Investigating the Role of Speaker Counter in Handling Overlapping Speeches in Speaker Diarization Systems

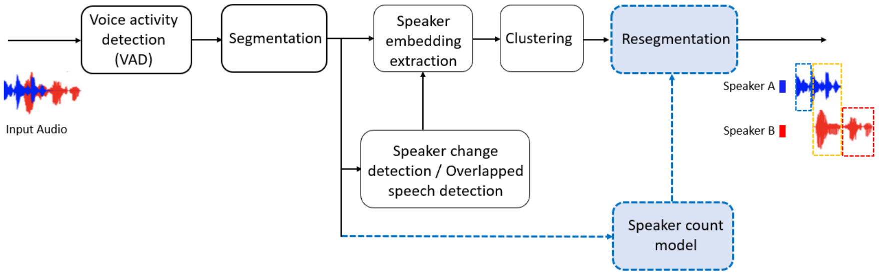
Thanh Thi-Hien Duong[1], Phi-Le Nguyen[2], Hong-Son Nguyen[3], and Ngoc Q. K. Duong[4]

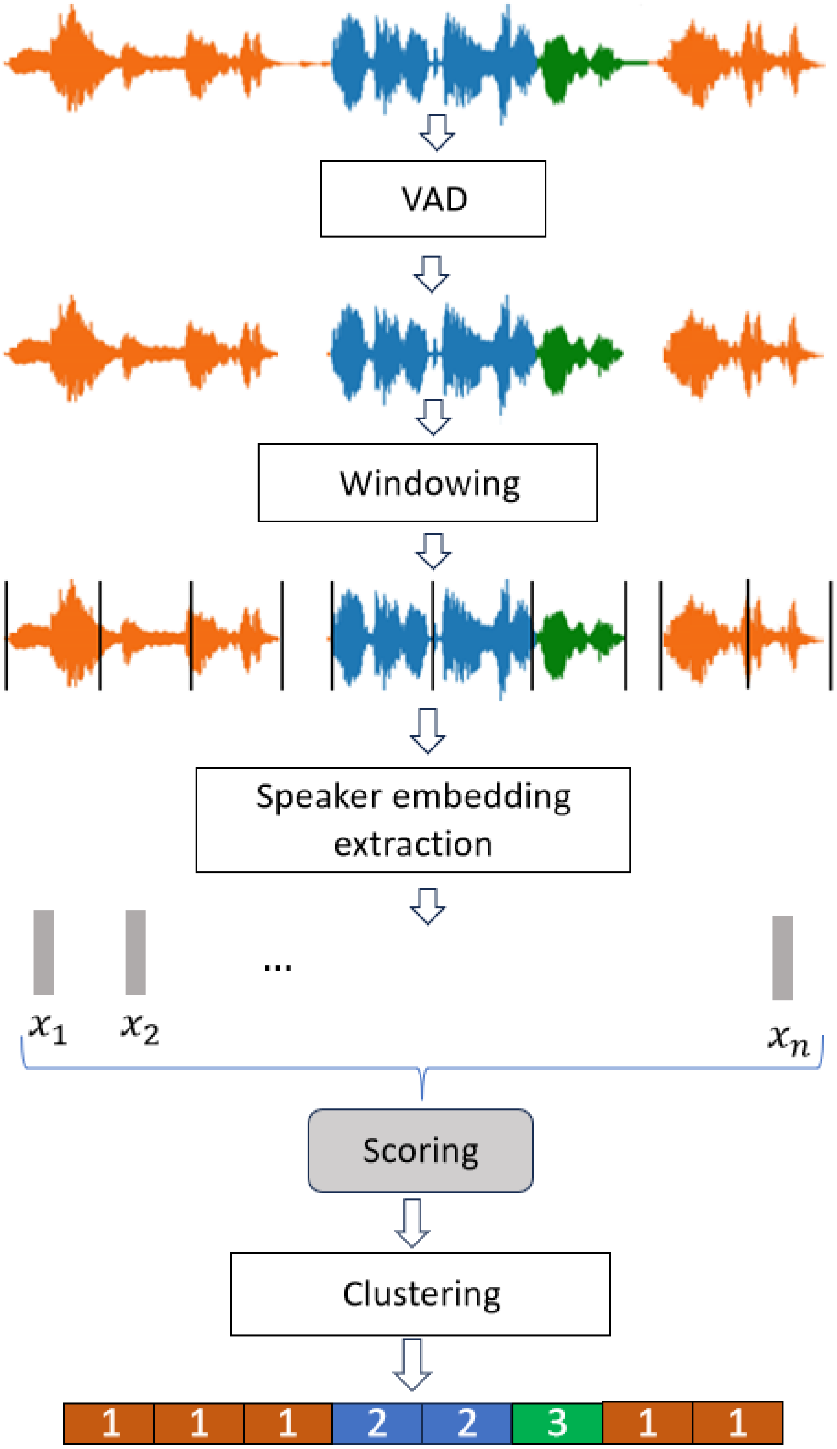
[1]Hanoi University of Mining and Geology
[2]University of Science and Technology of Hanoi
[3]Aimenext Join Stock Company
[4]Lacroix Impulse

August 17, 2023

Abstract

In real-life conversations, meetings, or debates, there are often situations where many people speak at the same time, leading to overlapping speech segments. Such overlapping speech is an extremely challenging problem for the speaker diarization task. The widely used clustering-based diarization approaches perform quite poorly under such situations due to their limited capabilities in handling overlapping speeches. This paper investigates a speaker diarization framework in which a new building block, called speaker count, is integrated. Such speaker counter predicts the number of active speakers in each analyzing audio window, then its output is used in the conventional re-segmentation step of the diarization pipelines in order to better label the active speakers in each considered segment. We also investigate the effect of the analyzing audio window size on diarization performance by theoretical analysis. We claim that the speaker count block ensures a lower diarization error rate when the analyzing window size is small enough. Experiment results obtained from two state-of-the-art diarization systems with different settings on two benchmark datasets, AMI Headset mix and DIHARD III, confirmed the effectiveness of the proposed approach.


Voice activity detection (VAD)
Segmentation
Speaker embedding extraction
Clustering
Resegmentation
Input Audio
Speaker change detection / Overlapped speech detection
Speaker count model
Speaker A
Speaker B


VAD
Windowing
Speaker embedding extraction
...
x_1
x_2
x_n
Scoring
Clustering
1
1
1
2
2
3
1
1

Hosted file

bibliography.bib available at https://authorea.com/users/653596/articles/660488-investigating-the-role-of-speaker-counter-in-handling-overlapping-speeches-in-speaker-diarization-systems

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ORIGINAL ARTICLE

Investigating the Role of Speaker Counter in Handling Overlapping Speeches in Speaker Diarization Systems

Thanh Thi-Hien Duong[1] | **Phi-Le Nguyen**[2] | **Hong-Son Nguyen**[3] | **Ngoc Q. K. Duong**[4]

[1]Department of Information Technology, Hanoi University of Mining and Geology, Vietnam

[2]School of Information Technology, Hanoi University of Science and Technology, Vietnam

[3]Aimenext Join Stock Company, Vietnam

[4]Lacroix Impulse, France

Correspondence

Thanh Thi-Hien Duong, Department of Information Technology, Hanoi University of Mining and Geology, No.18 Vien Street, Duc Thang Ward, Bac Tu Liem District, Hanoi 100000, Vietnam.
Email: duongthihienthanh@humg.edu.vn

Abstract

In real-life conversations, meetings, or debates, there are often situations where many people speak at the same time, leading to overlapping speech segments. Such overlapping speech is an extremely challenging problem for the speaker diarization task. The widely used clustering-based diarization approaches perform quite poorly under such situations due to their limited capabilities in handling overlapping speeches. This paper investigates a speaker diarization framework in which a new building block, called speaker count, is integrated. Such speaker counter predicts the number of active speakers in each analyzing audio window, then its output is used in the conventional re-segmentation step of the diarization pipelines in order to better label the active speakers in each considered segment. We also investigate the effect of the analyzing audio window size on diarization performance by theoretical analysis. We claim that the speaker count block ensures a lower diarization error rate when the analyzing window size is small enough. Experiment results obtained from two state-of-the-art diarization systems with different settings on two benchmark datasets, AMI Headset mix and DIHARD III, confirmed the effectiveness of the proposed approach.

KEYWORDS

speaker diarization, speaker counter, speaker embedding, resegmentation

1 | INTRODUCTION

In daily conversations, we often encounter situations where many people talk at the same time. These situations may be very short for interruptions or responses such as "sorry", "yeah", "uh-huh", or longer for arguments in debates or meetings. These overlapped speech cases challenge speech processing systems such as automatic speech recognition, speech separation, speaker diarization, etc. As an example in speech recognition, when knowing that multiple individuals speak at the same time, a source separation algorithm could be called at the front end to separate a target speech from the mixture before passing it to a recognition back-end[1,2].

This study deals with the task of speaker diarization in the multi-speaker audio recording situation, which aims to label speech timestamps with classes corresponding to speaker identity and answer the question "Who spoke when?"[3,4]. Speaker diarization is an essential component in speech processing systems such as *e.g.,* information retrieval, broadcast[5], telephonic conversation analysis[6], meeting analysis[7,8], and speech recognition[9].

Traditional diarization systems usually start by segmenting the input audio stream into uniform speech segments with the support of voice activity detection (VAD), then extracting the speaker embeddings from those fixed length segments, and finally performing speaker clustering on such extracted embeddings[10]. Such systems usually assign labels only to the person most likely to speak and omit others in overlapping segments. As a result, determining who is speaking in real-world situations with the occurrence of highly overlapping speeches is a difficult task even for the best-performing speaker diarization systems[11,12,13].

Recently, some systems use additional blocks such as speaker change detection and overlap detection to help the clustering algorithm[14,15]. It is noteworthy that there have been several studies on overlapping speech detection and its application in speaker detection. For example, Boakye *et al.* use an HMM-based segmenter to detect overlapping segments[3]. Huijbregts *et al.*

introduces a loudspeaker model based on GMM for overlap detection and investigates the system 'twice' to detect overlap first, and then uses it to refine tune speaker samples and perform tasks[16].

At the era of deep learning with more and more releases of annotated data, new end-to-end deep neural network (DNN)-based diarization model has emerged. Among such works can be mentioned the speaker recording system based on the area recommendation network (RPNSD)[17], in which the DNN simultaneously generates overlapping speech segmentation suggestions and computes the possibility of embedding people in their talk. Compared to standard logging methods, RPNSD provides a shorter pipeline and can handle overlapping speeches. In[18], authors present a novel speech separation framework that combines end-to-end speaker diarization and a convolutional time-domain speech separation network. This method obtains better diarization and separation performance compared to the baselines for both fixed and flexible numbers of speakers. Besides, some other notable approaches such as unbounded interleaved-state recurrent neural network (UIS-RNN) consist of supervised neural models based on speaker embeddings[19], discriminative neural clustering (DNC)[20], deep learning speech separation guided diarization based approaches[21,22] and target-speaker voice activity detection[23]. Some of these approaches have been shown to be particularly effective and ranked highly in the recent Third DIHARD Challenge[24].

While significant progress has been made on speaker diarization task so far, due to the complex acoustic scenes, a large amount of speech overlapped, and the lack of complete labeled data, speaker diarization still faces great challenges[25]. One of them is determining the exact number of speakers in each considered audio segment. Therefore, in this paper, we investigate the use of a new building block named speaker count to independently predict the number of active speakers in each audio segment. This helps the clustering algorithm to assign enough speakers in each audio segment and thus potentially offer better diarization performance. We present an extension of our previous work[26], to investigate the role of the speaker counter in handling overlapping speeches in speaker diarization systems. Our contributions are summarized as follows:

- We introduce a speaker diarization framework in which the speaker count model is integrated as a new building block. This block can be developed independently of other processing blocks, making it flexible to be integrated into any existing system.
- We provide some theoretical analysis and investigate the effect of analyzing window size to prove the potential benefit of the proposed speaker count block in terms of the diarization error rate (DER).
- We perform experiments on two benchmark datasets (AMI Headset mix and DIHARD III) with various analyzing window sizes, where the oracle speaker count block is integrated into two state-of-the-art speaker diarization systems. Diarization results confirm the potential benefit gained by the proposed block.

For the rest of the paper, we first introduce the proposed speaker diarization workflow and two baseline systems in Section 2. Then, section 3 provides theoretical analysis to prove the potential benefit obtained by the speaker count block. Experiment results and discussion are presented in Section 4. Finally, the conclusion is shown in Section 5.

2 | SPEAKER DIARIZATION APPROACH

2.1 | Proposed speaker count integrated workflow

Clustering-based speaker diarization systems often consist of three main processing blocks arranged in a pipeline structure: voice activity detection (VAD), speaker embedding extraction, and speaker clustering[9,27]. The VAD[28] is used for detecting portions of the audio signal containing voice only by removing non-voice segments such as background noise, music, silence, etc. The speaker embedding extraction block aims to capture the speaker's discriminative characteristics in a speech segment and encapsulates them into a speaker embedding. The clustering step is crucial to group the embeddings that belong to the same speaker and label speaker identities in each segment. Example of a general pipeline of the speaker diarization is described in Fig. 1-black boxes part, and more detail the output of each step with three speakers labeled as 1, 2, 3 is shown in Fig. 2.

State-of-the-art approaches[12,15] use additional blocks such as overlapped speech detection[3], speaker change detection[29], or re-segmentation[30] for a better diarization performance.

In case overlapped detection is used, the system consists of three major modules: voice activity detection, overlap detection, and speaker recognition. In the first phase, a silence detection model is used to identify silence durations and remove them. The non-silence speech then is divided into small segments by using a sliding window with a size of α, and the overlap ratio of β, where α and β are tunable parameters. As example in some implementations, α and β are set to 1 second and 50%, respectively. After segmentation, on the one hand, every segment is inputted into the overlap detection module to verify whether it is an

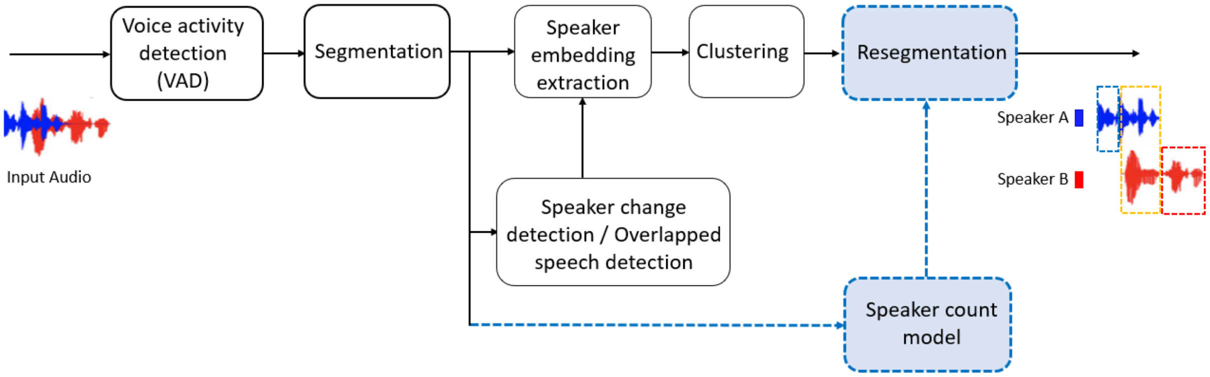

Input Audio
Voice activity detection (VAD)
Segmentation
Speaker embedding extraction
Clustering
Resegmentation
Speaker change detection / Overlapped speech detection
Speaker count model
Speaker A
Speaker B

FIGURE 1 General workflow of the speaker counter integrated diarization system.

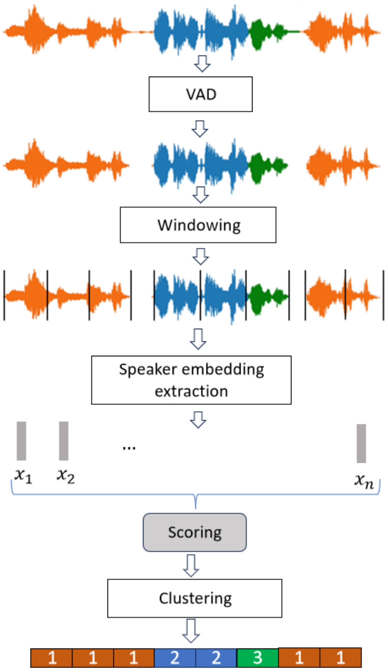


FIGURE 2 A typical speaker diarization pipeline.

overlapped speech or not. On the other hand, the segments are passed through the speaker recognition module to identify the speaker. The speaker recognition module is comprised of two steps. The first step, namely speaker embedding, is to extract speech features. Such feature vectors are then passed to the second step, named clustering, which performs a clustering algorithm to categorize the speakers into groups. The output of the speaker recognition module is the probability for a segment to be the voices of the speakers. Finally, the results of the overlap detection module and the speaker recognition module are concatenated to generate diarization result. Specifically, for a non-overlap segment, the speaker is the one determined by the speaker recognition with the highest probability; while for an overlapped segment, beside the primary speaker, we also assign a secondary speaker whose probability is the second highest.

Hereafter we present in more detail the re-segmentation step when the speaker count block is integrated into the proposed model.

2.2 | Resegmentation with speaker counter

Overlap detection can be seen as a binary version of the speaker count estimation problem, where the number of speakers equals one in the non overlap cases or more than one corresponds to the overlap cases. Therefore we apply a speakers count estimation model for the overlap detection problem.

In the considered approach, the speaker count block is integrated into diarization workflow to estimate the number of active speakers at each analysis window. Then, instead of just labeling one or two speakers with the highest scores resulted from the clustering block (as in case the conventional overlapped detection is used), resegmentation block will label speaker identities according to the exact speaker numbers reported from the speaker count block. The blue square-dot boxes in Fig. 1 indicate such steps considered in the paper.

2.3 | Baseline 1: Pyannote

Among a large number of approaches have been proposed recently, in this paper we choose two widely used ones as baseline to validate the effectiveness of the proposed approach. The first baseline is Pyannote library[15], which has been actively updated by the authors from its first creation. The second baseline is UIS-RNN[19], one of the reference for deep learning model. In both baselines the considered speaker count can be easily integrated as a new building block.

Pyannote has been developped since 2020 and maintained as a PyTorch library. Pyannote.audio is open-source toolkit written in Python and is part of pyannote for speaker diarization [‡]. It integrates a set of state-of-the-art neural building blocks that can be either trained separately or combined and jointly optimized to build speaker diarization pipelines: the end-to-end neural voice activity detection (VAD)[31], the speaker change detection[29], the overlapped speech detection[30], the speaker embeddings[32], and the Bayesian model-based clustering[15]. In one of the released model, the first three blocks were all trained on the DIHARD[33] and the AMI[34] datasets, while the speaker embedding was trained on the VoxCeleb1[35] and the VoxCeleb2[36] datasets. It should be noted that, Pyannote explores a metric learning approach to train speaker embeddings that are directly optimized for a predefined (usually is cosine) distance. This is different with earlier diarization systems[37,38] which use *x-vectors* extracted from a fixed-length sliding window as input to the clustering step. The Pyannote authors claim that their library does not require techniques like probabilistic linear discriminant analysis (PLDA) before clustering.

It is worth noting that, while each building block can be trained separately, Pyannote offers possiblity to combine them for end-to-end training in which the hyper-parameters are optimized jointly to minimize the diarization error rate. This joint optimization process has been shown to offer better results than the late combination of multiple building blocks that were tuned independently from each other[14].

2.4 | Baseline 2: UIS-RNN

It has been shown that DNN-based embeddings (a.k.a. *d-vectors*[39]) are more poweful than the the conventional handcrafted features (*i-vector*[40]) for the task[41,42]. This can be explained by the fact that DNN can be trained with large-scale datasets, which offers the resulting embedding robust against varying speaker identities and acoustic conditions. Thus, in the second baseline, we investigate the use of powerful DNN architectures for both three major steps: VAD , speaker embedding extraction[42] and speaker clustering[19]. The baseline is based on a fully supervised speaker diarization approach, named unbounded interleaved-state recurrent neural networks (UIS-RNN)[19]. However, differently from the original paper, we made some modifications in our implementation to improve the result. First, we use the pre-trained model developed for the Google WebRTC project[§] for VAD block. This VAD model has been known as one of the fastest VAD for real-time processing [¶]. Second, speaker embeddings are extracted from the state-of-the-art speaker recognition deep network[42] trained on the VoxCeleb1[35] and the VoxCeleb2[36] datasets. This network modifies ResNet in a fully convolutional way to encode 2D spectrograms of audio signals, followed by a NetVLAD/GhostVLAD layer[43] for feature aggregation along the temporal axis. It produces a fixed-length 512-dimensional output d-vector for each input audio segment. The implementation is provided by the author [#].

[‡] https://github.com/pyannote/pyannote-audio
[§] https://webrtc.org/
[¶] https://github.com/wiseman/py-webrtcvad
[#] https://github.com/WeidiXie/VGG-Speaker-Recognition

In this considered UIS-RNN approach, each speaker is modeled by an instance of RNN with shared parameters. As the total number of speakers in an audio recording is generally unknown, an unbounded number of RNN instances can be generated. It is claimed that Since all components of the system can be learned in a supervised manner, it is preferred over unsupervised systems in scenarios where training data with high quality time-stamped speaker labels are available. Furthermore, UIS-RNN can better handle complexities in speaker diarization since it automatically learns both the speaker changes and the number of speakers within each utterance via a Bayesian non-parametric process. In our experiment, we use the UIS-RNN implementation provided by the Google AI Blog[‖]. We would like to highlight that in our second baseline implementation, DNN architectures for the VAD and the speaker embedding could be considered to be more advanced than the ones used in the original UIS-RNN paper and closer to the recent advance in deep learning[19]. Thus, we argue that this can be served as a strong baseline to evaluate the potential of our proposed approach.

3 | THEORETICAL ANALYSIS

We analyze the diarization error rate (DER) obtained by the proposed approach using speaker count and comment on its gain or loss in different cases compared to the basic setup where only one speaker is considered to be active at each analysis window.

Let I be an input audio file with a duration of L seconds, according to[44], the DER of I is defined by:

$$DER_{spkcount} = \frac{\sum_{i=1}^{P} max[N_{ref}(i), N_{cnt}(i)] - N_{correct}}{\sum_{i=1}^{P} N_{ref}(i)}, \tag{1}$$

where $N_{ref}(i)$ and $N_{cnt}(i)$ are the ground truth and the detected number of speakers at i-th speaker count window, respectively, $P = \frac{L}{wlen}$ is the total number of window segment with window length $wlen$, and $i \leq P$.

Note that in this formula, the window size $wlen$ is not written as it appears in both nominator and denominator, and the segment border effect can be neglected with the assumption that $wlen$ is small compared to the audio segments with the same speaker activities determined by the first diarization block. Assuming further that the diarization block outputs the perfect appearance probability order of all speakers, we have $N_{correct} = min[N_{ref}(i), N_{cnt}(i)]$. When considering only one active speaker at every analysis window (*i.e.,* the one with highest probability outputted by the diarization system), the corresponding DER, denoted as DER_1, is defined as

$$DER_1 = \frac{\sum_{i=1}^{P} max[N_{ref}(i), N_{cnt}(i)] - 1}{\sum_{i=1}^{P} N_{ref}(i)} \tag{2}$$

We would like to see how much gain or loss we can obtain when using the speaker counter as an additional processing step from $DER_{spkcount} - DER_1$. Two separate cases can be seen as follows.

(a) $N_{cnt}(i) \leq N_{ref}(i)\ \forall i$. The DER in eq (1) can be written as

$$\begin{aligned} DER_{spkcount} &= \frac{\sum_{i=1}^{P} N_{ref}(i) - N_{cnt}}{\sum_{i=1}^{P} N_{ref}(i)} \\ &= DER_1 - \frac{\sum_{i=1}^{P} N_{cnt}(i) - 1}{\sum_{i=1}^{P} N_{ref}(i)} \end{aligned} \tag{3}$$

Note that thanks to the VAD block, non-speech segments are already cut before the diarization, and thus the speaker count system should be designed to detect at least one speaker at each window. Thus, with $N_{cnt}(i) \geq 1$ in all windows, the DER gain compared to DER_1 is $DER_{gain}^{(a)} = \frac{\sum_{i=1}^{P} N_{cnt}(i) - 1}{\sum_{i=1}^{P} N_{ref}(i)}$. With the ideal speaker count system where $N_{cnt}(i) = N_{ref}(i)\ \forall i$, the maximum gain in terms of DER we could expect is

$$DER_{gain-max} = \frac{\sum_{i=1}^{P} N_{ref}(i) - 1}{\sum_{i=1}^{P} N_{ref}(i)}. \tag{4}$$

This maximum gain logically implies $DER_{spkcount} = 0$. The $DER_{gain-max}$ is higher when the number of windows P is larger, meaning that with a smaller window length for speaker count detection, one can expect better diarization performance. This analysis is in line with our experiment results shown in Table 2 and Table 3.

[‖] https://github.com/google/uis-rnn

(b) $N_{cnt}(i) > N_{ref}(i)\ \forall i$. This is the case where the speaker count system counts more speakers than the ground-truth. The DER in eq (1) is then written as

$$\begin{aligned} DER_{spkcount} &= \frac{\sum_{i=1}^{P} N_{cnt}(i) - N_{ref}}{\sum_{i=1}^{P} N_{ref}(i)} \\ &= DER_1 - \frac{\sum_{i=1}^{P} 2 * N_{ref}(i) - N_{cnt}(i) - 1}{\sum_{i=1}^{P} N_{ref}(i)} \end{aligned} \tag{5}$$

As can be seen, with a very bad speaker count system that outputs $N_{cnt} > 2 * N_{ref}(i) - 1$ for most windows, the overall DER result will be worse as $DER_{spkcount} > DER_1$. However, this case might rarely happen in practice. On the other hand, when $N_{ref} < N_{cnt} \leq 2 * N_{ref}(i) - 1$ for most windows, the use of speaker count system offers the lower DER of $DER^{(b)}_{gain} = \frac{\sum_{i=1}^{P} 2*N_{ref}(i) - N_{cnt}(i) - 1}{\sum_{i=1}^{P} N_{ref}(i)}$. This gain is upper-bounded by $DER_{gain-max}$ since in this case $2 * N_{ref}(i) - N_{cnt}(i) - 1 \leq N_{ref}(i) - 1$, and the smaller window length, the higher DER gain could be expected.

As conclusion, the analysis shows that the benefit of the overlap speech detection is greater when (a) the input audio contains more segments with overlap and (b) the analyzing window size is smaller. These intuitions hold for general situation with more than two active speakers at each segment.

4 | EXPERIMENTAL

4.1 | Datasets

We evaluate the speaker diarization performance obtained by the baselines and the proposed approach on two benchmark datasets: AMI Headset mix[45] and DIHARD III[46] as below:

- AMI Headset mix dataset[45] is a widely used dataset for speaker diarization over the last decade. This dataset consists of 98 hours of meeting recordings from 180 speakers in total. The recordings were in English and recorded in three rooms with different characteristics. Distribution according to the number of simultaneous speakers, the AMI dataset includes 81% of the total voiced periods as single-speaker, 15% of the time as two-speaker, and the remaining 4% of the time as three or more speakers. This show that the two-speaker case occupies approximately 75% of the overlap segments. The dataset was split into 70% for training (68.6 hours), 15% for validation (14.7 hours), and 15% for evaluation (14.7 hours).
- DIHARD III dataset[46] is a recent benchmark used in the third DIHARD speech diarization challenge**. This dataset contains single-channel wide-band audio recorded from 11 different environments. Such recordings vary from very clean ones (*i.e.,* near-field recordings of reading audiobooks) to noisy, far-field ones. The dataset was split into about 40.7% for training (27.92 hours), 10.2% for validation (6.98 hours), and 49.1% for evaluation (33.65 hours). The statistics for both datasets are summarized in Table 1.

TABLE 1 Datasets statistics

Dataset	Train (hours)	Validation	Test
AMI Headset mix	68.6h	14.7h	14.7h
DIHARD III	27.92h	6.98h	33.65h

4.2 | Performance evaluation

DER: We use the pyannote.metrics toolkit[44] to evaluate the speaker diarization systems in terms of diarization error rate (DER). DER expresses the portion of the recording that is labeled incorrectly, including three possible types of errors: false alarm,

** https://dihardchallenge.github.io/dihard3/

missed detection, and speaker confusion, defined as follows:

$$DER = \frac{\text{false alarm} + \text{miss detection} + \text{confusion}}{total}, \quad (6)$$

where *falsealarm* denotes the duration of non-speech incorrectly classified as speech, *missdetection* denotes the duration of speech incorrectly classified as non-speech, *confusion* denotes the duration of speaker incorrectly classified, and *total* is the total duration of reference speeches. DER is expressed as a percentage, and the lower DER the better.

JER: We also use another metric developed by the DIHARD II competition[33], namely the Jaccard Error Rate (JER). JER first calculates the sum of false alarm and miss detection per speaker in the audio, then averaged it to compute JER. Specifically, JER is computed as follows:

$$JER = \frac{1}{N} \sum_{i}^{N_{ref}} \frac{FA_i + MISS_i}{TOTAL_i}, \quad (7)$$

where with the *i*-th speaker, *FA* denotes the total system speaker time not attributed to the reference speaker, *MISS* is the total reference speaker time not attributed to the system speaker, and *TOTAL* denotes the duration of the union of reference and system speaker segments. Similar to DER, JER is also expressed as a percentage, and the lower JER the better.

B3-F1: The third metrics for evaluating diarization system are B3-cubed precision, recall, and F1 score[47]. The B-cubed precision for a frame assigned to speaker A in the reference and speaker B in the system is the proportion of frames assigned to speaker B that is also assigned to speaker A. The same, B-cubed recall for a frame is the proportion of all frames assigned to speaker A that are also assigned to speaker B. Then, the overall precision and recall are the average of the frame-level precision and recall, and the overall F-1 is their harmonic average. Different from DER and JER, B3-F1 score the higher the better.

4.3 | Implementation details

Pyannote baseline: we use implementation codes, configuration files, and pre-trained models for all component blocks provided by the authors [††], due to they were already trained and validated on the AMI and DIHARD datasets. Speaker embeddings are extracted every 1-second sliding window with 512 dimensions for clustering. After that, we evaluate the performance of Pyannote baseline on the test set of the two considered datasets summarized in Table 1.

UIS-RNN baseline: With the second baseline model, we further augment the VoxCeleb1[35] and the VoxCeleb2[36] datasets with approximately 34 hours of Japanese speeches collected from Youtube and approximately 1,000 hours of English speeches from ST Chinese Mandarin Corpus[‡‡] for training the *d-vectors* speaker embedding. The spectrogram size is set varying corresponding from 2-second to 6-second temporal segments, which are random extracted from each utterance. Spectrograms are calculated by the short-term Fourier transform (STFT) with a sliding Hamming window of size 25 ms, a window shift of 10 ms, and 256 frequency bins. Then they are standardized by subtracting the average value and then dividing by the standard deviation of all frequency components in a single time step. For training the UIS-RNN clustering on the considered AMI headset mix and DIHARD III dataset, we set the sliding window for speaker embedding extraction is 1-second instead of 240 ms in the original model. The other training parameters are set the same as the original implementation. During the evaluation, speaker embeddings of dimension 512 are extracted every 1-second sliding window for clustering as the Pyannote baseline setting.

4.4 | Diarization results and discussion

In order to investigate the potential benefit of the proposed speaker count integration approach, for each baseline method, we compare the speaker diarization results of four different settings as follows:

[††] https://github.com/pyannote/pyannote-audio

[‡‡] http://openslr.org/38

TABLE 2 Speaker diarization results obtained by the Pyannote-based methods on AMI Headset mix and DIHARD III datasets. The original model is trained and provided by the authors.

		AMI Headset mix			DIHARD III		
Pyannote-based methods	**Window size (seconds)**	DER%	JER%	B3-F1	DER%	JER%	B3-F1
Original model	1	32,09	99.15	0.59	21.58	40.28	0.74
One speaker assignment	1	29.36	59.36	0.63	25.33	44.68	0.72
Oracle overlap detection	1	34.28	59.68	0.56	28.41	41.89	0.71
	0.8	32.46	58.09	0.57	27.86	41.61	0.71
	0.6	30.12	57.58	0.58	26.92	41.41	0.73
	0.4	28.85	57.64	0.6	25.81	41.21	0.75
	0.2	27.83	56.18	0.63	24.44	40.94	0.77
	Oracle speaker change	25.6	55.98	0.64	21.58	40.28	0.74
Oracle speaker count	1	29.13	58.44	0.61	25.67	39.07	0.71
	0.8	26.24	55.98	0.64	24.49	38.41	0.71
	0.6	25.33	54.82	0.65	23.27	37.00	0.72
	0.4	23.75	53.91	0.65	21.51	35.14	0.74
	0.2	21.87	52.37	0.65	19.31	33.18	0.77
	Oracle speaker change	20.62	51.05	0.65	16.73	30.95	0.80

TABLE 3 Speaker diarization results were obtained by the UIS-RNN-based methods on AMI Headset mix and DIHARD III datasets. The original UIS-RNN model implementation is provided by the Google AI Blog but trained by ourselves on the considered datasets.

		AMI Headset mix			DIHARD III		
UIS-RNN-based methods	**Window size (seconds)**	DER%	JER%	B3-F1	DER%	JER%	B3-F1
Original model	1	30.87	59.06	0.61	27.88	46.55	0.59
One speaker assignment	1	28.52	64.91	0.59	27.91	46.57	0.59
Oracle overlap detection	1	30.96	58.69	0.56	28.08	45.03	0.58
	0.8	30.7	58.09	0.57	27.57	44.89	0.58
	0.6	28.7	57.33	0.58	26.93	44.48	0.58
	0.4	27.64	56.92	0.6	26.14	44.19	0.59
	0.2	26.18	55.74	0.63	25.1	43.67	0.60
	Oracle speaker change	24.7	54.8	0.64	23.64	43.08	0.61
Oracle speaker count	1	28.41	55.44	0.65	28.32	41.85	0.58
	0.8	27.32	53.48	0.67	27.14	41.30	0.58
	0.6	24.27	51.78	0.67	25.94	40.09	0.59
	0.4	22.6	50.36	0.67	24.21	48.35	0.61
	0.2	21.03	49.12	0.67	21.97	46.64	0.63
	Oracle speaker change	18.74	46.32	0.7	19.28	0.70	0.67

- Original model: This is the baseline speaker diarization model, in which speaker embeddings and clustering are executed for every 1-second segment of the input audio file. With Pyannote model, all processing blocks use the pre-trained model published by the authors. With UIS-RNN model, we use pre-trained VAD model provided by the authors, while speaker embedding extraction and clustering models are trained by ourselves as described in Section 4.3.
- One speaker assignment: In this setting, all processing blocks are the same as the original model, except the clustering algorithm assigns only one speaker who has the highest probability for each 1-second segment.
- Oracle overlap detection: All processing blocks in this setting are the same as the original model, except that the *oracle* overlap detection is used instead of the trained DNN model. Assuming the overlap detection for each analyzing audio window is perfect (*i.e.,* known from the ground-truth), in order to evaluate the upper-bound diarization performance with the use of overlap detection block with different analyzing window sizes, we vary the window size as 0.2 seconds, 0.4 seconds, 0.6 seconds, 0.8 seconds, and 1 second to investigate its effect on diarization performance. In windows that have more than two active speakers, we assign the two ones with the longest active duration. The best diarization performance is obtained when using oracle speaker change and the speaker activity (*i.e.,* active or inactive) boundary is perfectly determined.

- Oracle speaker count: This case allows us to investigate the potential benefit of the proposed speaker counter when it is integrated into existing base systems. As the same oracle overlap detection setup, the speaker count window is varied as 0.2 seconds, 0.4 seconds, 0.6 seconds, 0.8 seconds, and 1 second to analyze its impact and get the best performance obtained using Oracle speaker change. The number of active speakers in each window, possibly more than two, is perfectly specified given the ground-truth.

Speaker diarization results obtained by the Pyannote baseline and the UIS-RNN baseline with four different setups on AMI Headset mix and DIHARD III datasets are shown in Table 2 and Table 3, respectively.

We first compare the results of the four system setups. It can be seen that the simple one speaker assignment setting offers better results in terms of DER, the most important evaluation metric, than the original model for both the two baselines on the AMI Headset mix dataset, which contains about 19% of multiple speaker cases. This shows that clustering is still very challenging for overlapping speeches, and discloses the necessity of speaker count building block. Besides, we can see the average diarization result is better on both the AMI Headset mix and DIHARD III datasets when overlap detection is used, especially with small window sizes. This is not surprising, cause of overlap detection block helps the model can correctly assign two speakers in the overlap segments. Finally, as expected, the proposed approach integrating speaker counter gets the best speaker diarization performance in terms of DER and F-1 score on both two baselines model and two experimental datasets. This confirms the role of the integrated speaker counter in handling overlapped speeches in the diarization systems.

Considering the effect of analysis window size on diarization performance, it is first noted that when the analyzing window size is 1 second, the general diarization results in both the oracle overlap detection and the oracle speaker count settings are not better than those obtained by the original models or the one speaker assignment case. The cause is in many analyzing windows, the reality overlapping speech duration is less than 1 second. Therefore, assigning two speakers in the overlap detection case, and multiple speakers in the speaker counter case for all such 1-second long windows is less accurate. As a results observed in Table 2 and Table 3, the smaller window size gets the better diarization performance, because it allows the speaker assignment to be closer to the ground-truth. The best performance is obtained by the proposed speaker count integrated approach with the oracle speaker change decision. Specifically, DER is as low as 20.62% for the Pyannote on the AMI Headset mix dataset, 16.73% for the Pyannote on the DIHARD III dataset, 18.74% for the UIS-RNN on the AMI Headset mix dataset, and 19.28% for the UIS-RNN on the DIHARD III dataset.

5 | CONCLUSION

This paper focus on addressing the problem of efficiently handling overlapping speech in speaker diarization systems. We first introduce a new building block to independently count the number of active speakers in each audio segment aiming to better label speakers. We then provide a theoretical analysis to demonstrate the upper gain of the proposed approach obtained with a speaker counter. Finally, we perform experiments on two widely used datasets, where the proposed speaker counter is integrated into two strong diarization baselines. The experimental results of different model settings have confirmed the potential benefit of the proposed speaker count integrated approach in real-world datasets. Future work could be devoted to developing and training a real DNN-based speaker count model, *e.g.,* motivated from the CRNN approach[48], for a practical speaker diarization application. Further interesting research direction could be to jointly optimize the proposed speaker count block with other processing blocks via an end-to-end DNN-based diarization system. This can be motivated by the recent work on speaker diarization with region proposal network[49].

ACKNOWLEDGMENTS

This work has been supported by the Hanoi University of Mining and Geology under grant number T21-01, based on decision No. 97/QD-MDC, and Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 07/2020/STS01

REFERENCES

1. Barker J, Vincent E, Ma N, Christensen H, Green P. The PASCAL CHiME speech separation and recognition challenge. *Computer Speech Language.*;27(3):621-633.
2. Li Y, Zheng X, Woodland PC. Self-Supervised Learning-Based Source Separation for Meeting Data. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2023.
3. Boakye K, Trueba-Hornero B, Vinyals O, Friedland G. Overlapped speech detection for improved speaker diarization in multiparty meetings. In: 2008:4353–4356.

4. Anguera X, Bozonnet S, Evans N, Fredouille C, Friedland G, Vinyals O. Speaker Diarization: A Review of Recent Research. *IEEE Transactions on Audio, Speech, and Language Processing.* 2012;20(2):356-370.
5. Vinals I, Ortega A, Lopez J, Miguel A, Lleida E. Domain adaptation of PLDA models in broadcast diarization by means of unsupervised speaker clustering. In: 2017:2829–2833.
6. India M, Hernando J, Fonollosa JA. Language Modelling for Speaker Diarization in Telephonic Interviews. *Comput. Speech Lang..* 2023;78(C).
7. Yella SH, Bourlard H. Improved overlap speech diarization of meeting recordings using long-term conversational features. In: 2013:7746–7750.
8. Neumann Tv, Kinoshita K, Delcroix M, Araki S, Nakatani T, Haeb-Umbach R. All-neural Online Source Separation, Counting, and Diarization for Meeting Analysis. In: 2019:91-95
9. Park TJ, Kanda N, Dimitriadis D, Han KJ, Watanabe S, Narayanan S. A review of speaker diarization: Recent advances with deep learning. *Computer Speech Language.* 2022;72:101317.
10. Anguera X, Bozonnet S, Evans N, Fredouille C, Friedland G, Vinyals O. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing.* 2012;20(2):356-370.
11. al. eGS. Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. In: ISCA 2018:2808–2812.
12. Ryant N, Church K, Cieri C, et al. Second DIHARD challenge evaluation plan. In: 2019.
13. Ryant N, Singh P, Krishnamohan V, et al. The Third DIHARD Diarization Challenge. 2021.
14. Yin R, Bredin H, Barras C. Neural Speech Turn Segmentation and Affinity Propagation for Speaker Diarization. In: 2018:1393–1397
15. Bredin H, Yin R, Coria JM, et al. pyannote.audio: neural building blocks for speaker diarization. In: 2020:7124-7128.
16. Huijbregts M, Leeuwen D, Jong F. Speech overlap detection in a two-pass speaker diarization system. In: 2009.
17. Huang Z, Watanabe S, Fujita Y, et al. Speaker Diarization with Region Proposal Network. In: 2020:6514-6518
18. Maiti S, Ueda Y, Watanabe S, et al. EEND-SS: Joint End-to-End Neural Speaker Diarization and Speech Separation for Flexible Number of Speakers. 2022.
19. Zhang A, Wang Q, Zhu Z, Paisley J, Wang C. Fully supervised speaker diarization. In: 2019:6301–6305.
20. Li Q, Kreyssig FL, Zhang C, Woodland PC. Discriminative Neural Clustering for Speaker Diarisation. In: 2021:574-581
21. Fang X, Ling ZH, Sun L, et al. A Deep Analysis of Speech Separation Guided Diarization Under Realistic Conditions. In: 2021:667-671.
22. Morrone G, Cornell S, Raj D, et al. Low-Latency Speech Separation Guided Diarization for Telephone Conversations. *2022 IEEE Spoken Language Technology Workshop (SLT).* 2022:641-646.
23. Medennikov I, Korenevsky M, Prisyach T, et al. Target-Speaker Voice Activity Detection: a Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario. In: 2020.
24. Ryant N, Singh P, Krishnamohan V, et al. The Third DIHARD Diarization Challenge. In: 2021:3570–3574
25. Yu F, Zhang S, Fu Y, et al. M2Met: The Icassp 2022 Multi-Channel Multi-Party Meeting Transcription Challenge. In: 2022:6167-6171
26. Duong TTH, Nguyen PL, Nguyen HS, Nguyen DC, Phan H, Duong NQK. Speaker count: A new building block for speaker diarization. *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC).* 2021:1149-1155.
27. Serafini L, Cornell S, Morrone G, Zovato E, Brutti A, Squartini S. An experimental review of speaker diarization methods with application to two-speaker conversational telephone speech recordings. *Computer Speech Language.* 2023;82:101534. doi: https://doi.org/10.1016/j.csl.2023.101534
28. Gelly G, Gauvain JL. Optimization of RNN-Based Speech Activity Detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing.* 2018(3):646–656.
29. Yin R, Bredin H, Barras C. Speaker Change Detection in Broadcast TV Using Bidirectional Long Short-Term Memory Networks. In: ISCA 2017.
30. Bullock L, Bredin H, García-Perera LP. Overlap-aware diarization: resegmentation using neural end-to-end overlapped speech detection. In: 2020:7114-7118.
31. Lavechin M, Gill MP, Bousbib R, Bredin H, Garcia-Perera LP. End-to-end Domain-Adversarial Voice Activity Detection. In: 2020.
32. Bredin H. pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems. In: 2017; Stockholm, Sweden.
33. Ryant N, Church K, Cieri C, et al. The Second DIHARD Diarization Challenge: Dataset, Task, and Baselines. In: 2019:978–982
34. Carletta J. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation.* 2007;41(2).
35. A. Nagrani WXAZ. VoxCeleb: Large-scale Speaker Verification in the Wild. *Computer Speech Language.* 2019.
36. Chung JS, Nagrani A, Zisserman A. VoxCeleb2: Deep Speaker Recognition. In: 2018.
37. Broux PA, Desnous F, Larcher A, Petitrenaud S, Carrive J, Meignier S. S4D: Speaker Diarization Toolkit in Python. In: ISCA 2018:1368–1372
38. Povey D, Ghoshal A, Boulianne G, et al. The Kaldi Speech Recognition Toolkit. 2011. IEEE Catalog No.: CFP11SRW-USB.
39. Wan L, Wang Q, Papir A, Moreno IL. Generalized End-to-End Loss for Speaker Verification. In: 2018:4879-4883.
40. Dehak N, Kenny PJ, Dehak R, Dumouchel P, Ouellet P. Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing.* 2011;19(4):788-798.
41. Garcia-Romero D, Snyder D, Sell G, Povey D, McCree A. Speaker diarization using deep neural network embeddings. In: 2017:4930-4934.
42. W. Xie JSCAZ. Utterance-level Aggregation For Speaker Recognition In The Wild. In: 2019:5791-5795.
43. Arandjelović R, Gronat P, Torii A, Pajdla T, Sivic J. NetVLAD: CNN architecture for weakly supervised place recognition. In: 2016.
44. Bredin H. pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems. In: 2017; Stockholm, Sweden.
45. Carletta J. Unleashing the Killer Corpus: Experiences in Creating the Multi-Everything AMI Meeting Corpus. *Language Resources and Evaluation.* 2007;41(2):181–190.
46. Ryant N, Church K, Cieri C, Du J, Ganapathy S, Liberman M. Third DIHARD challenge evaluation plan. In: 2020.
47. Bagga A, Baldwin B. Algorithms for Scoring Coreference Chains. In: 1998.
48. Grumiaux PA, Kitíc S, Girin L, Guérin A. High-resolution speaker counting in reverberant rooms using CRNN with ambisonics features. In: 2020.
49. Huang Z, Watanabe S, Fujita Y, et al. Speaker Diarization with Region Proposal Network. In: 2020.

AUTHOR BIOGRAPHY

Thanh Thi-Hien Duong received a Bachelor of Science in Informatics from the Hanoi National University of Education (HNUE), Vietnam, in 2000, and a Master of Science in Computer Science from the Hanoi University of Science and Technology (HUST), Vietnam, in 2009. She obtained her doctorate in Computer Science at HUST in 2019. She is currently a senior lecturer and researcher at the Hanoi University of Mining and Geology. Her research interest concerns signal processing and machine learning, applied to audio, image, and video.

Phi-Le Nguyen received a Bachelor of Engineering and a Master of Science from the University of Tokyo in 2007 and 2010, respectively. In 2019, she obtained her doctorate in informatics from The Graduate University for Advanced Studies, National Institute of Informatics in Tokyo, Japan. Dr. Phi Le Nguyen is currently the managing director of the International Research Center for Artificial Intelligence (BKAI) and a lecturer at Vietnam's Hanoi University of Science and Technology (HUST), School of Information and Communication Technology. Her research interests include communication network architecture and applied AI. Dr. Nguyen has published her research findings in top-tier journals and conferences, including ComNet, JNCA, IEEE Sensors, ACM TOSN, IEEE IM, and IEEE ICC. Her research has been published in more than sixty conferences and publications, and she has won numerous best paper awards, including ISSNIP'14, SoICT'15, and ICT-DM'19. She has participated in prestigious conferences as a TPC member, including Globecom, ICC, PRICAI, and WCNC.

Hong-Son Nguyen graduated from University of Engineering and Technology, Vietnam National University, Hanoi , in 2019 with a Bachelor's degree in Information technoloy. I have previously held the position of Senior Developer at Harveynash and served as a Technical Leader at Aimenext. Currently, I am a senior AI Engineer with extensive experience working with image processing, Natural Language Processing (NLP), and Audio processing technologies. My journey reflects a commitment to excellence, continuous learning, and a drive to make a difference in the tech industry.

Ngoc Q. K. Duong is currently a Senior Engineer at Lacroix Impulse in France. Prior to that, he worked as a senior researcher for InterDigital and Technicolor RD center in France for ten years. He held the HDR since 2020 and he obtained the Ph.D. degree at the French National Institute for Research in Computer Science and Control (INRIA), Rennes, France in 2011. His research interest concerns statistical signal processing, multimedia analysis, and machine learning. He is a senior member of IEEE and regularly serves on the technical program committee of various international conferences. He has received several research awards, including the IEEE Signal Processing Society Young Author Best Paper Award in 2012 and the Bretagne Young Researcher Award in 2015. He is the co-author of more than 50 peer-reviewed scientific publications and the co-inventor of about 30 granted and pending patents.