[[1]](#footnote-1)

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 analyze speech 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 overlapped speech. The same brain feature allows us to detect the number of simultaneously active sources. In order to quantify human level performance for this task we have designed a perception study that evaluates participants’ ability to accurately count multiple speakers in a single channel audio file. The influence of listening time and of familiar voices were also studied. The study was carried in 3 sessions, with the help of 31, 38 and 80 volunteers and significantly extends some of the early findings in existing literature.

Using the conclusions from the perception analysis, several 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 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

# INTRODUCTION

S

peaker diarization continues to be a difficult goal for applications used in the presence of multiple 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 as input both single channel audio as well as video recording of multiple people talking simultaneously and is able to extract the corresponding speech signal for each individual speaker. This alternative approach of using visual information to enhance speech diarization 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 audio source. This seems to suggest that blind source separation (BSS) and speaker diarization of overlapping speech could be possible with only the audio signal as input.

Discovering the capabilities of human SAA has not got intense 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 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 perception study using Romanian language with the goal to create an automated competing speaker counting system and to compare it with human performance. We used a group of 31 volunteers, and also observed the influence of listening time. Next, in [8], we used a different group of 38 volunteers and replicated the results, while quantifying the impact of using voices that are known to the listeners. A year later, the study presented in [7] confirmed our findings from [6] and [8]. At that point, we were able to refine the targets for an artificial speaker counting system in terms of maximum competing speakers and input duration. In order to compare human level performance with machine performance we repeated the perception study using the same number of active speakers and similar mixtures durations as we are targeting with the automated method. The third session was realized for English language with the help of 80 volunteers and is to our knowledge the largest study that observes the ability of humans to count the active speakers in a single channel mixture. The results of all the 3 sessions are discussed in Section II.

Knowing the limits of the human SAA, is useful because it can help technology research groups to set a first bar for automated systems aiming to accomplish similar tasks. The ability of counting competing speakers at a given time is important for enhancing the performance of multiple applications like speaker diarization, in-vehicle assistance, conflict detection, etc. Speaker diarization can be improved by tagging each input frame with the number of active speakers. This can boost the performance of algorithms by isolating single speaker portions which allows for modeling the voice profile of the participating speakers. Conflict detection assumes that during a conflictual situation, the participants talk simultaneously and loudly which can be detected by an overlapped speech detection engine.

A first step in this direction is represented by research efforts focused on a simpler task of detecting overlapped speech. One of the first studies on this topic [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 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. Reference [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 challenge is approached with a Convolutional Neural Network (CNN) in [55], where overlapped speech is detected on frames as short as 0.5 seconds.

In a previous work [19], we proposed a CNN for detecting overlapped speech on very short timeframes (25 milliseconds). Being able to tag frames this short, would be appealing for integration as an intermediary step after the voice activity detector (VAD) in systems that aim for more complex tasks like speaker diarization. In another practical example, 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. 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 7th 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] and [57], the authors explore several deep-learning topologies to solve the same problem and obtain results that are generally better than prior work. However, the study analyzes timeframes longer than one second (with five seconds as the main focus) which limits the potential for contributing to more practical applications. 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 rule estimate speaker counts up to 10.

The current paper builds on top of our prior work [6], [8] and [19]. In [6] and [8] we proposed a solution for competing speaker counting, but 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 CNNs to solve the same challenge. Furthermore, it targets sub-second timeframes up to 25 milliseconds. 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 short frame durations that we are targeting make our models more appealing for improving speaker diarization. For diarization, we could use the single speaker periods to build voice profiles. If the speaker counting engine operates on long timeframes, the probability of having overlapped speech on each timeframe increases and this challenges the goodness of the created voice profiles. Also, diarization methods can be improved by reducing the output word search space for each speaker by creating constraints given by the speaker count associated to each frame.

The rest of the paper is organized as follows. In Section 2 we will present the methodology and conclusions of the perception studies. Section 3 describes the speech corpus that was used to create the mixtures database that was used to train our models, along with the annotation procedure. Section 4 describes the feature sets that were used for building the models, along with the intuition behind selecting each feature set. Sections 5, 6 and 7 are dedicated to presenting the model architectures and to describing training and inference results. Finally, Section 8 is reserved for conclusions and ideas for future work.

# Counting competing speakers by humans

Humans find it natural to switch attention from one speaker to another in a multi-source 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 mentally exhausting task.

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 software applications that run the entire process.

## First Session

When designing the first session of the perception analysis study, we did not know the number of speech sources that a listener can accurately track. Therefore, the volunteers were provided recordings with up to 10 simultaneous speakers.

|  |  |  |
| --- | --- | --- |
| Fig. 1a. First session counting accuracy without restricting listening time | Fig. 1b. First session average listening time per speaker count | Fig. 1c. First session correlation between listening time and counting accuracy |

The application that drove the experiment let the participants know the minimum and maximum number of sources. We used 10 speech sources and combined them, and therefore there was the chance that listeners would learn some of the voices from previous recordings. The mixtures were created using both male and female voices and had durations of about 5 minutes for each speaker count. As we observed, the duration was more than enough for the experiment. The volunteers were not constrained to provide an answer on the perceived speaker count in a given time. We measured the response time however to determine if it is correlated with the accuracy. The listeners used headphones and were given the option of taking a break if they found the task too challenging. In this first session, we used 31 volunteers with the average age of 27 years and 4 months. Figures 1a, 1b and 1c highlight the results we obtained.

In fig. 1a, we can observe the counting accuracy for each number of active speakers in the mixtures. As expected, when there are only 2 speech sources, the volunteers were able to correctly count them in about 96% of the cases. Listeners were able to accurately count 3 speakers in approximately 61% of cases and after this point, the accuracy steadily drops until we see a sharp rise at 10 sources. This was expected because when the participants could not accurately count the source count, they assumed it’s the highest number. However, the counting accuracy for 10 competing speakers is 11% which is very close to the probability of guessing.

Fig. 1b shows the average listening duration for each number of simultaneous sources. Interestingly, we see an increase until 8 sources and after that we observe a sharp decline in listening time. We can speculate that this is because some of the volunteers gave up listening if they were overwhelmed by the number of sources. As the source count increases the audio signal resembles more to noise, and therefore it is expected that some listeners may just guess a number.

Fig. 1c shows the correlation between listening duration and aggregated counting accuracy collected for a random subset of volunteers. Visually, there is no observable impact of listening time over the counting accuracy. This can be explained by the fact that the participants were not constrained to a certain response time, and gave the answer when they were ready.

In addition to speaker counting, volunteers were asked to comment on the experience. We summarized their responses in Table I. For example, almost 80% of participants were able to learn voices from different mixtures. For example, if they followed a certain speaker in the mixture with 2 sources, they may have recognized it in mixtures with other source counts. Interestingly, 5 volunteers declared they heard foreign languages event though the experiment was entirely in Romanian.

## Second Session

In the second session of the experiment, that was carried a year later, and presented in [8], we aimed to investigate more if there is a correlation between listening time and accurately counting the simultaneous speech sources. We restricted the response time of the volunteers to 5, 10, 20, 40 and 80 seconds. We speculated that as the allowed answer time is higher, we would observe improved accuracy.

This session also studied if human listeners can count more accurately simultaneous speakers if they are familiar with the voices in the mixtures. In order to accomplish this goal, we used 38 participants. 23 of them were coworkers and were familiar with each other’s voices. We recorded 10 volunteers, created the mixtures and played them to the rest of the 13 coworkers. In addition, we ran the perception study with 15 volunteers from a different social group where participants were not familiar with the voices in the recordings. In total, we had 28 listeners: 13 listened to familiar voices and 15 listened to unfamiliar voices.

Fig. 2a. summarizes the competing speaker counting accuracy measured in the second session, for each of the selected listening durations. We can observe that even though we had different participants and used different mixtures compared to the first session, the results are highly similar. We still observe more than 95% accuracy in mixtures with 2 speakers, and about 60% accuracy in mixtures with 3.

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% |

|  |  |
| --- | --- |
| Fig. 2a. Second session counting accuracy with restrictions on listening time | Fig. 2b. Impact of using known voices, observed in second session |

The drop in counting performance in fig. 2a. has a similar trend to the one in fig. 1a. Interestingly, the correlation between listening time and accuracy is still not visible. While we can see that when volunteers were required a response in 5 seconds the precision was lower, the increase in accuracy with listening duration is not observed clearly. This may mean that 5 seconds is still a sufficient time to provide a reasonable accurate response.

Fig. 2b. highlights the impact of being familiar with the voices participating in the mixtures, on the accuracy. Contrary to our intuition, we observed that when the listeners were familiar with the voices in the recordings, the accuracy dropped. This can be explained by the fact that some of our volunteers declared that when they identified a known person, they attempted to follow that person’s speech, ignoring the other present voices.

## Motivation for the Third Session

After the first two sessions of the perception study, we have reasonable data that shows how human listeners cannot accurately identify more than 3-4 competing speakers when presented with a single channel recording. It appears that 5 seconds is sufficient to provide reasonable response accuracy, compared with longer listening durations. In addition, we observed that familiar voices tend to capture the attention of the listener who may ignore other voices present in the mixture. The data in Table I presents some of the strategies used by volunteers to count the competing speakers and also suggests the challenging nature of this task.

Given that humans have difficulties in counting more than 4 speech sources, it makes sense that when designing an automated method with the same goal, we should start with 4 competing speakers as the maximum limit. This also matches with most practical applications that might be enhanced by speaker counting. Since the first two sessions presented to participants mixtures with up to 10 sources, a comparison between the results obtained and the accuracy of a machine which has to select up until 4 maximum sources is not valid. This is one main reason for which the perception study needed to be reiterated with limiting the source count to 4.

The second motivation for a third session of the perception study was related to the fact that in order to create an automated method for competing speaker counting, we needed a large speech corpus. English language has far more resources that can be used in statistical methods than Romanian and therefore we designed our speaker counting engine for English. This implies reiterating the perception study. Nevertheless, it is interesting to observe if the conclusions we presented in the previous two sections, for Romanian language, correspond to similar ones for English.

## Third Session

In the last session of the perception study, we focused on the volunteers identifying correctly 1, 2, 3 and 4 simultaneous speakers. Compared to the previous experiments, we added a single speaker class because the automated method would be designed to identify frames with a single active speaker. This will increase the overall classification accuracy but we can analyze the confusion matrices in order to compare the results with the previous two sessions.

For the computer method we are targeting very short frame durations with the maximum analysis frame being 1000 milliseconds. In order to compare it with human level performance, the volunteers’ response time was restricted to 500, 1000, 2000 and 5000 milliseconds. We can directly compare the results obtained on 5000 ms. Recordings with the ones obtained in the second session. The mixtures were created using the *Librispeech dev-test corpus* [34].

|  |  |
| --- | --- |
| Fig. 3a. Third session competing speaker counting accuracy | Fig. 3b. Confusion matrices for the third session |

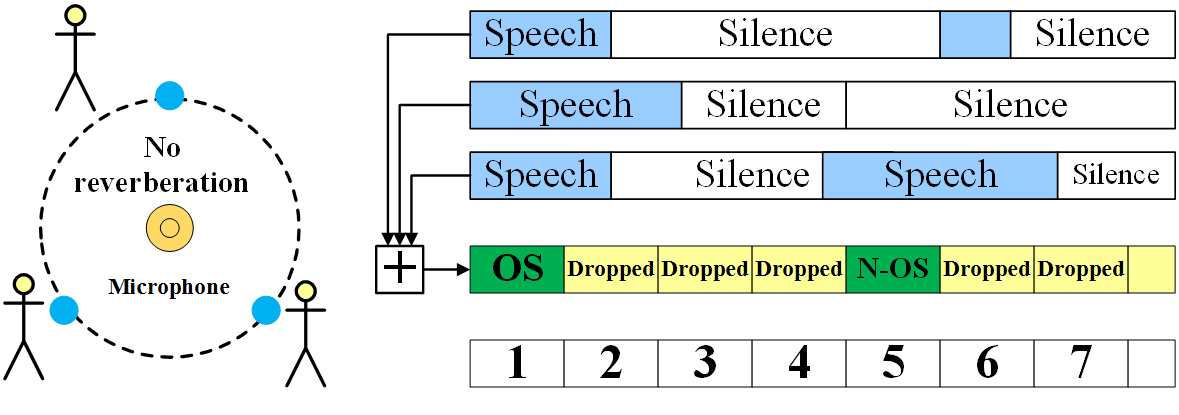


Fig. 6. Labeling speech mixtures frames

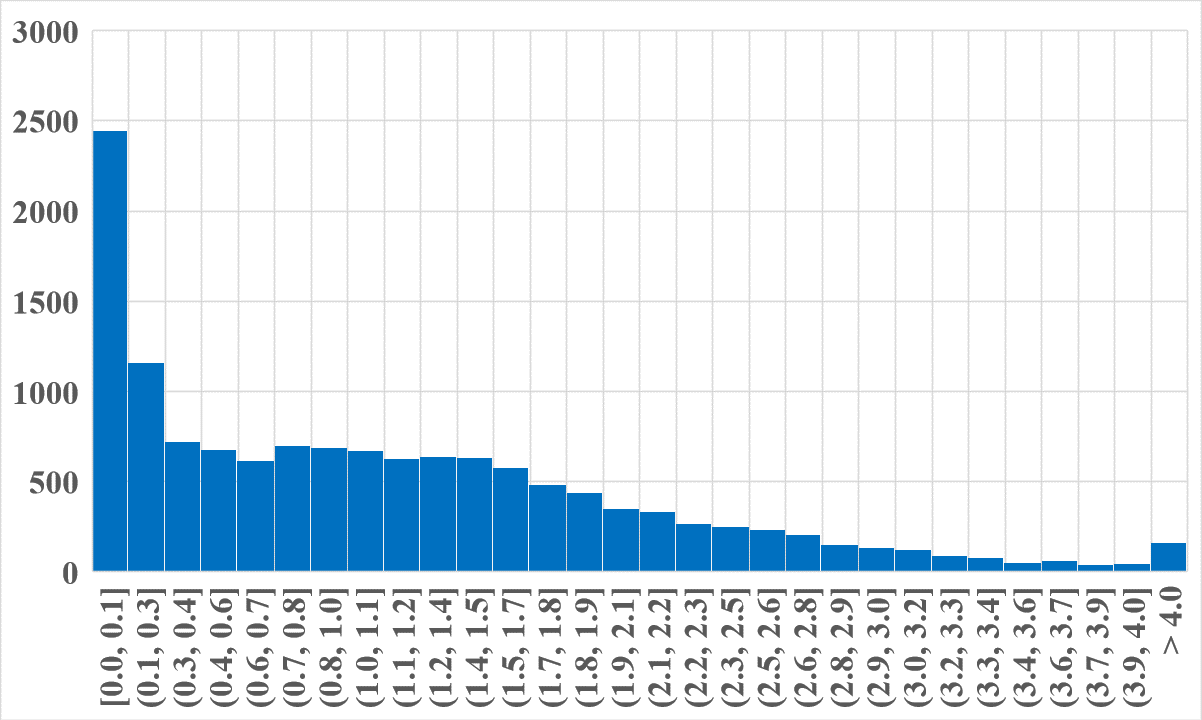


Fig. 7. Histogram with lengths of continuous speech computed on Librispeech test-clean dataset

# 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 safely assumed that the “amount” of speech is equally distributed throughout the entire recording for all speakers. However, in [19] and also in the current study we are targeting sub-second frame lengths and this raises different challenges.

## 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]. References [22], [55] and [57] also use artificial mixtures, and therefore, we decided to do the same. This 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 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 language can influence the experiment’s outcome.

## 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 audio. 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...

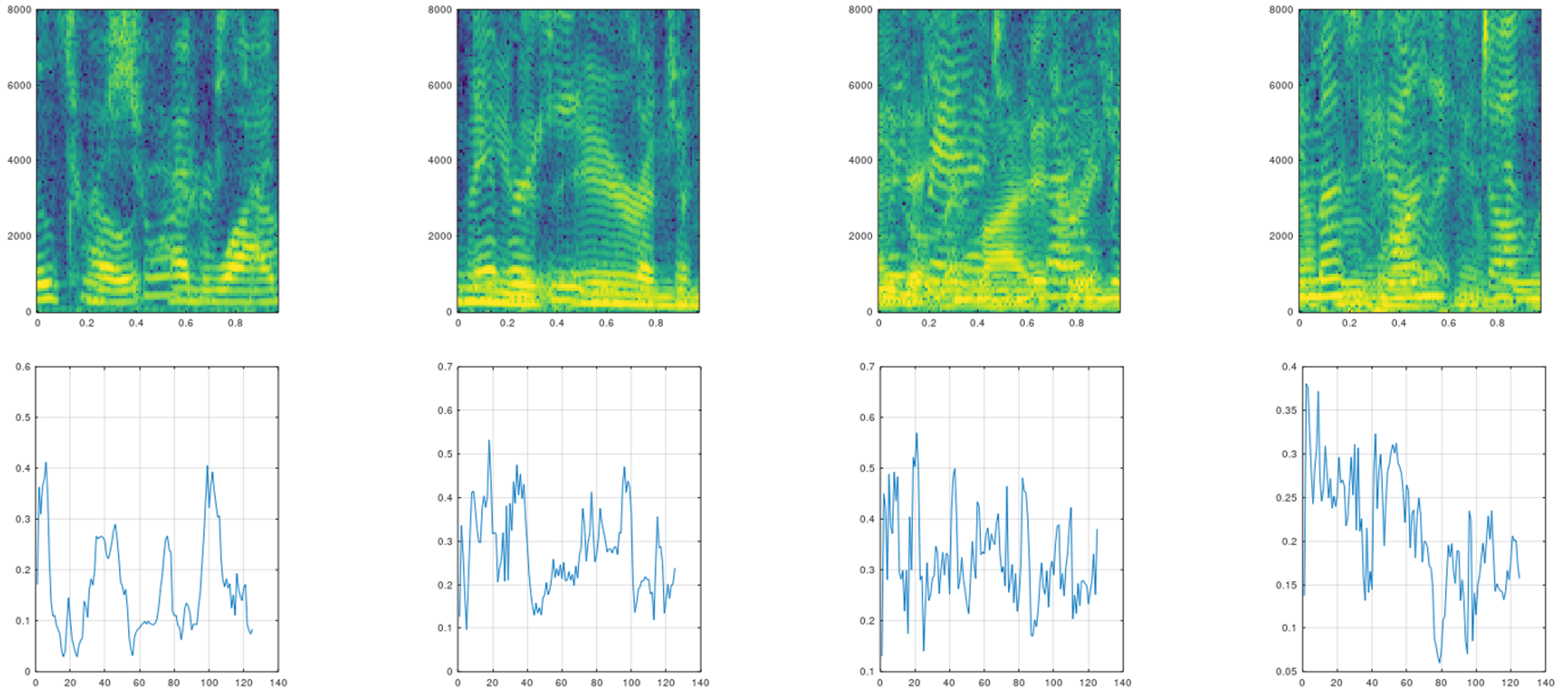


Fig. 8. Spectrogram and speech signal envelope examples for 1, 2, 3 and 4 simultaneously active speakers

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.

# 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 determine 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, 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 feature 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 can examine 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.

# Model architectures

When deciding on the model architecture, we were guided by the belief that if we are correct when selecting the type of 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, and that can spawn thousands of combinations, prior work and intuition played an important role in defining the model.

## 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 data 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] and [55] CNN’s are used 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] and [57]. 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] and [55], 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.

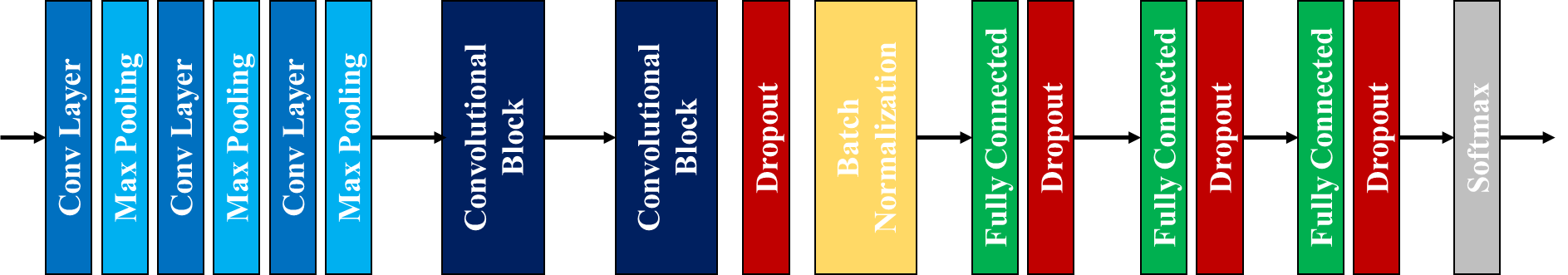


Fig. 9. Architecture of the CNN used for speaker counting. Convolutional blocks are a succession of 3 pairs of convolutional layers and max pooling layers

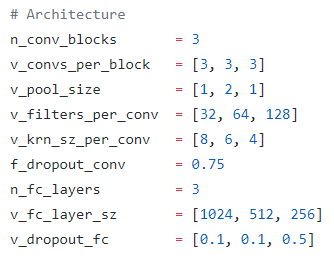


Fig. 10. Configuration section in the model training script

## Speaker counting architecture

Figure 9 illustrates the architecture of the model used for speaker counting. 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 [51] layer, following the convolutional blocks has the role of 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. The scripts used to prepare the datasets, to run training, validation and inference are open-source: *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.

# 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 determines a variable size feature vector for each frame length. In consequence, 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.

## 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 inference performance. For training the models we used *LibriSpeech dev-clean* and for inference we used *LibriSpeech test-clean*.

During training we used Adam algorithm, presented in [52] and implemented by the Keras API, in order to modify the weights. Adam optimizer was selected as it gained traction in the last years and is believed to be more efficient than other approaches at adapting the learning rate. We a learning rate of 0.0005 for all the models. During training, we monitored the loss function and the categorical accuracy. The batch size depended on the model size and the available compute device memory. 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.
* 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.

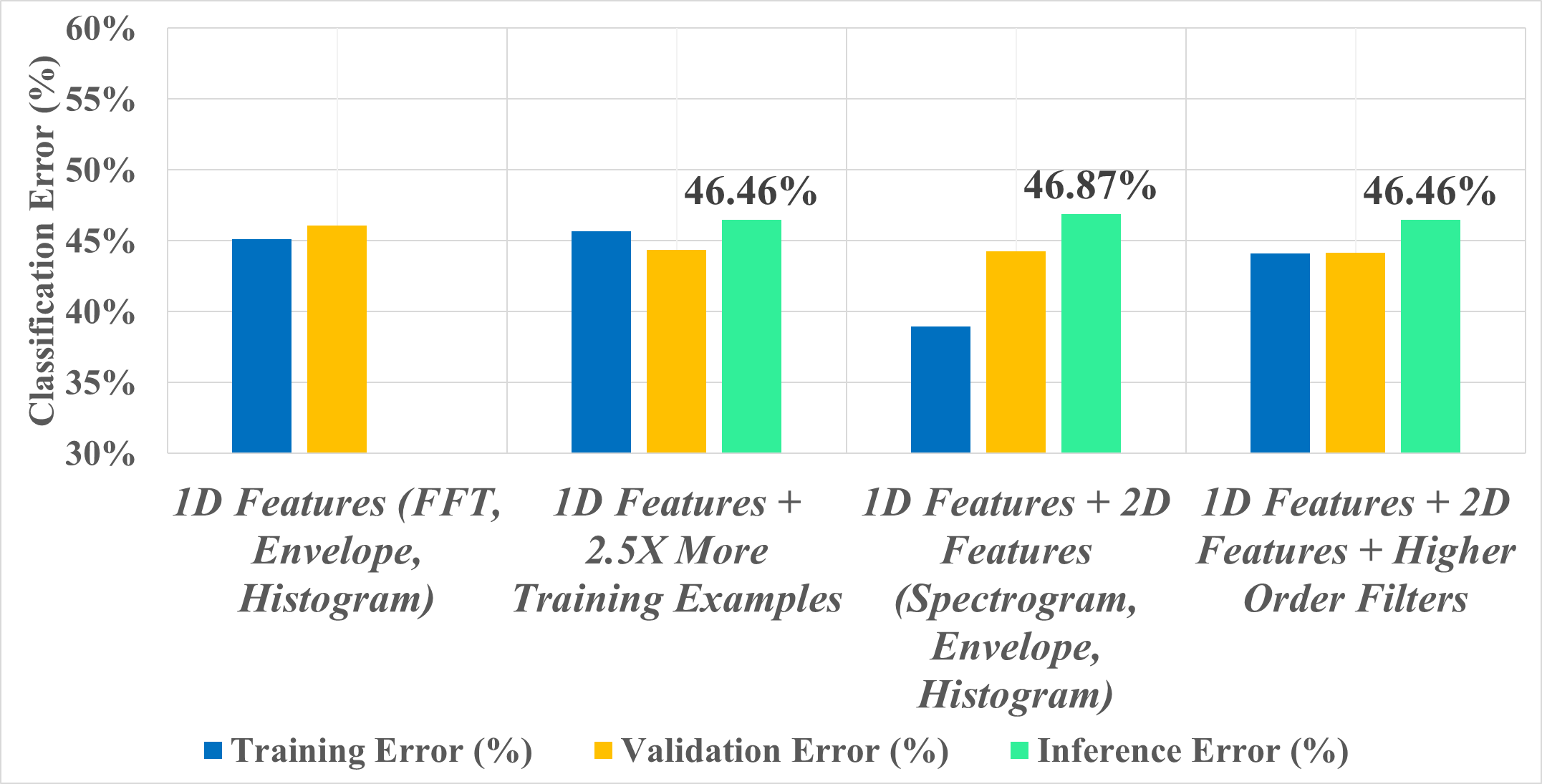


Fig. 11. Training steps for speaker counting on 100ms frames

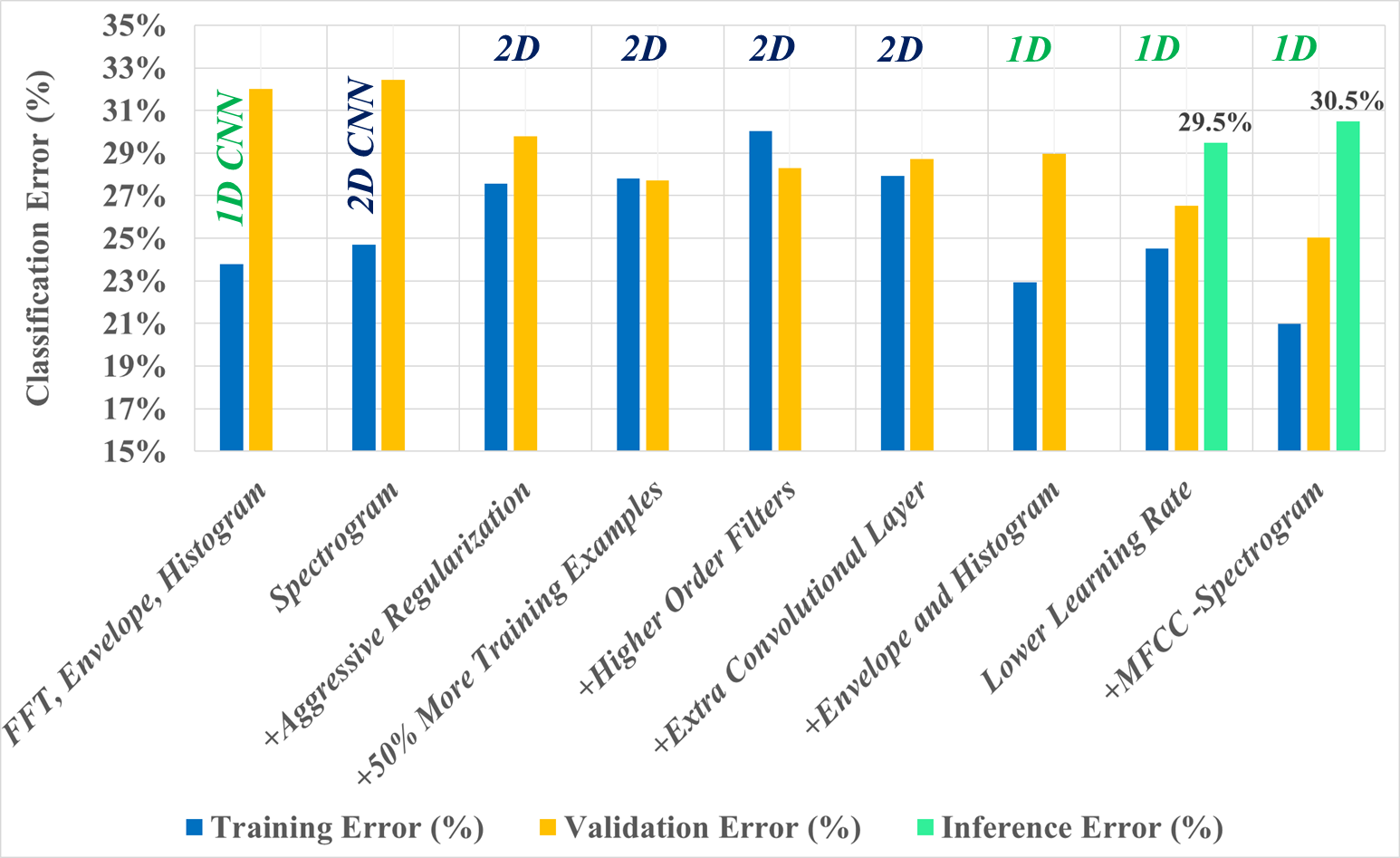


Fig. 12. Training steps for speaker counting on 500ms frames

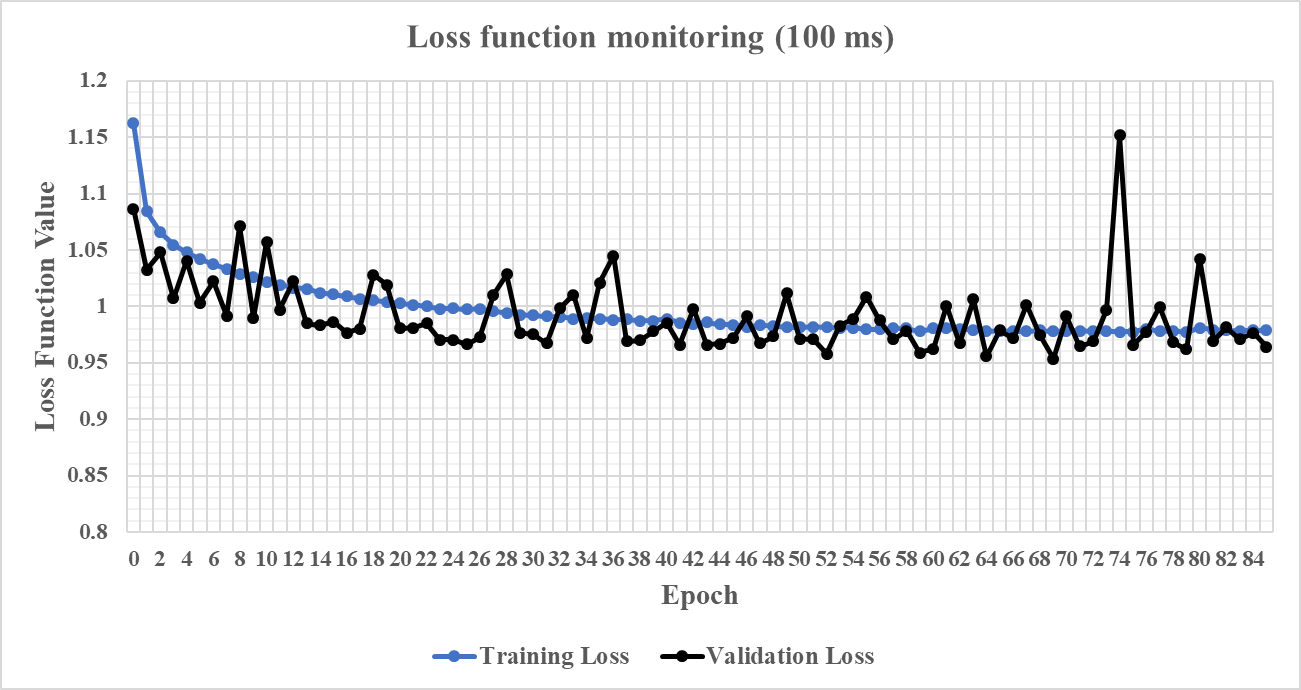


Fig. 13. Loss function decrease for 100ms model

* 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.

## Training steps for 100ms and 500ms speaker counting

Figures 11, 12 and 13 are illustrating the training steps we took in order to converge the models that 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 in Fig. 11, we used CNNs 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 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 lowest inference time as the final one.

In Fig. 13 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. Compared to the performance achieved by human listeners the CNN 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 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 used 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 human level performance for this isolated task, as described in section II.

# 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 timeframes. Once the models were trained, we ran inference using 100000 samples created from a new dataset, in our case *LibriSpeech test-clean*. This means that the inference was realized with a new set of speakers. The same models trained for speaker counting were also used to detect overlapped speech

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 |

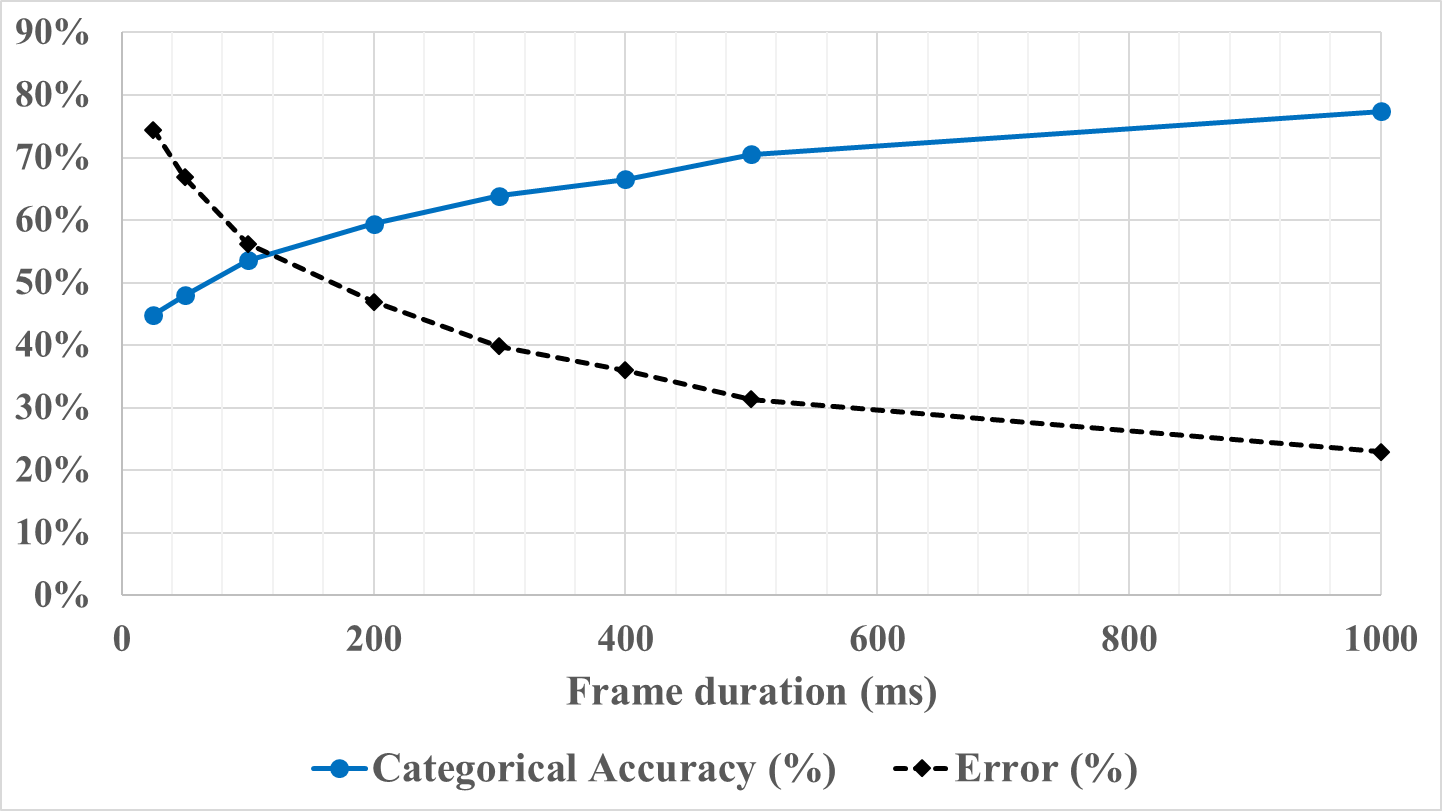


Fig. 14. Speaker counting accuracy as function of frame length

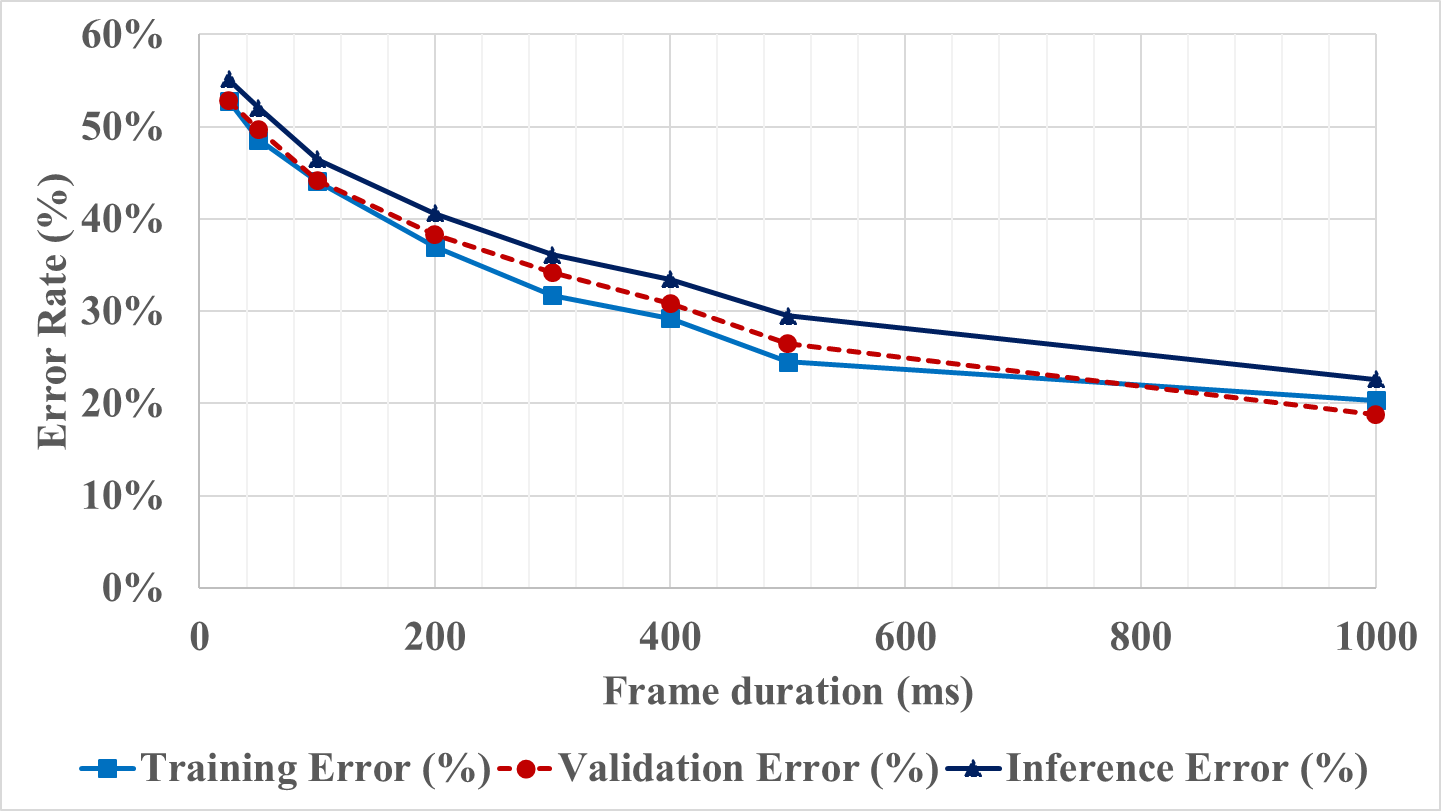


Fig. 15. Training, validation and inference errors for speaker counting

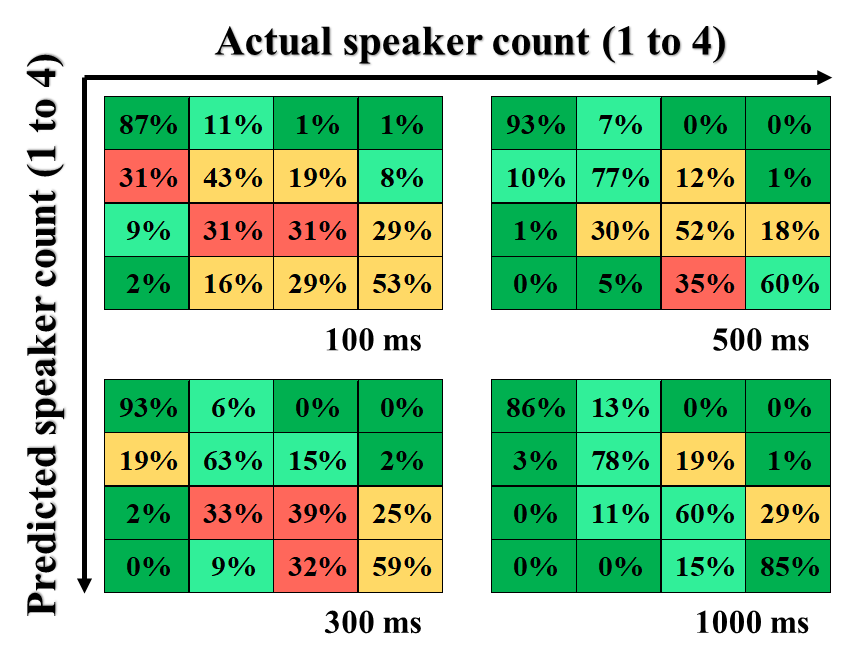


Fig. 16. Confusion matrices for speaker counting. Columns represent the actual speaker count, while the rows represent the predicted count.

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.

## Speaker counting

Figures 14 and 15 are probably the most important results of this work. We can see in Fig. 14 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 15 shows a relatively close error between training, validation and inference datasets which gives us confidence that the models have a good generalization capacity.

In Fig. 16 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] and [57] 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.

We think the strength of our approach relies in the fact that we trained a new model for each timeframe, 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 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. This can be performed efficiently by a VAD.

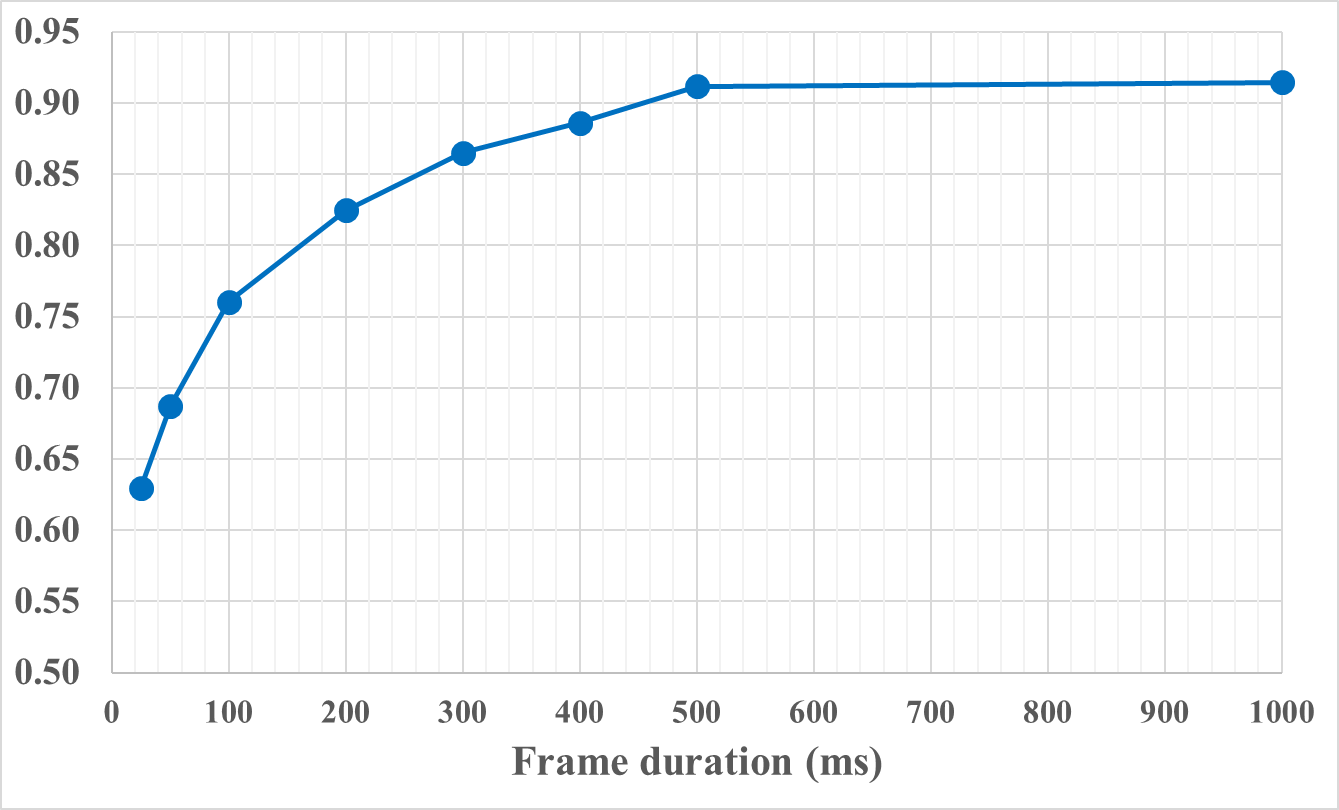


Fig. 17. Overlapped speech detection F-Score depending on frame duration

## 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 speech corpus for training and inference, that was acquisitioned in-house.

To compute how the models trained for speaker counting would fare at overlapped speech detection, we proceeded as follows: 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 maximum sources.

Figure 17 illustrates the inference results by varying frame length. For 25ms of speech, the obtained F-Score is 0.63 which is lower than what we reported in [19]. This is because in [19] the source count is lower and 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, the performance of the models at overlapped speech detection is very appealing with F-Scores 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 presented in [19]. This could be influenced by using the spectrogram instead of MFCC and also by using a much larger speech corpus for training.

# 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 a speaker counting inference accuracy which is better than scores published in current literature. We think this is likely due to our approach of training a new model for each targeted timeframe. 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 – Proc. 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 – Proc. 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, 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., 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
57. F-R. Stoter, S. Chakrabarty, B. Edler, E. A. P. Habets, “CountNet: Estimating the Number of Concurrent Speakers Using Supervised Learning”, In IEEE/ACM Transactions on Audio, Speech, and Language Processing, October 2018

1. 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.m.andrei@gmail.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. [↑](#footnote-ref-1)