Multimedia Tools and Applications (2023) 82:13667–13688
https://doi.org/10.1007/s11042-022-13746-7

Audio-video fusion strategies for active speaker detection in meetings

Lionel Pibre[1] · Francisco Madrigal[2] · Cyrille Equoy[2] · Frédéric Lerasle[2] · Thomas Pellegrini[1] · Julien Pinquier[1] · Isabelle Ferrané[1]

Received: 13 January 2021 / Revised: 15 March 2022 / Accepted: 29 August 2022 /
Published online: 28 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Meetings are a common activity in professional contexts, and it remains challenging to endow vocal assistants with advanced functionalities to facilitate meeting management. In this context, a task like *active speaker detection* can provide useful insights to model interaction between meeting participants. Detection of the active speaker can be performed using only video based on the movements of the participants of a meeting. Depending on the assistant design and each participant position regarding the device, active speaker detection can benefit from information coming from visual and audio modalities. Motivated by our application context related to advanced meeting assistant, we want to combine audio and visual information to achieve the best possible performance. In this paper, we propose two different types of fusion (naive fusion and attention-based fusion) for the detection of the active speaker, combining two visual modalities and an audio modality through neural networks. In addition, the audio modality is mainly processed using neural networks. For comparison purpose, classical unsupervised approaches for audio feature extraction are also used. We expect visual data centered on the face of each participant to be very appropriate for detecting voice activity, based on the detection of lip and facial gestures. Thus, our baseline system uses visual data (video) and we chose a 3D Convolutional Neural Network (CNN) architecture, which is effective for simultaneously encoding appearance and movement. To improve this system, we supplemented the visual information by processing the audio stream with a CNN or an unsupervised speaker diarization system. We have further improved this system by adding visual modality information using motion through optical flow. We evaluated our proposal with a public and state-of-the-art benchmark: the AMI corpus. We analysed the contribution of each system to the merger carried out in order to determine if a given participant is currently speaking. We also discussed the results we obtained. Besides, we have shown that, for our application context, adding motion information greatly improves performance. Finally, we have shown that attention-based fusion improves performance while reducing the standard deviation.

✉ Lionel Pibre
lionel.pibre@gmail.com

[1] IRIT, Université de Toulouse, CNRS, INP Toulouse, UT3, Toulouse, France

[2] LAAS-CNRS, UT3, Toulouse, France

Keywords Active speaker detection · Multimodal fusion · Deep learning · Audio processing · Video processing · Speech analysis

1 Introduction

Meetings are essential in our society, at universities and industries, where they are commonly held to coordinate professional matters such as projects, research, and funds, but can also be informal when people discuss everyday issues. Usually, at some point during a meeting, the person of interest is the one who is speaking because she/he is the center of attention of all the participants. This task of detecting the person(s) speaking at a given time is called active speaker detection. Active Speaker Detection is a multimodal analysis task that consists of analysing a video, determining if the movements of one of the faces appearing correspond to the speech signal contained in the audio track. This task can also be performed only on the video based on the movements of the participants. Indeed, in such a context, communication takes place not only through voice but also relies on non-verbal signs as gestures, orientation, etc. Analysing this audiovisual information to estimate the person of interest or the active speaker can be useful in scenarios, such as human-robot interaction [18] or human-machine multiparty dialogue [15] among others, so a system able to interpret both cues is desirable. Various approaches to analyse meeting data have been carried out in the literature [8, 31]. Most of them focus on analysing audio-only, which indeed is a source that provides rich information. For example, multiple works explore ways to recognize the speaker [38, 40], others propose diarization techniques [11] that allow partitioning the incoming audio stream into homogeneous segments, i.e. audio sections with a single speaker label. Likewise, some approaches process the audio, turning it into text [42]. As overlapping speech can lead to miss detections or recognition errors, audio detection may not be efficient enough, and the visual detection of the person may lead to some improvements.

Visual-only approaches for active speaker detection may encounter several difficulties. Indeed, for example, occlusions or facial movements such as facial expressions or yawns [14] can mislead these approaches. Nevertheless, the audio information combined with the video information helps to overcome these difficulties [32]. Indeed, the movement of the lips and more generally of the head can help to detect if a person is speaking. Here, lips movement is encoded by spatiotemporal features, commonly estimated through Convolutional Neural Networks (CNN).

In previous work [26] we have also addressed such an issue. Our objective here is to go further and focus on the fusion of representations coming from audio, through speaker diarization results, and video modalities.

2 Motivations

The research work presented here is part of an industrial project aiming at developing smart assistants to automatically produce meeting summaries (LinTO https://linto.ai/). The long-term project objective is to focus on discussions between participants to company meetings, in order to perform high-level processing like segmentation into dialogue acts, study of their relationships or production of meeting summaries.

Processing spontaneous multi-party meetings has always been considered as a challenging task [27] as they correspond to complex interaction situations not well structured unlike edited audiovisual contents coming from TV shows, TV-series or movies and which have also been the subject of research over the past decade [5, 13]. In multi-party meetings several speakers may be active at the same time (as illustrated in Figs. 1 and 2 respectively from audio and visual point of view), speech transcription becomes a very challenging task in such conditions, including spontaneous and overlapping speech, variability acquisition conditions due to distant speech or camera field of view.

Within the applicative context of the LinTO project, we chose to focus on "active speaker detection" by studying cues coming from non-verbal social signals and more particularly by associating face movements with participant's speaking status in order to strengthen detection robustness. We are interested in how audio and visual perception, their low-level cues and their fusion could bring information about interactions between participants. Besides, by working with vector representations and merging these low-level results we also wanted to study solutions that avoid, as much as possible, to process personal information or learn speaker models. We are also independent from the spoken language.

The "active speaker detection" task is far more complex than determining whether there is somebody speaking (Voice Activity Detection). We consider it, here, as a prerequisite for higher-level components that should form the processing pipeline leading to conversation analysis and automatic meeting summaries, which both depend on the language used during meetings.

Thus, we organized our work around three following aspects:

- merge visual and audio features such that we obtain a robust final detection;
- find the most suitable fusion approach for our problem;
- focus on the audio modality in order to study its impact, as an additional modality, and compare the performance between a supervised and an unsupervised method involving or not any speaker models.

As previously explained, active speaker detection can be carried out either by using video alone or by combining the visual modality with the audio modality. Our approach is based

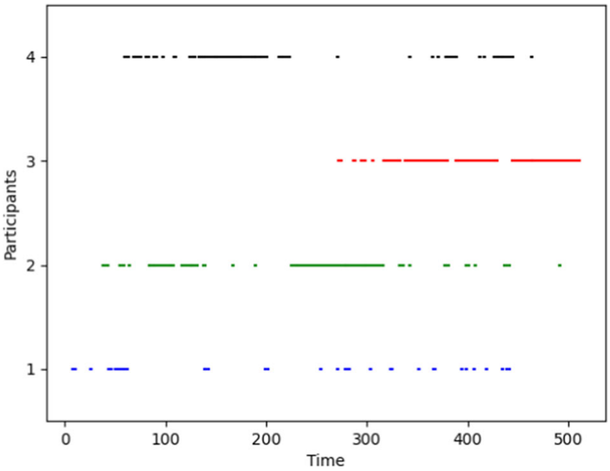


Fig. 1 Illustration of the type of result we wish to obtain. The abscissa axis represents the time in seconds, and the ordinate axis represents the speakers. We want to detect each time a participant becomes active, even if another participant is already active

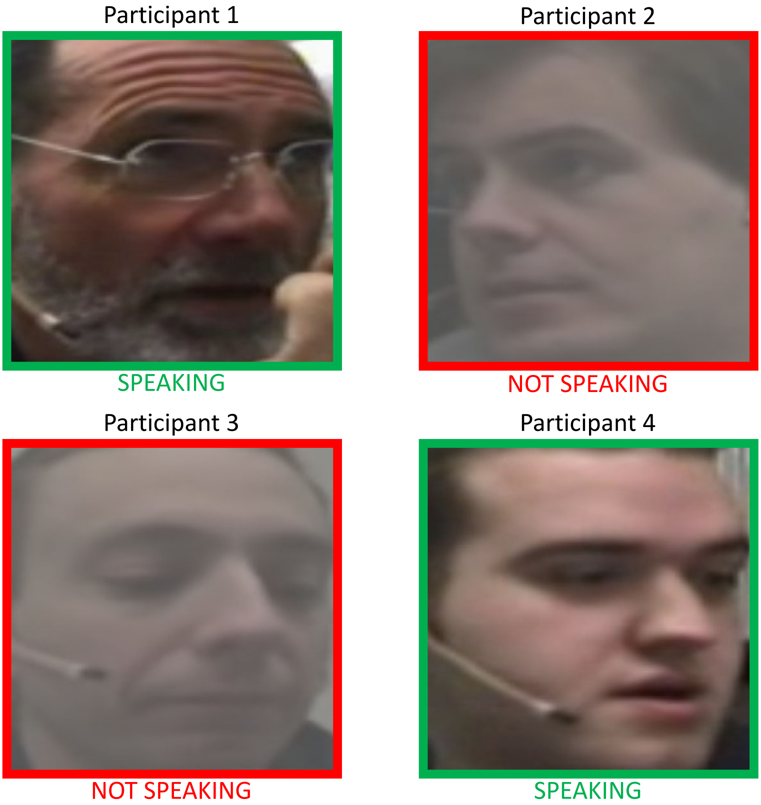


Fig. 2 Illustration of the type of result we wish to obtain. For this example, we look at the result obtained at a given time. In this case, we can see that participants 1 and 4 are talking, and participants 2 and 3 are listening and are therefore not active speakers

on the prior image detection of the face using state-of-the-art techniques. The principle is to analyse frames to characterize the orientation of the face in a continuous space in terms of pitch, yaw, and roll. Therefore, a "speaker" / "non-speaker" status can be inferred at once with only the pure visual percept.

However, in literature, there are few public multi-person audio-video datasets in our application context (meeting), which limits the scope of the evaluations and comparisons that we can perform. Nevertheless, the evaluations of the available datasets with increasing complexity, from mono to multi-person observed within the field of view, are encouraging for the different analyses and experiments that we have carried out. The main contributions of this paper are:

1. A pure visual speaker classifier, based on 3D CNNs, is applied in this original context. It takes as input video frames (i.e. clips) with the Red, Green, and Blue channels but also optical flow information.
2. The implementation and comparison of two fusion strategies to combine audio representations with the visual representations obtained. This shows that an attention-based fusion strategy [2, 25] between audio and video modalities significantly improves performance compared to a naive fusion strategy.

3. Our experiments show that an unsupervised speaker diarization system, used to produce audio representations, reaches almost the same performance as a neural network trained for this audio task in a supervised way, while requiring fewer parameters to train.

This paper has the following structure: Section 3 presents the related work and Section 4 we explain the processing that we apply to each of the modalities. Section 5 describes the strategies used for the fusion. Section 6 details the experimental protocol used, and the quantitative and qualitative results, including a discussion, are given in Section 7. Last, Section 8 concludes and mentions future work.

3 Related work

Communication situations and interaction, between humans commonly involve voice and gestures, encoding verbal and non-verbal signs, which computer process as audio and video signals respectively. Our work focuses on exploiting these signals to estimate the person (s) who is speaking at a given time, and point him/her then as the active speaker. From both signals, audio has been widely explored because it naturally captures whether somebody is speaking or not.

3.1 Audio-based methods

In literature, there is a large amount of work on speaker recognition [38]. For such purpose, one way to achieve it involves speaker diarization techniques [11] where the audio stream is partitioned into “speech” and “non-speech” segments, and where a speaker ID is assigned to each “speech” segment. In [3], Bonastre et al. proposed a speaker diarization method based on the binary key modeling, which transforms the audio into a feature representing the speaker within the binary space. Then, the diarization is performed by an iterative agglomerative clustering algorithm that forms segments of a same speaker. Patino et al. [29] improved the method by considering spectral clusterization. One of the main challenges is to be able to recognize the same person regardless of the intensity of the speaker’s voice, e.g. whispering, or the background noise which may alter speaker identification. Vestman et al. [38] did a deep taxonomy on different features that address these issues and proposed a sound time-varying feature that gives state-of-the-art results. With the rise of the Convolutional Neural Network (CNN), some proposals [20] have exploited such end-to-end solution for speaker diarization. The most recurrent architectures are those based on Long Short Term Memory (LSTM) Networks [33, 39] since they capture the variations in the voice of the speaker. One of the limitations of CNNs is that they are computationally expensive, generally requiring powerful GPUs to produce good results. In [41], the authors propose an end-to-end system at utterance-level for speaker verification. This method proposes a new “*thinResNet34*” trunk architecture, which incorporates a *GhostVLAD* layer allowing to aggregate features across time. The *thinResNet34* architecture has only 3 million parameters when a classic *ResNet-34* [17] has 22 million. It is trained and evaluated using the VoxCeleb dataset [28], an audiovisual dataset of short clips of interviews extracted from YouTube. In this dataset, with over 2000 hours of recordings and more than 7000 people, [41] has demonstrated the effectiveness of such a compact network by providing state-of-the-art performances.

However, whispers, background noises, or interspersed audio cause bad estimates. To overcome these difficulties, several approaches include visual features since those are not affected by these phenomena.

3.2 Video-based methods

Pure visual information is an important source to consider speaker recognition, especially if the audio is not available, corrupted, or unintelligible. Zhou et al. [44] give an in-depth review of advances in visual speaker recognition until 2014. The authors provided a list of datasets aimed for this purpose. Also, the presented methods are grouped according to the type of features used, highlighting three groups:

- Image-based group. Here raw pixels are transformed directly as visual features with the aid of methods such as Principal Component Analysis (PCA) [19, 24]. More recently, the authors of [34] have used neural networks for the detection of active speakers. To do so, it first uses a face detector, then the authors use the AlexNet [23] network to extract features on RGB color data for all found faces, and finally they give these features to a recurrent LSTM network.
- Motion-based group. Features describe the observed movement during speaking. Instead of creating handcrafted features, several proposals estimate the motion directly from the video with techniques such as Optical Flow (OF) [24].
- Geometric-based group. The features focus on the geometric information of a moving mouth. In [22], the movement is computed by measuring the distance between points detected over the mouth. Then, the difference of distances between successive images represents the motion. The main limitation of these methods is that any movement of the mouth is associated with a speaking status. So the number of false positives is commonly high. To overcome this problem, this motion information can be combined with the audio signal and/or the raw images.

3.3 Audiovisual-based methods

Audiovisual methods are robust to background noise and different speaking modes (whisper). Recently deep learning has provided advances in audiovisual speaker estimation by capturing the temporal relationships of visual and acoustic cues. The use of a recurrent neural network (RNN) is explored in [35] to extract video features using 2D CNN. Then, in a similar way as [22], they train an LSTM layer for each modality. The output of both layers is concatenated and the result is used to train a final LSTM layer. This process is known as early fusion, on the contrary, it is called late fusion when the outputs are merged at the end without any other training layer pursuing them. In [30], Petridis et al. compares both fusion methods in the context of speaker recognition. Afouras et al. [1] explore the impact of training audiovisual lip-reading network with different loss functions.

In the literature, the application of LSTM layers has been widely studied because it is good at capturing spatiotemporal information related to a speaker. Besides, other CNN architectures learn this aspect but are used in different classification contexts. Such is the case of the deep 3-Dimensional Convolutional Network (C3D) [36]; here the objective is to classify actions such as "*biking*", "*running*", among others. This 3D CNN model has been extended to other architectures, in [37] they study the use of a 3D residual neural network (*ResNet3D*). In both cases, the goal is to obtain a 3D feature that encodes simultaneously appearance and motion. We aim to study these networks in our context of speaker detection because these proposals show good performances with state-of-the-art methods. The classic CNN-based image classification approach learns from raw images. Some works have shown that the use of additional features like optical flow [4, 24, 37] or depth cue [4], help to improve estimation.

3.4 Synthesis

The main drawback of video-based applications (whether pure or in conjunction with audio) is that their evaluations are conducted in constrained situations where there is usually only one person, or few people looking straight at the camera. This scenario is not realistic in the context of a traditional meeting. Therefore, we propose to use a classifier, which considers spatiotemporal features, trained from videos in realistic meeting situations.

Additionally, the video can provide more information if we consider the social aspect among the meeting participants. For example, a person who speaking will move, have facial expressions and make gestures to support his or her arguments. That is why in this work, different fusion strategies have been studied. First, let's describe how each modality has been processed.

4 Mono-modality processing

4.1 Visual modality processing

Let's start by looking at the visual modality. We worked with two visual modalities: the RGB images of the videos, and the optical flow image calculated from these images.

4.1.1 Video-based network architecture

In order to obtain a visual representation, we consider a 3D CNN architecture. Since we want to capture the temporal relationship that exists between the images, we group several consecutive images to form a non-overlapped video clip of L frames. The input clip has a size $3 \times L \times H \times W$, where H and W are the height and width of the frame in the Red, Green, and Blue (RGB) channels. We relied on [36] to set this parameter and chose $L = 16$ as clip size, i.e. the number of frames. Thus, the clips are given as input to the networks.

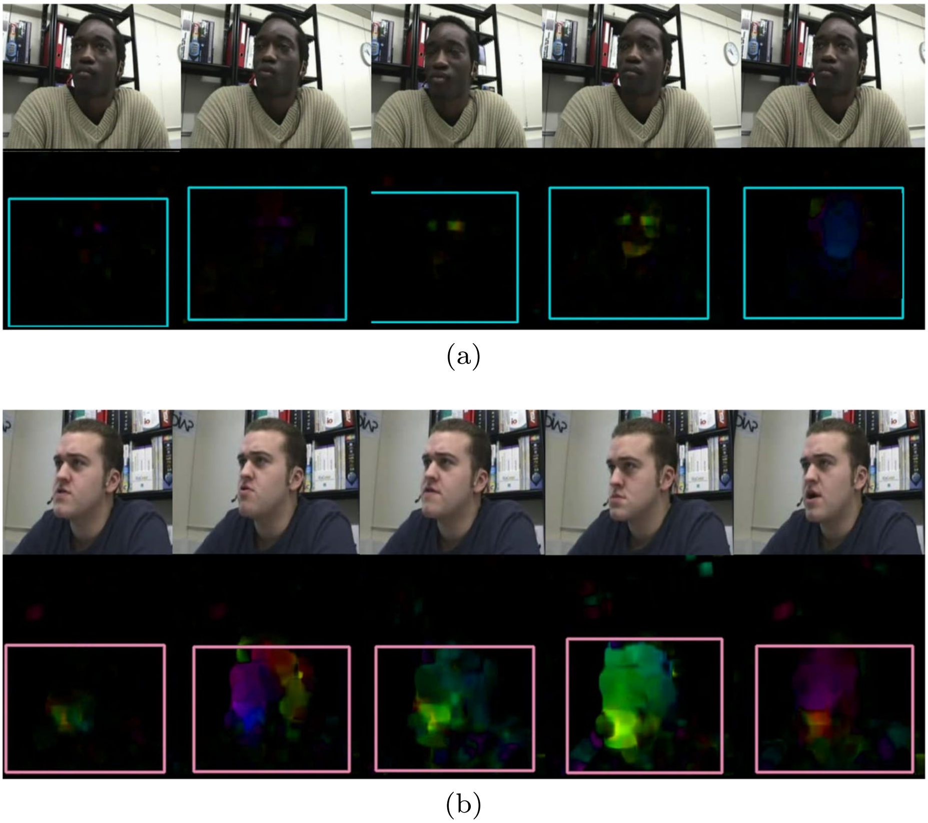
In comparison with 2D CNN, here we have a 4-dimensional tensor. The difference with a classic 2D CNN is that the convolution is performed in 3D, allowing the feature maps to learn temporal context. Thus, the convolution layers create a 3D feature where the initial part focuses on the appearance of the first images and the rest considers the salient motion [36].

In our case, we chose to use the *ResNet3D* architecture. This architecture takes up the idea of *ResNet* residual blocks [17] but using a 3D convolution instead of a 2D one. This preserves and propagates the temporal reasoning through the network layers. Input is a 4D tensor and the image size being the same as *ResNet2D*: $3 \times 16 \times 224 \times 224$. Here, we evaluate the *ResNet3D-18* version, i.e. considering 18 blocks. In the rest of this paper, we will call this method "Video-net". Video-net is trained with RGB images that show only one participant's face.

The output of this network indicates the probability that the analysed participant is speaking or not.

4.1.2 Optical flow calculation and processing

In order to add the facial movement information, we took inspiration from the works [4, 24, 37], and encoded this information through the optical flow. We calculated the optical flow



(a)

(b)

Fig. 3 We show in this figure two examples, (a) and (b), of images given to CNN. For these two examples, we have at the top the images of the original video and at the bottom the magnitude of optical flow images. We can clearly see that the person in example (a) is blinking while the person in example (b) is speaking

with the OpenCV[1] library. Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of objects or cameras. In our case the cameras are fixed, so it will be the movements of the participants.

We present two examples in Fig. 3, where we have for each example at the top the RGB images and at the bottom the magnitude of optical flow images. As you can see, the person in example (a) is blinking while the person in example (b) is speaking.

We trained a network with two branches: one branch for the raw image and one for the optical flow. Each of the branches is a *ResNet3D*. We remove the top of each network and concatenate both visual representations. After concatenation, two fully connected layers are trained.

[1] https://docs.opencv.org/3.4/index.html

4.2 Audio modality processing

Now let's focus on the two audio approaches we used.

4.2.1 Audio-net system

For the audio representation, we use the author's implementation of [41], named VGG-Speaker-Recognition framework.[2] The first part of this network (*thinResNet34*) takes as input an extract of the spectrogram computed from the audio file. This network uses the principle of shared weights on all extracts belonging to the same audio file.

The second part of this network is made up of a dictionary-based *GhostVLAD* layer [43] to aggregate features across time. The objective of this part is to obtain a vector of a fixed size as output of the network whatever the length of the audio file processed.

The spectrograms are calculated with the Librosa library[3] from audio clips corresponding to the duration of the video clip, i.e. 0.64 second, with 256 frequency components. The spectrogram is normalized by subtracting the mean and dividing by the standard deviation. The *thinResNet34* network was trained with Adam optimizer and an initial learning rate of $1e$-2.

This network produces an audio representation that is initially intended to respond to a speaker recognition task, or a speaker verification task. For our problem, the vector produced by this network will be merged with other vectors coming from other systems components applied to audio or video data. In the rest of this paper, we will call this method "Audio-net".

4.2.2 pyBK speaker diarization system

After carrying out a study on the existing speaker diarization system applied to meeting data from AMI, *pyBK* [29] was selected as showing the best performances. For this speaker diarization system, baseline acoustic features are MFCCs comprising 60 static coefficients computed from windows of 250 ms with a 100 ms shift and with a filter bank of 60 channels. The binary key background model (KBM) is determined from a pool of Gaussians, each estimated using windows between 0.5 and 2-second duration set dynamically to ensure a minimum of 1024 components. The size of the KBM after Gaussian selection is set to 320. The top number of Gaussians per frame is set to 10. We empirically set the number of initial clusters at 12. We used the Jaccard distance metric to select the output clustering solutions, and we employed the elbow method for the selection of the best number of clusters. All parameters were set experimentally.

To be more specific, for our approach, we used only the output of *pyBK* in our global system. Indeed, in the case of *pyBK*, we directly merge the result of this approach with Video-Net and Audio-net presented earlier. To do this we used hot vectors. A hot vector is a representation of categorical variables as binary vectors. Here, we took as vector size the maximum number of speakers found by *pyBK* applied on the whole set of meeting recordings. So we created hot vectors of a dimension $N_{max} + 1$, to be able to encode when no one is speaking. To merge this hot vector with Video-net and Audio-net, we use a Gated

[2] https://github.com/WeidiXie/VGG-Speaker-Recognition

[3] http://librosa.org/doc/main/index.html

Recurrent Unit [9] (GRU) layer proposed by Cho et al. [9]. Its role is to make each recurrent unit to adaptively capture dependencies of different time scales. The GRU has gating units that modulate the flow of information inside the unit.

The activation h_t^j of the GRU at time t and layer j is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\tilde{h}_t^j$:

$$h_t^j = \tilde{h}_t^j(1 - z_t^j) + z_t^j h_{t-1}^j \tag{1}$$

where an update gate z_t^j decides how much the unit updates its activation.

The update gate is computed by:

$$z_t^j = \sigma\left([\mathbf{W}_z\mathbf{x}_t]^j + [\mathbf{U}_z\mathbf{h}_{t-1}]^j\right) \tag{2}$$

where σ is a logistic sigmoid function, and $\mathbf{x_t}$ is an element of a given sequence $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_T})$. The logistic sigmoid function will transform the values between 0 and 1, allowing the gate to filter between the less-important and more-important information in the subsequent steps.

The candidate activation $\tilde{h}_t^j$ is computed as in (3) from [2].

$$\tilde{h}_t^j = \tanh\left([\mathbf{W}\mathbf{x}_t]^j + [\mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1})]^j\right) \tag{3}$$

where r_t is a set of reset gates, tanh is hyperbolic tangent, and $\odot$ is an element-wise multiplication. When off (r_t^j close to 0), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state.

The reset gate r_t^j is computed similarly to the update gate:

$$r_t^j = \sigma\left([\mathbf{W}_r\mathbf{x}_t]^j + [\mathbf{U}_r\mathbf{h}_{t-1}]^j\right) \tag{4}$$

Since the input of Video-net is clips of 16 frames, we have extracted the result vector of *pyBK* at the time of each frame. Thus, we have at the input of the GRU layer 16 vectors of size 8.

5 Fusion strategies

Each module, described in the former section, has been designed to produce feature vectors assumed to be characteristic of the speaking or non-speaking status of the current participant, considering a single modality. Because we are interested in studying the impact of additional audio cues to video results, different dimensions were taken into account for the vectors coming from Audio-net or pyBK (16, 32, 64, 128), while each output vector from Video-net, for RGB images or optical flow, was fixed to the same dimension (128). Fusion has been operated by combining those feature vectors as presented in Fig. 4 (vector concatenation). Two approaches were tested: a naive fusion as a baseline version and an innovative one, involving attention mechanisms.

5.1 Naive fusion

For this fusion, we have applied the easiest way to merge these vectors, by concatenating them. However, one of the vectors is likely to be more discriminating than others. This is why we made the size of the various vectors vary to be able to analyse the importance of each method on the fusion process.

We present, in Fig. 4, the global architecture used to perform the fusion. In the case of naive fusion, attention modules (in blue) are not used. The two top branches are associated with audio modality processing. These two branches are never activated at the same time, either Audio-net or *pyBK* is used. The two lower branches are associated with visual modality processing via Video-net. When RGB images and optical flow are used, a pre-fusion is carried out between the two representations coming from each Video-net network.

5.2 Attention-based fusion

Then, we proposed a second fusion strategy introducing attention layers, as shown in Fig. 4 (in blue). Each single modality representation $\mathbf{x}^m$ (m being one modality: Audio, RGB or OF) is sent to the input of an attention layer, which in turn, produces a new representation $\mathbf{o}^m$. These new representations are then merged in the fusion step.

The principle of the attention mechanism is to give more importance to the elements of a feature vector that are the most discriminating. In practice, attention is simply a vector, often the outputs of a dense layer using softmax function, as detailed below.

Let $\hat{\mathbf{x}}^m$ be the result of applying the softmax on the output vector $\mathbf{x}^m$ of a modality m with $\mathbf{x}^m = x_1^m \dots x_n^m$, where n is the representation dimension. $\hat{\mathbf{x}}^m$ is computed as:

$$\hat{\mathbf{x}}^m = \left(\frac{e^{x_i^m}}{\sum_j e^{x_j^m}} \right) \tag{5}$$

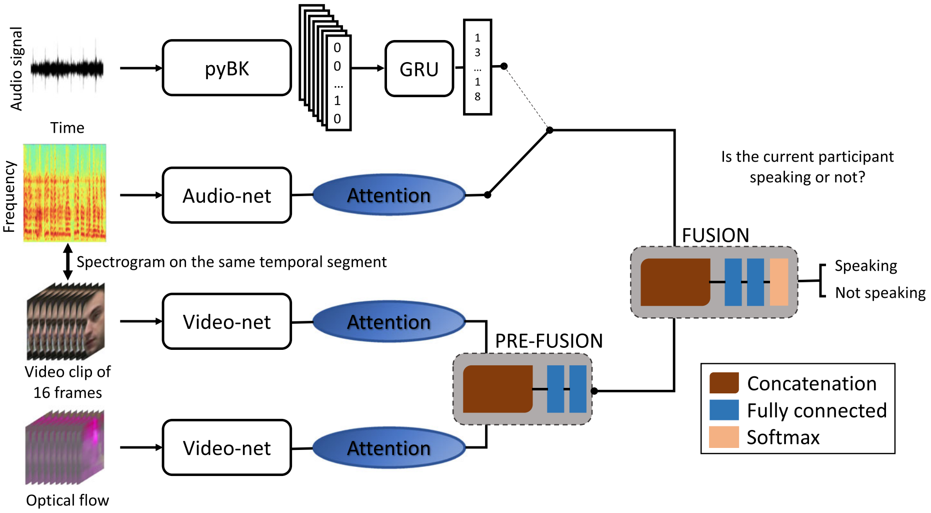


Fig. 4 Illustration of the architecture we used. The top two branches are associated with audio modality processing. In this work, *pyBK* is an alternative to Audio-net, so these two branches are never activated at the same time, either Audio-net or *pyBK* is used. The two lower branches are associated with visual modality processing via Video-net. Fusion is represented in grey boxes where the feature vectors are first concatenated (in brown) to feed two fully connected layers (in blue). The difference between the attention-based fusion strategy and the naive fusion approach relies on the use or not of attention mechanisms (blue ellipses)

Table 1 Information about the IDIAP remote control scenario meetings

# meetings	# participants	Total	Avg per meeting
38	4	17h 44min	28 min

Finally, to obtain the output vector $\mathbf{o}^m$ for this modality, we multiply term by term the vectors $\hat{\mathbf{x}}^m$ and $\mathbf{x}^m$:

$$\mathbf{o}^m = \hat{\mathbf{x}}^m \cdot \mathbf{x}^m \tag{6}$$

These $\mathbf{o}^m$ vectors are then concatenated: RGB with OF first in a pre-fusion step; and then with Audio. Concatenation results are then feeding two fully connected layers as shonw in Fig. 4.

6 Experimental protocol

Our experiments have been carried out using the Keras Python library on a workstation with an Intel Xeon E-2286G 4.0 GHz CPU, 64 Gb of RAM, and one NVIDIA GeForce RTX 2080 Ti. We also used the OSIRIM computing cluster[4] composed of 7 GPU servers each equipped with 2 Xeon 2640 V4 @ 2.40 GHz CPUs and 4 NVIDIA GeForce GTX 1080 Ti GPUs.

6.1 Dataset

Our first objective was to apply our approach to a dataset dedicated to meeting and freely available, while waiting for the internal dataset on professional meetings designed for LinTO, in order to compare the effectiveness of our approach on both datasets as done in [10]. So we have first evaluated our method using the AMI Corpus (Augmented Multi-party Interaction) [7]. This dataset consists of over 100 hours of meeting recordings. The recordings use a range of signals synchronized to a common timeline and recorded with different sensors: close-talking and far-field microphones, individual and room-view video cameras. It also includes outputs from a slide projector and an electronic whiteboard. During each meeting, the participants have also unsynchronized pens available to them that record what is written.

The meetings were recorded in English using three different rooms with different acoustic properties, and include mostly non-native speakers. To make sure that we keep the same meeting context (microphones, cameras...), we have selected the IDIAP “remote control scenario” meetings. The information for this scenario is presented in the Table 1.

This scenario includes 38 recorded meetings with four participants each, whose roles are: Project manager, Industrial designer, User interface designer and Marketing expert.

There are 4 cameras and each one records a single participant as shown in Fig. 5.

For the audio part, each participant is recorded with a headset with a microphone. These recordings are then mixed into a single audio file. In the case of our project, only one microphone will be used to record the meetings, so we chose to only use “mix headset” to be in the same application context and to simulate real conditions.

The AMI corpus has detailed ground truth (GT) at the audio level but has no information regarding the video. First, we use this dataset to train and evaluate the performances of the

[4] https://osirim.irit.fr

Fig. 5 Examples of images from the AMI dataset captured by individual cameras

visual-based networks following a cross-validation methodology. Thus, all recorded meetings are divided into five cross-validation folders: four are used to train the models and one for testing.

The CNNs are trained considering individual images only (no general view image) using RGB. The face is detected with *ResNetSSD FaceDetector* from the CAFFE library. The samples are grouped in clips of 16 frames successive and labeled as speaker or non-speaker according to the audio-based GT. The images are rescaled to 224 × 224 pixels which is the size required for *ResNet*-based network.

6.2 Evaluation metrics

To evaluate the different models we proposed and to measure the efficiency of our fusion strategies, we used the macro and micro area under the curve [6, 16] (AUC) applied to the receiver operating characteristic curve (ROC) which is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The TPR and FPR are defined by:

$$TPR = \frac{TP}{TP+FN} \tag{7}$$

$$FPR = \frac{FP}{FP+TN} \tag{8}$$

where TP is the number of true positive, $TP + FN$ is the number of real positive cases in the dataset, FP is the number of false positive, $FP + TN$ is the number of real negative cases in the dataset.

ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model will predict the correct class for a new sample.

Macro AUC calculates metrics for each label and computes its unweighted mean. This does not take label imbalance into account. Micro AUC globally calculates metrics by considering each element of the label indicator matrix as a label.

6.3 Assessment of the speaker diarization system

Concerning the results of *pyBK* on the task of speaker diarization, we did not look after a better performance on this task. Our goal was to figure out if we could improve the result of the active speaker detection task by adding this quickly-obtained information, without a learning phase as it is based on an unsupervised method.

To assess this speaker diarization system, we used a 5-fold cross-validation on all the audio files of the IDIAP remote control scenario meetings from the AMI corpus.

Table 2 Results obtained with *pyBK* applied on meetings from the IDIAP "remote control scenario"

Missed detection rate	False alarm rate	Confusion rate	DER
50.52%	3.46%	16.02%	70.0%

To evaluate pyBK on its diarization performances, we used the Diarization Error Rate (DER), which is the standard metric for evaluating and comparing speaker diarization systems. This measure was introduced for the NIST Rich Transcription Spring 2003 evaluation (RT-03S). It is defined as follows:

$$DER = \frac{False\ alarm + Missed\ detection + Confusion}{Total} \tag{9}$$

where *False alarm* is the duration of non-speech incorrectly classified as speech, *Missed detection* is the duration of speech incorrectly classified as non-speech, *Confusion* is the duration of speaker confusion, and *Total* is the total duration of speech in the reference.

As can be seen in Table 2, the results we have obtained with the parameters described above do not reach those of the state-of-the-art (for example a DER of 12.81% in [12]).

Besides, as we can see, the fact that the DER is as high comes mainly from the fact that we have a high rate of missed detection. Indeed, the false alarm rate is only 3.46%, and the confusion rate is 16,02%, which is relatively low. These scores mean that even if this approach misses half of the speech segments, the speech segments are generally attributed to the right speakers.

6.4 Training settings

Video-net and Audio-net have been trained with a batch size of 20 clips, the Adam optimizer [21], and an initial learning rate of 0.05. Training is stopped after 21 epochs.

Besides, since we did not have access to the weights of the Video-net model when we did our experiments, all our models were trained from scratch to have a fair comparison.

6.5 Evaluation protocol

We present in Fig. 6 how our evaluations were carried out.

For each meeting, here the IS1004a meeting, we evaluated our system by analysing in parallel the flow of each participant, using the same model (shared weights) for all the participants. For the audio part, we use the mix-headset audio recording that consists of voices from all the speakers, so we have a single audio file per meeting.

To summarize, for each evaluated meeting, we have 4 video streams, one for each participant, and a single audio stream, containing the voices of all participants.

7 Evaluation of our fusion strategies

In this section, we present the results obtained during our experiments. In Table 3, the size of the vector used for fusion is indicated after the name of the neural networks. For example, Video-net_128 + Audio-net_16 means that the output of Video-net is a 128 dimensional vector and the output vector of Audio-net is a 16 dimensional vector.

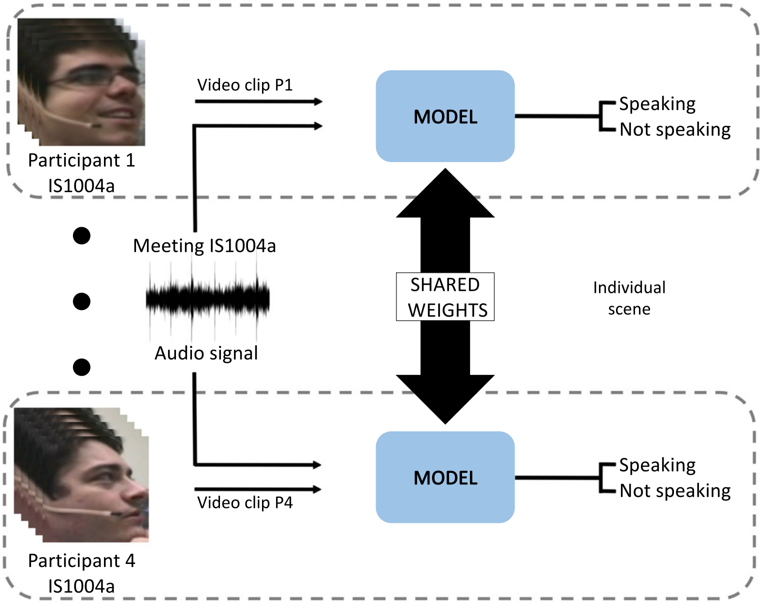


Fig. 6 Method used to carry out the evaluation of our different models

We have not experimented with Audio-net alone because the objective of this network is to support Video-net. By bringing information such as the fact that somebody is speaking and when there is a speaker change. We remind the reader that Audio-net is initially used for speaker recognition and does not allow the detection of the active speaker as it is.

7.1 Results related to Video-net

We want to evaluate the potential of visual information in the first step. We started by evaluating the performance of Video-net. As can be seen in the first row of Table 3, Video-net

Table 3 Active speaker detection results in terms of Macro AUC using Video-net and Audio-net

Fusion	Model	Macro AUC
	Video-net-RGB	72.4% ± 3.5
Naive		
	Video-net-RGB_128 + Audio-net_16	76.2% ± 4.4
	Video-net-RGB_128 + Audio-net_32	**76.8%** ± **4.7**
	Video-net-RGB_128 + Audio-net_64	76.6% ± 5.1
	Video-net-RGB_128 + Audio-net_128	76.4% ± 5.2
Attention		
	Video-ne-RGB_128 + Audio-net_16	79.4% ± 3.7
	Video-net-RGB_128 + Audio-net_32	78.6% ± 3.0
	Video-net-RGB_128 + Audio-net_64	**79.6%** ± **2.9**
	Video-net-RGB_128 + Audio-net_128	78.0% ± 2.2

The best result of each fusion is shown in bold

obtains an area under the curve equal to 72.4%. This result will be our baseline for the following experiments.

7.2 Fusion between Video-net and Audio-net

In Table 3, we present the results we obtained by merging Video-net and Audio-net. We first present those obtained with the naive fusion, and then the results with the attention-based fusion. As can be seen, merging representations slightly improves performance. Indeed, with this fusion, we manage to achieve at best a macro AUC of 76.8%, an improvement of more than 4% compared to Video-net alone.

When we look at attention-based fusion, we can see that there is a very strong improvement in performance. Indeed, for this fusion, we can see an improvement of up to 7.2% compared to Video-net alone. The use of attention layers on the output vectors of each modality make fusion more efficient.

Same evaluations were applied to video, audio, and optical flow.

7.3 Fusion between audio, video and optical flow

We present the results of merging video, audio, and optical stream in Table 4.

First of all, we can notice that processing the optical stream alone gives better performance than RGB data processing. Indeed, with optical flow we have a 3% improvement.

Then if we look at the performances obtained by merging video with optical stream. We can see that this greatly improves performances. In fact, adding this information gives a macro AUC of 81% for naive fusion and 82.4% for attention-based fusion. This gives a gain of more than 9% compared to Video-net alone.

Let us now focus on naive fusion as shown in Table 4 involving audio modality. We can see that the addition of Audio-net allows a gain of about 4%. However, when looking at the results for attention-based merging, we can see that the performances are similar ($\approx$ 82%).

Table 4 Active speaker detection results in terms of Macro AUC using Video-net, Audio-net and optical flow

Fusion	Model	Macro AUC
	Video-net-RGB	72.4% ± 3.5
	Video-net-OF	75.4% ± 4.0
Naive		
	Video-net-RGB_128 + Video-net-OF_128	81.0% ± 4.4
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_16	82.2% ± 4.0
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_32	81.6% ± 3.5
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_64	**83.4% ± 4.4**
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_128	82.4% ± 3.9
Attention		
	Video-net-RGB_128 + Video-net-OF_128	82.4% ± 2.3
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_16	82.4% ± 2.5
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_32	**84.0% ± 2.8**
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_64	82.5% ± 3.0
	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_128	83.0% ± 3.6

The best result of each fusion is shown in bold

This is also true for the fusion between Video-net and Audio-net. This fusion allows us to obtain the best performance with a macro AUC of 84%. However, we can notice that with this fusion we get a smaller standard deviation than with the naive fusion.

7.4 Alternative and lighter fusion between Video-net and *pyBK*

In our approach, we also tried to compare results when using an unsupervised method for speaker diarization based on *pyBK*. The same kinds of experiments were carried out and presented in Table 5.

As we can see, the naive fusion between Video-net-RGB and *pyBK* output vectors gives better performance than Video-net with a gain of about 3% in the best case. However, this fusion is less efficient than the one between Video-net and Audio-net. This does not seem significant but we can highlight one possible advantage which concerns the number of parameters involved in computation. This can be of interest if the process has to be embedded.

When we focus on attention-based fusion, we can see that we get the same performances as between Video-net-RGB and Audio-net, an improvement of up to 7% compared to Video-net alone.

When we focus on the fusion between Video-net, *pyBK*, and optical flow, we can see that the performances are not as good as when we used Audio-net. Indeed, with *pyBK*, the performances are 1% lower.

7.5 Discussion

As we saw earlier, by merging both visual and audio modalities (using *pyBK* or Audio-net) performances were higher than our baseline video-based system. Let us now go a little further in the analysis of our results.

As can be seen in Tables 3, 4, 5 and 6, we show the evolution of the macro AUC as a function of the repartition in the fusion of Video-net-RGB, and Audio-net, or *pyBK*'s output vector.

Table 5 Active speaker detection results in terms of Macro AUC using Video-net, and pyBK's output vectors

Fusion	Model	Macro AUC
	Video-net-RGB	72.4% ± 3.5
Naive		
	Video-net-RGB_128 + pyBK_16	**75.6%** ± **4.8**
	Video-net-RGB_128 + pyBK_32	74.2% ± 5.2
	Video-net-RGB_128 + pyBK_64	74.0% ± 3.3
	Video-net-RGB_128 + pyBK_128	74.2% ± 4.0
Attention		
	Video-net-RGB_128 + pyBK_16	79.2% ± 2.6
	Video-net-RGB_128 + pyBK_32	**79.4%** ± **2.7**
	Video-net-RGB_128 + pyBK_64	78.8% ± 3.7
	Video-net-RGB_128 + pyBK_128	78.0% ± 2.4

The best result of each fusion is shown in bold

Table 6 Active speaker detection results in terms of Macro AUC using Video-net, pyBK's output vectors, and optical flow

Fusion	Model	Macro AUC
	Video-net-RGB	72.4% ± 3.5
	Video-net-OF	75.4% ± 4.0
Naive		
	Video-net-RGB_128 + Video-net-OF_128	81.0% ± 4.4
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_16	81.8% ± 3.5
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_32	81.6% ± 3.5
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_64	**82.6% ± 2.9**
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_128	81.4% ± 3.7
Attention		
	Video-net-RGB_128 + Video-net-OF_128	82.4% ± 2.3
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_16	82.8% ± 2.9
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_32	81.8% ± 3.3
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_64	**83.0% ± 2.9**
	Video-net-RGB_128 + Video-net-OF_128 + pyBK_128	81.2% ± 2.9

The best result of each fusion is shown in bold

The variation of vector size has a small impact on the results. Indeed, the biggest difference one can encounter within the same fusion is only 1.8% with the Video-net-RGB_128 + Video-net-OF_128 + pyBK attention-based fusion.

An interesting point to observe on the overall results is that we did not obtain the best performance when Audio-net has a vector size of 128. This observation is also true when we merge Video-Net with the vector obtained with *pyBK*. Thus, we can deduce that the audio vector is only used here as a support for the video.

Moreover, interesting information to take into account is the number of parameters to be learned. Indeed, *pyBK* is merged with the other networks through a GRU layer. When the output of the GRU layer is set to a vector of size 16, the fusion of *pyBK* with a neural network requires only 66,512 parameters to be trained. Table 7 shows the number of parameters per model.

In this work, we also proposed a lighter alternative by merging Video-Net with the output vector of *pyBK*. Indeed, the Video-net-RGB_128 + *pyBK*_16 fusion has 33 million parameters to compute, while Video-net-RGB_128 + Audio-net_16 has 53 million. Moreover, having fewer parameters means that the learned model will be lighter, for example Video-net-RGB_128 + *pyBK*_16 weighs only 133 MB compared to 215 MB for Video-net-RGB_128 + Audio-net_16.

Table 7 Number of parameters per system

Model	#parameters
Video-net-RGB	33.21M
Audio-net_128	21.95M
Video-net-RGB_128 + Audio-net_16	53.56M
Video-net_128 + *pyBK*_16	33.28M

Table 8 Summary of all our results obtained by merging the different approaches

Fusion	Model	Macro AUC
	Video-net-RGB	72.4% ± 3.5
	Video-net-OF	75.4% ± 4.0
Attention	Video-net-RGB_128 + pyBK_32	79.4% ± 2.7
Attention	Video-net-RGB_128 + Audio-net_64	79.6% ± 2.9
Attention	Video-net-RGB_128 + Video-net-OF_128	82.4% ± 2.3
Attention	Video-net-RGB_128 + Video-net-OF_128 + pyBK_64	83.0% ± 2.9
Attention	Video-net-RGB_128 + Video-net-OF_128 + Audio-net_32	84.0% ± 2.8

Only the best fusion result between the different modalities is presented

In addition to the results presented in the Tables 3 to 6, we have calculated the micro AUC. We have not included these results because it is the same behavior, but it is 1% lower on all values.

We present, in Table 8, a summary of the best results we have obtained by merging the different modalities and approaches. As can be seen, all the results presented are the result of the attention-based fusion. As we have shown in our experiments, this fusion allows to make the most of each modality and thus to obtain the best detection of the active speaker.

8 Conclusion

This paper presents a framework for active speaker detection in a meeting context using two fusions between three approaches / modalities. During our experiments, we used only one audio representation at a time, either the feature extracted through a CNN (Audio-net) or the feature extracted with *pyBK*, a speaker diarization system. The other two representations used are based on visual information. We used the AMI corpus, and from that the video from each camera filming the participants individually of each meeting. We analysed the visual information through 3D CNN (Video-net), which is generally used to classify actions such as "cycling". We used this particular neural network to encode the spatiotemporal aspect of a participant in the active speaker detection task. The last feature we used was based on motion via optical flow. This feature is used in combination with video. In order to take this information into account, we made a joint learning with the video with a two-branch network.

We first evaluated Video-net-RGB and used this result as a baseline system. With this experiment, we obtained a macro AUC of 72.4%. We then merged Video-net-RGB with Audio-net. We show that merging these two modalities improves results by at least 4%. We have also noticed by combining these two modalities that the attention-based fusion gives better performances since in the best case, we have a gain of more than 7%. Next, we showed that adding motion information with the optical flow greatly increases the results. In fact, merging video with the optical stream allows reaching a macro AUC of 78.8%, an improvement of more than 6% compared to Video-net alone. Moreover, by adding the audio modality through Audio-net, we show that it further improves performance with a macro AUC of over 83% regardless of the fusion used.

In a second step, we carried out the same experiments by replacing Audio-net with a lighter alternative which is *pyBK* speaker diarization system. We have shown in the course of our experiments that this alternative allows us to reduce the number of parameters while maintaining performances very close to those obtained with Audio-net.

The continuation of this work consists in using our proposed approach on the project internal dataset with new issues. Indeed, in our project, there may be a higher number of participants. Moreover, the number of participants is not always the same. Another difference in the data is that in our case a single microphone is used for all participants. In the case of the AMI corpus data, each participant has a microphone. For this work we used the audio file mixing all the recordings in order to get closer to the recording conditions of our project. However, by using a single microphone for all the participants, the distance to the microphone of each participant will be different, so we will have other difficulties. Finally, the video acquired within the scope of the LinTO project were made with a 360 camera as in [10]. Applying our approach to this dataset, not yet available at the time of this submission, will be a way to assess its robustness and validate its integration in the pipeline dedicated to the LinTO project and its long-term objective.[5]

Funding This work was supported by the LinTO project (2018-2021) P169201-2658717// DOS0069247, funded by Bpi France, as part of the French project "PIA3: Programme d'Investissements d'Avenir 3".

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Afouras T, Chung JS, Senior A, Vinyals O, Zisserman A (2019) Deep audio-visual speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). https://doi.org/10.1109/tpami.2018.2889052
2. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd International conference on learning representations, ICLR. San Diego, USA. http://arxiv.org/abs/1409.0473
3. Bonastre JF, Anguera X, Sierra GH, Bousquet PM (2011) Speaker modeling using local binary decisions. In: Proc. Conference of the international speech communication association, Interspeech, pp 13–16. Florence. http://www.isca-speech.org/archive/interspeech_2011/i11_0013.html
4. Borghi G, Venturelli M, Vezzani R, Cucchiara R (2017) POSEidon: face-from-depth for driver pose estimation. In: Proc. IEEE Conference on computer vision and pattern recognition CVPR, Honolulu, pp 5494–5503. https://doi.org/10.1109/cvpr.2017.583
5. Bost X, Linarés G, Gueye S (2015) Audiovisual speaker diarization of TV series. In: Proc. IEEE International conference on acoustics, speech and signal processing, ICASSP, South Brisbane, pp 4799–4803. https://doi.org/10.1109/ICASSP.2015.7178882
6. Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2
7. Carletta J (2007) Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. Lang Resour Eval 41(2):181–190. https://doi.org/10.1007/s10579-007-9040-x
8. Chakravarty P, Zegers J, Tuytelaars T, Van hamme H (2016) Active speaker detection with audio-visual co-training. In: Proc. 18th ACM international conference on multimodal interaction, ICMI, Tokyo, pp 312–316. https://doi.org/10.1145/2993148.2993172

[5] https://www.irit.fr/SAMOVA/site/projects/current/linto/

9. Cho K, van Merriënboer B, Gulcehre C, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. ACL Conference on empirical methods in natural language processing, EMNLP, Doha, pp 1724–1734. https://doi.org/10.3115/v1/d14-1179
10. Chung JS, Lee BJ, Han I (2019) Who said that?: audio-visual speaker diarisation of real-world meetings. In: Proc. Conference of the international speech communication association, Interspeech, pp 371–375. Graz. https://doi.org/10.21437/Interspeech.2019-3116, https://www.isca-speech.org/archive/pdfs/interspeech_2019/chung19b_interspeech.pdf
11. Das A, Bhattacharjee U, Mitra DK (2017) One-decade survey on speaker diarization for telephone and meeting speech. International Journal of Scientific Research in Computer Science Engineering and Information Technology IJSRCSEIT 2(5)
12. Dubey H, Sangwan A, Hansen JH (2019) Transfer learning using raw waveform sincnet for robust speaker diarization. In: Proc. IEEE International conference on acoustics, speech and signal processing, ICASSP. IEEE, Brighton, pp 6296–6300. https://doi.org/10.1109/ICASSP.2019.8683023
13. el Khoury E, Sénac C, Joly P (2014) Audiovisual diarization of people in video content. Multimedia Tools and Applications MTAP 68(3):747–775. https://doi.org/10.1007/s11042-012-1080-6
14. Everingham MR, Sivic J, Zisserman A (2006) Hello! my name is... buffy" – automatic naming of characters in TV video. In: Proc. of the British machine vision conference, BMVC. British Machine Vision Association, Edinburgh, pp 899–908. https://doi.org/10.5244/c.20.92
15. Haider F, Campbell N, Luz S (2016) Active speaker detection in human machine multiparty dialogue using visual prosody information. In: Proc. IEEE Global conference on signal and information processing, GlobalSIP. IEEE, Washington, pp 1207–1211. https://doi.org/10.1109/globalsip.2016.7906033
16. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143(1):29–36. https://doi.org/10.1148/radiology.143.1.7063747
17. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proc. IEEE Conference on computer vision and pattern recognition, CVPR. IEEE, Las Vegas, pp 770–778. https://doi.org/10.1109/cvpr.2016.90
18. He W, Motlicek P, Odobez JM (2018) Deep neural networks for multiple speaker detection and localization. In: Proc. IEEE International conference on robotics and automation, ICRA, Brisbane, pp 74–79. https://doi.org/10.1109/icra.2018.8461267
19. Hong X, Yao H, Wan Y, Chen R (2006) A PCA based visual DCT feature extraction method for lip-reading. In: Proc. IEEE International conference on intelligent information hiding and multimedia signal processing, IIH-MSP, Pasadena, pp 321–326. https://doi.org/10.1109/iih-msp.2006.265008
20. Hruz M, Zajic Z (2017) Convolutional neural network for speaker change detection in telephone speaker diarization system. In: Proc. IEEE International conference on acoustics, speech and signal processing, ICASSP, New Orleans, pp 4945–4949. https://doi.org/10.1109/icassp.2017.7953097
21. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Bengio Y, LeCun Y (eds) 3rd International conference on learning representations, ICLR. San Diego, USA. http://arxiv.org/abs/1412.6980
22. Korshunov P, Halstead M, Castan D, Graciarena M, McLaren M, Burns B, Lawson A, Marcel S (2019) Tampered speaker inconsistency detection with phonetically aware audio-visual features. In: Proc. International conference on machine learning, ICML. Long Beach, USA
23. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, NIPS, Lake Tahoe, pp 1097–1105
24. Le N, Odobez JM (2016) Learning multimodal temporal representation for dubbing detection in broadcast media. In: Proc. ACM on multimedia conference, MM. ACM Press, Amsterdam, pp 202–206. https://doi.org/10.1145/2964284.2967211
25. Li S, Zou C, Li Y, Zhao X, Gao Y (2020) Attention-based multi-modal fusion network for semantic scene completion. In: AAAI
26. Madrigal F, Lerasle F, Pibre L, Ferrané I (2021) Audio-video detection of the active speaker in meetings. In: IEEE International conference on pattern recognition, ICPR, Milan, pp 2536–2543
27. Miró XA, Bozonnet S, Evans NWD, Fredouille C, Friedland G, Vinyals O (2012) Speaker diarization: a review of recent research. IEEE Transactions on Audio, Speech, and Language Processing TASLP 20(2):356–370. https://doi.org/10.1109/TASL.2011.2125954
28. Nagrani A, Chung JS, Xie W, Zisserman A (2020) Voxceleb: large-scale speaker verification in the wild. Comput Speech Lang 60:101027. https://doi.org/10.1016/j.csl.2019.101027

29. Patino J, Delgado H, Evans N (2018) The EURECOM submission to the first DIHARD challenge. In: Proc. Conference of the international speech communication association, Interspeech. ISCA, Hyderabad, pp 2813–2817. https://doi.org/10.21437/interspeech.2018-2172
30. Petridis S, Shen J, Cetin D, Pantic M (2018) Visual-only recognition of normal, whispered and silent speech. In: Proc. IEEE International conference on acoustics, speech and signal processing, ICASSP, Calgary, pp 6219–6223. https://doi.org/10.1109/icassp.2018.8461596
31. Ren J, Hu Y, Tai YW, Wang C, Xu L, Sun W, Yan Q (2016) Look, listen and learn — a multimodal lstm for speaker identification. In: Proc. Thirtieth AAAI conference on artificial intelligence, Phoenix, pp 3581–3587. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12386
32. Roth J, Chaudhuri S, Klejch O, Marvin R, Gallagher A, Kaver L, Ramaswamy S, Stopczynski A, Schmid C, Xi Z, Pantofaru C (2019) Supplementary material: AVA-ActiveSpeaker: an audio-visual dataset for active speaker detection. In: 2019 IEEE/CVF International conference on computer vision workshop, ICCVW, Seoul, pp 3718–3722. https://doi.org/10.1109/iccvw.2019.00460
33. Sarkar A, Dasgupta S, Naskar SK, Bandyopadhyay S (2018) Says who? deep learning models for joint speech recognition, segmentation and diarization. In: Proc. IEEE International conference on acoustics, speech and signal processing, ICASSP, Calgary, pp 5229–5233. https://doi.org/10.1109/icassp.2018.8462375
34. Stefanov K, Beskow J, Salvi G (2017) Vision-based active speaker detection in multiparty interaction. In: International workshop on grounding language understanding, GLU. Stockholm. https://doi.org/10.21437/GLU.2017-10, https://www.isca-speech.org/archive/pdfs/glu_2017/stefanov17_glu.pdf
35. Tao F, Busso C (2019) End-to-end audiovisual speech activity detection with bimodal recurrent neural models. Speech Comm 113:25–35. https://doi.org/10.1016/j.specom.2019.07.003
36. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: Proc. IEEE International conference on computer vision, ICCV, Santiago, pp 4489–4497. https://doi.org/10.1109/iccv.2015.510
37. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proc. IEEE Conference on computer vision and pattern recognition, CVPR, Salt Lake City, pp 1874–1883. https://doi.org/10.1109/cvpr.2018.00675
38. Vestman V, Gowda D, Sahidullah M, Alku P, Kinnunen T (2018) Speaker recognition from whispered speech: a tutorial survey and an application of time-varying linear prediction. Speech Comm 99:62–79. https://doi.org/10.1016/j.specom.2018.02.009
39. Wang Q, Downey C, Wan L, Mansfield PA, Moreno IL (2018) Speaker diarization with LSTM. In: Proc. IEEE International conference on acoustics, speech and signal processing, ICASSP, Calgary, pp 5239–5243. https://doi.org/10.1109/icassp.2018.8462628
40. Wu JD, Tsai YJ (2011) Speaker identification system using empirical mode decomposition and an artificial neural network. Expert Syst Appl 38(5):6112–6117. https://doi.org/10.1016/j.eswa.2010.11.013
41. Xie W, Nagrani A, Chung JS, Zisserman A (2019) Utterance-level aggregation for speaker recognition in the wild. In: Proc. IEEE International conference on acoustics, speech and signal processing, ICASSP. IEEE, Brighton, pp 5791–5795. https://doi.org/10.1109/icassp.2019.8683120
42. Yasir M, Nababan MN, Laia Y, Purba W, Gea A et al (2019) Web-based automation speech-to-text application using audio recording for meeting speech. In: Journal of physics: conference series, vol 1230, p 012081. IOP Publishing. https://doi.org/10.1088/1742-6596/1230/1/012081
43. Zhong Y, Arandjelović R, Zisserman A (2018) Ghostvlad for set-based face recognition. In: Proc. Asian conference on computer vision, ACCV, pp 35–50. Springer, Perth. https://doi.org/10.1007/978-3-030-20890-5_3
44. Zhou Z, Zhao G, Hong X, Pietikäinen M (2014) A review of recent advances in visual speech decoding. Image Vis Comput 32(9):590–605. https://doi.org/10.1016/j.imavis.2014.06.004

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.