPRNN	8	17.7	25.3	50
DPRNN + SR	8	18.8	31.3	50

Table 4: Speed (ms) and SI-SNRi (dB) of the DPRNN and the SR network on the WSJ0-2mix during inference. The speed is how long it takes for an RTX 2070 Super to separate 1 second given a 4 second mixture.

It is also notable that the SR network performs better with a lower chunk size. This is why we lowered the chunk size for our implementation to 50 instead of the 250 of the original SepFormer.

This behaviour is unexpected since the chunk size is a parameter that is only relevant to the separation network and not to the SR network. Table 4 shows, that for the separation, a lower chunk size actually leads to lower accuracy. The results suggest, that the SR network seems to be able to correct mistakes better, when a lower chunk size is used for the separation even though this lower chunk size leads to slightly worse estimations.

Although the results in Table 4 show how the SR network can be used to speed up other architectures, it should be noted that the heuristics of the SR network were removed for these experiments. Unlike the concept of the SR network itself, the heuristics may need adjusting for each architecture or are simply not needed to make the SR network work to its full potential in some cases. In order for the SR network to work with existing architectures, it is necessary to calculate multiple losses, similar to the multi-loss concept used in the SFSRNet. There are three mandatory losses. One of them is for optimizing the SR network and one of them is for optimizing the final estimations of the separation process. The third loss is for optimizing the estimations that are calculated before the SR network.

It is also necessary to normalize the estimations between negative 1 and 1 before they are processed by the SR network. This is because the outputs of the separation network are scaled incorrectly and one of the main reasons the SR network functions is the reference point the mixture brings. Therefore, the estimations need to be scaled the same way the mixture is.

Conclusion

This paper proposes a new neural network for source separation which utilizes super-resolution (SR). While the proposed network is exceeding state-of-the-art performance on the WSJ0-2mix and Libri2Mix tasks, the main contribution of the paper is the SR process which is able to improve any separation network. The secondary contribution is the multi-loss system where the sampling rate of the solutions increases after each separation block.

SR works well with audio source separation since the separation usually does not need the full resolution sequence and returning to the original sampling rate is simplified by having all the necessary information in the mixture. SR allows to either speed up or increase the accuracy of the network.

A limitation of our SR implementation is operating on only the magnitude and not the phase. This mirrors early source separation approaches. Finding a source separation compatible SR network which operates in the time-domain like in (Kuleshov, Enam, and Ermon 2017; Lee and Han 2021) or uses phase reconstruction (Hu et al. 2020) would be a logical next step.

[1] https://github.com/j-rixen/SFSRNet-Audio-samples

References

Bahmaninezhad, F.; Wu, J.; Gu, R.; Zhang, S.-X.; Xu, Y.; Yu, M.; and Yu, D. 2019. A Comprehensive Study of Speech Separation: Spectrogram vs Waveform Separation. In *Interspeech 2019*, 4574–4578. ISCA.

Bronkhorst, A. 2000. The Cocktail Party Phenomenon: A Review of Research on Speech Intelligibility in Multiple-Talker Conditions. *Acta Acustica united with Acustica*, 86: 117–128.

Chen, J.; Mao, Q.; and Liu, D. 2020. Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation. In *Interspeech 2020*, 2642–2646. ISCA.

Cosentino, J.; Pariente, M.; Cornell, S.; Deleforge, A.; and Vincent, E. 2020. LibriMix: An Open-Source Dataset for Generalizable Speech Separation. arXiv:2005.11262.

Garofolo, John S.; Graff, David; Paul, Doug; and Pallett, David. 1993. CSR-I (WSJ0) Complete.

Haykin, S.; and Chen, Z. 2005. The Cocktail Party Problem. *Neural Computation*, 17(9): 1875–1902.

Hershey, J. R.; Chen, Z.; Le Roux, J.; and Watanabe, S. 2016. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 31–35. Shanghai: IEEE.

Hu, S.; Zhang, B.; Liang, B.; Zhao, E.; and Lui, S. 2020. Phase-Aware Music Super-Resolution Using Generative Adversarial Networks. In *Interspeech 2020*, 4074–4078. ISCA.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Kolbaek, M.; Yu, D.; Tan, Z.-H.; and Jensen, J. 2017. Multitalker Speech Separation With Utterance-Level Permutation Invariant Training of Deep Recurrent Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10): 1901–1913.

Kuleshov, V.; Enam, S. Z.; and Ermon, S. 2017. Audio Super Resolution using Neural Networks. *CoRR*, abs/1708.00853.

Lam, M. W.; Wang, J.; Liu, X.; Meng, H.; Su, D.; and Yu, D. 2019. Extract, Adapt and Recognize: An End-to-End Neural Network for Corrupted Monaural Speech Recognition. In *Interspeech 2019*, 2778–2782. ISCA.

Lam, M. W. Y.; Wang, J.; Su, D.; and Yu, D. 2021. Sandglasset: A Light Multi-Granularity Self-Attentive Network for Time-Domain Speech Separation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5759–5763. Toronto, ON, Canada: IEEE.

Lea, C.; Vidal, R.; Reiter, A.; and Hager, G. D. 2016. Temporal Convolutional Networks: A Unified Approach to Action Segmentation. In Hua, G.; and Jégou, H., eds., *Computer Vision – ECCV 2016 Workshops*, volume 9915, 47–54. Cham: Springer International Publishing.

Lee, J.; and Han, S. 2021. NU-Wave: A Diffusion Probabilistic Model for Neural Audio Upsampling. arXiv:2104.02321.

Luo, Y.; Chen, Z.; and Yoshioka, T. 2020. Dual-Path RNN: Efficient Long Sequence Modeling for Time-Domain Single-Channel Speech Separation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 46–50. Barcelona, Spain: IEEE.

Luo, Y.; and Mesgarani, N. 2018. TaSNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 696–700. Calgary, AB: IEEE.

Luo, Y.; and Mesgarani, N. 2019. Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8): 1256–1266.

Nachmani, E.; Adi, Y.; and Wolf, L. 2020. Voice separation with an unknown number of multiple speakers. In *ICML 2020*, 2623–2634.

Narayanan, A.; and Wang, D. 2014. Investigation of Speech Separation as a Front-End for Noise Robust Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4): 826–835.

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

Roux, J. L.; Wisdom, S.; Erdogan, H.; and Hershey, J. R. 2018. SDR - half-baked or well done? *CoRR*, abs/1811.02508.

Subakan, C.; Ravanelli, M.; Cornell, S.; Bronzi, M.; and Zhong, J. 2021. Attention Is All You Need In Speech Separation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 21–25. Toronto, ON, Canada: IEEE.

Tzinis, E.; Wang, Z.; and Smaragdis, P. 2020. Sudo RM -RF: Efficient Networks for Universal Audio Source Separation. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. Espoo, Finland: IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, D.; and Chen, J. 2018. Supervised Speech Separation Based on Deep Learning: An Overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10): 1702–1726.

Yu, D.; Kolbaek, M.; Tan, Z.-H.; and Jensen, J. 2017. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 241–245. New Orleans, LA: IEEE.

Zeghidour, N.; and Grangier, D. 2020. Wavesplit: End-to-End Speech Separation by Speaker Clustering. arXiv:2002.08933.

Zhang, L.; Shi, Z.; Han, J.; Shi, A.; and Ma, D. 2020. FurcaNeXt: End-to-End Monaural Speech Separation with Dynamic Gated Dilated Temporal Convolutional Networks. In Ro, Y. M.; Cheng, W.-H.; Kim, J.; Chu, W.-T.; Cui, P.; Choi, J.-W.; Hu, M.-C.; and De Neve, W., eds., *MultiMedia Modeling*, volume 11961, 653–665. Cham: Springer International Publishing.