Investigation of Speech Separation Models Including Video Source

Vladislav Smirnov
FCS, HSE, Moscow
vmsmirnov@edu.hse.ru

Andrey Petukhov
FES, HSE, Moscow
aapetukhov_1@edu.hse.ru

Vera Buylova
FES, HSE, Moscow
vbuylova@edu.hse.ru

Last Edit Date: November 24, 2024

***Abstract*—This paper investigates state-of-the-art speech spearation models with a focus on audio-visual methods. We evaluate existing audio-only and audio-visual architectures, e.g. ConvTasNet, DPRNN, DPTNet, and extend their functionality by integrating visual source. We also experiment with existing target source separation model, VoiceFilter, and test the existing audio-visual solution, RTFSNet. Our experiments demonstrate a significant performance boost with utilization of AV fusion techniques with the existing audio-only solutions. The proposed AV-DPTN model achieves the highest quality among the tested models while maintaining computational efficiency. The codebase for this paper is publicly available on GitHub at https://github.com/teasgen/speech_separation.**

I. Introduction

Audio source separation (SS) is an area in deep learning with applications ranging from denoising the conference calls to reconstructing the individual audio source (e.g. voice, guitar) from a music track. The central challenge of SS lies in the separation a mixture of signals into its consistent components. This task is often categorized into blind source separation (BSS), when there is a need to separate a number of sources from one another, and target source separation (TSS), when there is a specific target source given by reference. In this paper, we'll be focusing on BSS methods. However, TSS methods are also suitable for the task of separating k sources if a ground truth audio for each speaker is provided. In this case, one can treat a problem as k independent TSS tasks.

The majority of TSS methods require a reference audio, which is a short, clean recording of the target speaker's voice. This audio serves as a unique identifier of the speaker (Wang et al., 2018[1]; Ge et al., 2020[2]). These methods minimize L2 loss between the predicted and reference spectrograms, thus, to predict an audio, one need to transform the spectrogram to the waveform to reconstruct the target signal.

In BSS problems, multiple sources are separated simultaneously, which creates a problem where the model must decide to which source corresponds each output. Permutation-Invariant Training (PIT)[3] minimizes the loss by considering all possible permutations of predicted and ground truth sources. This strategy is used in many time-domain methods (Luo & Mesgarani, 2019[4]; Luo et al., 2020[5]). However, due to its neglect of the frequency domain, these methods face limitations in a so-called cocktail party problem, when the audio is recorded in a noisy environment.

Many methods tried to extend the existing time-domain methods by including the frequency information (Afouras et al., 2018[6]). However these methods only increase the already high computational requirements and have proved to be inefficient.

In recent years, audio-visual (AV) learning has gained attention as an approach to enhance SS quality by utilizing visual information (e.g. lip movements) alongside with the audio one. The common approaches to work with audio-visual information are early fusion, where the features from both modalities are combined at the input stage for the joint processing (i.e. there is one model for both modalities), or late fusion, where the audio and visual features are pre-processed with different models. The current SOTAs (Li et al., 2022; Pegg et al., 2024) use the pretrained encoders for each domain, and then utilize the model that share information between adjacent processing layers (audio and video), thus, combining both early and late fusion.

In this work, we experiment with existing SOTA architectures to achieve a maximum separation quality for the mix audio of two speakers while trying to maintain low resource consumption. We conducted comprehensive experimental evaluations on the dataset provided by the teachers. The rest of the paper is organized as follows. Section II presents the existing solutions. In Section III, we describe the methodology used to conduct the experiments. In Section IV, we describe the conducted ablation studies and its results. We present the model that has achieved the highest quality in section V. Finally, we sum up our findings in Section VI.

II. Related Work

The audio-only source separation has driven a comprehensive research in this domain, producing the methods that have achieved a high separation quality. TasNet[4], includes a CNN encoder and LSTM layers that are used to process the uncompressed audio. This method is also utilized in ConvTasNet[5] that differed from the previous one by replacing the recurrent layers with a Temporal Convolutional Network (TCN), which allowed the model to handle the inputs of an arbitrary length. The model uses a linear encoder to

represent the speech waveform, applies masks using a TCN to separate speakers, and reconstructs the waveforms with a linear decoder. Empirical studies[7] have proved that adding deep encoder and decoder (i.e. sequential application of the convolutional layers) gives a significant quality boost to the model.

Although these models reach high separation quality, they have a large number of parameters and require a lot of time to train. Another architecture that is based on the TasNet approach uses the method that divides the audio into overlapping chunks and applies intra- and inter-chunk operations iteratively. This approach is used in Dual-path RNN[8], and it allows the model to capture both local and global dependencies. By replacing 1-D CNN layer used in TasNet with DPRNN, the model has reached a new state-of-the-art quality while also reducing the computational complexity.

Although achieving high performance metrics, both RNN and CNN methods face limitations in efficiently capturing the information about the context over long sequences. To address this, Dual-Path Transformer Network[9] was proposed. By integrating RNNs into the self-attention mechanism of transformers, DPTNet has reached a high performance results the problem of modeling of long speech sequences.

CTCNet[10], inspired by cortico-thalamo-cortical circuits, integrates audio and visual modalities by employing separate auditory and visual subnetworks to learn hierarchical representations, which are then fused through a thalamic subnetwork using top-down connections. This approach enables information sharing between modalities. RTFSNet[11] also uses the idea of the shared parameters. This model operates in the time-frequency domain using a DPRNN-like pipeline on compressed audio-visual inputs. It first applies the Short-Time Fourier Transform (STFT) to obtain the input representation and then integrates audio and visual information using an attention-based fusion mechanism. RTFSNet achieves state-of-the-art performance while significantly reducing the number of iterations and parameters.

VoiceFilter[1] is a target speech separation model that isolates a specific speaker's voice from a mixture using a short reference audio. The model generates a speaker embedding from the reference and uses it to condition a spectrogram masking network.

This paper builds upon existing research in both audio-only and audio-visual blind source separation. In this project,

- We developed a codebase for traning both audio-only and audio-vidual BSS and TSS models.
- Our experiments include evaluations of state-of-the-art audio-only models. We first started with the classic architectures (ConvTasNet and DPRNN). We then experimented with the enhanced versions of these models, such as Deep-Encoder-Decoder-Conv-TasNet and DPTNet.
- We used the VoiceFilter-based model to evaluate the suitability of the TSS pipeline for this task with.
- We extended our study to audio-visual models. We first started with a SOTA architecture RTFSNet. We then extended the previously mentioned audio models to the visual domain by combining the video and audio embeddings.

The last approach of extending DPTNet to the visual domain has proven to be the most efficient, balancing both low computational complexity and high separation quality.

III. Methodology

Dataset. The models were trained and tested on a two-speaker speech separation dataset consisting of mixed audios and ground truth audios for each speaker with a sampling rate of 16 kHz. The visual part of the dataset included videos of the lip movements for each speaker, represented as a series of 50 frames, each with a resolution of 96 × 96 pixels.

Encoders. For our video encoder, we utilized the pre-trained Multi-Scale TCN lip reading network[12]. This encoder takes a sequence of $T = 50$ grayscale video frames of size 96×96, corresponding to one video clip structured as an array of shape $(50, 96, 96)$. It outputs an embedding of size 512 for each frame, resulting in an embedding matrix of shape $(50, 512)$.

Training objective. The training objective is maximizing the scale-invariant signal-to-noise ratio (SI-SNR), which is calculated as:

$$\text{SI-SNR} = 10 \cdot \log_{10} \frac{\|\mathbf{s}_{\text{target}}\|^2}{\|\mathbf{e}_{\text{noise}}\|^2},$$

where:

$$\mathbf{s}_{\text{target}} = \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle}{\|\mathbf{s}\|^2} \mathbf{s},$$

$$\mathbf{e}_{\text{noise}} = \hat{\mathbf{s}} - \mathbf{s}_{\text{target}},$$

- $\mathbf{s}$ is the reference signal,
- $\hat{\mathbf{s}}$ is the estimated signal.

To handle the issue of matching predicted sources to their corresponding references, Permutation Invariant Training (PIT) is employed.

Evaluation metrics. We report the scale-invariant signal-to-noise ratio improvement (SI-SNRi)[13] that indicate the model's ability to reduce interference of the input audio. Additionally, we evaluate the output audio using widely used in source separation tasks metrics, perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI).

IV. Experimental Setup

A. *ConvTasNet*

The baseline model was exactly following the paper's[5] best non-causal configuration. The initial learning rate was set to 2×10^{-3}. AdamW was used as an optimizer. All of the models were trained for 400 epochs with 500 steps per epoch. However, for all of them the early stopping was applied as soon as the loss reached the plateau. The baseline configuration required 100 epochs to train. More complex models required more time to converge.

The application of the deep encoder and decoder was following the paper's[7] configuration with exponentially increasing (decreasing for decoder) dilation. This configuration was trained for 200 epochs.

The embeddings of two videos were compressed and concatenated to form a combined one, and then were linearly interpolated to the audio representation. The features were then normalized and added to the encoded audio. The algorithm described in Appendix A. Incorporating visual features reduced the convergence time to around 140 epochs to achieve a plateau.

Table I shows the performance of the systems with different parameters, from which we can conclude the following statements:

- Decreasing the gradient norm has lead to a faster convergence of the model.
- The utilization of the deep encoder-decoder architecture has increased the number of parameters but returned the better separation metrics. This configuration was chosen as a trade-off between performance and model size.
- Simple audio-visual feature fusion has given a significant quality boost to the model's performance and to the convergence speed.
- Training with an augmented dataset didn't increase the metric significantly.

TABLE I
THE EFFECT OF DIFFERENT CONFIGURATIONS IN CONV-TASNET.

Epochs	Max. Grad. Norm	Deep Enc-Dec	Audio-Visual	Size (M)	SI-SNR (dB)↑
100	8	×	×	5	6.21
200	5	✓	×	11.4	8.62
139	5	✓	✓	11.5	**11.07**

B. *DPRNN – like architectures*

The baseline model for these experiments was DPRNN model [8] for audio-only setup. All architecture parameters were the same as in the paper. The one and only one architecture change is added skip-connection between Encoder and Decoder (for skipping dprnn blocks). The step size always was equal to chunk size / 2. The initial learning rate was set to 2×10^{-3}. AdamW was used as an optimizer and Cosine Annealing Scheduler. All of the models were trained up to 100 epochs with 625 steps (full dataset) per epoch. Firstly, we experimented with different kernel sizes for audio length compression. The best size among [2, 4, 7] was 7, because the loss function converged significantly better during first 4 epochs with this value and achieved about 9.67 SI-SNR on validation after 90 epochs. Moreover, there were two experiments with different chunk size and shape of audio latent dimension, but they didn't provide better values.

The small number of epochs means the training process was stopped to prevent wasting Compute hours, when the ablation result was obvious.

After that we decided to change raw audio prediction to masking mix audio and tried Tanh-masking from DPTN paper [9].

TABLE II
THE EFFECT OF DIFFERENT CONFIGURATIONS IN BASE DPRNN

Kernel Size	Chunk Size	Latent Shape	Epochs	SI-SNR (dB)↑
2	150	64	3	-22
4	150	64	4	-0.59
7	150	64	4	1.03
7	150	64	93	**9.67**
7	250	64	4	-0.22
7	150	96	4	0.33

TABLE III
THE EFFECT OF MASKING DPTN MECHANISM IN BASE DPRNN

Masking	Epochs	SI-SNR (dB)↑
×	1	-1.84
×	17	**5.17**
✓	1	0.57
✓	17	4.24

As shown in Table III, despite good initial metrics, after 1 epoch the masking mechanism works worse for this model after several epochs.

C. *DPTN – like architectures*

The success and simplicity of DPTN model forced us to try it. The architecture parameters were as same as in the paper. To make experiments consistent we chose chunk size and kernel size equal to DPRNN experimetns, also the optimizer and LR were left the same, but we applied gradient clipping in transformer model to prevent gradient explosion. This approach didn't affect DPRNN training due to small gradient norm during most part of process.

The ablation of DPTN masking mechanism in previous stage stimulated to try raw audio prediction instead of paper's method. As shown in Table IV, DPTN is much better than DPRNN and the raw wav prediction is preferably. The number of epochs was 93 (plateau) for all experiments.

TABLE IV
DPTN VS DPRNN

Model	Masking	Max Grad Norm	Model Size (M)	SI-SNR (dB)↑
DPRNN	×	∞	2.6	9.67
DPTN	✓	10	2.8	10.43
DPTN	×	10	2.8	**11.45**

The audio-visual configuration was as same as in ConvTasNet experiments. Also for better converged we used gating mechanism for fusing audio and visual encodings. The gated fusion algorithm is presented in Appendix A.

TABLE V
THE EFFECT OF VISUAL FEATURES IN DPTN

Fusion type	Epochs (plateau)	Model Size (M)	SI-SNR (dB)↑
w/o visual	93	2.8	11.45
Identity	81	4.4	13.90
Scalar	81	4.4	14.03
Tanh scalar	81	4.4	**14.15**

In Table V you may see that gate fusion is slightly better than raw addition operation and visual features are extremely enhance model quality.

D. *RTFS-Net*

Another architecture that we implemented was RTFS-Net. The lightest model from the article [11], RTFS-Net-4, was used, resulting in a 0.8M parameters model. Although RTFS-Net-6 and other models have the same size (in number of parameters), they were not trained due to high memory demands of the TDANet [14] architecture which is used in audio and video processing. Training was performed with no more than 200 epochs, but early stopping was applied when the loss reached plateau. AdamW with learning rate equal to 10^{-3} was chosen as the optimizer and OneCycleLR as the scheduler. The maximum gradient norm was set to 8.

Following the paper's encoder architecture, we employed raw complex-valued STFT as audio encoder with Fourier transform size 256 and hop length 128. As a video encoder, we used a pretrained lipreader to extract 512-dimensional embeddings of the lip movements of speakers.

In contrast to the original paper, our network was trained to extract both speakers from the audio, i.e. the output of the network consisted of two clean audio samples. Instead of SRU, we used a 4-layer bidirectional LSTM.

Table VI presents training results for RTFS-Net.

TABLE VI
RTFS-NET TRAINING RESULTS.

R	Max. Grad. Norm	Size (M)	Video Embedding Size	SI-SNR (dB)↑
4	8	0.827	512	**7.65**

E. *VoiceFilter*

We attempted to adapt the VoiceFilter [1] audio-only architecture for an audio-visual task. The model was trained for no more than 100 epochs using the AdamW optimizer with a learning rate of 0.001, and OneCycleLRScheduler. To incorporate visual information, we replaced d-vector embeddings with video embeddings obtained from the pre-trained lipreader snv1x-tcn1x from the lipreading repository.

Given that the video embedding has dimensions $T_v \times D$, where $T_v = 50$ and $D = 1024$, the embeddings were further processed with a trainable GRU layer and a linear layer to compute the final d-vectors. For detailed descriptions, see Appendix B.

Despite experimenting with LSTM and GRU architectures, and applying various aggregation strategies for the recurrent layer outputs, the results were unsuccessful. Table VII summarizes the results under different loss functions. We conclude that further architectural changes are necessary for successful adaptation.

V. RESULTS

We have tried many Blind Speech Separation models and provided relevant ablation studies. During ablation studies, which are fully described in previous section, we got some insights:

1) Visual features have significantly boosted BSS model performance.
2) Predicting masks instead of raw audio performed noticeably worse in SI-SNRi terms when the validation loss went out to the plateau.

The best model is DPTN, which was expanded for using visual information and its final results are presented in Table VIII.

TABLE VII
VOICEFILTER TRAINING RESULTS UNDER DIFFERENT LOSS FUNCTIONS.

Loss	Size (M)	RNN	Max. Grad. Norm	D-vec size	SI-SNR (dB)↑
Si-SNR Wav	7.8	GRU	2	1024	0.16
L1 Spec	7.8	GRU	∞	1024	0.08
L2 Spec	7.8	LSTM	6	1024	-0.98

TABLE VIII
FINAL BEST MODEL AV-DPTN RESULTS ON VALIDATION SPLIT

Extracted	SI-SNR↑	SI-SNRi↑	PESQ↑	STOi↑	Model Size (M)	GFLOps
×	14.159	14.161	2.46	0.93	40.8	108.5
✓					4.4	51.4

'Extracted' means the video embeddings were previously extracted, otherwise SS model inference includes Video model inference.

VI. CONCLUSION

In this work, we conducted experiments with models for separating audio in a two-voice mixture. In addition, the models were adapted to incorporate an additional video input, and an approach for fusing audio and video features was developed. We hope that this work provides valuable insights into the application of speech separation.

VII. ACKNOWLEDGEMENTS

This work is a part of the Deep Learning for Audio course at HSE university.

REFERENCES

[1] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. Lopez Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," *arXiv preprint arXiv:1810.04826*, 2018.

[2] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "SpEx+: A Complete Time Domain Speaker Extraction Network," *arXiv preprint arXiv:2005.04686*, 2020.

[3] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation-invariant training (pit)," *arXiv preprint arXiv:1703.06284*, 2017.

[4] Y. Luo and N. Mesgarani, "TaSNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[5] ——, "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation," *arXiv preprint arXiv:1809.07454*, 2019.

[6] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep Audio-Visual Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[7] B. Kadıoğlu, M. Horgan, X. Liu, J. Pons, D. Darcy, and V. Kumar, "An Empirical Study of Conv-TasNet," *arXiv preprint arXiv:2002.08688*, 2020.

[8] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-Path RNN: Efficient Long Sequence Modeling for Time-Domain Single-Channel Speech Separation," *arXiv preprint arXiv:1910.06379*, 2019.
[9] J. Chen, Q. Mao, and D. Liu, "Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation," *arXiv preprint arXiv:2007.13975*, 2020.
[10] K. Li, F. Xie, H. Chen, K. Yuan, and X. Hu, "An Audio-Visual Speech Separation Model Inspired by Cortico-Thalamo-Cortical Circuits," *arXiv preprint arXiv:2212.10744*, 2022.
[11] S. Pegg, K. Li, and X. Hu, "RTFS-Net: Recurrent Time-Frequency Modelling for Efficient Audio-Visual Speech Separation," *arXiv preprint arXiv:2309.17189*, 2024.
[12] B. Martinez, P. Ma, S. Petridis, and M. Pantic, "Lipreading Using Temporal Convolutional Networks," *arXiv preprint arXiv:2001.08702*, 2020.
[13] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, 2006.
[14] X. H. Kai Li, Runxuan Yang, "An Efficient Encoder-Decoder Architecture With Top-Down Attention For Speech Separation," *arXiv preprint arXiv:2309.17189*, 2023.

APPENDIX

A. Gated Fusion Algorithm

```
Input:
    Audio embedding A ∈ ℝ^{T_audio, d_audio}
    Visual embedding V ∈ ℝ^{2, T_visual, d_visual}

Step 1: Compress visual embedding dimension
    V ← Linear(V)
    V ∈ ℝ^{2, T_visual, d_audio/2}

Step 2: Concatenate visual embeddings over hidden dimension
    V ← concat(V[0, ...], V[1, ...])
    V ∈ ℝ^{T_visual, d_audio}

Step 3: Linear interpolation over time dimension
    V ← Linear interpolation(V)
    V ∈ ℝ^{T_audio, d_audio}

Step 4: Fuse audio and visual features with gating mechanism
    AV ← A + gate_function(V)

Output: Feature embedding AV for SS model
```

The gate function may be one of

- Identity (used in TasNet and DPTN experiments)
- Learnable scalar (used in DPTN experiments)
- Tanh postprocessed learnable scalar (used in DPTN experiments)

B. VoiceFilter d-vector extraction from video embeddings

```
Input:
    Video data x ∈ ℝ^{B×T_v×H×W}
    RNN with hidden size H, output size H_out

Step 1: Extract video embedding
    e ← Lipreader(x)
    e ∈ ℝ^{B×T_v×D}

Step 2: Process embedding with GRU
    h, _ ← RNN(e)
    h ∈ ℝ^{B×T_v×H}

Step 3: Compute d-vector using the last RNN step
    d ← Linear(h[:, −1, :])
    d ∈ ℝ^{B×H_out}

Output:
    d-vector d
```

The RNN is either LSTM or GRU