Overlapped speech detection and competing speaker counting – humans vs. deep learning

Valentin Andrei, Horia Cucu, and Corneliu Burileanu, *Member, IEEE*

***Abstract*—A natural evolution of applications that use speech processing and analysis is to improve their robustness to multi-speaker environments. Humans use selective auditory attention and can easily switch focus from one source to another even when listening to a single channel recording with speakers talking simultaneously. In order to quantify human level performance for this task we have designed a perception experiment that evaluates participants' ability to accurately count multiple speakers in a single channel recording and studied the influence of listening time and of using familiar voices. The experiments use a higher number of subjects than similar reported attempts and are confirmed by some of the findings in existing literature.**

Using the conclusions from the perception experiment, a set of convolutional neural networks were trained to estimate the number of competing speakers on speech timeframes ranging from 25 milliseconds to 1 second. The same models were also instructed to tag overlapped speech and we observed F-Score values up to 0.91. For both tasks, the proposed methods lead to lower error than existing approaches and require smaller timeframes. Compared with human listeners, the neural networks can count speakers more accurately by analyzing a considerably shorter recording.

***Index Terms*— selective auditory attention, overlapped speech, speaker counting, convolutional neural networks**

I. Introduction

SPEAKER diarization continues to be a difficult goal for many applications that have to process portions of speech signals with multiple simultaneously active speakers. In [1], the authors from Google and Carnegie Mellon University, make the following statement: "*Additionally, as is standard in literature, we exclude overlapped speech (multiple speakers speaking at the same time) from our evaluation*" when presenting the performance of a speaker diarization system. Furthermore, in [2], a Google Research team presents a deep learning approach that takes both a single channel audio signal as well as a video recording of multiple people talking simultaneously and is able to separate the speech and assign it to each individual speaker. This alternative approach of using visual information to enhance speech analysis demonstrates the challenging nature of the task. However, the human brain has selective auditory attention (SAA) or selective hearing (SH), and is able to switch focus from one active speaker to another even when presented with a single channel recording. This seems to suggest that blind source separation (BSS) and speaker diarization of overlapping speech should be possible without additional information other than the audio signal.

The human SAA performance has not got focused attention from the technology research community, with few existing references on this topic. Being an interdisciplinary problem, we can find some literature that studies SAA and SH with the goal of explaining neuro-physiological features of various individuals, like in [3] and [4]. Reference [5] presents one of the first studies that tries to determine how many simultaneous speakers a human can detect accurately, using Japanese language, and 21 volunteers – males and females. The study also aims to quantify the impact of listening time on the detection accuracy.

In a previous work [6], we presented a similar experiment for Romanian language with the goal of creating an automated system for counting competing speakers and comparing it with human performance. We used a group of 31 volunteers, and the influence of listening time was also observed. As a follow-up, in [8], we used a different group of 38 volunteers and replicated the results, while also quantifying the impact of listening to voices that are known to the speakers. The study presented in [7] confirms our findings from the mentioned studies.

Knowing the limits of the human SAA, is useful because it can help technology research groups to set a first bar for automated systems trying to accomplish the same tasks. The ability of counting competing speakers at a given time is important for enhancing the performance of multiple applications like speech diarization, speech recognition, in-vehicle assistance, crowd-sensing, conflict detection, etc.

A first step in this direction is represented by some research efforts focused on a simpler task of detecting overlapped speech, and we highlight some of the attempts made in the last decade. One of the first studies designed to create solutions for detecting overlapped speech [9] – presents a Hidden Markov Model with Gaussian Mixture Model outputs (HMM-GMM) that uses well established features like Mel-Frequency Cepstral Coefficients (MFCC), Root Mean Square (RMS) and Linear Predictive Coding (LPC) coefficients to improve diarization error. In an environment with only 2 speakers, the

Submitted for review on 10th November 2018.

Valentin Andrei was during his Ph.D. with the Speech and Dialogue Research Laboratory from University Politehnica of Bucharest, Romania. He is now with Intel Corporation in Santa Clara, California, USA (email: valentin.andrei@intel.com).

Horia Cucu is with the Speech and Dialogue Research Laboratory, University Politehnica of Bucharest, Romania (email: horia.cucu@upb.ro).

Corneliu Burileanu is with the Speech and Dialogue Research Laboratory, from University Politehnica of Bucharest, Romania (email: corneliu.burileanu@upb.ro.

study in [10] presents up to 94% diarization accuracy boosted by the detection of overlapped speech, but without quantifying the impact of the enhancement. A similar effort that uses a Support Vector Machine (SVM) with engineered features is presented in [11]. In [12], identification of overlapped speech is intended to improve in-vehicle safety assistance. This study claims the highest detection F-Score we were able to find in existing literature, and we speculate that it is because it uses artificially mixed recordings from the TIMIT database [13]. References [14] – [17] have the same goal and we observed they all use hand engineered features. Similarly, [18] presents one of the first studies that approaches the problem using deep learning, by feeding well established features like MFCC and LPC to a Long Short-Term Memory (LSTM [23]) Recurrent Neural Network (RNN) to detect overlapped speech frames. The same challenged is approached with a CNN in [55], where overlapped speech is detected on frames as short as 0.5 seconds.

In a previous work [19], we proposed a convolutional neural network for detecting overlapped speech on very short timeframes (25 milliseconds). Being able to tag frames that would be appealing for integration as an intermediary step after the voice activity detector (VAD) in systems that aim for more complex tasks like speech recognition and diarization. Moreover, having high resolution when tagging timeframes is essential for estimating correctly the ratio of overlapping to non-overlapping speech for longer audio files, which was proven useful in conflict detection [53] and [54].

The next step in analyzing overlapped speech is to be able to count the number of competing speakers at a given time, and to be able to do that by analyzing a timeframe as short as possible. Surprisingly, there are rather few attempts to this problem in existing literature. In [20], the authors have detected an identifiable peak modulation pattern that is expected to decrease as the speaker count in a timeframe increases, and exploit this finding to estimate the number of active speakers. Reference [21] proposes a simpler approach, by directly estimating competing speaker count using the 7^{th} MFCC coefficient. In [24], [25] and [26] the authors propose various applications that try to count active speakers and use this information for crowd-sensing. In [22], the authors use a bi-directional LSTM to solve the same problem and obtain results that are generally better than prior work. However, the study analyzes long timeframes (five seconds) which limits the potential for contributing to a more practical application. In addition, it targets up to 10 speakers which as observed in [5] – [8] is far beyond what a human can accurately detect.

In [6] and [8] we presented an algorithm that uses dynamic time warping (DTW) on spectrograms to create a measure of similarity between a given input recording and a set of single speaker references. We observed that as the number of competing speakers increases, the similarity to the references drops and we used this to estimate speaker counts up to 10.

The current paper builds on top of our prior work [6], [8] and [19]. Even though in [6] and [8] we proposed a solution for competing speaker counting, the requirements in terms of analysis duration were high (more than one second) and the target of 10 competing speakers was unpractical for current applications' goals. Also, the studies were focused on Romanian Language and the amount of audio data used for the experiments was limited. The current paper is focused on English Language and uses convolutional neural networks to solve the same challenge. Furthermore, it targets sub-second timeframes up to 25 milliseconds. The results that will be presented in the following pages are encouraging and make our trained models more appealing for improving speaker diarization and speech recognition in overlapped speech environments. Another new contribution is a comparison with the work presented in [19], by using the same models trained for speaker counting to detect overlapped speech.

The rest of the paper is organized as follows. In Section 2 we will present the perception experiments methodology and their conclusions. Section 3 will briefly present the non-deep-learning algorithm we proposed for speaker counting along with reasons that motivated our pursuit to improve the solution. Section 4 describes the speech corpus that was used to create the mixtures along with the annotation procedure and is followed by Section 5 where we describe the feature sets that were used for building the models, along with the intuition behind selecting each feature set. Sections 6, 7 and 8 are dedicated to presenting the model architectures and to describing training and inference results. Finally, Section 9 is reserved for conclusions and ideas for future work.

II. Counting competing speakers by humans

Humans find it natural to switch attention from one speaker to another in a competing speakers' environment but when our volunteers were asked to listen to a single channel recording and count active speakers, the feedback we received is that this is a difficult and mentally exhausting task.

A. The Experiment Application

One of the key challenges that occurs when performing perception experiments, is to make sure every participant is subjected to the exact same conditions. Therefore, in order to enforce repeatability, we have implemented a software application that runs the entire process. The listeners have to answer the following questions for each recording they listen:

1) "How many speakers did you count in the recording?"
2) "Did you recognize any of the voices?"
3) "What discussion topics did you recognize"?
4) "Do you have any other comments?"

The speech mixtures used by the application combine both male and female voices. Each speaker was given a topic from a broad spectrum (e.g. religion, technology, economics, physics, biology). There are no topics shared by speakers.

The participants were instructed that there can be maximum 10 simultaneous speakers in a recording, but they were also told that there is no rule that each participant would be exposed to all source counts, and that each count in a recording is generated randomly. This was done to prevent side judgements.

Every time the listener plays a recording, the application randomly selects a duration from 5, 10, 20, 40 and 80 seconds. The sequence of durations was the same for each listener. This strategy was introduced in [8]. Initially, in [6], each attendee had unlimited time to give a response, and we hoped to see a

correlation between response time and correctness. However, we discovered that there was no correlation because the

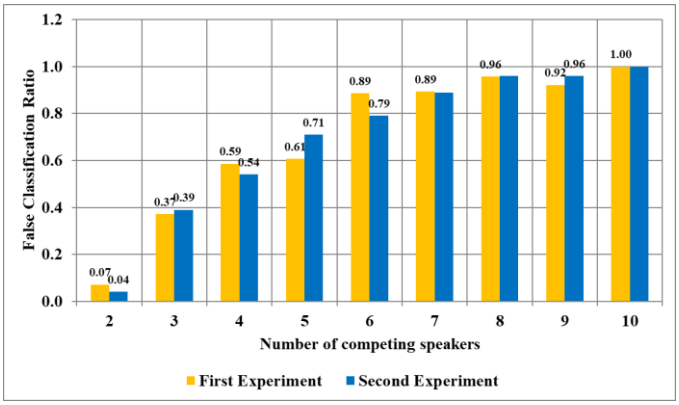

Fig. 1. Speaker counting error rate produced by human listeners, comparison between experiments in [6] and [8]

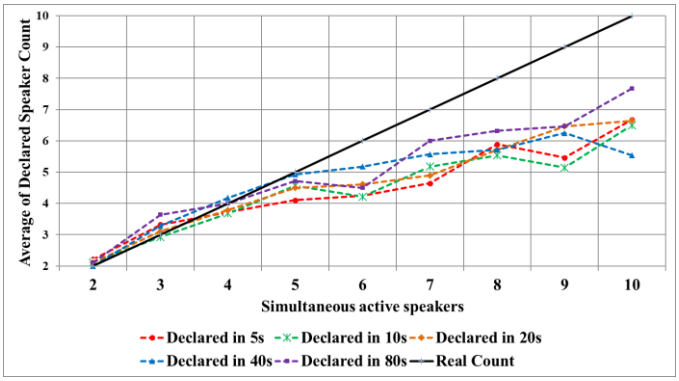

Fig. 2. Declared speaker count for each listening duration

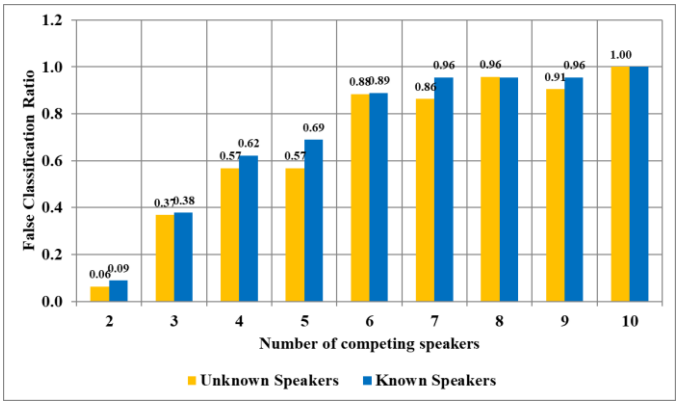

Fig. 3. Influence of listening to known voices over counting error rate

dispersion of response times was very wide. Therefore, in [8] we decided to limit the listening time for each recording.

The participants were also given the possibility to pause the test after listening steps if they accumulated fatigue, since we were interested in receiving answers as accurate as possible. The application saves all the answers in an encrypted output file and sends it to our team via e-mail.

In the second session of this experiment, in [8], we reported the score to the participants when they had finished all the recordings and observed that this increased their interest.

B. Counting Results

Figure 1 compares the false classification ratio (FCR) between the experiments we presented in [6] and [8]. When computing the FCR metric, if a listener guessed the speaker count wrongly, the response was tagged as false no matter how close the answer was. We can observe that the classification error is very similar between the 2 sessions, even though the groups of volunteers and the multi-speaker mixtures were different. Moreover, we can see that when more than 4 speakers are active simultaneously, the estimations of the participants become unreliable.

TABLE I
LISTENING EFFECTS REPORTED BY PARTICIPANTS

Effect	% of volunteers reported it
Learned voices from previous recordings	78%
Recognized a known person	67%
Was able to follow transmitted information	53%
Recognized different genders	46%
Just guessed where speaker count was high	39%
Detected different speech paces	32%
Reported hearing other languages	17%
Used silence periods to identify new speakers	14%
Reported words that are repeating	10%

Figure 2 shows the influence of the listening time over the counting accuracy. We can draw several conclusions from this diagram. The first is that the correlation between duration and accuracy is rather weak. While we can see that the line corresponding to 5 seconds recordings is on average showing the highest error, we can also look at recordings for 8 competing speakers where 5 seconds produced even better results than listening to 40 seconds. We can strengthen this conclusion by stating that correlation between listening duration and accuracy drops with source count. We might also be tempted to say that a faster response based on instinct has chances of being a better one than an answer based on a long and focused listening period, but in order to draw this conclusion we would probably need hundreds of volunteers to clearly illustrate a pattern.

A second observation from Fig. 2 is that participants overestimated the count for 2 and 3 speakers but underestimated it for more than 4 speakers. This is in line with findings from [5], fact suggested even by the paper's title: "*One, two, many – judging the number of concurrent talkers*". Related to this finding, we were in doubt whether to let the volunteers know that there are maximum 10 speakers, or not. We chose to tell them because we feared that when listening to a high-count recording, they would have been tempted to try totally random responses. We speculated that knowing there is a limit that seemed tangible to detect, they will be motivated to focus and try to give their best response. This is probably one reason for why this underestimation of counts occurs for more than 5 sources.

Figure 3 seems to suggest that detecting known voices in the recording, marginally increases the FCR metric. This effect is most noticeable for 5 sources where there is a 21% increase in FCR. We can speculate that this effect is because when participants detected a known voice, they were tempted to switch their auditory attention to that source and follow it, while ignoring the other sources. However, even if the mixtures are created by known persons to the volunteers, this does not significantly change the conclusion of the counting

experiment, with 4 sources still being a difficult target to produce reasonable estimations for.

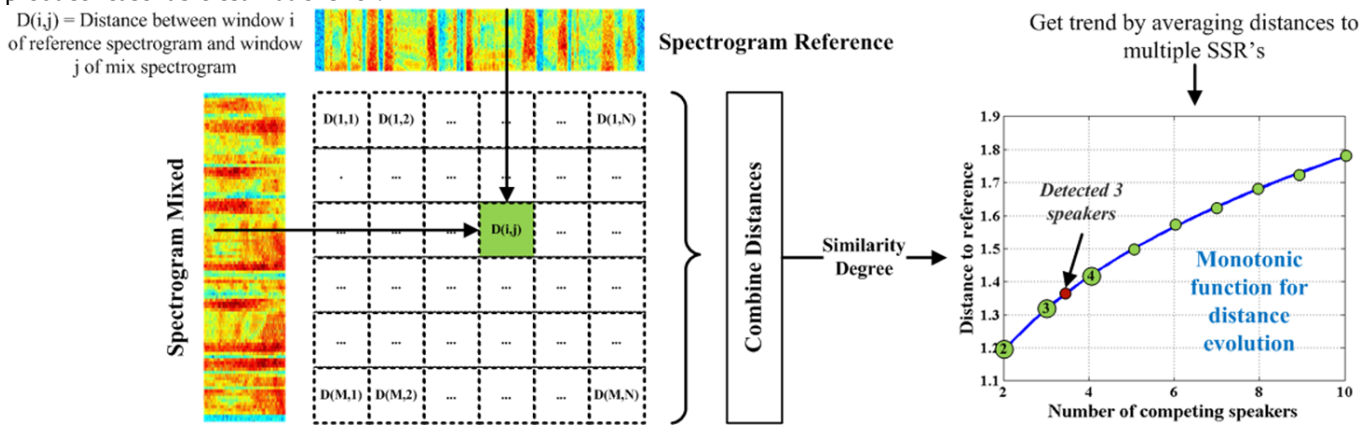

Fig. 4. Estimating speaker count using a similarity degree towards single speaker references

C. *Secondary Conclusions*

Table I highlights some of the main feedback notes that were provided by the participants after the experiment. We can see that their majority are "strategies" used to improve their counting accuracy. For example, in mixtures where there are combined voices of different gender speakers or voices of speakers with different speech paces, it is easier for a listener to detect some of these features. Detecting different paces and using silence periods to count other speakers are probably related.

The fact that 39% of the participants declared that they just guessed where they couldn't figure out the count is a confirmation that informing the listeners that there are maximum 10 speakers was a better decision, since a guess without an upper limit may have yielded even responses of 15, 20 or more speakers. Also, given that 17% of participants – more than 5 – reported hearing different languages (even when that was not the case) we can confirm that the task is mentally exhausting.

It is important to note that more than 50% of the attendees were able to track the subject of various portions in the recording, and that 78% were able to learn voices from previous recordings. This was possible because indeed some of the voices were used in multiple recordings. These two notes are probably a cause for increased FCR since both effects can "distract" the listener from only counting all the speakers. We expect only minor changes to the conclusions of this study, should it be repeated with more than 40 participants and more mixture data. We believe that mainly because of results displayed in Fig. 1, where introducing known voices in the mixture is an important distraction factor but does not change the overall conclusion: human listeners find it very challenging to accurately count more than 4 active simultaneous speakers in a single channel recording.

III. Speaker Counting Using Single Source Templates

We will briefly go over a non-deep-learning strategy for speaker counting [6], [8] because we think the idea behind this solution is worth mentioning.

The proposed algorithm assumes that as the number of sources in a speech mixture increases, the similarity between the audio signal and a single speaker reference decreases. Therefore, we need a metric that can quantify how different 2 recordings are. Ideally, we should be able to find a bijection between the distance metric (or similarity degree) and the speaker count.

Figure 4 describes our approach. The first step is to compute the spectrograms of both the mixture and the single speaker reference. Then, we compute the distance for each frame in the mixture's spectrogram, to each frame in the reference's spectrogram. The metric we selected is dynamic time warping (DTW) distance. After all the DTW computations, we have a matrix of distances from which we can extract the final similarity degree. In [6] and [8] we just summed all the elements in the distances matrix. Once we computed the similarity degree we can map it to a pre-computed curve that predicts speaker count based on distance to references. It is important to mention that we are using multiple single speaker references when computing the distances. For experiments in [6] and [8], we used 5 references.

We can see the curve we obtained in Fig. 5, where for each speaker count we averaged more than 1000 distances between all the mixtures and all the single speaker recordings for 5 seconds of signal. The obvious fact is monotonically increasing distances from the reference set, as the speaker count grows. Although the trend is clear, we admit that the standard deviation of the analyzed data does not allow for an error-proof classification.

The counting performance we obtained with this algorithm is illustrated in more detail in [8]. False classification ratio depends as expected of the frame length. For lengths of 5 seconds we obtained around 0.3 FCR, which means 30% of samples were labeled incorrectly. The error drops with the increase of analysis duration. In [8] we also analyzed the influence of noise. The conclusion was that the presence of noise shifts up the curve we obtained in Fig. 5 because the differences between amplitudes of frequency points accumulate during the DTW distance algorithm. Therefore, in the presence of noise, we have to compute different curves associated to various signal to noise ratios.

This approach is extremely computationally demanding, part because of the high number of DTW distances that need

to be computed, and part because DTW is a routine that cannot be easily optimized for massively parallel architectures, due to loop carry dependencies. Therefore, this algorithm is merely a first step demonstrating that speaker counting on a single channel recording is a solvable problem.

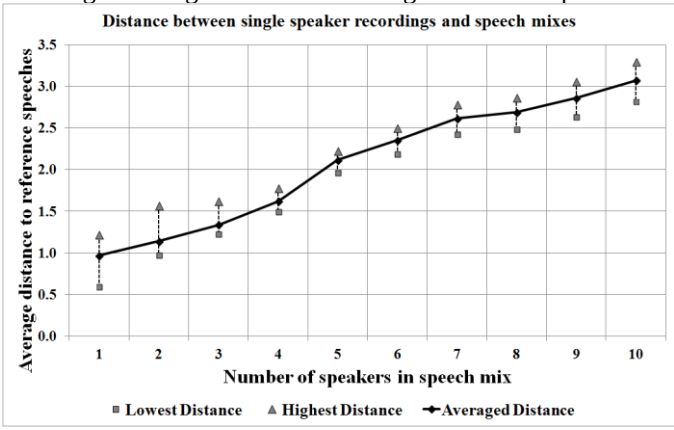

Fig. 5. The evolution of distances between mixtures and references as speaker count grows

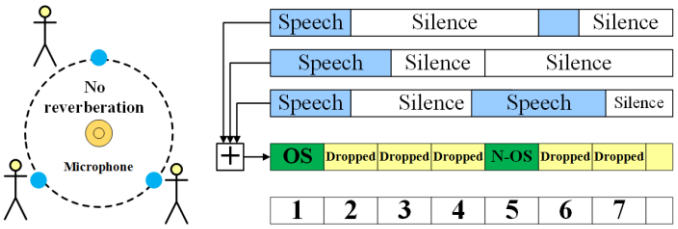

Fig. 6. Labeling speech mixtures frames

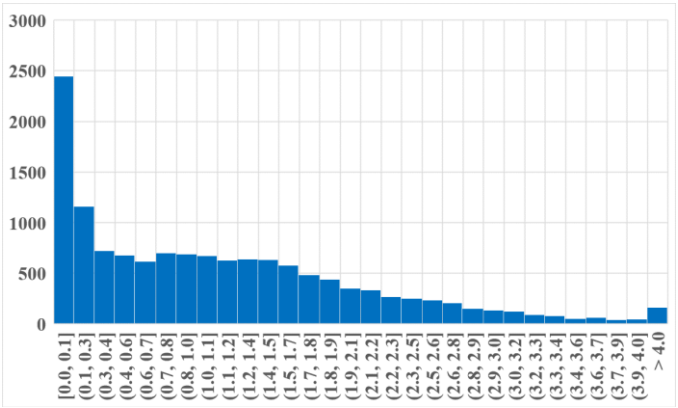

Fig. 7. Histogram with lengths of continuous speech computed on Librispeech test-clean dataset

IV. Preparing the Dataset for Deep-Learning Solutions

The first critical step we need to perform in order to ensure the success of a deep-learning solution to the proposed problems, is to make sure that the training and inference datasets are properly labeled. In [6] and [8], since the targeted frame lengths are large, we can safely assume that the "amount" of speech is equally distributed throughout the entire recording for all speakers. However, in [19] and also in the current paper we are targeting sub-second frame lengths and this raises different challenges.

A. Mixtures Creation

There were two options for building the input dataset: one was finding a corpus of recordings where human volunteers were actually talking simultaneously while being recorded, and the other one was to create a set of artificial mixtures. The first option was satisfied by the AMI Corpus, presented in [27], and used in [9], [17] and [18] to create a model designed to detect overlapped speech. We explored AMI Corpus but we had to apply blind source separation algorithms in order to extract the active periods for each speaker and the results were not satisfactory. There was too much interference caused by other voices farther away, to the target speaker's microphone.

It is critically important to be able to tag the input speech frames correctly with either the overlap status or the speaker count, in order to ensure the training convergence. This fact is demonstrated by more accurate overlapped speech detection presented in [10] and [16], and therefore, we decided to create a set of artificial mixtures, which enables the possibility of creating a huge number of input samples for training the model.

Figure 6 describes the mixing process. The main idea is that no frames with a partially active speaker were used to create a mixture. We labeled the frame with either the speaker count or with the overlap status. In order to detect the active periods of speech, we down-sampled the signal from 44.1kHz to 16 kHz and we used the Voicebox collection of scripts illustrated in [30], which implements a voice activity detector (VAD), based on the work presented in [31], [32], [33]. When combining sources, a normalization was applied to avoid clipping effects.

Our mixing strategy is analogue to a case where speakers are equally distanced from a single microphone, and they are in an environment with no reverberation. In future studies, we may study the impact of reverberation, by using artificial techniques for simulating a room, like in [28] and [29].

In [19], where we trained a convolutional neural network (CNN) for detecting overlapped speech, we used Romanian language, and a limited number of 10 male speakers for creating the training samples and 5 different male speakers for the inference samples. For the experiments presented in this paper, we used the *Librispeech* corpus [34], which comprises utterances in English produced by both male and female speakers. We used the *dev-clean* subset for training the models and the *test-clean* subset for inferencing. Both subsets are produced by different groups of 40 speakers each and contain about 5 hours of speech. Because in previous experiments we used a different language than in the ones to be presented next, we have the opportunity to conclude if the proposed methods apply to multiple languages.

B. Selecting maximum speaker count and frame duration

When creating the mixtures, we had to decide the maximum number of speakers we will count per frame. We decided to count up to 4, and the models were not trained to detect silence. Our models will label frames as belonging to one of four classes: 1, 2, 3, 4 speakers. We picked 4 speakers as we considered it to be a reasonable upper limit for conversational speech. Also, as the results presented in [5] - [8] show, and as reiterated in section II, humans find it very challenging to estimate accurately more than 3 simultaneous speakers in a

single channel signal. Being able to count more than 4 sources is probably more useful for niche applications, like crowdsensing. Our ultimate goal with these experiments is to be able to improve the robustness to multi-speaker environments for applications like speaker diarization, speech recognition, conflict detection, in-vehicle assistance, etc...

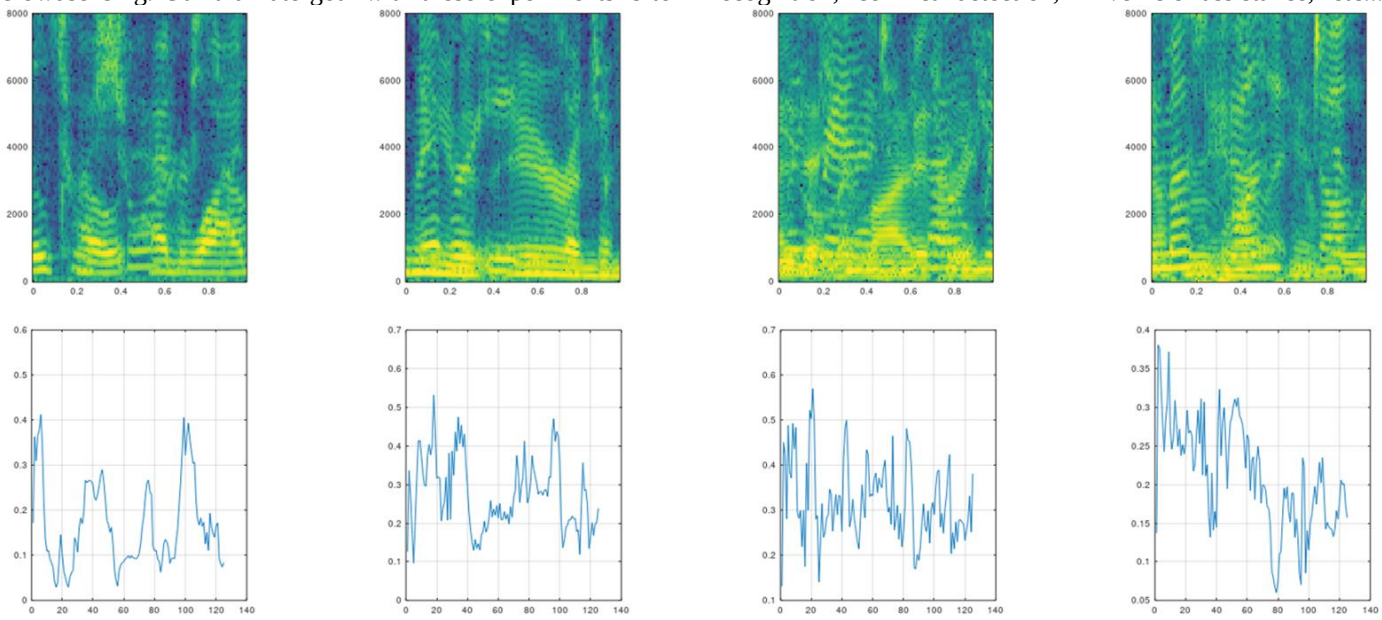

Fig. 8. Spectrogram and speech signal envelope examples for 1, 2, 3 and 4 simultaneously active speakers

Selecting the maximum and the minimum frame durations for which we want to optimize the models were also interesting decisions. A longer frame length has higher chance of containing more than one word so therefore a silence period will "pollute" the frame. The histogram from Fig. 7 was computed on *Librispeech test-clean* and counts the number of continuous speech segments of each length. As we can see, the probability of encountering a segment where a speaker is continuously active more than 2 seconds is small. The main majority of segments is less than one second. Figure 7 findings are perfectly in line with the study presented in [35], where word duration is measured for people speaking while exposed to various levels of stress.

Given these findings, and taking into account that having to analyze frames longer than 1 second would make our method less appealing for latency critical applications, we decided to optimize the models for frame lengths up to one second.

When selecting the lower limit, we want to be able to accurately count speakers for frames as short as possible, but this limit is given by the quantity of information stored in the signal window. In [19], we were able to detect overlapped speech for frames as short as 25 milliseconds so we kept the same limit for speaker counting.

V. Feature Set Selection

Speech analysis applications using deep-learning still generally rely on some feature extraction steps. We can speculate that this is caused by the nature of the speech signal which can cause frequent convergence to local minima during training. In addition, speech signals are often described as being quasi-periodic [36], [37], and this property can add redundant data to the input samples, increasing unnecessarily the model size, thus leading to computing inefficiencies.

In the deep learning era, there are studies that challenge the perception that feature extraction is needed, for example in [44] and [45], which seem to suggest that very deep neural networks can learn a set of features as representative as ones previously used in state-of-the-art literature. To demonstrate that, the authors of [44] and [45] just use the unprocessed raw audio signal as input to the model.

For our experiments, we have not excluded using the unprocessed mixture signal as input to the training phase, but we have discovered that using processed features yields better accuracy and faster convergence. Next, we will describe some of the features we selected for the experiments and the intuition behind the selection decision.

- *Unprocessed speech signal* – By using the entire frame to form the input samples, we have at least the guarantee that we are using the maximum amount of information we have for classification. The downside of this is that the number of trainable model parameters will increase heavily, especially if the first layer of the model will be a fully connected one. This is why we experimented with raw speech only for very short frames, where we needed as much quantity of information as possible. For future experiments, we are considering a novel technique for enabling the usage of raw speech frames as input vectors, presented in [38]. Similarly to convolutional layers, the model described in [38] uses chaining of *sinc* filters which basically selects various frequency bands, making the compute more efficient.

- *Signal envelope* – The intuition behind this feature is that as the number of active speakers grows, the envelope of the signal will show more frequent local minima and maxima due to the combining of single speaker signals. In Fig. 8, the set of waveforms in the bottom row represents

the envelope of recordings for 1 to 4 competing speakers. Our assumption seems to be reflected, with the waveforms corresponding to 1, 2 or 3 speakers showing clearly an increase in the number of local minima or maxima, but with envelopes for 3 and 4 competing speakers being hard to separate using this criterion. When computing the signal's envelope, we used the Hilbert transform.

- *Histogram of speech signal* – The hypothesis behind selecting this metric is that a mixture with 2 or more competing speakers is likely to have more frequent regions of high power. Also, this is a relatively small feature because we selected only 50 bins for the power levels.

- *Time-frequency spectrum* – The spectrogram continues to be a very popular feature in machine learning based speech analysis applications [39], [40], [42], [43], [55]. In [39], the spectrogram is used on a frame by frame basis by feeding it to the model, while in [40] and [55], the spectrogram of a signal is used as an image. As also speculated in [55], as speaker count grows, the spectrogram becomes denser and converges more to an image representing noise. However, we learned that this property is not always visible when representing the time-frequency spectrum as an image, as we did in Fig. 8. We can see that for one and two speakers, there appear to be slightly more darker regions, but we cannot observe a clear pattern for three and four competing speakers. For overlapped speech detection, subtle differences caused by speaker count do not pose such a difficult challenge as exemplified in [55]. Still, even if the spectrogram loses phase information compared to raw signal, it provides a high amount of information. In some cases, where the analysis duration is small, we experimented replacing the spectrogram with just the frequency spectrum and we observed no significant accuracy differences.

- *MFCC* – From their introduction in [41], Mel-frequency cepstral coefficients (MFCC) have been the norm in speech processing systems. Even though challenges to this status-quo appear in several studies like [44] and [45], MFCC used as feature vectors are still shown to provide slightly better word error rate (WER) compared to when using raw signal as feature set. In the current paper, we explore using the MFCC for long analysis durations, where using the spectrogram would generate a very high number of input features, increasing the model size exponentially. While we noticed that for shorter durations, spectrogram-based models yield slightly higher accuracy, the difference is not impressive compared to MFCC based models.

We explored also other feature sets like auto-regressive (AR) model coefficients or perceptual linear prediction (PLP) cepstral coefficients, but our results show that they do not improve the accuracy and add computational costs. This is why the statistics presented in the following paragraphs are obtained by training our models using various combinations of the mentioned feature sets. In cases where input frames are very short, we expand the input features by adding their squared values or combinations between various components – for example in [19], when we used this strategy for 25 milliseconds frames.

VI. Model Architectures

When deciding on the model architecture, we were guided by the belief that if our intuition is correct when selecting the type of the neural network, a large model trained with sufficient data will yield good accuracy. Given the number of hyperparameters that can be explored when designing a neural network, that can spawn thousands of combinations, prior work and intuition played an important role in defining the model.

A. CNN or LSTM?

We considered that traditional machine learning techniques like support vector machines (SVM), hidden Markov models (HMM), k-means clustering, self-organizing maps (SOM) and even multi-layer perceptron models (MLP) do not have the required structure complexity to model the relationship between the features in the input in order to produced high accuracy. Therefore, the main decision was whether to use a convolutional neural network (CNN) or a recurrent neural network (RNN).

Convolutional neural networks were conceptualized for more than 20 years, for example in [46], but they gained traction among researchers in the last years, especially due to the increase in compute resources, which followed and at some points even outpaced Moore's Law [47]. In speech recognition systems they were used successfully in research like [43], [48] and [49]. In [19], we used CNN's for detecting overlapped speech and in [38] a novel CNN architecture is proposed for speaker recognition. Even though CNN's are extremely powerful for image analysis applications, we can see there are important good results that demonstrate their usefulness in speech-based systems.

A subclass of RNN's, long-short term memory networks, were successfully used in the last few years for sequence modeling where there is a correlation between elements in the sequence, like speech recognition or natural language processing. Even though they gained attention lately, the concept is also at least 20 years old, and one of the first cited papers is [23]. According to [50], RNN's are used in well-established speech recognition applications like Amazon Alexa, Google's Assistant or Apple's Siri. Another powerful example of LSTM's used in speech analysis systems is presented in [39]. Speaker counting was approached using LSTM's in [22]. Interestingly, even though LSTM models are considered the norm in sequence modeling, [56] challenges this assumption and demonstrates CNNs can even outperform LSTMs.

For speaker counting or overlapped speech detection on short timeframes, we believe CNNs are a better solution. This is because during conversational speech, a speaker may become active at any time, suggesting that signal frames in a sequence should be regarded as uncorrelated. With longer signal windows, the correlation between timeframes in the

sequence may be higher, which justifies the usage of LSTMs, like in [22], where 5 seconds frames are used. Also, there is a general belief that CNNs are faster by design which is appealing for an application where latency is critical. In addition, we obtained encouraging results for overlapped speech detection with CNNs in [19] and after that, [55] presented another CNN based approach for the same task with promising results.

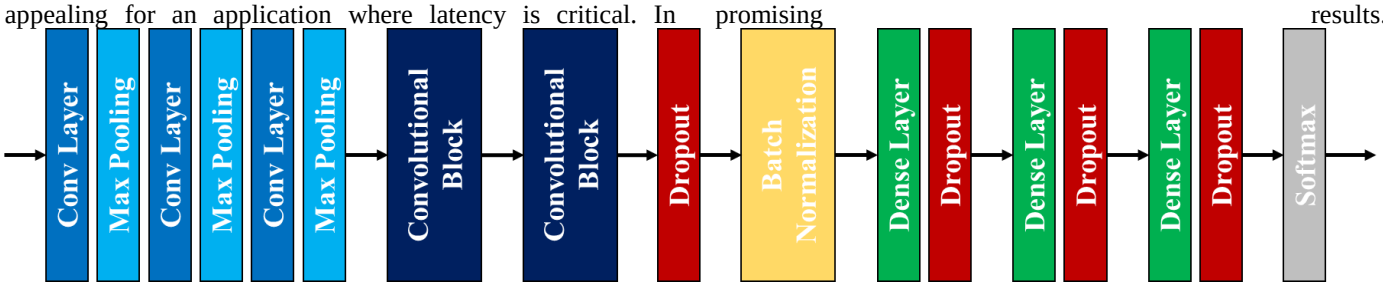

Fig. 9. Architecture of the CNN used for speaker counting. Convolutional blocks are a succession of 3 pairs of convolutional layers and max pooling layers

```
# Architecture
n_conv_blocks       = 3
v_convs_per_block   = [3, 3, 3]
v_pool_size         = [1, 2, 1]
v_filters_per_conv  = [32, 64, 128]
v_krn_sz_per_conv   = [8, 6, 4]
f_dropout_conv      = 0.75
n_fc_layers         = 3
v_fc_layer_sz       = [1024, 512, 256]
v_dropout_fc        = [0.1, 0.1, 0.5]
```

Fig. 10. Configuration section in the model training script

B. *Speaker counting architecture*

Figure 9 illustrates the architecture of the model we used for speaker counting experiments. In [19] we used a smaller variation of the same architecture for overlapped speech detection.

The topology we used is a fairly common CNN architecture, with examples in [43], [45] or [49]. We experimented with using both 1D and 2D convolutional layers and we observed that 1D filters produced at least the same accuracy as 2D filters, at a slightly reduced computational cost. Figure 10 shows the configuration section of the model training script. All the models were created using TensorFlow (www.tensorflow.org) and Keras API (https://keras.io/).

As it can be seen, the first parameter that can be configured is the number of convolutional blocks. A convolutional block contains a configurable number of convolutional layers and max-pooling layers. The number of filters and their size can be set per block. After the convolutional blocks we have a dropout operator applied with variable rate. The batch normalization layer, introduced in [51], following the convolutional blocks has the role of essentially speeding-up the convergence. It becomes even more important in training the model using reduced precision, like 16-bit floating point representation. Batch normalization also has some regularization effect. This is why the preceding dropout layer may be redundant but since the dropout rate is variable, we decided to keep it to increase the flexibility degree. Finally, the topology is completed by a variable number of fully connected layers with dropout operators attached. The dropout steps on fully connected layers were observed to have a critical role in preventing overfitting. Similarly, the number of hidden units and dropout rates are configurable. The Softmax layer is used as the CNN produces a one-hot vector of size 4, associated with 1, 2,3 or 4 concurrent speakers per frame. All the scripts used to prepare the datasets, to run training, validation and inference are made open-source at: *https://github.com/valentinandrei/saa_experiments*.

In [19], we used a variation of the topology presented in Fig. 9 for overlapped speech detection. We used a single convolutional block without max-pooling and we used a higher number of fully connected layers.

VII. Model Training

For speaker counting we are targeting frame lengths of 25, 50, 100, 200, 300, 400, 500 and 1000 milliseconds. We are using spectrogram, signal envelope or MFCC among the feature sets, and this causes a variable size feature vector for each frame length. Due to this we trained a model for each analysis duration. To reduce the number of hyperparameter searches, we used 100ms and 500ms as proxies for the rest of the models.

A. *Training process guidelines*

We divided the dataset used for model building in the *training* and *validation* subsets. Since the dataset is basically unlimited due to the countless possibilities of mixing the input sources, we used only 4% for the validation subset, leaving the rest for training. At minimum, we used 600000 samples in the dataset for model building, and therefore validation was performed on at least 24000 samples which has a reasonable prediction power on how inference will be performed on a different dataset. For training the models we used *LibriSpeech dev-clean* datasets and for inference we used *LibriSpeech test-clean* and mixtures were created as described in section IV.

During training we used Adam algorithm, presented in [52] and provided by the Keras API, in order to implement the gradient-based optimization of the objective function. Adam optimizer was used as it gained traction in the last years and believed to be more efficient than other approaches at adapting the learning rate. We used across the spectrum of tracked frame lengths a learning rate of 0.0005. While the model was being trained, we monitored the loss function and the categorical accuracy. Regarding the batch size, we adapted to the available computing resources and that varies across all the models that we trained. We followed three steps that guided the entire training process.

- We monitored the loss function for a sufficient number of epochs to make sure we don't miss steep drops in training and validation error. We observed that by saving the model checkpoint based on loss function value instead of categorical accuracy, inference performance is closer to validation set performance.

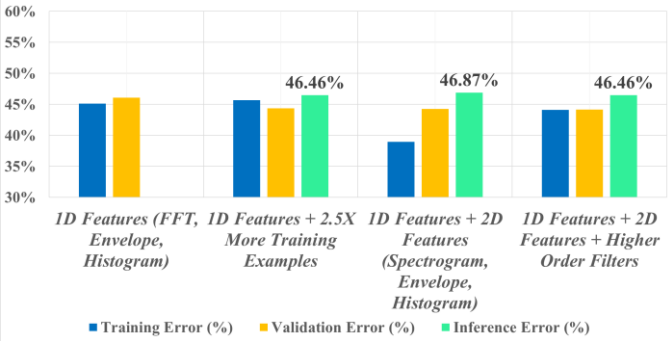

Fig. 10. Training steps for speaker counting on 100ms frames

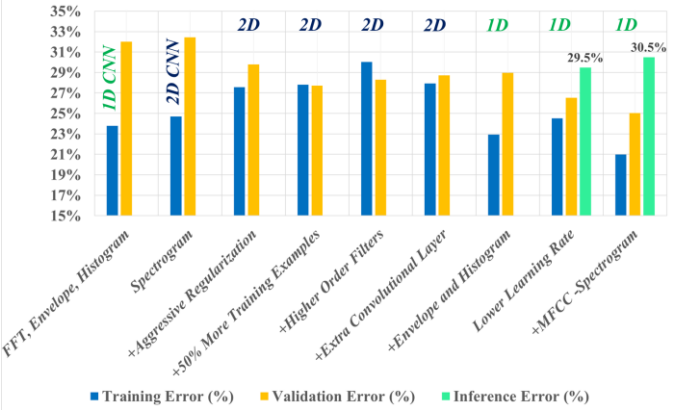

Fig. 11. Training steps for speaker counting on 500ms frames

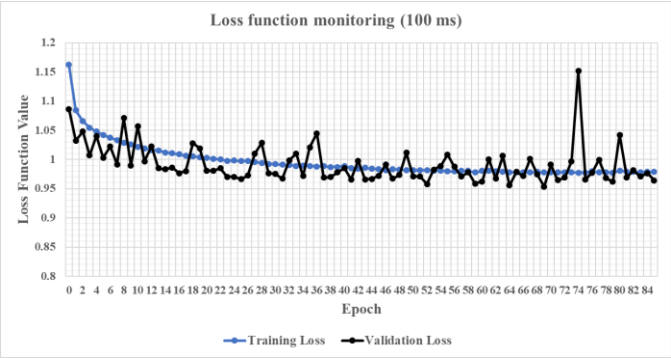

Fig. 12. Loss function decrease for 100ms model

- If the bias is high, that means the CNN fails to find predictable correlations between input features, and this can be either due to insufficient training examples or due to using a model with too few trainable parameters. If increasing model size or adding more datasets does not help, we also had the option of adding more components to the feature vector.

- If the variance is too high, that means that the CNN "maps" itself too well on the input data causing overfitting and we solved these issues by increasing regularization through dropout rates.

B. Training steps for 100ms and 500ms speaker counting

Figures 10, 11 and 12 are illustrating the training steps we took in order to converge the models we used to count competing speakers on frames of 100ms and 500ms. Each set of 2 or 3 bars in the charts can be viewed as an incremental step over the previous set.

For 100ms we used a CNN with only 1D convolutional layers. We started with using only the FFT, raw signal envelope and histogram as feature set components and we obtained around 46% validation error. Next, we retrained the model with 2.5 times more training examples and only saw a 1.5% drop in validation error. In the following step we replaced the FFT with the spectrogram but we did not see any significant reduction in validation or inference set errors. At this point we tried to increase the number of trainable parameters and we increased the order of the filters used in the 1D convolutional layers and the model performance was not improved. We therefore considered the checkpoints for the last 3 steps as candidates for speaker counting on 100ms and selected the one with the smallest inference time as the final one.

In figure 12 we can see that the loss function was monitored for a sufficient number of epochs so that we did not miss a late steep decline of the model error.

The final inference error for speaker counting on 100ms is 46.46%. We reiterate the fact that we are targeting counting up to 4 simultaneous speakers, and, compared to the performance achieved by human listeners we can state that the model we trained achieves similar level of counting accuracy by analyzing a frame smaller by at least an order of magnitude.

As illustrated in Fig. 12, for 500ms we took more steps since we expected a much lower model error. Initially the model had only 1D convolutional layers and used, as in the 100ms case, the FFT, signal envelope and histogram as features. We continued by replacing the feature vectors with the spectrogram of the signal frame and moved to 2D convolutional layers, treating the spectrogram as an image – as observed also in [40]. At this step we continued to see a large difference between training and validation error indicating that the model was overfitting the dataset and needed more aggressive regularization. Increasing dropout rates reduced the validation error with 2% and increased the training error. We followed with 3 incremental steps by adding 50% more training examples, increasing the model size with higher filter orders and adding another convolutional layer. These 3 steps did not show a significant variation in validation error. The next attempt was to add more features like the raw signal envelope and histogram and get back to using 1D convolutional layers. This step caused a sharp drop in the training error of about 4% but increased variance since validation error remained the same as in the previous steps. Slightly increasing regularization and reducing the start learning rate of the optimizer caused a more than 2% drop in validation error. This last step brought us to the final model that we used for inference. As a side effort, we replaced the spectrogram with MFCC in order to reduce the model size and improve latency and even though we observed slightly higher validation error, the inference performance was better for the

previous model. In the end we stopped at 29.5% classification error for counting speakers on 500ms timeframes, which is considerably better than what human listeners can achieved as described in section II.

VIII. Inference performance

As stated in the previous chapter, we used the models we trained for speaker counting on 100ms and 500ms frame lengths as proxies to select the hyperparameters for a wider spectrum of

TABLE II
FEATURES DESCRIPTION FOR EACH MODEL

Frame (ms)	Feature Vector Size	Features
25	278	SPECGRAM, ENVELOPE, HIST
50	582	SPECGRAM, ENVELOPE, HIST
100	1267	SPECGRAM, ENVELOPE, HIST
200	2787	SPECGRAM, ENVELOPE, HIST
300	4258	SPECGRAM, ENVELOPE, HIST
400	5727	SPECGRAM, ENVELOPE, HIST
500	7197	SPECGRAM, ENVELOPE, HIST
1000	3837	MFCC, ENV, HIST

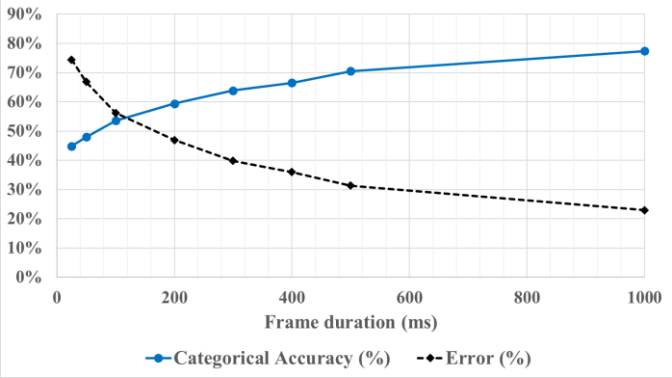

Fig. 12. Speaker counting accuracy as function of frame length

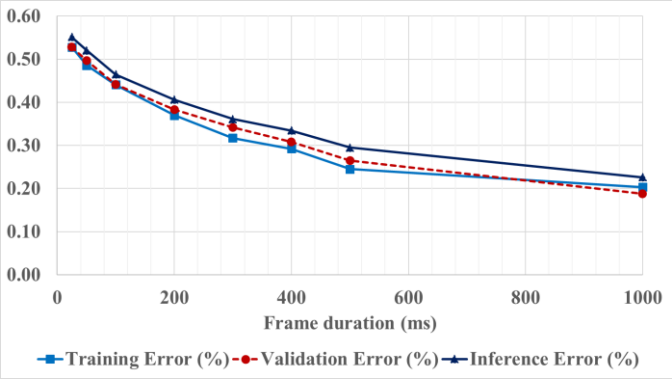

Fig. 13. Training, validation and inference errors for speaker counting

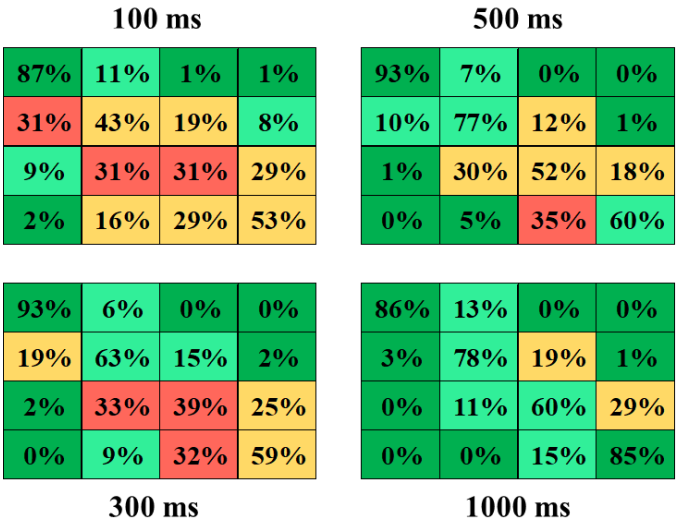

Fig. 14. Confusion matrices for speaker counting

analysis durations. Once the models were trained, we ran inference using 100000 samples created from a new dataset, in our case *LibriSpeech test-clean*. The same models trained for speaker counting were also used to detect overlapped speech and, therefore, we will present two sets of results in the following sub-sections.

Table II summarizes the feature vectors for all the models trained and presented in this section. We can observe how vector size increases with frame length due to the usage of the spectrogram and signal's envelope as features. For small frames, we reduced the spectrogram's window. After 500ms, we replaced the spectrogram with MFCC due to the growth of the vector size which became prohibitive for training the model. For 500ms we measured the impact of switching from spectrogram to MFCC and noticed that it is small enough to be compensated by the training and inference speed.

A. Speaker counting

Figures 12 and 13 are probably the most important results of this work. We can see in Fig. 12 that for 1 second frames, the speaker counting error gets close to 20%, while for sub-second frame lengths we see a steady drop in the error rate with the increase of the targeted duration. Figure 13 shows a relatively close error between training, validation and inference datasets which gives us confidence that the models have on overall a good generalization capacity.

In Fig. 14 we can analyze the confusion matrices for some of the trained models. We can clearly see excellent accuracy when the CNNs are being presented with a mixture with a single source, and good accuracy when the mixture has 4 sources. The models are challenged when having to distinguish more subtle differences between 2 and 3 speakers. Even though a mixture with 4 sources should have similar properties compared with mixtures with 2 or 3 sources, counting 4 speakers is more accurate because the models tend to predict "one or many" competing speakers. This is due to the fact that single source recordings are much more "different" than ones with competing speakers.

We believe that we are revealing the best results for speaker counting in single channel mixtures in existing literature. We

can compare with [22] where the authors achieve around 21% error for counting up to 4 simultaneous speakers on frames of 5 seconds. The 21% value was derived considering the counting accuracy reported for 2, 3 and 4 competing speakers. Our trained CNNs achieve the same level of error on frames of 1 second.

When comparing to human level performance we need to normalize the results we reported in Fig. 1 and Fig. 3. The listeners had to guess a speaker count, being given a limit of 10 speakers, while the CNN had only 4 classes to select the answer from. Therefore, if we consider only the counting accuracy reported for 1-4 speakers, the error is less than 24%. Our proposed CNN achieves this error level for a frame length between 500ms and 1 second, while the human listeners need on average at least 10 seconds, according to Fig. 2.

Intuitively, we think the strength of our approach relies in the fact that we trained a new model for each analysis duration, and we limited the number of active speakers to 4, based on human selective auditory attention performance. We recognize this may be a drawback for client applications where storing all the

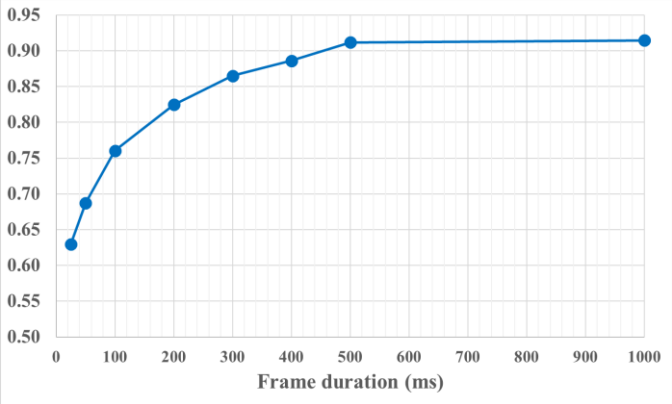

Fig. 15. Overlapped speech detection F-Score depending on frame duration

models is not possible. Another important factor that influenced the results is the fact that we did not use silence samples when training the model because we considered that the model should focus on detecting subtle differences between input examples.

B. Overlapped speech detection using the same models

In [19] we presented a CNN based approach for detecting overlapped speech. The topology had the same structure as the one presented in Fig. 9 with a single convolutional block and 6 fully connected layers. We targeted very small timeframes like 25ms where we obtained an F-Score of 0.72. This score was achieved using MFCC, signal envelope, FFT, AR coefficients and squared features. The main downside of the proposed methodology was that we used a non-standard dataset for training and inference, that was acquisitioned in-house.

We used the models trained for speaker counting to tag overlapped speech frames. During inference, if the CNN predicts 2, 3 or 4 competing speakers, the frame is labeled as overlapped speech. We expect slightly different results than [19], given the fact that the number of competing speakers was limited at 3, while for this study we use 4 sources as a limit.

Figure 15 illustrates the inference results varying with frame length. For 25ms of speech, the obtained F-Score is 0.63 which is lower than what we published in [19]. Intuitively, this is because the source count and by the fact that in [19] we use more features. For 100ms, we obtained an F-Score of 0.76 which is very close to the 0.78 value reported in [19]. Starting from 200ms frames, the performance of the models for overlapped speech detection is very appealing with values higher than 0.8. At 500ms we reached a plateau with the F-Score being 0.91, which is considerably higher than the 0.8 published in [19]. This could be due to using the spectrogram instead of MFCC and also due to using a much larger database.

CONCLUSIONS AND FUTURE WORK

This work concludes a set of several experiments designed to create automated methods for competing speaker counting and overlapped speech detection in single channel recordings. As seen in the sections dedicated to results, this is a task where a machine can surpass human level performance, essentially by being able to analyze much shorter frames of the input signal.

On mixtures longer than 5 seconds, two groups of 31 and 38 listeners demonstrate a classification error for up to 4 competing speakers of slightly less than 24%. With the proposed deep-learning approach, we achieved 22% error with just one second of analysis, and 29% error by counting on 500ms frames.

Also, we believe that we are presenting an inference accuracy for speaker counting which is better than scores published in current literature. We think this fact is likely due to our approach of training a new model for each targeted frame duration. This should not limit the potential for adopting the model into practical usages, because when designing an application, the speech signal is likely to be segmented with a fixed frame length, and a single pre-trained model can be used. It is true that one alternate solution is to count speakers on longer frames using models trained for 25ms or 50ms for example, though if we are to consider the frames completely independent, we don't believe this is a promising approach.

Another conclusion of this study is that during training, if limiting the number of competing speakers that can be counted, the confusion matrix converges toward predicting with higher accuracy one speaker and the maximum number of speakers. We believe this is because the differences between mixtures with 2 or more competing speakers are subtle. In future experiments, this finding can perhaps be exploited by a set of cascading CNN architectures that work as a binary classifier between 2 consecutive speaker counts.

Lastly, using the same models for overlapped speech detection, demonstrated F-Scores up to 0.91 on 500ms timeframes and higher than 0.8 for frames larger than 200ms. This reported performance is also higher than what we were able to find in existing prior work.

REFERENCES

[1] Wang, Quan et al. "Speaker Diarization with LSTM." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2018): 5239-5243.

[2] A. Ephrat et al., "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation", ACM Transactions on Graphics, 37(4): 112:1-112:11 (2018)

[3] Coch, D., Sanders, L. D., Neville, H. J., "An event related potential study of selective auditory attention in children and adults", Journal of Cognitive Neuroscience, Vol 17, Nr. 4, 2005

[4] Gomes, H., Duff, M., Ramos. M, Molholm, S, Foxe, J., Halperin, J., "Auditory selective attention and processing in children with attention deficit/hyperactivity disorder", Journal of Clinical Neurophysiology, August 2011

[5] M. Kashino and T. Hirahara, "One, two, many – judging the number of concurrent talkers" J. Acoust. Soc. Am., vol. 99, no. 4, pp. 2596–2603, 1996.

[6] V. Andrei, H. Cucu, A. Buzo, C. Burileanu, "Detecting the number of competing speakers – human selective hearing versus spectrogram distance based estimator," INTERSPEECH 2014 – 15th Annual Conference of the International Speech Communication Association Proceedings, 2014, pp. 467 – 470

[7] T. Kawashima and T. Sato, "Perceptual limits in a simulated cocktail party" Attention, Perception and Psychophysics, vol. 77, no. 6, pp. 2108– 2120, 2015

[8] V. Andrei, H. Cucu, A. Buzo, C. Burileanu, "Counting competing speakers in a timeframe – human versus computer", INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association Proceedings, 2015, pp. 3999-4003

[9] K. Boakye, B. Trueba-Hornero, O. Vinyals, G. Friedland, "Overlapped speech detection for improved speaker diarization in multiparty meetings", ICASSP 2008 – IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings, 2008

[10] W. Tsai, S. Liao, "Speaker Identification in Overlapped speech", Journal of Information Science and Engineering, 2010, pp. 1891-1903

[11] R. Vipperla, J. T. Geiger, et. al., "Speech overlap detection and attribution using convolutive non-negative sparse coding", ICASSP 2012 – Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2012

[12] N. Shokouhi, A. Sathyanarayana, S. O. Sadjadi, J. H. L. Hansen, "Overlapped speech detection with applications to driver assessment for in-vehicle active safety systems", ICASSP 2013 – Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2013

[13] Garofolo, John, et al., "TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1", Philadelphia: Linguistic Data Consortium, 1993.

[14] S. H. Yella, H. Bourlard, "Overlapped speech detection using long-term conversational features for speaker diarization in meeting room conversations", IEEE/ACM Transactions on Audio, Speech and Language Processing, December 2014, Vol. 22, No. 12

[15] N. Shokouhi, A. Ziaei, A. Sangwan, J. H. L. Hansen, "Robust overlapped speech detection and its application in word-count estimation for Prof-Life-Log data", ICASSP 2015 – Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2015

[16] S. A. Chowdhury, M. Danieli, G. Riccardi, "Annotating and categorizing competition in overlap speech", ICASSP 2015 – Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 5136-5121

[17] J. T. Geiger, F. Eyben, et. al., "Using linguistic information to detect overlapped speech", INTERSPEECH 2013 – 15th Annual Conference of the International Speech Communication Association Proceedings, 2013

[18] J. T. Geiger, F. Eyben, B. Schuller, G. Rigoll, "Detecting overlapped speech with Long Short-Term Memory Recurrent Neural Networks", INTERSPEECH 2013 – 14th Annual Conference of the International Speech Communication Association Proceedings, 2013

[19] V. Andrei, H. Cucu and C. Burileanu, "Detecting overlapped on short timeframes using deep learning", INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association Proceedings, 2017

[20] T. Arai, "Estimating number of speakers by the modulation characteristics of speech," in Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 2. IEEE, 2003, pp. II–197.

[21] H. Sayoud and S. Ouamour, "Proposal of a new confidence parameter estimating the number of speakers-an experimental investigation," Journal of Information Hiding and Multimedia Signal Processing, vol. 1, no. 2, pp. 101–109, 2010.

[22] F. R. Stoter, S. Chakrabarty, B. Edler, and E. A. P. Habets, "Classification vs. regression in supervised learning for single channel speaker count estimation" in accepted to ICASSP, 2018

[23] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, Nov. 1997

[24] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, "Crowd++: Unsupervised speaker count with smartphones," in Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 2013, pp. 43–52

[25] A. Agnessens, I. Bisio, F. Lavagetto, Et al. Speaker count application for smartphone platforms, Proc. of the 5th IEEE International Conference on Wireless Pervasive Computing, pp. 361-366, 2010

[26] P. Kannan, S.P. Venkatagiri, Et Al., Low cost crowd counting using audio tones, Proc. of 10th ACM Conference on Embedded Network Sensor Systems, pp. 155-168, 2012s

[27] J. Carletta, "Announcing the AMI Meeting Corpus", The ELRA Newsletter 11(1), January–March 2006, p. 3-5

[28] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," The Journal of the Acoustical Society of America, vol. 65, no. 4, pp. 943–950, 1979.

[29] E. A. P. Habets, "Room impulse response (RIR) generator," https: //github.com/ehabets/RIR-Generator, 2016.

[30] M. Brookes, "Matlab Toolbox for Speech Processing", Department of Electronics and Engineering Imperial College of London, 2002, http://www.ee.ic.ac.uk/hp/staff/dmb/ voicebox/voicebox.html

[31] J. Sohn, N. S. Kim, and W. Sung. "A statistical model-based voice activity detection." IEEE Signal Processing Lett., 6 (1): 1–3, 1999. doi: 10.1109/97.736233

[32] Ephraim, Y. & Malah, D. "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator", IEEE Trans Acoustics Speech and Signal Processing, 32(6):1109-1121, Dec 1984

[33] Rainer Martin. "Noise power spectral density estimation based on optimal smoothing and minimum statistics.", IEEE Trans. Speech and Audio Processing, 9(5):504-512, July 2001

[34] V. Panayotov, G. Chen, D. Povey, S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books", Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), http://www.openslr.org/12/

[35] Greenberg, S., Carvey, H.M., Hitchcock, L. and Chang, S. "Temporal properties of spontaneous speech - A syllable-centric perspective", submitted to Journal of Phonetics (based on a presentation at the ISCA Workshop on Temporal Integration in the Perception of Speech, Aix-en-Provence, April 5, 2002

[36] T. Nakatani, M. Miyoshi, and K. Kinoshita, "One microphone blind dereverberation based on quasiperiodicity of speech signals," in NIPS 16. 2004.

[37] Vasilakis M.; Stylianou, Y. Spectral jitter modeling and estimation. Biomedical Signal Processing and Control 2009, 129.

[38] M. Ravanelli, Y. Bengio, "Speaker Recognition from Raw Waveform with SincNet", https://arxiv.org/abs/1808. 00158v2 Accepted at SLT 2018

[39] D. Amodei, S. Ananthanarayanan, et. al. "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin", Proceedings of the 33rd International Conference on Machine Learning, 2016, JMLR: W&CP Vol. 48

[40] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," arXiv preprint arXiv:1609.07132, Sep. 2016

[41] S. Davis and P. Mermelstein, "Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences," IEEE Trans. ASSP, vol. 28, no. 4, pp. 357–366, 1980

[42] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pretrained deep neural networks for large-vocabulary speech recognition," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 30–42, Jan. 2012

[43] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 12, pp. 2263– 2276, Dec. 2016

[44] Sainath, Tara N., Ron J. Weiss, Andrew W. Senior, Kevin W. Wilson and Oriol Vinyals. "Learning the speech front-end with raw waveform CLDNNs." INTERSPEECH (2015).

[45] Hoshen, Yedid, Weiss, Ron J., and Wilson, Kevin W. “Speech acoustic modeling from raw multichannel waveforms”. In ICASSP, pp. 4624–4628. IEEE, 2015
[46] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks. MIT Press, 1995
[47] G. Moore. Progress in digital integrated electronics. In IEDM Tech. Digest, pages 11–13, 1975.
[48] Abdel-Hamid, Ossama, Mohamed, Abdel-rahman, Jang, Hui, and Penn, Gerald. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In ICASSP, 2012
[49] Sainath, Tara N., rahman Mohamed, Abdel, Kingsbury, Brian, and Ramabhadran, Bhuvana. Deep convolutional neural networks for LVCSR. In ICASSP, 2013
[50] S. Martin, “What is the difference between CNN’s and RNN’s”, https://blogs.nvidia.com/blog/2018/09/05/whats-the-difference-between-a-cnn-and-an-rnn/
[51] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In Proceedings of ICML, pages 448–456, 2015
[52] D. P. Kingma, J. Ba, “Adam: A method for stochastic optimization”, 3rd International Conference for Learning Representations, San Diego, 2015, https://arxiv.org/abs/1412.6980
[53] F. Grezes, J. Richards, and A. Rosenberg, “Let me finish: Automatic conflict detection using speaker overlap,” in Proceedings of Interspeech, 2013
[54] S. Kim, F. Valente, M. Filippone, and A. Vinciarelli, “Predicting Continuous Conflict Perception with Bayesian Gaussian Processes,” in IEEE Transactions on Affective Computing, 2014
[55] E. Kazimirova, A. Belyaev, “Automatic detection of multi-speaker fragments with high time resolution.” In 19th Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 1388-1392. 2018
[56] S. Bai, J.Z. Kolter, and V. Koltun. "Convolutional Sequence Modeling Revisited." Seventh International Conference on Learning Representations, 2018