Dual-path Attention is All You Need for Audio-Visual Speech Extraction

Zhongweiyang Xu*, Xulin Fan*, Mark Hasegawa-Johnson

University of Illinois at Urbana-Champaign
{zx21, xulinf2, jhasegaw}@illinois.edu

Abstract

Audio-visual target speech extraction, which aims to extract a certain speaker’s speech from the noisy mixture by looking at lip movements, has made significant progress combining time-domain speech separation models and visual feature extractors (CNN). One problem of fusing audio and video information is that they have different time resolutions. Most current research upsamples the visual features along the time dimension so that audio and video features are able to align in time. However, we believe that lip movement should mostly contain long-term, or phone-level information. Based on this assumption, we propose a new way to fuse audio-visual features. We observe that for DPRNN [1], the inter-chunk dimension’s time resolution could be very close to the time resolution of video frames. Like [2], the LSTM in DPRNN is replaced by intra-chunk and inter-chunk self-attention, but in the proposed algorithm, inter-chunk attention incorporates the visual features as an additional feature stream. This prevents the upsampling of visual cues, resulting in more efficient audio-visual fusion. The result shows we achieve superior results compared with other time-domain based audio-visual fusion models.

Index Terms: Target Speaker Extraction, Audio-Visual Attention

1. Introduction

It is possible to separate multiple sources using permutation invariant training [3]. Results can be improved by applying temporal convolutional networks on time-domain learnable features [4]. Time-domain features have extremely high temporal resolution; dual-path recurrent neural network (DPRNN) models [1] [2] [5] [6] use intra-chunk RNNs to model short-term dependencies, while inter-chunk RNNs model long-term dependencies. These models are now widely available for use as default separators for any speech enhancement or extraction task.

One problem of these speech separation systems is that even when they’re able to separate sources from mixtures, there is no clear definition of which source is the target source. Visual lip movement is able to provide such information. Humans fuse audio and video speech pre-consciously [7], so it is natural to assume that, if a facial video is available, the visible person is the one who should be attended.

Audio-visual speech technology research includes a wide range of topics, including speech recognition, speech reconstruction, speech separation, and speech extraction. Audio-visual speech recognition (AVSR) can be performed on large in-the-wild datasets with reasonable accuracy [8] [9] [10] [11] [12], thus lip movement is able to provide enough information to recognize sentences. In some sense, however, speech recognition may be an easier task than speech extraction, because there exist an infinite number of human utterances which would all correspond to the same sentence.

Complementary to AVSR is the problem of estimating speech from silent videos. Video-to-speech synthesis can be performed with quality that has reasonable word error rate (WER) and perceptual quality (PESQ), but that differs a lot from the original sound [13] [14] [15] [16]. One source of the difference is the lack of speaker information when only using video cues: if speaker information is provided, the model is able to generate utterances with much better performance [17].

Extracting speech from a mixture is a more constrained problem than video-to-speech synthesis. Audiovisual speech extraction is typically performed using mid-level or low-level feature fusion, i.e., concatenating or multiplying the audio and video features [18] [19] [20] [21] [22] [23] [24]. [18] [24] use concatenation fusion of frequency domain audio features and visual features to estimate separation/extraction masks on the mixture STFT. LSTM is used as the masking network. [22] uses the same feature and fusion methods but uses an encoder-decoder structure to directly output target speech. [19] tries to add speaker embedding into the network of [19] to add more robustness. [20] [23] [21] try to incorporate a time domain separation network [4] with mid-fusion into audio-visual speech enhancement. All these mid-level feature fusion algorithms beg the question: at what sampling rate should the fusion take place? Most methods do not downsample the audio features very much in the time dimension, therefore they must upsample the video.

In our paper, we propose a dual-path attention model tackling the audio-visual speech extraction problem. Dual-path attention [2] is similar to the DPRNN model [1], in that features are fused across time both intra-chunk and inter-chunk. In the original DPRNN, this dual-path approach was meant to extract both short-term (local) and long-term (global) information. In addition to the original intuition, this dual-path approach allows audio-visual fusion only in the inter-chunk path, which makes sense because we assume visual cues only contain relatively long term phone-level information.

In each intra-chunk module, we only use self-attention of the audio mixture features to extract the audio’s local information. In each of the inter-chunk steps, audio-visual cross-attention is used to fuse audio and visual features, while self-attention is also used in this module. Our model shows superior results in terms of SI-SNR, even training and testing on variable numbers of speakers for a large dataset collected in the wild.

2. Model Architecture

2.1. Audio-visual Encoder

The audio encoder is a single 1-D convolutional layer with similar structure to TasNet [4], which takes raw audio with length $T$ as input and outputs an STFT-like embedding $E_a \in \mathbb{R}^{T' \times D_a}$ with feature dimension $D_a$ and time dimension $T'$.

$$E_a = \text{conv1d}(x)$$

(1)

On the visual side, we extract the face features $E_v \in \mathbb{R}^{T' \times D_v}$. $T'$ and $D_v$ are different for each dataset.
\[ \mathbb{R}^{T_v \times D_v} \] from video frames using a pretrained MTCNN face detection model \[ \text{[25]} \] and FaceNet model \[ \text{[26]} \], where \( T_v \) denotes number of frames and \( D_v \) is the video feature dimension.

### 2.2. Masking Network

Our model is based on generating a mask on the mixture encoding. The masking network consists of a chunk operation, \( N \) dual-path attention modules, an overlap and add operation to transform the chunk back to the encoding dimension, and then sigmoid activation is applied to get the mask. The overview is shown in Figure 1A.

To prepare the audio embedding for the dual-path attention module, we follow the design of DPRNN \[ \text{[1]} \] and divide \( E_v \) into overlapping chunks with 50% overlap. By concatenating these chunks in a new time dimension, we reshape the matrix \( E_v \in \mathbb{R}^{T' \times D_v} \) into the 3-tensor \( C_0 \in \mathbb{R}^{S \times K \times D_v} \), where \( S \) is the number of overlapping chunks and \( K \) is the chunk size.

The audio 3-tensor and the visual feature matrix are then fed into \( N \) connected dual-path attention modules. The dual-path attention module is covered in detail in section 2.3 below and is the main part of the masking network. Each dual-path attention module outputs two tensors which have the same dimensionality with \( C_a \) and \( E_v \), respectively. Residual connections are applied after each dual-path attention module. Each module’s output is fed as input to the next module, until all \( N \) modules are all applied. Assume \( C_{a0} \) and \( E_{v0} \) are the outputs of the \( i \)th dual-path attention module and \( C_{a0} = C_{a1}, E_{v0} = E_{v1} \), then for \( i \in [1, N] \) we have:

\[
\hat{C}_{ai}, \hat{E}_{vi} \leftarrow \text{DualPathAttention}(C_{ai-1}, E_{vi-1}) \tag{2}
\]

\[
C_{ai} \leftarrow \hat{C}_{ai} + C_{ai-1}, E_{vi} \leftarrow \hat{E}_{vi} + E_{vi-1} \tag{3}
\]

After \( N \) dual-path attention modules, the overlap-add operation with 50% overlap is applied to transform the 3-tensor to a 2-D encoding with the same dimension as the audio encoder output. The 2-D encoding is passed through a sigmoid to get a mask; the Hadamard product of the mask and the audio encoder output is the encoding of the target speech.

### 2.3. Dual-Path Attention

Our dual-path attention module consists of \( N \) dual-path submodules where each dual-path submodule consists of both intra-chunk and inter-chunk attention modules. At each dual-path submodule, the audio features \( C_a \) will first go through the intra-chunk attention module with self-attention. Then the extracted local features will be fed into the inter-chunk attention module to model long-term temporal structure and audio-visual correspondence. We include residual connections after each intra-chunk and inter-chunk module to help convergence. The architecture of the whole dual-path attention module is shown in Figure 2.

#### 2.3.1. Intra-chunk Attention Module

The intra-chunk attention module aims to learn local temporal structure of the chunked audio feature. It consists of \( N_{\text{intra}} \) layers, where each layer takes the chunked audio feature \( C_a \in \mathbb{R}^{S \times K \times D_v} \) as input and outputs a tensor with the same size. Inside each layer, a MultiHeadAttention (MHA) and a position-wise feedforward layer are applied to each of the \( S \) chunks separately, while residual addition and LayerNorm (LN) are applied after both MHA and FeedForward, so for all \( s \in [1, S] \):


\[ C_a[s, :] \leftarrow \text{LN}(C_a[s, :] + \text{MHA}(C_a[s, :])) \]  

\[ C_a[s, :] \leftarrow \text{LN}(C_a[s, :] + \text{FeedForward}(C_a[s, :])) \]

2.3.2. Inter-chunk Attention Module with Modality Fusion

The inter-chunk attention module aims to deal with long-term audio-visual information. Similar to the intra-chunk attention module, each inter-chunk attention module has \( N_{\text{inter}} \) encoder layers. However, different from the intra-chunk attention module, each layer in the inter-chunk attention module receives audio chunked features \( C_a \in \mathbb{R}^{S \times K \times D_a} \) and visual features \( E_v \in \mathbb{R}^{S \times D_v} \) as input, and then applies both self-attention for each modality and cross-attention between modalities. This makes sure that each modality can attend to both itself and the other modality, which allows both information fusion and learning long-term temporal structures.

Self-attention for the visual information \( E_v \in \mathbb{R}^{S \times D_v} \) is a simple attention module as proposed in the original transformer paper [27]. query, key, and value with dimension \( \mathbb{R}^{S \times (h \times D_k)} \) are generated from linear transformations on \( E_v \), where \( S \) is the time dimension, \( D_k \) is the visual feature dimension, \( h \) is the number of heads, and \( D_k \) is the internal dimension for each head. In the following formulas, linear transformation is abbreviated as LT.

\[
\text{query}_{vv}, \text{key}_{vv}, \text{value}_{vv} = LT_{D_v \rightarrow (h \times D_k)}(E_v)
\]

\[
x_{vv} = \text{Attention}(\text{query}_{vv}, \text{key}_{vv}, \text{value}_{vv})
\]

Self-attention for the chunked audio features \( C_a \in \mathbb{R}^{S \times K \times D_a} \) is similar to the visual side, but with an extra K dimension. We just do multthead-attention for each slice on the K dimension \( C_{a_k} \in \mathbb{R}^{S \times D_a} \) and concatenate the results.

\[
\text{query}_{aa_k}, \text{key}_{aa_k}, \text{value}_{aa_k} = LT_{D_a \rightarrow (h \times D_k)}(C_a[; \cdot, k, ;])
\]

\[
x_{aa_k}[; k, :] = \text{Attention}(\text{query}_{aa_k}, \text{key}_{aa_k}, \text{value}_{aa_k})
\]

For cross-attention between modalities, although the chunking step helps us match the long-term time dimension with size \( S \) for the video and audio input, the audio input still has an extra short-term dimension with size \( K \) (chunk size) compared to the video input. We find that doing cross-attention between each slice of the audio chunked features and the video features requires a large amount of computation and does not provide a significant performance gain. That’s because for attention computation with long-term information, the attention computation is similar for each audio feature slice \( C_a[; k, ;] \). So to do cross-attention efficiently, we choose to collapse the K dimension of the audio features by a single \( 1 \times 1 \) convolutional operation and use the collapsed audio feature \( C_{va} \in \mathbb{R}^{S \times D_a} \) as the audio modality for cross-attention.

\[
C_{va} = 1 \times 1 \text{Conv}(C_a)
\]

\[
\text{query}_{va} = LT_{D_a \rightarrow (h \times D_k)}(E_v)
\]

\[
\text{query}_{va}, \text{key}_{va}, \text{value}_{va} = LT_{D_a \rightarrow (h \times D_k)}(C_{va})
\]

All the keys, values, and queries used in cross-attention have the same dimension \( \mathbb{R}^{S \times (h \times D_k)} \), so the cross-attention computation is just a simple attention computation similar to the one in the visual information self-attention step:

\[
x_{va} = \text{Attention}(\text{query}_{va}, \text{key}_{va}, \text{value}_{va})
\]

We add self-attention result and cross-attention result for each modality. On the visual side, \( x_{vv} \) and \( x_{va} \) have same dimension \( \mathbb{R}^{S \times (h \times D_k)} \), so simple element-wise addition works. On the audio side where self-attention result \( x_{aa_k} \) has an additional K (Chunk Size) dimension, we need to broadcast \( x_{aa_k} \) in this dimension so that the result of addition has dimension \( \mathbb{R}^{S \times K \times (h \times D_k)} \). Then we use a linear transformation to turn the audio and video embeddings with features dimension of \( (h \times D_k) \) back into \( D_a \) and \( D_v \) respectively:

\[
C_a \leftarrow \text{LN}(C_a + LT_{(h \times D_k) \rightarrow D_a}(x_{aa_k} + x_{va}))
\]

\[
E_v \leftarrow \text{LN}(E_v + LT_{(h \times D_k) \rightarrow D_v}(x_{va} + x_{va}))
\]

\[
C_a \text{ and } E_v \text{ are further fed into a position-wise feedforward layer consisting of two consecutive fully connected layers. Add and Norm operations are applied before and after the feedforward layer. The pipeline is demonstrated in the formula below:}
\]

\[
C_a \leftarrow \text{LN}(C_a + \text{FeedForward}(C_a))
\]

\[
E_v \leftarrow \text{LN}(E_v + \text{FeedForward}(E_v))
\]

2.4. Audio Decoder

We do the overlap-add operation with 50% overlap on the output of the final inter-chunk attention module to transform the chunked audio features with dimension \( \mathbb{R}^{S \times K \times D_a} \) back into \( \mathbb{R}^{T' \times D_a} \), as mentioned in section 2.2. Then we apply sigmoid on the embedding to output a mask to be applied to the mixture encoding. Finally, the masked audio features is fed into the decoder which is a transposed convolutional layer to output the separated audio. Below \( \odot \) represents Hadamard product.

\[
M = \text{Sigmoid}(\text{Overlap\&Add}(C_a \odot))
\]

\[
x_{\text{extracted}} = \text{TransposedConv1d}(M \odot E_a)
\]

3. Experiments

3.1. Experimental Settings

Hyperparameters are set as follows: \( N = 3 \), \( N_{\text{intra}} = 4 \), \( N_{\text{inter}} = 4 \), \( K = 160 \), \( D_a = 256 \), \( D_v = 512 \). In the video processing stage, MTCNN extracts a 160×160 patch for each image frame. The pretrained FaceNet model then further processes the video frames and turns each frame into a feature embedding with 512 dimensions. On the audio side, the audio encoder has a window size of 16, stride of 8, and output dimension of 256. Cropped facial regions are then fed together with audio features into the sequence of \( N = 3 \) dual-path attention modules, each of which contains \( N_{\text{intra}} = 4 \) intra-chunk and \( N_{\text{inter}} = 4 \) inter-chunk attention modules.

The configuration of multi-headed attention is the same for both intra-chunk and inter-chunk modules, each with \( h = 8 \) heads of internal dimension \( D_k = 64 \). After the attention calculation, linear layers are applied to the attention output with respectively 256 dimensions for audio and 512 dimensions for video. The FeedForward layer is just two linear layers with the intermediate feature dimension \( D_L = 1024 \). For pretraining process with the GRID dataset [23], the attention field covers the entire sequence. For later training and validation with the LRS3 dataset where each utterance has an 8-second duration, we apply a mask during attention calculation to limit the attention field to a 5 second duration around the current timestep.
3.2. Pretraining

Our final evaluation of the audio-visual model is done on the LRS3 dataset [29]. However, we notice that LRS3 is a difficult task for the audio-visual model for three reasons. First, the video channel of LRS3 is not ideal: many image frames are non-frontal faces, and some miss the face entirely. Second, the audio channel of LRS3 has significant reverberation which makes it harder to separate out the original audio. Third, the number of speakers in LRS3 is very large, therefore it is rare for the same speaker to be repeated from one batch to the next, which makes it harder for the network to acquire useful gradients during the initial training phase.

We therefore choose to pretrain our model on the GRID corpus [28], a smaller and easier dataset, before moving on to LRS3. The GRID corpus consists of 33 speakers (one speaker missing) where each speaker has 1000 3-second long utterances of simple sentences and video capturing the frontal face area. We pretrain on GRID for 10 epochs with the same training configurations mentioned below before moving to LRS3 dataset.

3.3. Training

For the LRS3 dataset, we choose a subset of 33,000 utterances out of the whole LRS3 dataset and clip the audio and video to be 8 seconds per utterance, with a 16 kHz audio sampling rate and 25 video frames per second. From the 33000 extracted utterances, we use 27000 utterances for training, 3000 for validation, and 3000 for testing. Thus we have 60 hours of LRS3 training data, 6.67 hours for validation, and 6.67 hours for testing. The 25 video frames per second. From the 33000 extracted utterances, we use 27000 utterances for training, 3000 for validation, and 3000 for testing. Thus we have 60 hours of LRS3 training data, 6.67 hours for validation, and 6.67 hours for testing.

We use SI-SNR [32] as both the training objective and evaluation metric. We use Adam optimizer [33] with initial learning rate 1e-4. We decay the learning rate by a factor of 2 when the validation SI-SNR stops improving for three consecutive epochs. The whole training process is in mixed precision with batch size 1 on one single V100 GPU. The entire training process lasts for 50 epochs.

4. Results and discussion

We evaluate our model on our LRS3 testing set with 2, 3, 4, or 5 speakers respectively. Since we were unable to find published models for any previous audio-visual speaker extraction algorithm, we created audio-visual and audio-only baselines to match published near-state-of-the-art systems using ConvTasnet [4]. Our audio-visual baseline is shown as AV-ConvTasnet in Table 1. The AV-ConvTasnet baseline uses the same 512-dimensional visual feature as our model, but its audio encoder is set to $N = 512, L = 32$, following the notation and recommendations in [4]. Then we upsample the visual features by repeating in the time dimension and concatenate them with the output of the audio encoder. The concatenated audio-visual features are then fed into the Temporal Convolutional Network with $B = 256, H = 1024, Sc = 256, P = 3, X = 8, R = 3$, also following the notation in [4]. The pre-training and training are all performed identically to our model.

The audio-only baseline is an audio-only ConvTasnet, trained on our LRS3 training set using the permutation-invariant training (PIT) [3] SI-SNR objective. The configuration is $N = 512, L = 32, B = 128, H = 512, Sc = 128, P = 3, X = 8, R = 3$ following the notation in [4]. Since the conventional PIT training assumes a fixed number of sources, we train four models corresponding to 2, 3, 4, 5 speakers. This means the audio-only baseline models assumes a known number of speakers. The final performance and comparison are summarized in Table 1.

Table 1: Speech Extraction Scores for our Model and two Baselines.

| MODELS          | 2-SPK | 3-SPK | 4-SPK | 5-SPK | 512-SPK |
|-----------------|-------|-------|-------|-------|---------|
| Mixture         | -0.05 | -3.36 | -5.37 | -6.66 |
| ConvTasnet      | 11.01 | 10.11 | 9.50  | 9.03  |
| ConVtasnet      | 11.23 | 10.18 | 9.49  | 9.00  |
| Ours            | 18.10 | 17.41 | 17.01 | 16.60 |

As shown in Table 1 our model surpasses the two baseline models by a large margin. It’s able to achieve 18.1dB of SI-SNR improvement for the 2-speaker case where the target speech is actually recorded in the wild with reverberation. Also, the SI-SNR improvement does not degrade much when the number of speakers increases, thus our model has the potential to behave robustly in real-world settings.

5. Conclusion

In this paper, we propose a new approach to tackle modality fusion when the resolutions of different modalities vary. In addition, we propose a speech extraction model utilizing this technique which achieves significant gain over our baseline models.

By using the chunking process proposed by Dual-path RNN [1], we are able to match the $S$ (Number of chunks) dimension in the high-resolution modality with the time dimension of the low-resolution modality. We are able to match the time resolution between the audio and visual modalities naturally and Inter-chunk attention with modality fusion can be done readily without additional upsample operations. Utilizing this dual-path attention mechanism as our separation module, our audio-visual model is able to achieve superior performance over other time-domain speech extraction systems.

In the future, more serious conditions like environmental noise, reverberation, or multi-lingual mixtures can be further tested. Also, certain modifications need to be done to make the model run in real-time and in a semi-causal manner.
6. References

[1] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation,” 2020.

[2] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, “Attention is all you need in speech separation,” 2021.

[3] M. Kolbeek, D. Yu, Z.-H. Tan, and J. Jensen, “Multi-talker speech separation with utterance-level permutation invariant training of deep recurrent neural networks,” 2017.

[4] 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, p. 1256–1266, Aug. 2019. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2019.2915167

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

[6] S. E. Chazan, L. Wolf, E. Nachmani, and Y. Adi, “Single channel voice separation for unknown number of speakers under reverberant and noisy settings,” 2020.

[7] H. McGurk and J. MacDonald, “Hearing lips and seeing voices,” Nature, vol. 264, no. 5588, pp. 746–748, 1976.

[8] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, “Lip reading sentences in the wild,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.367

[9] 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, p. 1–1, 2019. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2018.2889052

[10] B. Shi, W.-N. Hsu, K. Lakhotia, and A. Mohamed, “Learning audio-visual speech representation by masked multimodal cluster prediction,” in International Conference on Learning Representations, 2022. [Online]. Available: https://openreview.net/forum?id=Z1Qlm11uOM

[11] P. Ma, S. Petridis, and M. Pantic, “End-to-end audio-visual speech recognition with conformers,” ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7613–7617, 2021.

[12] B. Garcia, B. Shillingford, H. Liao, O. Siohan, O. de Pinho Forin Braga, T. Makino, and Y. Assael, “Recurrent neural network transducer for audio-visual speech recognition,” in Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop, 2019. [Online]. Available: https://arxiv.org/abs/1911.03480

[13] H. Akbari, H. Arora, L. Cao, and N. Mesgarani, “Lip2audspec: Speech reconstruction from silent lip movements video,” CoRR, vol. abs/1710.09798, 2017. [Online]. Available: http://arxiv.org/abs/1710.09798

[14] K. Vougioukas, P. Ma, S. Petridis, and M. Pantic, “Video-driven speech reconstruction using generative adversarial networks,” in INTERSPEECH, 2019.

[15] A. Ephrat, T. Halperin, and S. Peleg, “Improved speech reconstruction from silent video,” 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 455–462, 2017.

[16] A. Ephrat and S. Peleg, “Vid2speech: speech reconstruction from silent video,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017.

[17] K. R. Prajwal, R. Mukhopadhyay, V. Namboodiri, and C. V. Jawahar, “Learning individual speaking styles for accurate lip to speech synthesis,” CoRR, vol. abs/2005.08209, 2020. [Online]. Available: https://arxiv.org/abs/2005.08209

[18] T. Afouras, J. S. Chung, and A. Zisserman, “The conversation: Deep audio-visual speech enhancement,” 2018.

[19] ——, “My lips are concealed: Audio-visual speech enhancement through obstructions,” 2019.

[20] J. Wu, Y. Xu, S.-X. Zhang, L.-W. Chen, M. Yu, L. Xie, and D. Yu, “Time domain audio visual speech separation,” 2019.

[21] H. Sato, T. Ochiai, K. Kinoshita, M. Delcroix, T. Nakatani, and S. Araki, “Multimodal attention fusion for target speaker extraction,” 2021.

[22] A. Gabbay, A. Shamir, and S. Peleg, “Visual speech enhancement,” 2018.

[23] Y. Luo, J. Wang, L. Xu, and L. Yang, “Multi-Stream Gated and Pyramid Temporal Convolutional Neural Networks for Audio-Visual Speech Separation in Multi-Talker Environments,” in Proc. Interspeech 2021, 2021, pp. 1104–1108.

[24] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, “Listening to listen at the cocktail party,” ACM Transactions on Graphics, vol. 37, no. 4, p. 1–11, Aug. 2018. [Online]. Available: http://dx.doi.org/10.1145/3197517.3201357

[25] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” IEEE Signal Processing Letters, vol. 23, no. 10, p. 1499–1503, Oct. 2016. [Online]. Available: http://dx.doi.org/10.1109/LSP.2016.2603542

[26] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298682

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” CoRR, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[28] M. Cooke, J. Barker, S. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” The Journal of the Acoustical Society of America, vol. 120, no. 5, pp. 2421–2424, 2006. [Online]. Available: https://doi.org/10.1121/1.2229005

[29] T. Afouras, J. S. Chung, and A. Zisserman, “Lrs3-ted: a large-scale dataset for visual speech recognition,” in arXiv preprint arXiv:1809.04696, 2018.

[30] 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), 2015, pp. 5206–5210.

[31] 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.

[32] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “Sdr–listen at the cocktail party,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 626–630.

[33] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.