Eurographics / ACM IMX Joint Workshop WICED x Cinemotions (2023)
H.-Y. Wu, S. Jolly, A. Bruckert and L. Lévêque (Editors)

Assessing active speaker detection algorithms through the lens of automated editing

Rohit Girmaji[1], Sudheer Achary[1], Adhiraj Deshmukh[1], Vineet Gandhi[1]

[1]CVIT Lab,The International Institute of Information Technology - Hyderabad, India

Abstract

This paper addresses the challenge of active speaker detection in automated video editing and highlights the limitations of current audio-only and audio-visual speaker detection methods in handling unseen data with overlapped speakers, speaker occlusions, low video resolution, and random noises. Firstly, we select the BBC Old School Dataset, a comprehensive dataset introduced for automated video editing, and annotate it with active speaker labels. We propose an audio-based nearest neighbour algorithm that utilizes additional inputs, such as audio samples of each speaker and faces, to predict and track the active speaker. We evaluate the effectiveness of our approach on the BBC Old School Dataset by utilizing f rame-level speaker accuracy, which we consider a more suitable metric in the context of video editing. We observe that this simple setup outperforms the current state-of-the-art methods in predicting the active speaker. By incorporating these methods into our speaker-based editing approaches, we also notice that our method closely approximates the output obtained using ground truth annotations.

CCS Concepts

• ***Computing methodologies*** → ***Computational photography;*** *Supervised learning; Object identification;*

1. Introduction

Awareness of the person speaking at a given moment (active speaker) is crucial in video editing, especially in dialogue-driven scenes. Simpler editing styles like Dragnet (described by Murch in his book [MW01]) rely entirely on the active speaker information. In Dragnet style, both the video and audio of each character's entire line are included within each edit, and the editing process is akin to a tennis match, with rapid back-and-forth cuts that leave no space for reaction. The other more sophisticated cuts like L-cuts or J-cuts also hinge upon the knowledge of the active speaker, where the voice of the person seen in the outgoing shot continues, or the sound of the speaker who is about to be shown is heard before the cut happens. Just like human editors, accurate knowledge of the active speaker is essential for the realm of automated editing.

Naturally, majority of prior research efforts towards automated editing heavily rely upon active speaker information [LDTA17, MKSG20, GRLC15]. However, identifying and detecting the active speaker remains challenging for automated editing systems, unlike human editors who can do so effortlessly. Most existing methods either assume that the active speaker information is available [GRLC15] or utilize handcrafted features and methods to detect it [LDTA17]. To this end, evaluating the applicability of current state-of-the-art active speaker detection algorithms in the context of automated video editing is essential. This work uses the Old School Dataset(OSD) [old22] to assess different ASD algorithms. OSD is professionally collected by BBC research and is arguably the most comprehensive benchmark for automated editing.

Each camera view in the Old School Dataset provides zoomed-out wide-angle shots of the scene, covering activity from multiple actors. The automated editing task involves both virtual camera simulations [GVR*14] and camera selection (both in terms of view and the virtual shot [MKSG20]). The wide-angle views provide an apt and varied scenario for assessing ASD algorithms, deviating from typical medium-close-up shots (e.g., newsroom). We manually annotate the OSD with active speaker labels. We then evaluate different ASD algorithms in two ways. First, we evaluate them on frame-level accuracies for correctly detecting the active speaker(s). It is aimed to bring insights into the applicability of different ASD approaches in wide-angle shots, where the face sizes are small and have varying poses. Second, we use the noisy, active speaker information for the task of shot selection in OSD. We compare them against the edits obtained using the ground truth speaker labels and evaluate them with the expert ground truth edits provided with the OSD dataset.

We evaluate three classes of ASD algorithms: (a) Audio-Visual algorithms [TPD*21], (b) Diarization based algorithms, with manual cluster assignments, and (c) Nearest Neighbour-based speaker verification given a few reference utterances from each of the actors. In the audio-only algorithms (i.e., (b) and (c)), we additionally utilize a single-face photo of each of the actors to localize them in the scene. We plug the output of each of these algorithms into a

simplified computational editing algorithm that aims to maximize the visibility of the active speaker while avoiding fast/jump cuts. We want to emphasize that the goal of our work is not to present a comprehensive editing algorithm, and we realize that simplified editing does not reflect the grammar of complex exchanges that go on all the time in even the most ordinary conversations. Our work aims to instead reflect upon the role ASD algorithms play in editing and the extent of their usability in the current form.

Our evaluation suggests that the performance of the audio-visual ASD algorithms remains below par in the studied setup. We find that the minimal nearest neighbour-based algorithm combined with a face verification/tracking strategy provides the best performance. We also find that when combined with a dynamic programming optimization based editing framework, it comes close to the editing performance achieved while using the ground-truth speaker labels. The experiments suggest that having additional information (e.g. few utterances and one photo of each actor) helps alleviate the challenges for ASD in the studied setup. Overall, our work makes the following contributions:

1. We annotate BBC's Old School Dataset with active speaker labels. We also provide annotations for background noise, buzzer press, silence etc. allowing them to be used in automated editing research.
2. We set up and evaluate three different classes of ASD algorithms for active speaker detection in the OSD. We employ an off-the-shelf audio-visual algorithm and adapt purely audio-based algorithms for ASD by utilizing face verification combined with visual tracking.
3. We present comprehensive results and discussion on the efficacy of these algorithms when employed with a dynamic programming-based editing framework.

The remaining sections of the paper are structured as follows: Section 3 will provide a detailed explanation of the OSD Dataset and the process of annotation. Following that, in section 4, we will delve into the methods used for ASD and video editing algorithms that were considered for this study. The evaluation of ASD algorithms will be discussed in the experiments section, along with the video editing techniques used on the OSD Dataset using various inputs and approaches.

2. Related Works

Active speaker detection (ASD) algorithms aim to identify active speakers in a scene [RCK*20]. ASD requires input from both visual and audio modalities. If we have beforehand knowledge of the actor's faces and their corresponding voices in a scene, then ASD can be performed by pure audio-based diarization [CHN*20b]. In the wild setting, where such information is unknown, recent methods apply deep neural networks on face tracks to detect if the voice is synchronized with the lip and face movement [CZ17]. Our work investigates a controlled setup where the numbers of actors and their identities are known upfront; hence, both approaches are applicable. We briefly review the advancements in speaker diarization and ASD below. We then discuss automated editing algorithms that utilize active speaker information.

2.1. Speaker Diarization

Speaker diarization is the process of separating an audio recording that contains multiple speakers into distinct segments based on the speaker's identity. Most speaker diarization systems [SGR15, GRSS*17, WDW*18] consist of multiple relatively independent components (a) a speech segmentation module, which removes non-speech segments, (b) a module to extract speaker-discriminative embeddings [DKD*10, WWPM18], (c) a clustering module and (d) a refinement module which enforces additional constraints to further refine the diarization results [SGR15]. More recent attempts have aimed to consolidate these modules and train diarization in an end-to-end manner. The end-to-end Neural Diarization (EEND) family of approaches [FKH*19a, FKH*19b, BYC*20, BL21] model diarization as a multi-label classification problem using permutation-invariant training.

2.2. Audio Visual ASD

In the early days of Audio Visual Automatic Speaker Detection (ASD), basic visual features such as upper body [SBM21] and facial [PIT*16] movements were utilized to predict the active speaker. However, the effectiveness of this method was limited due to the weak correlation between body movements and speech activity. Later, the combination of audio and visual information proved to be much more beneficial in performing ASD [EML*18]. Audio Visual Fusion techniques approached the task by assigning speech to one of the speakers in a video [ACM*20]. Some methods view ASD as a classification model that evaluates each speaker in the video and outputs an active speaker label for each of them [AOCZ20]. Lately, various deep learning architectures have been proposed like [TPD*21] with attention mechanism, and Graph Neural Networks [MRT*] have been developed and have provided significant performance improvements in ASD.

2.3. Video Editing

Several previous automatic editing methods employ speaker information for editing. Classical computational editing systems [IOM95, LRGC01, RGC01, RBB08], for example, use speaker detection algorithms to determine who is talking and select a camera known to have a shot of that person. More recently, [GRLC15] proposed an optimization-based approach for automatically creating well-edited movies from a 3D animation. They employ speaking as one of the main action categories. Work by [LDTA17] proposes an editing framework based on a user-specified set of film-editing idioms. They employ idioms like speaker visibility, which ensures that the speaker of each line of dialogue is visible. The work by Moorthy et al. [MKSG20] suggests that speech-based editing scores highly with respect to conveying actor emotions. We take a similar approach to [MKSG20] and replace gaze potential with speaker potential. Our work is also related to previous works that pose video editing as a discrete optimization problem, solved using dynamic programming [EDRM07], [GRLC15], [LCCR11], [MCB16].

Figure 1: Screenshots of a scene from 4 different camera views in the Old School BBC Dataset. Screenshots from Camera 1 to Camera 4 can be seen from left to right.

3. BBC Old School Dataset

We take the publicly available BBC Old School Dataset for our study. The dataset consists of raw footage of a multi-camera shoot of a game show called *Old School*. Screenshots from different camera views of the dataset videos can be visualized in [Figure-1]. The dataset also provides processed videos of the raw footage, which are multiple takes of different scenes of the show. They also provide a human-edited programme as a benchmark for automated editing systems. We use the Edit Decision List (EDL) corresponding to the human-edited programme to extract videos of a total duration of 30 minutes from the processed shoot videos. We provide ground truth speaker annotations for these videos taken from one camera view, and these speaker annotations are the same for all the camera views.

3.1. Annotation Process

We use the open-source project VIA tool for annotations. Its GUI and annotation process can be visualized in [Figure-2]. We follow the below steps for annotation:

3.1.1. Generating Face and Voice Tracks

For face crop tracks generation, we follow the steps described in the face track annotator of VIA tool. We first automatically generate face tracks using VGGFaceTracker [KMSZ12, RHGS15]. Then we manually filter, select and update the annotations. For all the videos, we manually generate voice tracks by watching the video and listening to the audio stream concurrently. Human annotators refined the start and end of speech segments to get accurate labels for the segments. We merge neighbouring segments of the same speaker if the gap between the segments is less than 1 second. We also consider the speech segments even if it is less than 1 second and generate voice tracks and labels for such segments. In addition to the speaker's voice tracks annotation we also provide labels for off-screen speakers, and non-speech sounds like bell ring, buzzer. As a final step, the annotations are manually verified by 3 annotators to get quality labels.

3.1.2. Labelling the face and voice tracks

As all the videos in the dataset have the same set of speakers, video level identity labels of speakers are the same across all the videos.

3.2. Dataset Statistics

The dataset consists of 18 video clips with a total duration of about 30 minutes. There are 5 unique identities with voice tracks and faces. There is only one off-screen speaker in the data. *Bell ring* and *Buzzer* sounds are the non-speech sounds.The total duration of overlapped speech segments and speech activity in the video amounts to 1.3 minutes and 25 minutes respectively.

4. Methodology

We describe different types of active speaker detection methods used and the ways we rely on speaker information in the task of video editing with different approaches

4.1. Speaker detection methods

We have used Audio Visual Active Speaker Detection (ASD), Audio-based Speaker Diarization and Audio-based KNN Classifier methods for speaker detection

4.1.1. Audio Visual Active Speaker Detection(ASD)

Active Speaker Detection (ASD) models predict the active speaker at the frame level. An Active Speaker in a video is a speaker whose face is visible and audible simultaneously. We consider TalkNet [TPD*21] an ASD model for our experiments. TalkNet is a classification model that takes cropped faces in the video and corresponding audio as input and outputs active speaker labels for each face. The model consists of a feature representation frontend and a speaker detection backend. The frontend generates visual and audio spatio-temporal features and the classifier backend consists of an inter-modal cross-attention and self-attention mechanism followed by a classifier head to generate the active speaker label.

4.1.2. Audio-based Speaker Diarization

In our work, we employ the recent state-of-the-art EEND method proposed by Bredin et al. [BYC*20, BL21].The model trained on the *AMI*, *VoxConverse*, and *DIHARD* datasets [MCK*05, CHN*20a, RSK*21] is used in our experiments.

4.1.3. Audio-based KNN Classifier

We use K-Nearest Neighbour (KNN) Classifier for our approach. We take three 10 second audio samples of each speaker to form the search space and perform nearest neighbour search of a test sample to get the class label prediction. We take the majority class label of the top-K nearest neighbours as the test sample label.

We generate X-vector speaker embeddings [SGRM*18] with a pre-trained TDNN model using SpeechBrain [RPP*21]. These embeddings will now become our search space for the KNN algorithm.

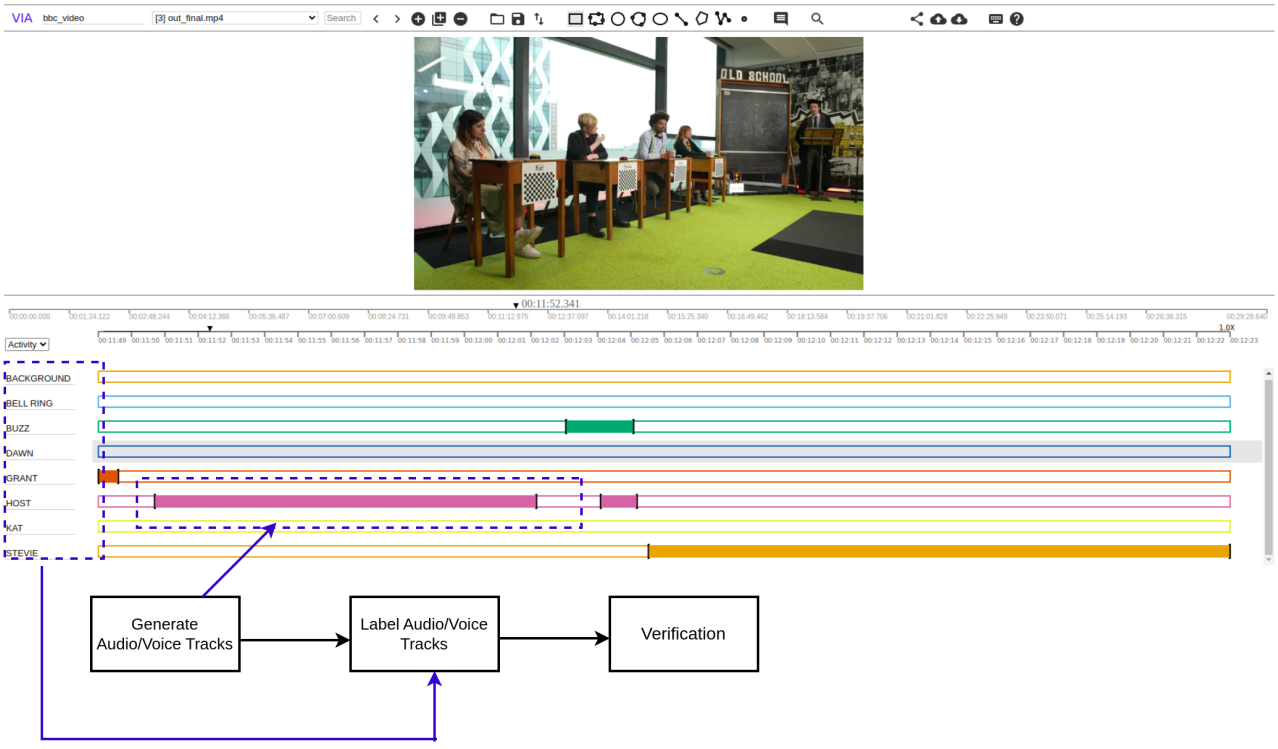

Figure 2: Annotation tool and process for speaker annotations. Background is the off-screen speaker, Bell ring is the non-speech sound.

For a given video, we perform a sliding window approach and predict labels for each segment through K-nearest neighbour search. For segments with more than one predicted label, we take the label with the highest similarity measure. We use cosine similarity as the similarity measure.

4.2. Video Editing

Here we describe the ways we leveraged active speaker information (*Speaker Potential*)in the task of video editing with two different approaches, i.e. Speaker Guided Greedy Editing(SGE) and Speaker Guided Optimization Editing(SOE). We also describe the pre-processing steps like Shot Generation which is required for both the approaches

4.2.1. Shot Generation

In the BBC old school data set, we are given multiple camera views for a video and a human-edited video. We employ the algorithm [GVR*14] for shot generation. For example, in a 3 actor video, we generate three 1-shots (shots with a single actor), two 2-shots (shots with 2 actors) and one master shot (shot that includes all actors). Similarly, for n actor video we generate a total of $\frac{n(n+1)}{2}$ shots.

For shot generation, we need tracks of the actors in a frame (bounding boxes). We use ByteTrack [ZSJ*22], which is a Multi-Object Tracking method that estimates bounding boxes and identities of objects in videos. They adopt an object detector YOLOX [GLW*21] to detect the bounding boxes with a confidence score. In the first step of the algorithm, the bounding boxes are divided into high-confidence and low-confidence score bounding boxes. In the second step, it tries to associate the high-confidence score bounding boxes with the tracklets and then associates the low-confidence score bounding boxes with the unmatched tracklets.

We use ByteTrack for its efficiency and its ability in handling special scenarios like person occlusions, crossing past each other etc, which are comprised in OSD.

4.3. Speaker Potential

Speaker potential [2] quantitatively measures the importance of each shot at every time instant. Previous works [LDTA17] and [GRLC15] estimate the actions/emotions in a given shot by either relying on additional meta-data or bottom-up computational features. [JSSH15] and [RKGS18] have shown that the gaze data recorded from users enables effective localization of focal scene events. We extend this idea to calculate speaker potential similar to that of gaze potential in [MKSG20] of each shot s using active speaker information at each time frame t.

Assuming we have active speaker information (the actor who spoke at that instant of time) per each frame, we determine the speaker potential for each shot. We adopt a bottom-up approach, we first calculate the speaker potential of (lower-order shots) single shots, which capture individual actors using [1].

$$S(s_t^x) = \begin{cases} \lambda & x \text{ is speaker} \\ 0 & otherwise \end{cases} \quad (1)$$

where s_t^x refers to a shot s at time t that contains a single actor x

Speaker potential for shots with multiple actors (higher-order shots) is then computed from the speaker potentials of constituent lower-order shots using [2].

$$S(s_t^{ab}) = S(s_t^a) + S(s_t^b) - |S(s_t^a) - S(s_t^b)| \quad (2)$$

where s_t^{ab} refers to a shot s at time t that contains set of actors $\{a,b\}$

For instance, speaker potentials of two 1-shots $S(s^a)$ and $S(s^b)$ can be used to compute the speaker potential of a 2-shot $S(s^{ab})$. Similarly, speaker potentials of two 2-shots $S(s^{ab})$ and $S(s^{bc})$ can be used to compute speaker potential of a 3-shot $S(s^{abc})$

It can be seen in equation [2] if there is only one speaker 1-shot with the active speaker gets high speaker potential. If a and b are the active speakers, the 2-shot($S(s^{ab})$) that contains both the active speakers gets the maximum speaker potential.

4.3.1. Speaker Guided Greedy Editing

For speaker-guided greedy editing(SGE) we select the shot that best captures the speaker from the generated shots. For speaker information, we use manual annotations and active speaker detection networks. When more than one person speaks simultaneously, their combined shot is selected. The algorithm continues with its current selection until a change of speaker occurs. A minimum shot duration (l) is enforced to avoid rapid shot transitions. If a silence lasting L seconds is detected, we display a wide shot during that particular segment of silence.

4.3.2. Speaker Guided Optimization Editing

GAZED [MKSG20] is an end-to-end system to automate the video editing process for staged performances. It outputs an edited video that adheres to common cinematic principles and is aesthetically pleasing to watch. In the regular Gazed approach we use gaze potentials which are then combined with other terms that model cinematic principles like avoiding jump cuts, rhythm (pace of shot transitioning), avoiding transient shots etc. But with speaker-guided optimization editing(SOE), we rely on active speaker information to build speaker potential [2] instead of the human gaze. Speaker potentials assign a higher cost to shots with an active speaker, which will enforce the *dynamic programming* optimization framework to select the shot with greater speaker potential by constraining other cinematic principles. The underlying algorithm poses shot selection as a discrete optimization problem, which examines the importance of each of the multiple shots generated for every video frame, while adhering to cinematic principles like avoiding cuts between overlapping shots (termed jump cuts), avoiding rapid shot transitions, maintaining a cutting rhythm, etc. Cinematic principles are modelled as penalty in the term $E_e(s_{t-1}, s_t)$ where s_t represents shot s at time frame t. This penalty term is the sum of three different costs, namely shot transition cost, shot overlap cost, and cutting rhythm cost as described in [MKSG20]. The final solution is obtained via a search for the optimal path through an editing graph. In SOE setting the shot selection will ensure to follow cinematic principles and provide an aesthetically pleasing experience to watch which isn't constrained in SGE approach. As there could be incorrect active speaker predictions from active speaker detection networks, cinematic-motivated penalty terms do also work as an error recovery mechanism (to not make bad mistakes) in some scenarios. Unlike the greedy approach, the optimization-based method cannot run in a real-time setting as it needs to construct a complete cost matrix to get a minimal cost path. This approach has a higher memory footprint compared to the greedy approach as it needs to build an editing graph for optimization framework and backtrack for the optimal cost path. For a video with f frames and n actors, it will have a space complexity of $O(2^n - 1 * f)$. We also propose that our SOE framework with cinematic principle penalizing terms is open to extension for other useful information such as action etc, but not limited to human gaze or active speaker.

Formally, given a sequence of frames $t = [1..T]$, the set of generated shots (rushes) $S_t = \{s_i^t\}_{i=1}^{2^n-1}$ and the active speaker information a_t corresponding to active speaker a at time frame t, our algorithm selects a sequence of shots $\varepsilon = \{r_t\},\ r_t \in S_t$ for each frame t minimising the objective function [3]

$$E(\varepsilon) = \Sigma_{t=1}^{T} - ln(S(r_t)) + \Sigma_{t=2}^{T} E_e(r_{t-1}, r_t) \quad (3)$$

where $S(r_t)$ that represents speaker potential for each shot and $E_e(r_{t-1}, r_t)$ represents cost for transitioning from one shot to another.

We solve equation [3] using dynamic programming. The algorithm outputs a sequence of shots r_t (where r is the selected shot at time frame t) from the set of shots generated over time $\{S_t\}$. We build a cost matrix $C(r_t, t)$ where $r_t \in s_t^i$ and $t = [1..T]$, each cell is computed with recurrence relation [4]

$$C(r_t, t) = \begin{cases} -ln(S(r_t)) & t = 1 \\ min_k[C(r_k, t-1) - ln(S(r_t)) + E_e(r_k, r_t)] & otherwise \end{cases} \quad (4)$$

For each cell in the matrix, we compute and store the minimum cost to reach it. Once the matrix is built, we then perform backtracking to retrieve the sequence of optimal shots.

5. Experiments

We perform all our experiments on the timeline edit videos from OSD. Our experiments are of two parts. We first predict the speakers at frame level using TalkNet, pyannote and KNN algorithm on the dataset videos and evaluate using frame level speaker accuracy. In the second part, we perform video editing using the speaker predictions from all three models from the first part and ground truth speaker data. We evaluate the three ASD approaches in the context of video editing using frame level editing accuracy.

	Duration(sec)	Frame Level Accuracy					
		TalkNet(ASD)				Pyannote	KNN
		cam1	cam2	cam3	cam4	All cameras	
Full video	1768.64	54.51%	52.10%	52.76%	50%	73.71%	83.5%
Clip #1	192.76	58.29%	57.38%	51.97%	53.13%	84.98%	89.41%
Clip # 2	80.52	56.83%	56.20%	54.44%	54.32%	89.87%	91.15%
Clip #3	153.92	55.38%	53.12%	54.41%	53.12%	33.10%	88.73%
Clip #4	106.84	53.28%	49.47%	53.80%	46.63%	90.12%	86.00%
Clip #5	140.72	49.41%	47.46%	52.62%	51.51%	13.95%	86.17%
Clip #6	23.64	49.67%	47.66%	52.39%	51.02%	52.70%	86.28%
Clip #7	85.2	47.76%	46.09%	50.63%	52.91%	17.31%	85.54%
Clip #8	6.8	47.70%	46.06%	50.47%	52.65%	40.35%	85.63%
Clip #9	103.04	47.48%	46.07%	49.44%	51.75%	49.82	83.97%

Table 1: Frame Level Accuracy for the Full Video and Specific Segments.

		Editing Accuracy with TalkNet predictions		Editing Accuracy With GT Speaker Annotations		Editing Accuracy with KNN Speaker Predictions	
	Duration(sec)	Speaker Greedy	Speaker Optimization	Speaker Greedy	Speaker Optimization	Speaker Greedy	Speaker Optimization
		Cam 4	Cam 4	All Cameras		All Cameras	
Clip #1	192.76	46.99%	50.33%	71.63%	79.85%	66.33%	77.94%
Clip # 2	80.52	50.05%	58.06%	66.54%	76.49%	61.22%	76.39%
Clip #3	153.92	61.23%	66.76%	82.77%	88.00%	71.88%	78.79%
Clip #4	106.84	47.06%	61.40%	80.50%	85.37%	65.64%	71.93%
Clip #5	140.72	45.92%	50.45%	67.89%	74.68%	55.38%	72.40%
Clip #6	23.64	60.67%	66.88%	88.40%	92.74%	75.96%	80.57%
Clip #7	85.2	47.77%	52.75%	71.45%	74.21%	58.76%	66.71%
Clip #8	6.8	51.71%	58.91%	66.00%	68.42%	64.88%	68.42%
Clip #9	103.04	42.17%	52.56%	64.33%	65.75%	58.95%	60.48%

Table 2: Editing Frame level accuracies with ground truth and predicted speaker annotations using various methods.

We experiment with an Audio Visual ASD model TalkNet. TalkNet is pretrained on TalkSet data pretrained on AVA-Active Speaker dataset. We predict active speakers at frame level using this model. These models essentially predict the bounding box of the active speaker faces. To get the person-id for the predicted bounding box, we use faces of the actors and tracking (ByteTrack) information to associate the predicted bounding box with person-id.

For experiments related to Audio-based speaker diarization(pyannote) we rely on the open-source python library for speaker diarization called pyannote-audio. We used the official pretrained (as described in section 4.1.2) Speaker Diarization pipeline from *pyannote* available on *Hugging-Face*.

We take three 10 second audio samples of each of the 5 speakers in the OSD for KNN algorithm. For a given video, we take sliding window approach with window length of 0.8 second and a stride of 0.4 second. For each segment, we predict the speaker by KNN search. We take top-3 nearest neighbours based on the cosine similarity score and use majority voting strategy to get the speaker for the segment. We experimented with different k values and found optimal performance for $k = 3$.

We evaluate our audio-based speaker detection approach in the task of video editing. For this, we take two existing editing algorithms namely speaker-based greedy editing(SGE) and speaker-guided optimization editing(SOE).

In SGE, we use minimum shot duration(l) of 1.5 second and silence duration(L) of 2 second.

In SOE, we use speaker potential function $S(s_t^x)$ [2] in optimization framework [3]. We constrain our optimization framework with a minimum shot duration (l) of 1.5 second. In the scenarios where there is silence or no clear speaker, similar to SGE approach speaker potential ranks master shot a higher cost and a lower cost for the other shots.

5.1. Evaluation metrics

Frame-level speaker accuracy for a video is calculated by dividing the number of frames with correct speaker prediction by the total number of frames with voice activity.

Frame-level Editing accuracy is calculated by dividing the number of frames that have a correct match with ground truth in terms of the subjects shown in the video edit by the total number of frames in the video.

6. Results and Discussions:

We conducted experiments using three different methods to detect speakers and report metrics for each of these methods in Table 1. We assessed the accuracy of TalkNet, Pyannote, and KNN approaches for the full video and 9 segments of the video. We chose these 9 segments based on varying visual and acoustic conditions. Specifically, we selected segments that contained non-speech sounds such as buzzers, bell rings, and clapping, as well as instances of overlapping speech, and instances where all actors' speech was covered.

Based on the data presented in Table 1, it can be concluded that the KNN method outperforms both Pyannote and TalkNet. One reason for this could be that KNN uses audio samples of the speakers as extra input for predictions, while Pyannote doesn't use any extra data. TalkNet had the lowest performance, likely due to the complex visual conditions present in the videos such as face occlusions and low resolution. The performance of TalkNet varied depending on the camera views, indicating that its performance is influenced by visual features that change with different camera angles.

We conducted video editing experiments on the OSD dataset using speaker information obtained from ground truth annotations as well as speaker predictions obtained from the KNN and TalkNet methods. We use videos from one camera view for these experiments. Table 2 presents the frame-level editing accuracy results for the greedy and optimization approaches across 9 segments. The data in Table 2 suggests that the input speaker information plays a significant role in video editing and that the speaker is the most important factor in the video editing decision-making process. According to the metrics, approximately 70-80% of the frames in an edited video contain the speaker. Our results indicate that the speaker optimization approach outperforms the greedy approach. Furthermore, there is a visible correlation between speaker accuracy and editing accuracy.

7. Conclusions

This study evaluates various ASD methods and presents a simple audio-based approach that can outperform existing methods in video settings, such as those found in the OSD dataset. The KNN approach outperforms both diarization and audio-visual ASD methods. This study also emphasizes the importance of speakers in the video editing process and assesses the effectiveness of different ASD models in this task. Our experiments demonstrate that speaker information significantly influences video editing output. Our proposed approach requires audio samples and actors' face images as additional inputs for predicting and tracking the active speaker. We believe that audio samples and face images of actors are often available before the editing process. A potential future direction would be to investigate other factors that affect the video editing task with the goal of achieving fully automated video editing and developing robust ASD models that can perform well in various video settings, thereby improving the task of video editing.

References

[ACM*20] ALCÁZAR J. L., CABA F., MAI L., PERAZZI F., LEE J.-Y., ARBELÁEZ P., GHANEM B.: Active speakers in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 12465–12474. 2

[AOCZ20] AFOURAS T., OWENS A., CHUNG J. S., ZISSERMAN A.: Self-supervised learning of audio-visual objects from video. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16* (2020), Springer, pp. 208–224. 2

[BL21] BREDIN H., LAURENT A.: End-to-end speaker segmentation for overlap-aware resegmentation. In *Proc. Interspeech 2021* (2021). 2, 3

[BYC*20] BREDIN H., YIN R., CORIA J. M., GELLY G., KORSHUNOV P., LAVECHIN M., FUSTES D., TITEUX H., BOUAZIZ W., GILL M.-P.: pyannote.audio: neural building blocks for speaker diarization. In *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing* (2020). 2, 3

[CHN*20a] CHUNG J. S., HUH J., NAGRANI A., AFOURAS T., ZISSERMA A.: Spot the conversation : Speaker diarisation in the wild. arXiv : Arxiv2007.01216. 3

[CHN*20b] CHUNG J. S., HUH J., NAGRANI A., AFOURAS T., ZISSERMAN A.: Spot the conversation: speaker diarisation in the wild. *arXiv preprint arXiv:2007.01216* (2020). 2

[CZ17] CHUNG J. S., ZISSERMAN A.: Out of time: automated lip sync in the wild. In *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13* (2017), Springer, pp. 251–263. 2

[DKD*10] DEHAK N., KENNY P. J., DEHAK R., DUMOUCHEL P., OUELLET P.: Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing 19*, 4 (2010), 788–798. 2

[EDRM07] ELSON, DAVID, RIEDL, MARK: A lightweight intelligent virtual cinematography system for machinima production. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (2007), vol. 3, pp. 8–13. 2

[EML*18] EPHRAT A., MOSSERI I., LANG O., DEKEL T., WILSON K., HASSIDIM A., FREEMAN W. T., RUBINSTEIN M.: Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619* (2018). 2

[FKH*19a] FUJITA Y., KANDA N., HORIGUCHI S., NAGAMATSU K., WATANABE S.: End-to-end neural speaker diarization with permutation-free objectives. *Interspeech* (2019). 2

[FKH*19b] FUJITA Y., KANDA N., HORIGUCHI S., XUE Y., NAGAMATSU K., WATANABE S.: End-to-end neural speaker diarization with self-attention. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2019), IEEE, pp. 296–303. 2

[GLW*21] GE Z., LIU S., WANG F., LI Z., SUN J.: YOLOX: exceeding YOLO series in 2021. *CoRR abs/2107.08430* (2021). URL: https://arxiv.org/abs/2107.08430, arXiv:2107.08430. 4

[GRLC15] GALVANE Q., RONFARD R., LINO C., CHRISTIE M.: Continuity editing for 3d animation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA* (2015), Bonet B., Koenig S., (Eds.), AAAI Press, pp. 753–762. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9662. 1, 2, 4

[GRSS*17] GARCIA-ROMERO D., SNYDER D., SELL G., POVEY D., MCCREE A.: Speaker diarization using deep neural network embeddings. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 4930–4934. 2

[GVR*14] GANDHI, VINEET, RONFARD, REMI, GLEICHER M.: Multi-clip video editing from a single viewpoint. In *Proceedings of the 11th European Conference on Visual Media Production* (2014), pp. 1–10. 1, 4

[IOM95] INOUE T., OKADA K.-I., MATSUSHITA Y.: Learning from tv programs: Application of tv presentation to a videoconferencing system. In *Proceedings of the 8th annual ACM symposium on User interface and software technology* (1995), pp. 147–154. 2

[JSSH15] JAIN E., SHEIKH Y., SHAMIR A., HODGINS J. K.: Gaze-driven video re-editing. *ACM Trans. Graph. 34*, 2 (2015), 21:1–21:12. URL: https://doi.org/10.1145/2699644, doi:10.1145/2699644. 4

[KMSZ12] KLÄSER A., MARSZAŁEK M., SCHMID C., ZISSERMAN A.: Human focused action localization in video. In *Trends and Topics in Computer Vision: ECCV 2010 Workshops, Heraklion, Crete, Greece, September 10-11, 2010, Revised Selected Papers, Part I 11* (2012), Springer, pp. 219–233. 3

[LCCR11] LINO C., CHOLLET M., CHRISTIE M., RONFARD R.: Computational model of film editing for interactive storytelling. In *Interactive Storytelling - Fourth International Conference on Interactive*

Digital Storytelling, ICIDS 2011, Vancouver, Canada, November 28 - 1 December, 2011. Proceedings (2011), Si M., Thue D., André E., Lester J. C., Tanenbaum T. J., Zammitto V., (Eds.), vol. 7069 of *Lecture Notes in Computer Science*, Springer, pp. 305–308. URL: https://doi.org/10.1007/978-3-642-25289-1_35, doi:10.1007/978-3-642-25289-1_35. 2

[LDTA17] LEAKE M., DAVIS A., TRUONG A., AGRAWALA M.: Computational video editing for dialogue-driven scenes. *ACM Trans. Graph. 36*, 4 (2017), 130:1–130:14. URL: https://doi.org/10.1145/3072959.3073653, doi:10.1145/3072959.3073653. 1, 2, 4

[LRGC01] LIU Q., RUI Y., GUPTA A., CADIZ J. J.: Automating camera management for lecture room environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), pp. 442–449. 2

[MCB16] MERABTI B., CHRISTIE M., BOUATOUCH K.: A virtual director using hidden markov models. *Comput. Graph. Forum 35*, 8 (2016), 51–67. URL: https://doi.org/10.1111/cgf.12775, doi:10.1111/cgf.12775. 2

[MCK*05] MCCOWAN I., CARLETTA J., KRAAIJ W., ASHBY S., BOURBAN S., FLYNN M., GUILLEMOT M., HAIN T., KADLEC J., KARAISKOS V., KRONENTHAL M., LATHOUD G., LINCOLN M., MASSON A. L., POST W., REIDSMA D., WELLNER P.: The ami meeting corpus. Methods and Techniques in Behavioral Research. 3

[MKSG20] MOORTHY K. L. B., KUMAR M., SUBRAMANIAN R., GANDHI V.: GAZED- gaze-guided cinematic editing of wide-angle monocular video recordings. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020* (2020), Bernhaupt R., Mueller F. F., Verweij D., Andres J., McGrenere J., Cockburn A., Avellino I., Goguey A., Bjøn P., Zhao S., Samson B. P., Kocielnik R., (Eds.), ACM, pp. 1–11. URL: https://doi.org/10.1145/3313831.3376544, doi:10.1145/3313831.3376544. 1, 2, 4, 5

[MRT*] MIN K., ROY S., TRIPATHI S., GUHA T., MAJUMDAR S.: Intel labs at activitynet challenge 2022: Spell for long-term active speaker detection. 2

[MW01] MURCH, WALTER: *In the Blink of an Eye*, vol. 995. Silman-James Press Los Angeles, 2001. 1

[old22] Old school dataset. BBC Research and Development. 1

[PIT*16] PATRONA F., IOSIFIDIS A., TEFAS A., NIKOLAIDIS N., PITAS I.: Visual voice activity detection in the wild. *IEEE Trans. Multim. 18*, 6 (2016), 967–977. URL: https://doi.org/10.1109/TMM.2016.2535357, doi:10.1109/TMM.2016.2535357. 2

[RBB08] RANJAN A., BIRNHOLTZ J., BALAKRISHNAN R.: Improving meeting capture by applying television production principles with audio and motion detection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), pp. 227–236. 2

[RCK*20] ROTH J., CHAUDHURI S., KLEJCH O., MARVIN R., GALLAGHER A., KAVER L., RAMASWAMY S., STOPCZYNSKI A., SCHMID C., XI Z., 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)* (2020), IEEE, pp. 4492–4496. 2

[RGC01] RUI Y., GUPTA A., CADIZ J. J.: Viewing meeting captured by an omni-directional camera. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), pp. 450–457. 2

[RHGS15] REN S., HE K., GIRSHICK R., SUN J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems 28* (2015). 3

[RKGS18] RACHAVARAPU K. K., KUMAR M., GANDHI V., SUBRAMANIAN R.: Watch to edit: Video retargeting using gaze. *Comput. Graph. Forum 37*, 2 (2018), 205–215. URL: https://doi.org/10.1111/cgf.13354, doi:10.1111/cgf.13354. 4

[RPP*21] RAVANELLI M., PARCOLLET T., PLANTINGA P., ROUHE A., CORNELL S., LUGOSCH L., SUBAKAN C., DAWALATABAD N., HEBA A., ZHONG J., CHOU J.-C., YEH S.-L., FU S.-W., LIAO C.-F., RASTORGUEVA E., GRONDIN F., ARIS W., NA H., GAO Y., MORI R. D., BENGIO Y.: SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624. arXiv:2106.04624. 3

[RSK*21] RYANT N., SINGH P., KRISHNAMOHAN V., VARMA R., CHURCH K., CIERI C., DU J., GANAPATHY S., LIBERMAN M.: The third dihard diarization challenge. INTERSPEECH. 3

[SBM21] SHAHID M., BEYAN C., MURINO V.: S-VVAD: visual voice activity detection by motion segmentation. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021* (2021), IEEE, pp. 2331–2340. URL: https://doi.org/10.1109/WACV48630.2021.00238, doi:10.1109/WACV48630.2021.00238. 2

[SGR15] SELL G., GARCIA-ROMERO D.: Diarization resegmentation in the factor analysis subspace. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), IEEE, pp. 4794–4798. 2

[SGRM*18] SNYDER D., GARCIA-ROMERO D., MCCREE A., SELL G., POVEY D., KHUDANPUR S.: Spoken language recognition using x-vectors. In *Odyssey* (2018), vol. 2018, pp. 105–111. 3

[TPD*21] TAO R., PAN Z., DAS R. K., QIAN X., SHOU M. Z., LI H.: Is someone speaking?: Exploring long-term temporal features for audio-visual active speaker detection. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021* (2021), Shen H. T., Zhuang Y., Smith J. R., Yang Y., César P., Metze F., Prabhakaran B., (Eds.), ACM, pp. 3927–3935. URL: https://doi.org/10.1145/3474085.3475587, doi:10.1145/3474085.3475587. 1, 2, 3

[WDW*18] WANG Q., DOWNEY C., WAN L., MANSFIELD P. A., MORENO I. L.: Speaker diarization with lstm. In *2018 IEEE International conference on acoustics, speech and signal processing (ICASSP)* (2018), IEEE, pp. 5239–5243. 2

[WWPM18] WAN L., WANG Q., PAPIR A., MORENO I. L.: Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 4879–4883. 2

[ZSJ*22] ZHANG Y., SUN P., JIANG Y., YU D., WENG F., YUAN Z., LUO P., LIU W., WANG X.: Bytetrack: Multi-object tracking by associating every detection box. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII* (2022), Avidan S., Brostow G. J., Cissé M., Farinella G. M., Hassner T., (Eds.), vol. 13682 of *Lecture Notes in Computer Science*, Springer, pp. 1–21. URL: https://doi.org/10.1007/978-3-031-20047-2_1, doi:10.1007/978-3-031-20047-2_1. 4