How to Design a Three-Stage Architecture for Audio-Visual Active Speaker Detection in the Wild

Okan Köpüklü[1], Maja Taseska[2], Gerhard Rigoll[1]

[1] Technical University of Munich
[2] Microsoft Corporation

Abstract

Successful active speaker detection requires a three-stage pipeline: (i) audio-visual encoding for all speakers in the clip, (ii) inter-speaker relation modeling between a reference speaker and the background speakers within each frame, and (iii) temporal modeling for the reference speaker. Each stage of this pipeline plays an important role for the final performance of the created architecture. Based on a series of controlled experiments, this work presents several practical guidelines for audio-visual active speaker detection. Correspondingly, we present a new architecture called ASDNet, which achieves a new state-of-the-art on the AVA-ActiveSpeaker dataset with a mAP of 93.5% outperforming the second best with a large margin of 4.7%. Our code and pretrained models are publicly available [1].

1. Introduction

Fusion of audio and video modalities has been shown to provide promising solutions to long-standing challenging problems. These include among others, speaker diarization [16], biometrics [7], and action recognition [15, 38]. Similar to other tasks, Audiovisual Active Speaker Detection (AV-ASD) has also long been studied in literature [9, 10]. A particularly challenging flavor of this problem is AV-ASD in the wild, where speech is to be detected and assigned to one of possibly multiple active speakers at each instant in time. Clearly, fusing the complementary discriminative information from audio and video modalities is crucial: visual-only approaches can easily be mistaken by other face/mouth motions such as eating, yawning or emotional expressions. Audio-only approaches, although able to perform source clustering and separation [18, 46], aren't sufficiently robust to count the number of speakers and assign speech to the correct source. This is especially challenging with a single microphone input in acoustically adverse conditions, typically encountered in practice.

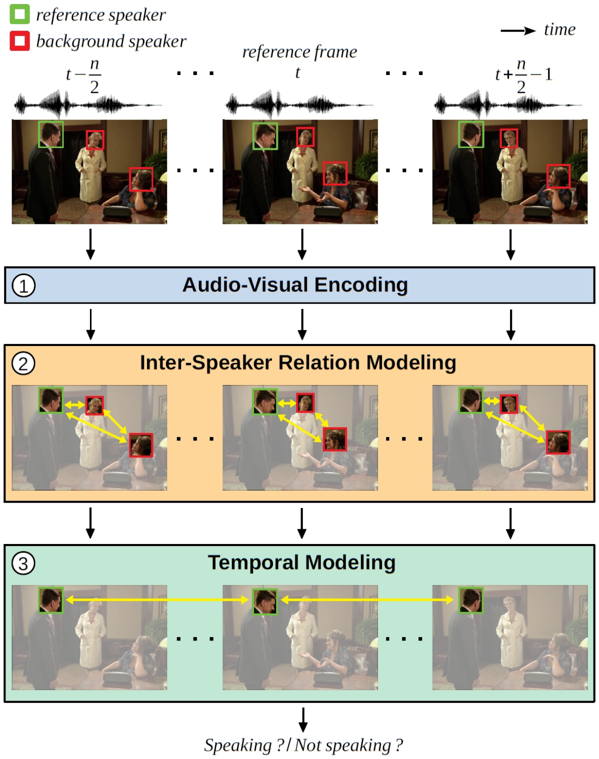

Figure 1. Audio-visual active speaker detection pipeline. The task is to determine if the reference speaker at frame t is *speaking* or *not-speaking*. The pipeline starts with audio-visual encoding of each speaker in the clip. Secondly, inter-speaker relation modeling is applied within each frame. Finally, temporal modeling is used to capture long-term relationships in natural conversations. Examples are from AVA-ActiveSpeaker dataset [42].

Recently, the AVA-ActiveSpeaker dataset [42] provided the first large-scale standard benchmark for audio-visual active speaker detection in the wild. Recent research [1, 32] indicates that active speaker detection in the wild requires (i) integration of audio-visual information for each speaker, (ii) contextual information that captures inter-speaker relationships, and (iii) temporal modeling to exploit long term relationships in natural conversation. In this paper, we con-

[1] https://github.com/okankop/ASDNet

solidate this three-stage pipeline for audio visual speaker detection, illustrated in Fig. 1, and study the importance of each stage in detail.

Contributions. We propose a novel three-stage pipeline for audio-visual active speaker detection in the wild. Our architecture, named ASDNet, sets a new state-of-the-art result on AVA-ActiveSpeaker dataset with a 93.5% mAP, and outperforms the second best method [32] with a large margin of 4.7% mAP (Section 4.5). As part of ASDNet, we propose

(1) architectures for the audio and video backbones of the audio-visual encoder (Section 3.2), that haven't been previously explored for active speaker detection;

(2) a simple, yet effective inter-speaker relation modeling mechanism (Section 3.3);

(3) In addition, we provide detailed ablation study and guidelines for tuning all components of ASDNet. The study includes comparison to the state of the art for the two novel components mentioned above, as well as evaluation of various Recurrent Neural Network (RNN) architectures for temporal modeling (Section 4.2.).

2. Related Work

We present the related work in two parts: (i) audio-visual feature extraction in various applications, and (ii) contributions that address active speaker detection in the wild and its challenges.

2.1. Audio-visual feature extraction

Audio. A common approach to extract features in speech and audio research in different applications, is to use Convolutional Neural Networks (CNNs) and RNNs with log-Mel or Short-Time Fourier Transform (STFT) spectrograms as inputs [13]. The popularity of these fixed transforms is due to their success in traditional speech and audio processing and the fact that they extract relevant information from first principles. Furthermore, the image-like configuration of the spectrograms allows employing network architectures well-known from computer vision applications. Particularly, in AV-ASD, this allows to use similar audio and video backbone architectures [1, 32].

Based on the interpretation of CNNs as a data-driven filterbank, researchers have applied CNNs directly on the audio waveforms to capture discriminative information for the task at hand [12, 30]. Such an approach in the context of AV-ASD has been used for an audio backbone in [2]. However, these approaches need much more data and computational resources that the ones exploiting spectrograms. With the goal to exploit the best from both worlds, researchers have come up with learnable, but yet constrained transformations of raw audio data. Examples include Harmonic CNNs used for music tagging, and the SincNet architecture proposed in [41]. The latter was successfully used in several audio applications [27,33,36]. To our best knowledge, this promising architecture hasn't been used in the context of AV-ASD.

Video. Active speaker detection using only video modality can be viewed as action recognition task. Prior to CNNs, action recognition research was dominated by hand-crafted features [28,29,49], combined with Fisher Vector representations [39] or Bag-of-Features histograms [8]. Ever since AlexNet [26] won the ImageNet Challenge [43], hand-crafted features were mostly abandoned in favor of features extracted by CNNs. This trend extended to video analysis tasks as well, including action recognition. Initially, due to the absence of a large-scale video dataset, architectures for action recognition could benefit from pretraining on the very-large ImageNet dataset [11]. The first intuitive approach was to treat video frames as multi channel input to 2D-CNNs [21, 45]. Other approaches include extraction of frame-level features with a 2D-CNN, followed by a spatiotemporal modeling mechanism [23].

With the availability of large-scale video datasets such as Kinetics [3], Moments-in-Time [37], and Jester [34], 2D-CNNs were replaced by 3D-CNNs, to better capture temporal information and motion patterns within video frames. A 3D-CNN architecture was first proposed in [20] by Ji *et al.* Since then, many 3D-CNN architectures for video recognition tasks followed, such as C3D [47], I3D [3], P3D [40], R(2+1)D [48], SlowFast [14], etc. In [17], the effect of dataset size on performance is investigated for several 3D-CNN architectures. Inflated versions of popular resource-efficient 2D-CNN architectures are analyzed for video classification tasks in [24]. In this work, we explore variants of 3D-CNNs for the AV-ASD task.

Fusion. The extracted modality-specific features can be combined at data level [25], feature level [35] or decision level [45]. The fusion that we apply in this work can be considered as feature level fusion, since we keep processing fused features at inter-speaker relation modeling and temporal modeling mechanisms afterwards.

2.2. Active speaker detection in the wild

Audio-visual active speaker detection is a specific case of source separation [4, 50], where audio and visual signals are leveraged jointly to assign a speech segment to its speaker. For this task, initial approaches [9, 10] use datasets collected in controlled environments. With the availability of AVA-ActiveSpeaker dataset [42], the research community was able to shift towards active speaker detection in the wild.

Audio-visual feature extraction is the first step in top-performing frameworks for active speaker detection [1, 6, 32, 42, 54]. A two-backbone approach has established it-

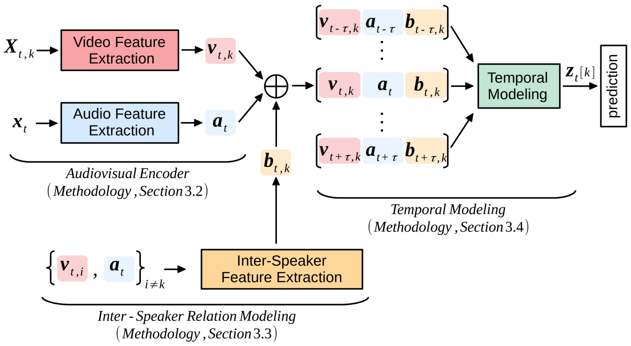
X t,k
Video Feature Extraction
v t,k
x t
Audio Feature Extraction
a t
Audiovisual Encoder (Methodology, Section 3.2)
b t,k
v t-τ,k a t-τ b t-τ,k
v t,k a t b t,k
v t+τ,k a t+τ b t+τ,k
Temporal Modeling
z t[k]
prediction
Temporal Modeling (Methodology, Section 3.4)
{v t,i , a t} i≠k
Inter-Speaker Feature Extraction
Inter-Speaker Relation Modeling (Methodology, Section 3.3)

Figure 2. Overview of the three-stage pipeline in ASDNet.

self as a standard architecture due to its versatility [45]. With a good audio-visual feature extraction and RNN-based temporal modeling, the authors in [6] achieved competitive performance on the AVA-ActiveSpeaker dataset. Temporal modeling constitutes an integral part of recent active speaker detection pipelines [1, 6, 42, 54]. Often neglected is the context information that can be obtained by modeling inter-speaker relationships. Researchers have only recently proposed methods to exploit the context information [1,32].

3. Methodology

Drawing inspiration from the insights in recent research, we seek to establish a general pipeline that incorporates audio-visual encoding, inter-speaker (context) modeling, and temporal modeling. By designing an appropriate architecture for each component, we are able to exceed the state-of-the art performance on the AVA-ActiveSpeaker dataset.

3.1. Notation and Overview

Let K denote the total number of speakers in a given clip. The data available to the active speaker detection system at time t is a set $\mathcal{X}_t = \{\boldsymbol{X}_{t,1}, \boldsymbol{X}_{t,2}, \dots \boldsymbol{X}_{t,K}, \boldsymbol{x}_t\}$, where $\boldsymbol{X}_{t,k} \in \mathbb{R}^{n \times 3 \times d_h \times d_w}$ is a tensor of face crops corresponding to the k-th speaker. The height and width of the face crops are denoted by d_h and d_w, 3 is the RGB channels and n is the number of consecutive face crops centering time instant t. The vector $\boldsymbol{x}_t$ contains the samples of the audio track corresponding to the duration of the video input. Given the input data, the objective is to produce a binary vector $\boldsymbol{z}_t$, where $\boldsymbol{z}_t[k] = 1$ if the k-th speaker is detected as *speaking* at time frame t, and $\boldsymbol{z}_t[k] = 0$ otherwise.

A high-level overview of our pipeline that maps the raw data $\mathcal{X}_t$ to the predictions $\boldsymbol{z}_t$ is illustrated in Fig. 2. Next, in Sec. 3.2-3.4, we zoom in on the design of the three pipeline components. In Sec. 3.5, we discuss the training strategy that enables an end-to-end inference: from face crops and an audio waveform, to a prediction *speaking* or *not speaking* for each speaker in the video clip.

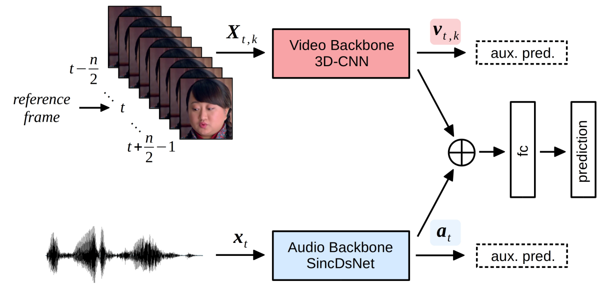

Figure 3. Audio-visual encoder architecture. Visual input $\boldsymbol{X}_{t,k}$ and audio input $\boldsymbol{x}_t$ are fed to the respective backbones to produce features $\mathbf{v}_{t,k}$ and $\mathbf{a}_t$. A concatenated feature vector $\mathbf{v}_{t,k} \bigoplus \mathbf{a}_t$ is fed to a fully connected layer which produces a prediction if speaker k is speaking at time t. Prediction heads are removed after training and are not part of the global picture in Fig. 2.

3.2. Audio-Visual Encoder Architecture

Our audio-visual encoder is illustrated in Fig. 3. The stack of face thumbnails $\boldsymbol{X}_{t,k}$ consists of n frames, $X_{t-\frac{n}{2},k}, \dots, X_{t,k}, \dots, X_{t+\frac{n}{2}-1,k}$, and the size of the audio input vector $\boldsymbol{x}_t$ is determined by the number of video frames, the video frame rate, and the audio signal sampling rate. The encoder produces an embedding vector by concatenating the modality-specific embeddings

$$\mathbf{v}_{t,k} = f_v(\boldsymbol{X}_{t,k}; w_v), \quad \mathbf{a}_t = f_a(\boldsymbol{x}_t; w_a), \tag{1}$$

where f_v and f_a are neural networks with trainable parameters w_v and w_a, respectively.

The concatenated features $\mathbf{v}_{t,k} \bigoplus \mathbf{a}_t$ are fed into a fully connected layer to get final predictions. To train the audio-visual encoder, we apply cross-entropy loss between the predictions and ground-truth labels. To ensure that consistent discriminative features are extracted from both modalities, we apply auxiliary classification networks after each backbone, following previous works [1, 32, 42]. The auxiliary networks are also trained with cross-entropy loss. The final loss becomes $L_{final} = L_{av} + L_a + L_v$. After training is completed, supervision heads are discarded and only the audio-visual backbone is used to extract features $\mathbf{v}_{t,k}$ and $\mathbf{a}_t$ for all speakers and time instants.

While the described high-level architecture is similar to that of existing audio-visual encoders [1,32,42], our contribution lies in the choice and design of the video and audio backbones, discussed next.

Video backbone. Movements of mouth and facial muscles are indicative of active speaking. Hence, to fully exploit the available video data, it is important to accurately model motion patterns. To this end, we propose using a 3D-CNN as the visual encoder function f_v, in contrast to the state-of-the-art approacches that apply 2D-CNNs [1,6,32,42,54]. As part of our study, we experimented with various resource-efficient and high-performance 3D-CNN

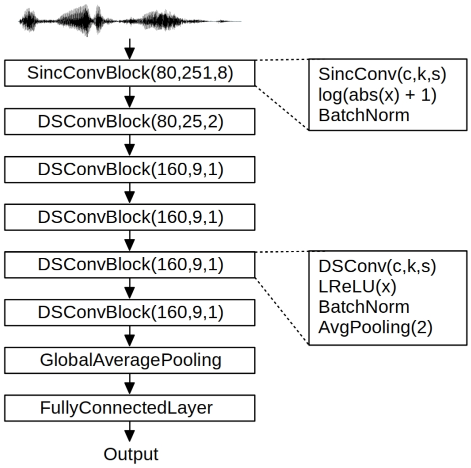

Figure 4. Audio encoder utilizing Sinc Convolutions (SincConv) and Depthwise Separable Convolutions (DSConv). The convolution parameters, *c, k, s* corrspond to the number of output channels, kernel size, and stride, respectively.

architectures [24] and found 3D-ResNeXt-101 to be the best performing candidate for our video backbone. Further insights from our investigation are discussed in Section 4.1.

Audio backbone. For the audio encoding backbone, the majority of existing AV-ASD approaches [1, 6, 32, 42, 54] extract Mel Frequency Cepstral Coefficients (MFCC) from the raw signal, and use the MFCCs as input to 2D-CNNs. In contrast, we propose using an audio backbone architecture that directly operates on raw audio signal via sinc convolutions [41]. In this manner, the system doesn't require a dedicated filterbank and directly exploits all available audio information. This is not the case in existing approaches, where phase information is often discarded after the filterbanks. After sinc convolutions, we apply log-compression, i.e., $y = log(abs(x) + 1)$. This non-linearity has been effective in other raw audio processing tasks as well [27, 53]. The features extracted by the sinc-convolutions are used as input to Depthwise Separable Convolutional (DSConv) blocks with Leaky-ReLU nonlinearity [52]. Our full audio encoder architecture, referred to as *SincDSNet*, is shown in Fig. 4. Features after the global average pooling are extracted as the audio features $\mathbf{a}_t$. The advantage of the proposed raw-audio backbone over existing feature-based backbones is experimentally demonstrated in Section 4.1.

3.3. Inter-Speaker Relation Modeling (ISRM)

The audio-visual encoder extracts features for each individual speaker separately - the features for speaker k do not contain visual information from the remaining speakers in the frame. However, features belonging to background speakers contain complementary information that improves the system performance, as shown in [1]. In this paper, we propose a method to aggregate information from the background speakers efficiently.

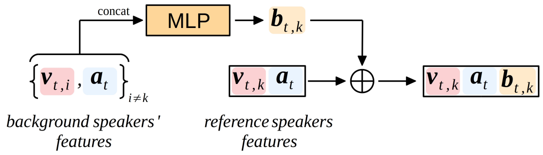
concat
MLP
$b_{t,k}$
$\{v_{t,i}, a_t\}_{i \neq k}$
background speakers' features
$v_{t,k}$ a_t
reference speakers features
$v_{t,k}$ a_t $b_{t,k}$

Figure 5. Inter-speaker relation modeling architecture. For reference speaker k at time instant t, we extract background features $\mathbf{b_{t,k}}$ by passing the concatenated features of background speakers through one layer MLP. Extracted features are then concatenated to reference speakers video features and audio features.

Consider a reference speaker k and m background speakers in the scene at time t. The output of the audio-visual encoder for the reference speaker is $[\mathbf{v}_{t,k}, \mathbf{a}_t]$. To incorporate information from background speakers, we propose to extract an additional feature vector $\mathbf{b}_{t,k}$ using a single-layer perceptron, as illustrated in Fig. 5. The input to the MLP are the concatenated audio-visual embeddings from all background speakers at time t. Note that the number m is fixed from the system's perspective: if there are less than m background speakers at time t, the encoder features are populated with zero vectors. If there are more than m speakers, only m are randomly selected. In this manner, the input dimension of the MLP is fixed. The final feature vector $[\mathbf{v}_{t,k}, \mathbf{a}_t, \mathbf{b}_{t,k}]$ is fed to the temporal model. An experimental study of the proposed ISRM, and comparison to the approach in [1] is provided in Section 4.2.

3.4. Temporal Modeling

Speaking is a coherent action in time: if a person is speaking at previous or future time instants, it is likely that the person is speaking at the current time instant. This is also valid for *remaining silent* action. Therefore, temporal modeling is crucial for accurate active speaker detection.

We experimented with several RNN-based temporal modeling architectures: Long Short-Term Memory (LSTM) [19], Gated Recurrent Unit (GRU) [5], Simple Recurrent Unit (SRU) [31] and their bidirectional versions. For the uni-directional methods, the reference frame is at the end of the input, while for the bidirectional methods it is at center of the input. The hidden state vector of the recurrent block at the reference frame is fed to a fully connected layer to produce a binary output $\mathbf{z}_t[k] \in \{0, 1\}$ (i.e. active speaker or not). In case speakers' features are not available for the selected time window, similar to [1] we apply same padding to the beginning or to the end. Out of all methods, Bidirectional-GRU performs best and becomes our final choice in temporal modeling stage.

3.5. Training Details

Training Audio-Visual Encoding Backbones. We train our audio-visual encoder using ADAM optimizer [22] for

Audio Backbone	Video Backbone	mAP
2D-ResNet-18	2D-ResNet-18	79.0
2D-ResNet-18	3D-ResNet-18	83.9
SincDSNet	2D-ResNet-18	80.8
SincDSNet	3D-ResNet-18	**86.1**

Table 1. Performance comparison of different audio and video backbones. Input length of 8-frames is used for all evaluations.

Audio Backbone	Params	MFLOP
SincDSNet	0.15M	13.8
2D-ResNet-18	11.2M	19.2

Table 2. Complexity comparison of different audio backbones.

70 epochs. Batch size is selected as highest possible number that fits to a single NVIDIA Titan XP GPU for different backbones. However, gradients are accumulated reaching to effective batch size of 192 before doing backward propagation. The learning rate is initialized with 3×10^{-4} and dropped by a factor of 10 at every 30 epochs. For video input, we apply random cropping, random horizontal flipping and color transformations as data augmentation at the training time. Finally, video input is reshaped to the resolution of 160×160. The audio signals are sampled at 16 kHz. 3D-CNNs are pretrained on Kinetics [3], 2D-CNNs are pretrained with ImageNet [11], and SincDSNet is trained from scratch. Once the training is finished, prediction heads are discarded and the features $\mathbf{v}_{t,k} \in \mathbb{R}^{512}$ and $\mathbf{a}_t \in \mathbb{R}^{160}$ are used to train the ISRM and the temporal model.

Training ISRM and Temporal Modeling. We used ADAM optimizer with cross-entropy loss to train the ISRM and the temporal model. We train for 10 epochs, with batch size of 256. The learning rate is initialized with 3×10^{-6} and dropped by 10 at 5th epoch. The MLP in the ISRM extracts the feature $\mathbf{b}_{t,k} \in \mathbb{R}^{128}$ independent from the number of background speakers. For the temporal model, we used two recurrent layers with hidden state dimension of 128, which experimentally proved to be optimal for our system.

Our final architecture ASDNet is implemented in PyTorch and all experiments are performed using a single NVIDIA Titan Xp GPU.

4. Experiments

Dataset. The AVA-ActiveSpeaker dataset [42] is the first audio-visual active speaker dataset collected in the wild. It contains 262 15-minute videos from Hollywood movies, recorded at 25-30 fps, 120 of which are used for training, 33 for validation, and 109 for testing. The videos consist of 3.65 million human-labeled frames, where face crops belonging to the same speaker are aggregated to create face tracks, and each face crop is annotated with *speaking* or *not-speaking* label. This results in 38.5 hours of face tracks with the corresponding audio signal. The number of speakers in the videos is time-varying, and a significant portion of face crops has resolution less than 100 pixels, making the dataset considerably challenging.

	Video Backbone	Params	GFLOP	mAP
32-f	3D-ResNeXt-101	48.6M	13.2	**88.9**
	3D-ResNet-18	33.2M	10.3	87.4
16-f	3D-ResNeXt-101	48.6M	14.1	**88.9**
	3D-ResNet-18	33.2M	11.2	87.5
8-f	3D-ResNeXt-101	48.6M	13.2	86.7
	3D-ResNet-18	33.2M	10.3	86.1
	2D-ResNet-18	11.2M	0.9	80.8
	3D-MobileNetV1 2.0x	13.9M	0.6	81.6
	3D-MobileNetV2 1.0x	2.1M	0.7	85.1
	3D-ShuffleNetV1 2.0x	4.6M	0.7	85.0
	3D-ShuffleNetV2 2.0x	3.9M	0.6	84.2

Table 3. Comparison of video backbones for different clip lengths. SincDSNet is used at the audio backbone, and face crop resolution is 160 × 160.

Evaluation Metric. We use the official ActivityNet evaluation tool that computes mean average precision (mAP). Unless stated otherwise, we use AVA-ActiveSpeaker validation set for our evaluations.

4.1. Audio-Visual Encoder Evaluation

In this section, we investigate the advantage of the proposed audio and video backbones, compared to backbones used in state-of-the-art active speaker detection systems. The encoder architecture is of utmost importance: the overall performance of the AV-ASD pipeline can only be as good as the extracted features. For these experiments, ISRM and temporal modeling are not used.

Which encoder architectures should be used? Following recent works [1, 6, 32, 42, 54], we take 2D-ResNet-18 architecture as the audio and video backbones of a baseline encoder. Inputs to the video backbone are stacked face crops, and inputs to the audio backbone are MFCCs, corresponding to a length of eight frames. This baseline achieves 79.0 mAP as shown in Table 1.

To demonstrate the benefit of applying 3D convolution kernels, we keep the baseline audio backbone and replace 2D-ResNet-18 by 3D-ResNet-18. *This change alone brings improvement of 4.9 mAP over the baseline.* The improvement is achieved solely due to the ability of the 3D convolution kernels to capture motion patterns in the video data.

Similarly, to evaluate the benefit of SincDSNet as the proposed audio backbone, we keep the baseline video back-

# Speakers	0	1	2	3	4	5
mAP	92.6	93.1	**93.4**	**93.4**	**93.4**	**93.3**

Table 4. Performance of inter-speaker relation modelling for different number of background speakers.

Method	Temporal Model	mAP
NonLocal [1]		87.2
NonLocal [1]	✓	92.8
ISRM (ours)		89.0
ISRM (ours)	✓	**93.4**

Table 5. Comparison of inter-speakers relation modeling methods.

bone and replace the 'MFCC + 2D-ResNet-18' audio backbone by SincDSNet. *This change brings improvement of 1.8 mAP over the baseline*, thanks to the partially learnable feature extraction by SincDSNet, operating on the raw audio data. Importantly, SincDSNet has 75 times less parameters than 2D-ResNet-18 and requires less floating point operations (FLOPs), as shown in Table 2.

Finally, our audio-visual encoder that uses both 3D-ResNet-18 and SincDSNet as backbones, achieves 7.1 mAP improvement over the baseline.

Can we use resource-efficient video encoders? One can attribute the performance boost achieved by 3D-ResNet-18 backbone to its increased number of parameters and FLOPs. Therefore, we have used several resource efficient 3D CNNs [24] as video backbone. We report their performance at the bottom of Table 3. Notably, all 3D CNN architectures achieve better performance than 2D-ResNet-18. For isntance, although 3D-MobileNetV2 1.0x contains much smaller number of parameters (approx. 7x less) and less FLOPs compared to 2D-ResNet-18, it achieves around 4 mAP better performance.

We have also experimented with deeper and computationally more expensive 3D-ResNeXt-101 architecture to check how much performance can be increased. 3D-ResNeXt-101 shows 0.6 mAP improvement over 3D-ResNet-18 when 8-frames input is used.

How does clip length affect performance? Although we used 8-frames clips to train our audio-visual backbones, longer clips would provide larger temporal context. In Table 3, we compare clip lengths of 8-frames, 16-frames and 32-frames for the best performing 3D-ResNeXt-101 and 3D-ResNet-18 video backbones. To maintain similar complexity, we removed the initial temporal downsampling for 8-frames input, and inserted an additional temporal downsampling to the initial convolution layer for 32-frames input. Applying 16-frames clip length brings a performance improvement of 1.4 mAP and 2.2 mAP over 8-frames clip length for 3D-ResNet-18 and 3D-ResNeXt-101, respectively. Using 32-frames clip length does not show same performance improvement over using 16-frames. We suspect that inserting additional temporal downsampling hinders backbones ability to capture motion patterns.

Background features	mAP
only reference frame	93.4
neighbouring window of 9 frames	**93.5**

Table 6. Performance comparison when background speakers' features at different number of frames are leveraged.

4.2. Inter-Speaker Relation Modeling Evaluation

In this section, we investigate the performance of the proposed ISRM and compare it to an existing approach [1] for context modelling. These experiments include the full ASDNet pipeline (encoder, ISRM, and a temporal model), where the temporal model, if present, is a Bidirectional-GRU with sequence length of 64.

How many background speakers to use for ISRM? We experimented with different number of background speakers for ISRM, and the results are reported in Table 4. In general, increasing the number of background speakers features increases the performance. ISRM increases the performance by 0.8 mAP compared to the case where only reference speaker's features are used with temporal modeling (0 background speaker case). In the rest of our experiments, we use three background speakers in the ISRM module.

How does our ISRM compare to existing approaches? In Table 5, we provide a comparison of our ISRM approach to the *NonLocal* [51] approach proposed in [1]. *NonLocal* captures relationships between all the speakers within clip, whereas our ISRM approach captures relationships between speakers only within reference frame. When used alone, after the audio-visual backbones, neither *NonLocal* nor our ISRM approach bring significant performance improvement (*NonLocal* even degrades the performance). However, ISRM contributes additional 0.8 mAP compared to a system that uses only temporal modeling.

Can ISRM benefit from neighbouring frames? At ISRM, we do not have to use background speakers' features at only reference frame. Neighbouring frames relative to the reference frame can also provide useful information for ISRM. Therefore we have used background speakers' features at neighbouring window of 9 frames, which shows a modest 0.1 mAP improvement as reported in Table 6. For the rest of the paper, we use 9 neighbouring frames at ISRM.

Method	Sequence Length	mAP
Bidirectional-GRU	64	**93.5**
Bidirectional-LSTM	64	93.4
Bidirectional-SRU	64	93.2
GRU	32	92.8
LSTM	32	92.7
SRU	32	92.7

Table 7. Performance comparison of temporal modeling methods.

Seq. Length	8	16	32	64	128
mAP	92.0	92.8	93.3	**93.5**	**93.5**

Table 8. Performance comparison of using different sequence lengths at the training of Bidirectional-GRU.

4.3. Temporal Modeling Evaluation

Which RNN architectures are most suitable? Table 7 shows the performance comparison of different RNN blocks used for temporal modeling. All one-directional methods takes 32-frames features as input and last output is used as input to final fc layer (reference frame is placed to the last of input sequence). For bidirectional methods, we have used 64-frames features as input and center output is used as input to final fc layer (reference frame is placed at the center of input sequence). Compared to their bidirectional versions, one-directional methods perform around 0.7 mAP worse. Out of all methods, bidirectional-GRU achieves the best performance.

What should be the length of the input sequence? We have experimented with different sequence lengths and reported results in Table 8. In general, using larger sequence length does not hurt the final performance. However, after sequence length 64, the performance converges to 93.5 mAP.

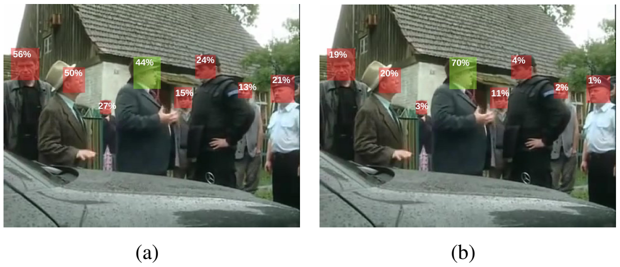

Figure 6. The network predictions for speaking probabilities of each speaker (a) after only audio-visual encoding (b) after temporal modeling and ISRM are also applied. Ground truths of *speaking* and *not-speaking* classes are denoted with green and red rectangles, respectively.

#	Speaker Video Feat.	Audio Feat.	ISRM Feat.	Temporal Modeling	mAP
1	✓				78.8
2		✓			49.3
3	✓	✓			88.9
4	✓	✓		✓	92.6
5	✓	✓	✓		89.6
6		✓	✓		64.5
7		✓	✓	✓	67.8
8	✓	✓	✓	✓	**93.5**

Table 9. Contribution of each component to the final performance.

Encoder Clip Length	ISRM and Temporal Modeling	mAP
8-frames	✗	86.7
16-frames	✗	88.9
8-frames	✓	93.4
16-frames	✓	93.5

Table 10. Effect of encoder clip length on the final performance. SincDSNet and 3D-ResNeXt-101 are used for audio and video backbones, respectively.

4.4. Component-wise Analysis

How does each component contribute to the performance? We investigated the contribution of each component to the final performance in Table 9. We highlight several findings: **(i)** Without ISRM and temporal modeling, suitable backbones alone achieve 88.9 mAP, which is better than any other state-of-the-art approach; **(ii)** ISRM and temporal modeling improve the performance by 0.7 mAP and 3.7 mAP when they are applied alone, respectively, showing the importance of both stages in the pipeline; **(iii)** In rows 6 and 7 in Table 9, we investigated the importance of ISRM stage by evaluating the performance without using reference speakers video features. Accordingly, even without looking reference speaker's face, information acquired from background speakers and audio enables our architecture to achieve around 68 mAP. This shows that ISRM is an indispensable part of our pipeline; **(iv)** When ISRM and temporal modeling are applied together, our architecture achieves the best performance with 93.5 mAP.

The contribution of temporal modeling and ISRM stages is visually illustrated in Fig. 6. With only audio-visual encoding, each speaker is analyzed independently and predictions for speaking probabilities are made without contextual and long-term temporal information in Fig. 6 (a). After applying temporal modeling and ISRM stages, the ASDNet predictions of speaking probabilities for *not-speaking* speakers drop and *speaking* speaker increases considerable as shown in Fig. 6 (b).

	Method	mAP
validation set	ASDNet (ours)	**93.5**
	Causal ASDNet (ours)	90.6
	MAAS-TAN [32]	88.8
	Chung et al. [6]	87.8
	ASC [1]	87.1
	Zhang et al. [54]	84.0
	Sharma et al. [44]	82.0
	Roth et al. [42]	79.2
test set	ASDNet (ours)	**91.7**
	Chung et al. [6]	87.8
	ASC [1]	86.7
	Zhang et al. [54]	83.5
	Roth et al. [42]	82.1

Table 11. Comparison with state-of-the-art methods on the AVA-ActiveSpeaker dataset. mAP results are calculated with the official evaluation tool as explained in [42].

Method	Number of Faces		
	1	2	3
ASDNet (Ours)	**95.7**	**92.4**	**83.7**
MAAS [32]	93.3	85.8	68.2
ASC [1]	91.8	83.8	67.6
Baseline [42]	87.9	71.6	54.4

Table 12. Performance comparison by number of visible faces on each frame.

Method	Face Size		
	S	M	L
ASDNet (Ours)	**74.3**	**89.8**	**96.3**
MAAS [32]	55.2	79.4	93.0
ASC [1]	56.2	79.0	92.2
Baseline [42]	44.9	68.3	86.4

Table 13. Performance comparison by face size.

How does the clip length affect performance? Increased encoder clip length (16-frames instead of 8-frames using 3D-ResNeXt-101 video backbone) improves the performance by 2.2 mAP if ISRM and temporal modeling are not applied. However, in the complete pipeline this improvement reflects to a marginal 0.1 mAP improvement in the final performance, which is shown in Table 10. This shows that increased encoder clip length shifts the improvement that could have been provided by temporal modeling to the encoder. This might not be desirable if complexity is important at the design of the architecture since doubling encoder clip length means doubling the complexity.

Can ISRM be placed after temporal modeling? If necessary, the order of ISRM and temporal modeling can be changed, which results in only a 0.1 mAP performance degradation.

Can we make the full pipeline causal? The complete pipeline can be made causal by placing the reference frame to the last place of the input for encoder and temporal modeling stages; and by not using neighbouring frames background speakers' features at ISRM. So that, no future information is used for the active speaker detection of the current frame. Causal pipeline achieves 90.6 mAP, which is still better than any state-of-the-art approach.

4.5. Comparison with the State-of-the-art

How does ASDNet compare to state-of-the-art methods? We compare the performance of ASDNet with several state-of-the-art methods in Table 11. For the final ASDNet, we used 16-frames clips at the audio-visual encoding stage, 3 background speakers with 9 neighbouring window at the ISRM stage, and bidirectional-GRU with 64-frames sequence length at the temporal modeling stage. ASDNet outperforms the second best approach by 4.7 mAP on the validation set, and by 3.9 mAP on the test set of AVA-ActiveSpeaker dataset.

How does number of faces affect the performance? Increased number of faces makes the active speaker detection task more challenging and the performance of ISRM becomes more critical. ASDNet outperforms all other state-of-the-art methods for all different face numbers as shown in Table 12. Superiority of ASDNet becomes more significant as number of faces increases.

How does face size affect the performance? Performance comparison for face size, which is set as small for [0, 64), medium for [64, 128), and large for [128, ∞) pixels, is shown in Table 13. ASDNet outperforms all other state-of-the-art methods for all different face sizes. Superiority of ASDNet becomes more significant for smaller faces.

5. Conclusion

In this paper, we scrutinized the task of Audio-Visual Active Speaker Detection and proposed a three-stage architecture, called ASDNet. With the proposed audio-visual encoder and the inter-speaker relation modelling mechanism, ASDNet outperforms the previous state-of-the-art with significant 4.7 mAP and 3.9 mAP on the validation and test set of AVA-ActiveSpeaker dataset, respectively. To make the final design and hyperparameter choices for ASDNet, we followed insights from carefully designed experiments each targeted a specific aspect of the system. Each of these experiments was discussed in the paper. We believe that these insights can be useful for other complex audio-visual tasks as well that require context and temporal modeling.

References

[1] Juan León Alcázar, Fabian Caba, Long Mai, Federico Perazzi, Joon-Young Lee, Pablo Arbeláez, and Bernard Ghanem. Active speakers in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12465–12474, 2020. 1, 2, 3, 4, 5, 6, 8

[2] I. Ariav and I. Cohen. An end-to-end multimodal voice activity detection using wavenet encoder and residual networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):265–274, 2019. 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 5

[4] Punarjay Chakravarty, Sayeh Mirzaei, Tinne Tuytelaars, and Hugo Van hamme. Who's speaking? audio-supervised classification of active speakers in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 87–90, 2015. 2

[5] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 4

[6] Joon Son Chung. Naver at activitynet challenge 2019–task b active speaker detection (ava). *arXiv preprint arXiv:1906.10555*, 2019. 2, 3, 4, 5, 8

[7] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *Proc. Interspeech 2018*, pages 1086–1090, 2018. 1

[8] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004. 2

[9] Ross Cutler and Larry Davis. Look who's talking: Speaker detection using video and audio correlation. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, volume 3, pages 1589–1592. IEEE, 2000. 1, 2

[10] Trevor Darrell, John W Fisher, and Paul Viola. Audio-visual segmentation and the cocktail party effect. In *International Conference on Multimodal Interfaces*, pages 32–40. Springer, 2000. 1, 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 5

[12] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968. IEEE, 2014. 2

[13] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *ACM Trans. Graph.*, July 2018. 2

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 2

[15] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10457–10467, 2020. 1

[16] Israel D Gebru, Sileye Ba, Xiaofei Li, and Radu Horaud. Audio-visual speaker diarization based on spatiotemporal bayesian fusion. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1086–1099, 2017. 1

[17] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 2

[18] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016. 1

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[20] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012. 2

[21] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[23] Okan Köpüklü, Fabian Herzog, and Gerhard Rigoll. Comparative analysis of cnn-based spatiotemporal reasoning in videos. *arXiv preprint arXiv:1909.05165*, 2019. 2

[24] Okan Kopuklu, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3d convolutional neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2, 4, 6

[25] Okan Kopuklu, Neslihan Kose, and Gerhard Rigoll. Motion fused frames: Data level fusion strategy for hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2103–2111, 2018. 2

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2

[27] Ludwig Kürzinger, Nicolas Lindae, Palle Klewitz, and Gerhard Rigoll. Lightweight end-to-end speech recognition from raw audio data using sinc-convolutions. *Proc. Interspeech 2020*, pages 1659–1663, 2020. 2, 4

[28] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005. 2

[29] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[30] Jongpil Lee, Taejun Kim, Jiyoung Park, and Juhan Nam. Raw waveform-based audio classification using sample-level cnn architectures. *arXiv preprint arXiv:1712.00866*, 2017. 2

[31] Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. Simple recurrent units for highly parallelizable recurrence. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 4

[32] Juan León-Alcázar, Fabian Caba Heilbron, Ali Thabet, and Bernard Ghanem. Maas: Multi-modal assignation for active speaker detection. *arXiv preprint arXiv:2101.03682*, 2021. 1, 2, 3, 4, 5, 8

[33] Chang-Le Liu, Sze-Wei Fu, You-Jin Li, Jen-Wei Huang, Hsin-Min Wang, and Yu Tsao. Multichannel speech enhancement by raw waveform-mapping using fully convolutional networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1888–1900, 2020. 2

[34] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[35] Qiguang Miao, Yunan Li, Wanli Ouyang, Zhenxin Ma, Xin Xu, Weikang Shi, and Xiaochun Cao. Multimodal gesture recognition based on the resc3d network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3047–3055, 2017. 2

[36] Simon Mittermaier, Ludwig Kürzinger, Bernd Waschneck, and Gerhard Rigoll. Small-footprint keyword spotting on raw audio data with sinc-convolutions. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7454–7458. IEEE, 2020. 2

[37] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):502–508, 2019. 2

[38] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10317–10326, 2020. 1

[39] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010. 2

[40] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017. 2

[41] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028. IEEE, 2018. 2, 4

[42] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, et al. Ava active speaker: An audio-visual dataset for active speaker detection. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4492–4496. IEEE, 2020. 1, 2, 3, 4, 5, 8

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2

[44] Rahul Sharma, Krishna Somandepalli, and Shrikanth Narayanan. Crossmodal learning for audio-visual speech event localization. *arXiv preprint arXiv:2003.04358*, 2020. 8

[45] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1*, pages 568–576, 2014. 2, 3

[46] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 334–340, 2018. 1

[47] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2

[48] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2

[49] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 2

[50] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John R Hershey, Rif A Saurous, Ron J Weiss, Ye Jia, and Ignacio Lopez Moreno. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking. *Proc. Interspeech 2019*, pages 2728–2732, 2019. 2

[51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 6

[52] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 4

[53] Neil Zeghidour, Nicolas Usunier, Iasonas Kokkinos, Thomas Schaiz, Gabriel Synnaeve, and Emmanuel Dupoux. Learning filterbanks from raw speech for phone recognition. In *2018 IEEE international conference on acoustics, speech and signal Processing (ICASSP)*, pages 5509–5513. IEEE, 2018. 4

[54] Yuan-Hang Zhang, Jingyun Xiao, Shuang Yang, and Shiguang Shan. Multi-task learning for audio-visual active speaker detection. 2, 3, 4, 5, 8