DENOISING IN DUAL-PATH-RNN-BASED SPEECH SEPARATION

Anna Grillenberger (s213637), Daniel Schober (s212599), Huiyu Lu (s212648), Pablo Táboas (s212556)

ABSTRACT

We present two new training approaches for the state-of-the-art models in speech separation for usage in noisy environments. Additionally, the models are tested on different noise categories and intensities. The new methods of using an additional output channel for noise data, and employing a cascade approach, both outperform the current method for known noise data. For unknown noise, the performance depends heavily on the noise category and the noise intensity.

Index Terms— Speech separation, Deep learning, Denoising, Recurrent neural networks

1. INTRODUCTION

Speech separation is the problem of isolating one or more speech signals from a mixture of sounds, where the mixture may include multiple speakers, background noise, and other interfering signals [1]. The ability of the human brain to separate the desired conversation in a room full of people was first noted by Cherry in 1953 [2], and the task is since then often called the ”cocktail-party-problem”. This is challenging for a technical device, especially in noisy environments, where the quality of the speech signals is degraded and the interfering sounds may mask or interfere with the speech. Despite these challenges, there is a growing interest in developing algorithms and systems for speech separation, as it has many potential applications in areas such as speech recognition, hearing aids, and telecommunication. In recent years, solving this task using deep learning approaches gained popularity, specifically treating it as a supervised learning problem.

In this work, we focus on the problem of dealing with noise in supervised voice separation from a single microphone, based on the state-of-the-art architectures SVoice [3] and DPRNN [4]. In the supervised learning approach, the dataset contains mixed audio, individual voices, and noise data. This is used to train a model to separate new mixed audio containing unseen speakers and background noise.

The models in the mentioned papers are trained only with output channels for a specified number of speakers, without explicitly using the known noise data. Both models' results show a performance gap when applied to noisy conditions. To overcome these limitations, we propose two approaches to using state-of-the-art architectures in noisy environments. First, we suggest a 2+1 channel model, that uses an additional output channel for the given noise. Second, a cascade approach consisting of two individual models, which first separates the speakers from the background noise, and then separates the voices of the speakers.

In addition to the results of the performance of the SVoice model using the WHAM! and WHAMR! datasets shown in [3], we present experiments of our trained models on different and unseen environmental and speaker background noise categories from the Microsoft Scalable Noisy Speech Dataset (MS-SNSD). Testing is also done for different noise intensities by varying the input speech-to-noise ratio (SNR).

Our main contributions are: (i) a 2+1 channel and a cascade approach to use state-of-the-art speech separation models in noisy conditions, (ii) results of the SVoice and DPRNN models for speech separation on a different dataset (LibriMix), and (iii) performing detailed testing of the models for different noise categories and intensities.

2. RELATED WORK

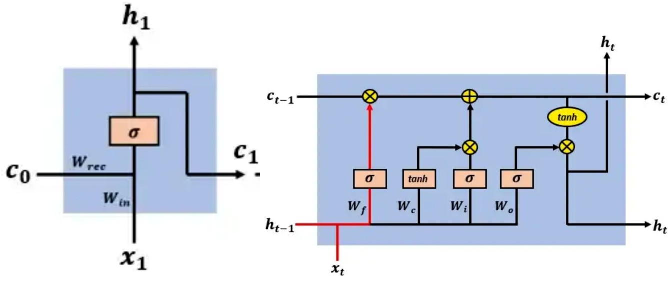
Recently, the interest in research in deep learning-based speech separation has progressed from standard time-frequency domain methods to time-domain approaches where the magnitude and phase information are jointly modeled and optimized [5][6][7]. As the time-frequency representation has several drawbacks including that the phase and magnitude of the signal are decoupled, and a long latency appears when calculating the spectrogram, a time-domain audio separation network (TasNet) was proposed by Yi Luo and Nima Mesgarani [8] significantly outperforming previous time-frequency methods in terms of separating speakers in mixed audio. It employs three components: an encoder, a separator, and a decoder. Firstly, TasNet creates a representation of the input signal that is optimized for identifying different speakers using a convolutional encoder. Then, a separator estimates masking matrices (weighting functions) for each target speaker for the actual speech separation. As a final step, a linear 1-D transposed convolutional decoder converts the modified encoder representation back to the sound waveform.

As usually, the input sequences are extremely long with tens of thousands or more waveform samples, effective modeling including long-term temporal dependencies poses a great challenge in conventional methods such as RNNs and 1-D CNNs [9]. Especially, 1-D CNNs with a fixed receptive field are not able to utilize time dependencies of the whole sequence [10].

As a solution, Luo et al. introduced a dual-path RNN (DPRNN) [4], using the same encoder and decoder architecture as TasNet but exchanging the 1-D CNN layers for separation by a network of Dual Path RNN blocks. After splitting the input sequence into shorter chunks, the data is modeled using two blocks of bi-directional RNNs. The first block models the data points of each chunk (intra-chunk RNN) independently to include local time dependencies and the second block uses inter-chunk RNN to combine the information from all the chunks for global modeling. By fully utilizing the global information via the inter-chunk RNNs, superior performance with 4.6% relative improvement with respect to scale-invariant signal-to-noise ratio (SI-SNR) on the WSJ0-2mix dataset can be achieved with a 49% smaller model size.

Focusing on an unknown number of multiple speakers, Nachmani et al. present a new method called SVoice [3], greatly outperforming the current state-of-the-art approaches including TasNet and DPRNN. Similar to the DPRNN, the SVoice model employs blocks of RNNs applied sequentially after each other. But instead of calculating the loss only on the final output, they discovered that it is beneficial to evaluate the error after each RNN, obtaining a compound loss that reflects the reconstruction quality after each layer. Additionally, two RNNs run in parallel in each block where the output of both is concatenated with the layer input that undergoes a bypass (skip) connection. As the output voices can switch between output channels, they propose a new permutation invariant loss that is based on a voice representation network trained on the same training set.

Besides the presented approaches where speech separation and denoising are unified in one single framework (multitasking), another method is widely used in research, called cascade. In comparison to the multitask models whereby converging the separated output channels to the clean speaker signal, noise is disregarded at the same time as the speakers are separated, in the cascade approach these tasks are split. Firstly, the input speech is processed by a speech-denoising model to remove noise components. The output mixture of multiple speakers without noise is then subsequently processed by a speech separation model to separate the individual sources. Liu et al. present a method where a denoising module is added as a front-end processor in order to facilitate speech enhancement [11]. As an outcome, the denoising module leads to a substantial performance gain across various noise types, and even better generalizations in noise-free conditions are achieved. Another two-stage model based on Conv-TasNet uses deep dilated temporal convolutional networks (TCN) to deal with the effects of noises and interference speakers separately, outperforming one-stage separation baselines substantially [12].



(a) RNN cell

(b) LSTM cell

Fig. 1: Comparison between the RNN and the LSTM cell.

3. MODEL ARCHITECTURE

After studying the algorithms explained in section 2, the SVoice architecture is chosen to be used and adapted in this research, as it achieves the best performance compared to other state-of-the-art approaches in speech separation. In the following, we will present some of the main advantages of the SVoice architecture. A key feature is the use of bidirectional LSTM cells. One of the main issues of RNNs is the vanishing or exploding of the gradient that comes with back-propagation. In short, RNNs are capable, in principle, to store past inputs to produce the currently desired output. For long-term sequences, the training process requires many time steps making the training error vanish as it gets propagated back [13]. On the other hand, the exploding problem describes the phenomenon that as the gradient is backpropagated through the network, it may grow exponentially from layer to layer. Either the step size is too large for updates to lower layers to be useful or it is too small for updates to higher layers. [14]

LSTM has three gates that update and control the cell states (forget, input, and output gates). The forget gate controls what information to forget in the cell state, the input one controls what new information will be encoded into the cell state, and the output gate controls what information encoded in the cell state is sent to the network as input in the following time step. In an LSTM, the state vector c_t can be written as

$$c_t = c_{t-1} \otimes f_t \oplus \tilde{c}_t \otimes i_t \tag{1}$$

It is the presence of the forget gate's vector of activation in the gradient term along with additive gradient structure (derived from Equation 1) which allows the LSTM to find a parameter update at any time step that prevents the error gradients from vanishing.

The LSTMs are used in a MultCat block whose schema is shown in Figure 3. In each of the two dual-path-blocks, two separate bidirectional LSTMs element-wise multiply their outputs and finally concatenate the input to produce the module output. The first block models the data points of each chunk (intra-chunk) to include local time dependencies and the second block models each data point of all the chunks

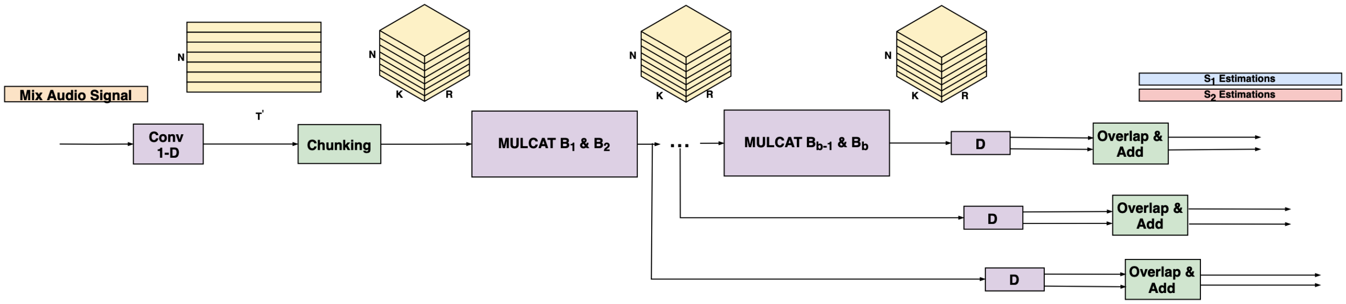


Fig. 2: General schema of the SVoice model.

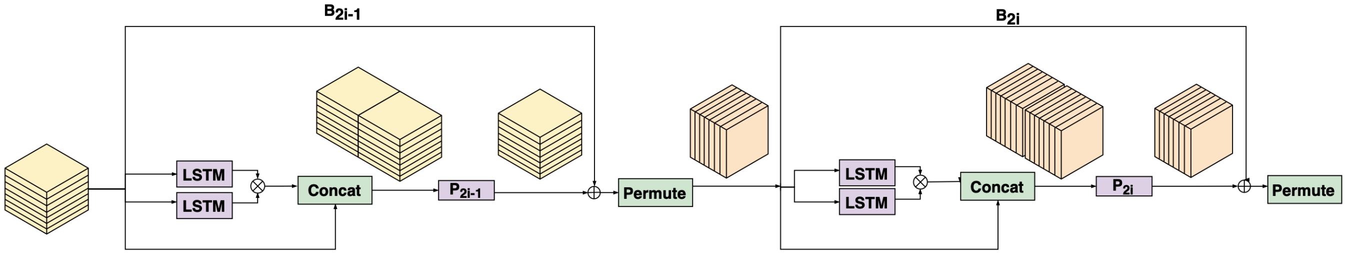


Fig. 3: Multiply and concat block (MulCat)

(inter-chunk) to combine the information from all the chunks for global modeling. In this way, the time dependencies of the full sequence are incorporated.

The loss function of the model uses SI-SNR (scale-invariant signal-to-noise ratio) and the main point is to obtain a value as high as possible. The SI-SNR is defined as

$$SI\text{-}SNR(s,\hat{s}) = 10\,log_{10}\frac{\|\tilde{s}_i\|^2}{\|\tilde{e}_i\|^2} \tag{2}$$

where $\widetilde{s_i}$ and $\widetilde{e_i}$ are defined as

$$\widetilde{s_i} = \frac{\langle s_i, \widehat{s_i}\rangle\, s_i}{\|s_i\|^2}\text{ , and } \widetilde{e_i} = \widehat{s_i} - \widetilde{s_i}$$

with the clean speaker input sources s, and the estimated speaker channels $\hat{s}$. The model has multiple input speech signals. Since the interest is to separate them (regardless of the output order), the loss is computed for the optimal permutation of the different output channels, i.e. the SI-SNR value will be the highest value obtained for the signals. This loss function is denoted as uPIT (utterance level permutation invariant training) [15] and is given as

$$l(s,\hat{s}) = -max\frac{1}{C}\sum_{i=1}^{C} SI\text{-}SNR(s_i\tilde{s}_{\pi(i)}), \tag{3}$$

where π is the optimal permutation of the C different output channels. Moreover, the loss function is calculated multiple times along the decomposition process. More specifically, the output of every odd MulCat block is decoded, which allows the model to calculate the uPIT loss multiple times, resulting in a so-called multiloss. This can be calculated as follows:

$$l(s,\{\hat{s}_j\}_{j=1}^{b/2}) = \frac{1}{b}\sum_{j=1}^{b/2} l(s,\hat{s}_j), \tag{4}$$

where b is the number of MulCat blocks.

Since the SVoice paper achieved good results and performed hyperparameter optimization with high computational resources, we decided to choose the same hyperparameters instead of changing them aimlessly with a probable decrease in performance. Instead, we focus on an extensive evaluation with different test scenarios and different datasets to answer our research question. For further investigations, it would be interesting to optimize the hyperparameters specifically for noisy environments.

4. EXPERIMENTAL SETUP

In this section, we will outline the steps for configuring and conducting our experiments, and present the detailed results of these experiments in the following section. As Figure 4 shows, a separate model is trained for each experimental scenario with a corresponding number of output channels.

All models have the SVoice architecture as a base. For the multitasking approach, which combines the denoising and

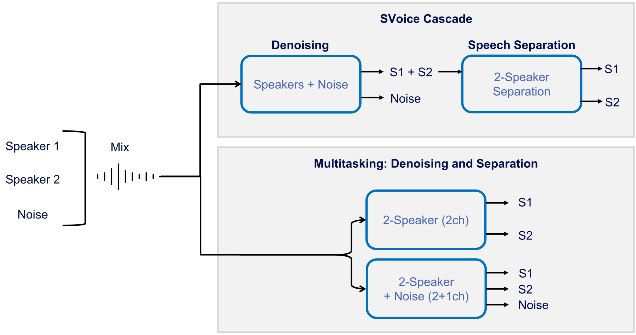


Fig. 4: Experimental protocol

source separation task in one model, two variations are evaluated. One 2ch-model where only the separated speakers are defined as output and a 2+1ch-model where an additional output channel for noise is added. In that way, the model also learns to characterize the noise sound profile. In addition to comparing these two model architectures with SVoice as a base, the DPRNN architecture is evaluated as well to investigate if the different denoising approaches show similar results with different speech separation architectures.

Moreover, a cascade approach will be implemented where two models are trained separately, one for denoising and one for speech separation. The denoising model is trained with an input mixture of 2 speakers and noise, having an all-speakers-channel and a noise channel as output. The speech separation model is trained on clean speaker mixtures without noise to separate the sources in one output channel for each speaker. Later in testing, both models are applied after each other.

4.1. Evaluation framework

Training and Test Data We trained our models on a LibriMix dataset of 2 speakers with ambient noise and then tested it on a held-out set of unseen mixed signals. LibriMix dataset consists of two- or three-speaker mixtures combined with ambient noise samples from WHAM! [16]. To reduce the computational cost, the waveforms were down-sampled to 8 kHz. For each audio, we have the clean waveform of speaker 1, speaker 2, and noise parts, i.e. each of the sources. The mixture is the sum of those three parts. The first 800 audios form the train set, and the remaining 200 audios form the valid set. Here, 4 to 6-second-long segments were used.

For additional testing, we collected six different categories of background noise from the MS-SNSD dataset (Typing, Nature, Cafe, Babble, Washing Machine, Square) [17]. We manually mix the different noises with the LibriMix speaker data. For the creation of the test set, we also vary the input SNR to analyze the effect of noise intensities. Here, we use five different SNR values.

4.2. Training Procedure

Model Configurations For DPRNN, we trained our model using the Asteroid toolkit. Asteroid is a Pytorch-based audio source separation toolkit whose support for audio pre-processing techniques and efficient processing of large amounts of data can simplify the training and evaluation process [18]. To use the SVoice model, we adapted the training and evaluation pipeline provided by [3].

Training Setup and Hyper-Parameters All model training is performed on DTU HPC, and both DPRNN and SVoice use the same training and validation sets. The DPRNN models are trained with 16 V100 GPUs with 8GB of RAM, while the SVoice models were trained with 8 V100 GPUs. We used batch sizes of 16 and 4 for DPRNN and SVoice, respectively. The optimizers are ADAM with a learning rate of 1e-3 and 5e-4, respectively. Due to limited available training time, the DPRNN models were trained for 100 epochs, while the training for the SVoice models was stopped after 20 epochs. Due to the decoding after every odd MulCat block to calculate the multiloss in SVoice, the training for the SVoice models requires much more time.

4.3. Quality of Separation

For the evaluation of the proposed models, we calculate the commonly used scale-invariant signal-to-noise ratio improvement (SI-SNRi) score. SI-SNRi is typically measured in decibels (dB) and is an important performance metric for speech separation systems, as it directly impacts the accuracy and quality of the separation. It is computed as follows:

$$SI\text{-}SNRi(s,\hat{s},x) = \frac{1}{C}\sum_{i=1}^{C} SI\text{-}SNR(s_i,\hat{s_i}) - SI\text{-}SNR(s_i,x)$$

where x are unseen mixture files with noise, s the speaker input sources, and $\hat{s_i}$ the estimated channels. For a fair comparison between the 2ch and 2+1 channel models, the SI-SNRi value for the 2+1ch model is only calculated using the speaker channels. Looking at the model outputs, we have found that the noise channel is usually separated well and similar to the input noise data, which would therefore lead to a positively biased SI-SNRi value for the 2+1ch model. Hence, we disregard the predicted noise channel in our testing pipeline before the calculation of the SI-SNRi value.

A positive SI-SNRi value indicates that the estimated channel has a higher SI-SNR than the original mixture signal, while a negative SI-SNRi indicates that the estimated channel has a lower SI-SNR. An SI-SNRi of zero means that the estimated speaker channel is equally well separated as in the provided mixture file.

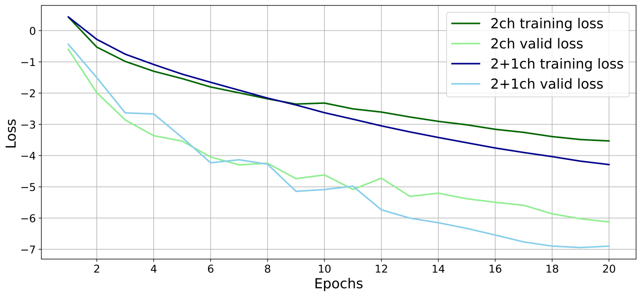


Fig. 5: Training curves of the SVoice 2ch and 2+1ch models trained with LibriMix data and WHAM! noise.

5. EXPERIMENTAL RESULTS

In this section, we first provide our experimental results of the different approaches tested on 200 unseen samples of the LibriMix dataset. After, we show the results of the SVoice 2ch and 2+1ch approaches for different noise categories and noise intensities.

Since the baselines from the mentioned publications were trained on much more data and higher computational resources, we use our own trained models from SVoice and DPRNN as a baseline to get comparable results (hereinafter referred to as SVoice 2ch and DPRNN 2ch). The baselines were trained with the same parameters and number of epochs.

5.1. Comparison of Model Architectures

The trend of the loss values for both models trained with SVoice is presented in figure 5. Since the loss values are decreasing during training, a learning of the target behavior can be detected. Whereas, in the beginning of training, the loss values of the two models are similar, in later epochs the 2+1 model shows a better performance. It can also be observed that the curves of both models are not completely flattened at epoch 20, where we stopped the training due to high training times. This took around 48 hours per model. Hence, better results can be expected for training with more epochs.

The results are reported in Table 1. It can be seen that our 2+1ch approach outperforms the 2ch approach used in the publications for both SVoice and DPRNN by a sizable margin. Also, the models trained with the SVoice architecture outperform the results obtained by DPRNN on the new dataset. It should be mentioned that the obtained SI-SNRi values are much lower than the ones shown in [3] (15.2 and 13.9) since more training data and time were used by the authors.

In addition, the described cascade approach for SVoice performs better than both the 2ch and the 2+1ch model. This approach requires a bigger model size and has a higher computational demand since two models are trained individually.

The distribution of the SI-SNRi values for the 200 samples used in testing for the mentioned approaches can be seen in the whisker plot in figure 6.

Approach	Model size	SI-SNRi (dB)
SVoice 2ch	7.5M	5.22
SVoice 2+1ch (Ours)	7.5M	6.15
DPRNN 2ch	3.6M	4.59
DPRNN 2+1ch (Ours)	3.6M	5.27
SVoice Cascade (Ours)	15M (2x7.5M)	7.19

Table 1: Peformance of the different approaches. All values were obtained by our own trained models.

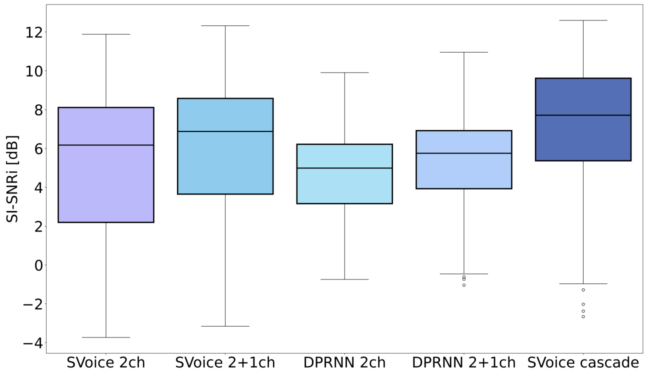


Fig. 6: Whisker plot of the results of the speech separation on 2-speaker LibriMix data with WHAM! noise

5.2. Generalizing in Different Noise Settings

Next, we compared the performance of the SVoice 2ch and the proposed SVoice 2+1ch approach under different, unseen noise settings using the MS-SNSD dataset. The SI-SNRi results for the six chosen categories for an input SDR of 0 are presented in figure 7.

It can be seen that the 2+1ch approach has similar or slightly worse results than the 2ch approach for five of the six categories. A tendency, that one of the approaches works better on environmental noise (typing, nature, washing machine) or on human speaking background noise (cafe, babble, square) can not be observed. However, the 2+1ch model clearly outperforms the 2ch model for the *Babble* noise category. This was observed for all tested noise intensities. Further investigations into the different noise samples showed that the *Babble* noise is the most similar to the noise used in the WHAM! dataset, which might explain the good performance. Given the number of training samples and the variety of noise data our model was trained on, the 2+1ch model is not able to generalize well in different noise categories.

In addition to adding unseen noise categories to the speaker files, we also varied the input SNR, which changes the intensity of the noise in relation to the speakers. The lower the SNR, the higher the noise intensity in the testing files. The results of the 2ch and 2+1ch models tested with files with noise from the category *Café* with five different noise intensities are presented in figure 8. It can be seen that

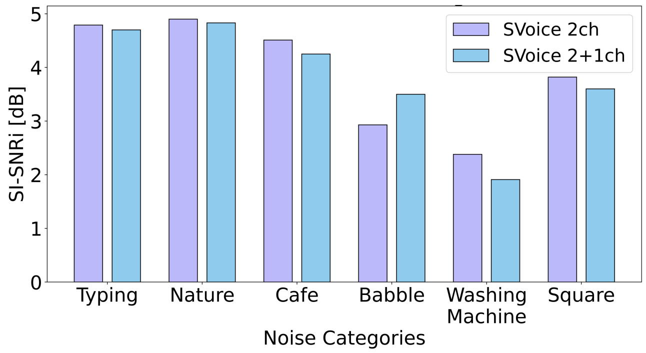


Fig. 7: Speech separation performance of the SVoice 2ch and 2+1ch models for unknown noise data with different categories from the MS-SNSD dataset with an input SDR of 0.

the 2+1ch model performs better than the 2ch model for low noise intensities, while it is worse for samples with high noise intensities. This behavior can be observed for all tested noise categories except for *Babble* noise, where the 2+1ch model shows better results for the SI-SNRi for all five input SNR.

The performance of both models depends heavily on the noise intensity. In general, the lower the noise intensity, the higher the SI-SNRi. However, for both models, a flattening of the curve can be seen between SNR 0 and 5. This might be because the SNR for our training data set lies in this range, and therefore the models can not generalize well for higher noise intensities.

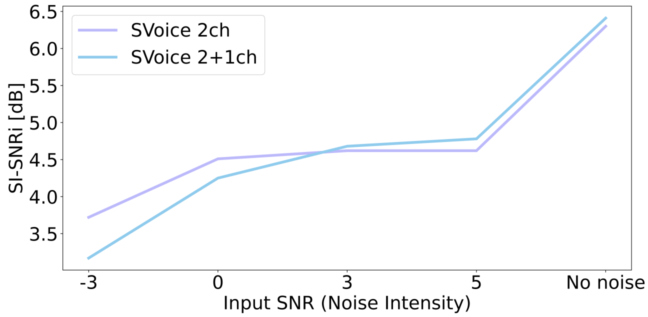


Fig. 8: Speech separation performance of the SVoice 2ch and 2+1ch models for unknown noise data from the category Café from the MS-SNSD dataset with five different input noise intensities.

6. DISCUSSION AND FUTURE WORK

In this study, we evaluated the performance of various deep-learning models for speech separation in noisy environments. Unlike previous work, we trained models for both SVoice and DPRNN with an additional noise channel. For known noise, the obtained results for the models trained with an additional noise channel are better than those trained with the existing method, by a sizable gap. This suggests that explicitly modeling the noise in the training process can be beneficial for improving speech separation and denoising performance. For further analysis, our new training approach should be used with the same data and computational resources as in [3], to find out if the baseline of an SI-SNRi of 15.2 can be outperformed.

However, for unknown noise conditions, the results of models with an additional noise channel were highly dependent on the specific noise category and intensity. Our models performed well for the noise category they were trained on (background speakers), but may not generalize well to other types of noise. We find that the 2+1ch model typically outperforms the 2ch model at low noise intensities. In further work, the training dataset should be created with a high variety of noise categories and input noise intensities and investigated if the new model is able to generalize better.

Additionally, we found that the SVoice model outperformed the DPRNN model on a new dataset (LibriMix), which confirms the results presented in [3], indicating that SVoice may be a particularly effective approach.

In addition, we also evaluated the effectiveness of multi-task learning compared to a cascade approach. We find that the cascade approach performs slightly better than multi-task learning at the cost of increased computational demand. For this, two different models needed to be trained. In further work, a new training procedure or a new cascade model could be developed, which combines the two steps in a single model, so both tasks are improved simultaneously while training.

Overall, these findings provide valuable insights into the capabilities and limitations of different approaches for training state-of-the-art models for speech separation in noisy environments.

The code of this project can be found in this github-repository: `https://github.com/AnnaGr-Git/DL_Denoising_SpeechSeparation`

7. REFERENCES

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," 2017.

[2] E.C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," in Journal of the Acoustical Society of America, 1953, vol. 25, p. 975–979.

[3] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," 2020.

[4] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," 2019.

[5] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," arXiv preprint arXiv:1806.03185, 2018.

[6] 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, vol. 27, no. 8, pp. 1256–1266, 2019.

[7] F. Bahmaninezhad, J. Wu, R. Gu, S. Zhang, Y. Xu, M. Yu, and D. Yu, "A comprehensive study of speech separation: spectrogram vs waveform separation," arXiv preprint arXiv:1905.07497, 2019.

[8] Y. Luo and N. Mesgarani, "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). IEEE, 2018, pp. 696–700.

[9] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5457–5466.

[10] S. Bai, Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.

[11] Y. Liu, M. Delfarah, and D. Wang, "Deep casa for talker-independent monaural speech separation," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 6354–6358.

[12] C. Ma, D. Li, and X. Jia, "Two-stage model and optimal si-snr for monaural multi-speaker speech separation in noisy environment," arXiv preprint arXiv:2004.06332, 2020.

[13] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 6, no. 02, pp. 107–116, 1998.

[14] G. Philipp, D. Song, and J. Carbonell, "The exploding gradient problem demystified-definition, prevalence, impact, origin, tradeoffs, and solutions," arXiv preprint arXiv:1712.05577, 2017.

[15] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 10, pp. 1901–1913, 2017.

[16] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Librimix: An open-source dataset for generalizable speech separation," arXiv preprint arXiv:2005.11262, 2020.

[17] C. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan, and J. Gehrke, "A scalable noisy speech dataset and online subjective test framework," arXiv preprint arXiv:1909.08050, 2019.

[18] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F. Stöter, M. Hu, and J. Martín-Doñas, "Asteroid: the pytorch-based audio source separation toolkit for researchers," arXiv preprint arXiv:2005.04132, 2020.