Gimeno *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2020) 2020:5
https://doi.org/10.1186/s13636-020-00172-6

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Multiclass audio segmentation based on recurrent neural networks for broadcast domain data

Pablo Gimeno*, Ignacio Viñals, Alfonso Ortega, Antonio Miguel and Eduardo Lleida

Abstract

This paper presents a new approach based on recurrent neural networks (RNN) to the multiclass audio segmentation task whose goal is to classify an audio signal as speech, music, noise or a combination of these. The proposed system is based on the use of bidirectional long short-term Memory (BLSTM) networks to model temporal dependencies in the signal. The RNN is complemented by a resegmentation module, gaining long term stability by means of the tied state concept in hidden Markov models. We explore different neural architectures introducing temporal pooling layers to reduce the neural network output sampling rate. Our findings show that removing redundant temporal information is beneficial for the segmentation system showing a relative improvement close to 5%. Furthermore, this solution does not increase the number of parameters of the model and reduces the number of operations per second, allowing our system to achieve a real-time factor below 0.04 if running on CPU and below 0.03 if running on GPU. This new architecture combined with a data-agnostic data augmentation technique called mixup allows our system to achieve competitive results in both the Albayzín 2010 and 2012 evaluation datasets, presenting a relative improvement of 19.72% and 5.35% compared to the best results found in the literature for these databases.

Keywords: Audio segmentation, Recurrent neural networks (RNN), LSTM, Broadcast data

1 Introduction

In the last few years, there has been a huge increase in the generation of multimedia content and large audiovisual repositories. That is the reason why automatic systems that can analyse, index and retrieve information in a fast and accurate way are becoming more and more relevant. In this context, audio segmentation systems are introduced. The main purpose of audio segmentation is to obtain a set of labels in order to separate an audio signal into homogeneous regions and classify them into a predefined set of classes, e.g., speech, music or noise.

This definition is really wide, and it includes several kinds of systems depending on the classes taken into account in the classification. For example, a speech activity detector (SAD) is considered if a binary speech/non-speech segmentation is performed. Another example of audio segmentation is the speaker diarisation task, that aims to separate different speakers in an audio stream defining one class per speaker. In this paper, we focus on a more generic segmentation task where our objective is to classify the audio signal as speech, music, noise or a combination of these. Other speech technologies applications such as automatic speech recognition (ASR) or speaker recognition can benefit from audio segmentation as a first pre-processing stage improving its performance in real-world environments by being provided with an accurate labelling of audio signals.

This works presents an approach to the multiclass audio segmentation task based on the use of recurrent neural networks, namely the well-known long short-term memory (LSTM) networks. Furthermore, we introduce a new block in the neural architecture seeking to remove redundant temporal information, reducing the number of operations per second and without increasing the number of parameters of the model.

The remainder of the paper is organised as follows: a review on some of the previous work on audio segmentation is done in Section 2. Our proposed RNN-based

*Correspondence: pablogj@unizar.es
ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, C/ Pedro Cerbuna 12, 50009 Zaragoza, Spain

audio segmentation system is described in Section 3. The database description and the metrics used to evaluate our results are exposed in Section 4. The experimental results obtained with our system are explained in Section 5. Finally, a summary and conclusions are presented in Section 6.

2 Previous work

2.1 Audio segmentation approaches

A generic audio segmentation system comprises two different steps: the feature extraction method and the segmentation and classification strategy. A wide review of the features and the segmentation methods applied in the literature is provided by Theodorou in [1].

Feature extraction is the first step in an audio segmentation system. The correct representation of the different acoustic classes is influenced by the set of acoustic features used in the system, both in frequency and time domains. If we focus on the time span they represent, features can be classified into frame-based and segment-based features. On the one hand, frame-based features represent short periods of time (10 to 30 ms) and are usually considered in speech related tasks, where each short segment can be considered to be stationary over that frame. Traditional Mel Frequency Cepstrum Coefficients (MFCC) or Perceptual Linear Prediction (PLP) are proposed in a great amount of works [2–4]. A musical approach to feature extraction is proposed in [5], where timbre, rhythm, pitch and tonality coefficients are computed for each frame. On the other hand, segment-based features are computed over longer periods of time (0.5 to 5 s). For example, in [6] segment features are extracted by fitting frame-based features to a reference model using a histogram equalisation transformation. The variation of the spectrum flux and the variation of the zero crossing rate are proposed in [7] as segment-based features. In [8], the zero crossing rate averaged in segments of 2 s is used to discriminate speech and music in radio recordings.

Once feature vectors are extracted, the next step is dealing with the detection and classification of the segments. Depending on how the segmentation is performed, audio segmentation systems can be classified into two main groups: the segmentation and classification approach and the segmentation by classification approach. In the following lines, both of them are explained:

- *Segmentation and classification*: In this group of systems, segmentation is performed in two separate steps. First, class boundaries are detected using a distance metric and then each delimited segment is classified in a second step. This approach is also known as distance-based segmentation in the literature. One of the best known distance metric is the Bayesian Information Criterion (BIC). For example, [9] applies a BIC-based segmentation to generate a sequence of language-dependent segments. In [10], BIC is used to generate a break point for every speaker. We can cite additional examples of distance metrics used in the literature such as the generalised likelihood tatio (GLR), that is computed to segment speaker and music information in real time in [11], or the cosine distance, combined with a local self-similarity analysis in [12] to detect speech/music transitions. Recent work has introduced the use of neural distance metrics, like the one proposed in [13] for speaker change detection.
- *Segmentation by classification*: in this approach the segmentation is performed by classifying consecutive fixed length audio segments, so that labels are produced directly by the classifier as a sequence of decisions. A set of well-known classification techniques have been developed for this task. A Gaussian Mixture Model (GMM) approach is proposed in [14] together with a maximum entropy classifier. In [15], authors use support vector machines (SVM) to separate speech and music in radiophonic audio streams. Multistage decision trees are used in [16] with the same objective of discriminating speech and music. The factor analysis (FA) technique, usually applied in speaker verification, is adapted to audio segmentation domain by Castán et al. in [17] obtaining relevant results for broadcast domain data.

In both approaches to audio segmentation, original segmentation boundaries are usually refined applying a resegmentation model. This is done to prevent sudden changes in segmentation labels. Some resegmentation strategies rely on hidden Markov models (HMM), like in [18] where different features from a previous stage are combined using an ergodic HMM to produce the final segmentation hypothesis. A less sophisticated solution is proposed in [19] where transitions happening in fragments smaller than 1 s are filtered.

2.2 Neural networks in audio segmentation

Recently, scientific community has experienced the exponential growth of neural networks due to the advances in hardware and the higher availability of data for training. Since the early 2010s, they are being increasingly applied to speech technologies. For example, several works focusing on acoustic modelling for ASR have been presented [20, 21]. In speaker recognition, the use of deep neural network-based representations is an active research topic [22, 23]. If we focus on the audio segmentation task, various authors have already proposed a segmentation by classification system based on neural networks. Concerning feed forward networks, some examples can be cited: a multilayer perceptron is trained using genetic

algorithms in [24] to perform a multiclass audio segmentation. Meinedo presents in [25] a multilayer perceptron-based classifier integrated in a system that combines SAD, speaker segmentation and clustering. More recently, a SAD for broadcast domain is proposed in [26] using the same multilayer perceptron model. Convolutional neural networks (CNN), commonly related to image recognition and image classification applications, are also being applied to audio segmentation and classification. These implementations usually rely on time-frequency representations of the audio signal that are treated as channels in an analogy with image processing systems. This is done, for example, in [27], where a database consisting of different environmental sounds is classified using CNNs extracting mel-spectograms from the input signals. In [28], CNNs are applied to a speech/music segmentation task showing a significant improvement over SVM and GMM systems. A recent work has also applied Mel-based convolutional kernels to the music detection task [29].

Recurrent neural networks are significantly useful when dealing with temporal sequences of information because they are able to model temporal dependencies introducing a feedback loop between the input and the output of the neural network. The long short-term memory (LSTM) networks [30] are a special kind of RNN that introduces the concept of memory cell. This cell is able to learn, retain and forget [31] information in long dependencies. This capability makes LSTM networks a very powerful tool to carry out long- and short-term simultaneously. Two LSTM networks are combined in a bidirectional LSTM (BLSTM) network to model causal and anticausal dependencies. One of them processes the sequence in the forward direction, while the other one processes the sequence backwards. Both LSTM and BLSTM networks have been successfully applied in speech technologies in several sequence modelling tasks such as ASR [32], language modelling [33] or speaker verification [34].

Concerning audio segmentation tasks, most of the LSTM-based systems deal with speech/non-speech classification. The first approach to SAD using LSTM network was presented in [35], where authors demonstrated the feasibility of this approach in a real-life noisy speech from Hollywood movies. A similar neural architecture is used in [36] to implement a noise-robust vowel-based SAD. A more recent paper combines both speech and music detection using recurrent LSTM networks [37]. Our latest work in SAD, used in the first DIHARD diarisation challenge [38] and the Albayzín 2018 diarisation challenge [39], is also based on a BLSTM classifier. Our proposal ranked in 4th position out of 10 participant teams in track 2 [40], and 1st out of 6 participant teams in the closed condition set [41], respectively.

2.3 Audio segmentation challenges

The field of music information retrieval has actively contributed to the improvement of audio segmentation research. The Music Information Retrieval Evaluation eXchange (MIREX) [42] is a set of music-related technological evaluations hosted annually since 2005. The different tasks proposed have changed since its first edition. Here, we cite some of the proposed in the most recent MIREX 2019: audio fingerprinting, audio classification, chord estimation or cover song identification. Concerning audio segmentation, MIREX 2018 proposed a music and/or speech detection task consisting in finding segments of music and speech in a signal. Several subtasks were considered for this task, ranging from separate detection of music or speech, a combination of both or relative loudness estimation of music. The winners of the music detection task used an approach based on a CNN combined with some rule-based smoothing techniques [43]. For the music and speech detection task, the best performing system fused different features using a multilayer perceptron model [44]. It must be mentioned that the use of external data sources for training is allowed in the MIREX evaluations.

It is also interesting to mention the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge that aims to advance research in the recognition of sound scenes and individual sound sources in realistic soundscapes. The most recent version (DCASE2019) [45] proposed five different tasks in this domain. In the sound event localisation and detection task, direction-of-arrival (DOA) estimation and audio segmentation are combined. Eleven different acoustic classes were considered such as speech, car noise or dog barks. State of the art results are obtained with a consecutive ensemble of convolutional recurrent neural networks (CRNNs) [46]. The sound event detection in domestic environment task is the closest to the task proposed in this paper, where starting and ending times must be given for each class in the audio signal. The best performing system in this task used a neural architecture based on CNNs, implementing an attention mechanism to extract an acoustic embedding used then for classification in a teacher-student framework [47]. For the two tasks presented here, the use of external data was not allowed for the development of the submitted systems.

Dealing with broadcast domain content can be challenging because such documents contain different audio sequences with a very heterogeneous style. Several speech conditions and domains can be found, from studio recordings to outdoor speech to telephonic quality, with several noises overlapped in all cases. A variety of acoustic effects and background music are also likely to appear. Many international evaluation campaigns have already faced the challenge of working with broadcast data: the ESTER

evaluations campaigns [48] aimed to evaluate automatic broadcast news rich transcriptions system for the French language proposing a transcription task, a segmentation task and an information extraction task. In the COST278 multilingual evaluation [49] for segmentation and speaker clustering, a database consisting of broadcast news from 10 European countries was used.

Concerning the Iberian languages, the Albayzín international campaigns have proposed a wide range of challenges from speech transcription to spoken term detection. The audio segmentation task was first introduced in the Albayzín evaluations in 2010 for a broadcast news environment [50]. The objective was to segment an audio signal into five classes: speech, music, speech with noise, speech with music and others. In this context, our paper is focused on this task, aiming to incorporate recurrent neural networks (RNN) as the main component of the segmentation system.

More recent Albayzín challenge corpora was released in 2012, namely the CARTV dataset [51], with around 20 h of audio from Aragón radio archive. The annotation for this dataset is slightly different from the one proposed in the Albayzín 2010 as labels can be music, speech and noise. Furthermore, the overlap of two or more of them is allowed. Taking into account this differences and the fact that data is coming from a different acoustic domain, we report the results of our final system on this dataset too, in order to explore the generalisation capabilities of our proposal.

3 System proposal

Proven the performance of LSTM networks in a binary classification task like SAD [35, 36], our previous work in audio segmentation aimed to replicate these results in a multiclass environment, like the one proposed in the Albayzín 2010 audio segmentation evaluation. Preliminary results in [52] have shown the feasibility of LSTMs in this task. That is why in this paper we aim to explore the behaviour of recurrent neural architectures for this task in a wider sense, including improvements on our previous results, both in performance and in computational complexity. Our proposal to enhance the neural architecture is the incorporation of a new block that we named "Combination and Pooling" block, that is described in the following subsections.

Furthermore, as several works have shown the improvement in performance obtained when using data augmentation techniques in audio classification [53, 54], we aim to incorporate this kind of techniques in our system too. Due to the data restriction imposed by the Albayzín 2010 evaluation, no further data can be used in training in order to be comparable with other systems. In this context, we introduce the mixup augmentation [55], which combines linearly pairs of examples to generate new virtual examples. In this work, we also explore the behaviour of this technique in our segmentation system.

We propose an RNN-based segmentation by classification system. Our approach combines the modelling capabilities of BLSTM networks with a resegmentation module to get smoothed segmentation hypothesis. In the following lines, we describe the three different blocks our system comprises: feature extraction, an RNN-based classifier and the final resegmentation module.

3.1 Feature extraction

Concerning the feature extraction, we are using a frame-based approach where we combine a traditional perceptual set of features with some musical theory motivated features that may help our system discriminate classes that contain music. In a preliminary preprocessing step, the audio is resampled at 16 kHz and converted to a single channel input. Then, log Mel filter bank energies and the log energy of each frame are extracted. Considering an audio input sampled at 16 kHz, Mel filters span across the frequency range in between 64 Hz and 8 kHz.

Additionally, Mel features are combined with chroma features [56]. These features are a time-frequency representation specially suited for music where the entire spectrum is projected onto 12 bins representing the 12 distinct semitones of the chromatic musical scale. Due to its robustness to variations in tone or instrumentation, combined with its capability to capture melodic and harmonic information, chroma features have been applied in different musical information retrieval applications. For example, chroma are usually extracted in most chord recognition applications [57, 58]. Chroma features are extracted using the openSMILE toolkit [59].

All features are computed every 10 ms using a 25 ms Hamming window. First- and second-order derivatives of the features are computed using 2 FIR filters of order 8 to take into account the dynamic information in the audio signal. Finally, feature mean and variance normalisation are applied at recording level.

3.2 Recurrent neural network

The central idea of our proposed segmentation system is the use of RNNs as the classification algorithm in a segmentation by classification approach. We propose two different variations of this RNN classifier: the first one, already proposed in [52] and that is going to be considered as our baseline system, is described in Fig. 1. As shown, the neural architecture is mainly composed by one or more stacked BLSTM layers with 256 neurons each. The outputs of the last BLSTM layer are then independently classified by a linear perceptron sharing its values (weights and bias) among all time steps.

The details of the proposed "Combination and Pooling" block are presented in Fig. 2. The main idea behind it is

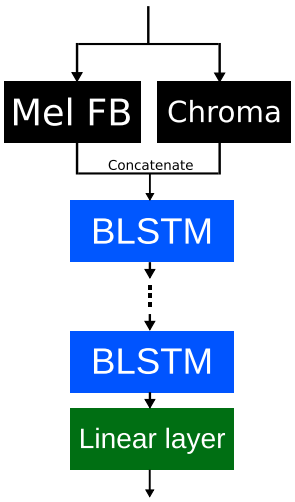

Fig. 1 BLSTM-based neural architecture description for the baseline RNN classifier

to reduce the redundant temporal information through a time pooling mechanism, while at the same time, a more appropriate representation is learned through a 1D convolutional layer. Furthermore, we propose three different variations of this block. The first one combines both the temporal pooling and the 1D convolutional layer, while the other two variations only use time pooling or a 1D convolutional layer respectively, to evaluate its separate influence on the system too.

In Figure 3, we present the second approach to the RNN classifier, incorporating our proposed "Combination and Pooling" block. The linear layer is configured in the same conditions as in the baseline system.

All systems have been trained and evaluated using finite length sequences (3 s, 300 frames), limiting the delay of dependencies that the network may take into account. These sequences have a length of 3 s with an advance of 2.5 s, thus 0.5 s are overlapped. In order to generate the final prediction, the first half of this overlapped part is taken from the previous window, and the second half is taken from the next window. This way the labels corresponding to the boundaries of each fragment are discarded as they may not be reliable. However, the neural network emits a segmentation abel for each frame processed at the input for the first system, and one segmentation label for each N frames processed when using the pooling setup, being N the temporal pooling factor applied.

The neural networks are trained using adaptive moment estimation (Adam) optimiser due to its fast convergence properties [60]. Also, cross entropy criterion is chosen as loss function, as usually done in multiclass classification tasks. Data are shuffled in each training iteration seeking to improve model generalisation capabilities. All the neural architectures in this paper have been evaluated using the PyTorch toolkit [61].

In addition to the emitted RNN segmentation labels, we are also considering the final linear layer scores for each class in order to perform the resegmentation step. In our results, we evaluate two different points in our system: the neural network output by using the RNN-emitted labels and the final labels produced by the resegmentation module.

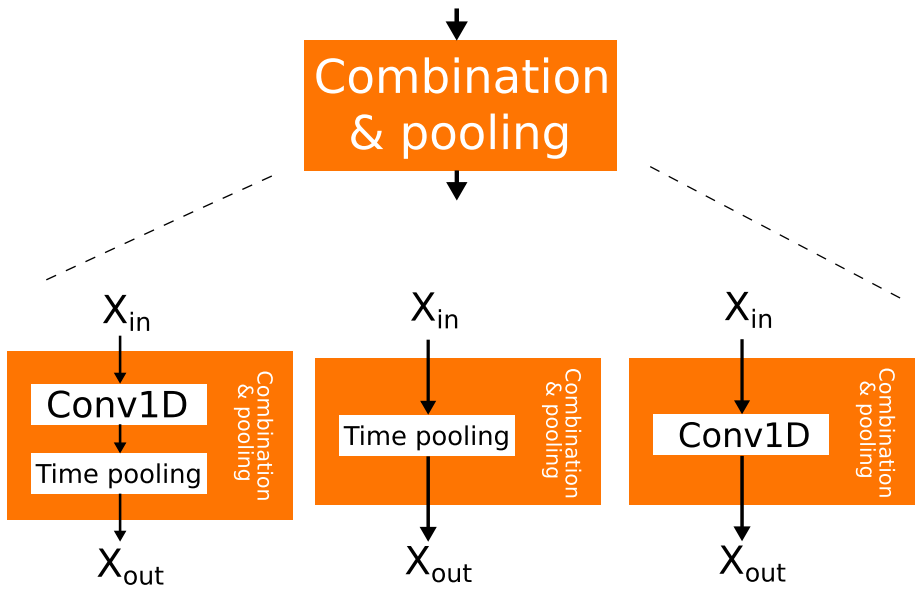

Fig. 2 Description of the proposed "Combination and Pooling" block and its three variations in this study

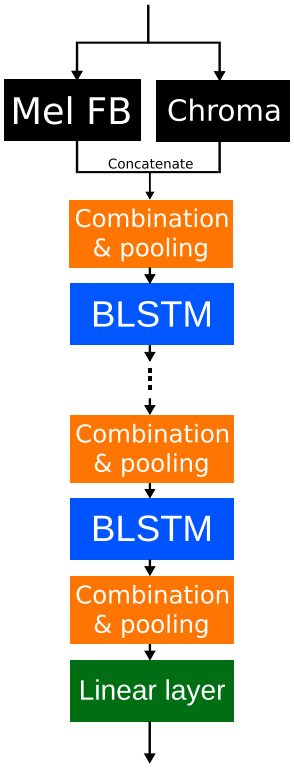

Fig. 3 Alternative BLSTM-based neural architectures including the proposed combination and pooling block

3.3 Resegmentation module

The RNN output may contain high-frequency transitions which are unlikely to occur in highly temporal correlated signals such as human speech or music. Aiming to avoid spurious changes in the segmentation hypothesis, we incorporate a resegmentation module in our system. Our implementation is based on an ergodic hidden Markov model where each class is modelled trough a state in the Markov chain. Every state is represented by a multivariate Gaussian distribution with full covariance matrix. No a priori information is required for this block to be fully functional because statistical distributions are estimated using the labels hypothesised by the RNN for each file in the database.

The neural network output may result in a noisy estimation of class boundaries. Aiming to avoid high-frequency transitions, neural network scores are downsampled by a factor L using an L order averaging filter. This filter is implemented as a zero-phase FIR filter [62] to avoid delays in the output signal. Moreover, each state in the Markov chain consists of a left-to-right topology of N_{ts} tied states that share the same statistical distribution. These two strategies allow us to impose a certain amount of inertia in the output, forcing a minimum segment length before a class change occurs. This length can be computed as follows:

$$T_{\min} = T_s L N_{ts} \tag{1}$$

where T_s is the neural network output sampling period, L is the downsampling factor, and N_{ts} is the number of tied states per state in the Markov chain.

4 Experimental setup

The experimental setup for our experiments is based on the proposed originally in the Albayzín 2010 audio segmentation evaluation. A complete description of the task, results obtained by the participants and a description of the different approaches used can be found in [50]. In the following lines we describe the database used in the evaluation and the metrics taken into account in our experiments. We also present the CARTV database introduced in the Albayzín 2012 evaluation [51], and that is used in the final part of our experimentation to check the generalisation capabilities of our proposal.

4.1 Databases description

As it has been explained previously, the Albayzín evaluation campaigns have proposed several speech and audio processing related tasks in the last decade. Audio segmentation is one of the proposed tasks from 2010 to 2014, focusing mainly on separating speech, music and noise. For this purpose, two datasets were released: the 3/24 TV dataset, released for the 2010 evaluation and coming from broadcast television domain, and the CARTV dataset, used in the 2012 evaluation and obtained from radio recordings. They share a set of common characteristics and some minor differences that are explained in this subsection.

The Albayzín 2010 database is part of a set of broadcast news recordings broadcast originally in 2009 by the Catalan TV channel 3/24 TV. Data was originally collected by the TALP Research Centre from the Universitat Politècnica de Catalunya. The full database includes around 87 h of manually annotated audio sampled at 16 kHz and it is divided in 24 files of around 4-h length each. Two thirds of the database are available for training, making a total of 58 h in 16 different sessions, while the remaining third, 29 h in 8 sessions, is used for test purposes. Furthermore, we reserve 15% of training subset for training validation, which translates to a total of 49 h of audio for training and 9 h for validation. The evaluation plan defines five different acoustic classes distributed as follows: 37% for clean speech (sp), 5% for music (mu), 15% for speech over music (sm), 40% for speech over noise (sn) and 3% for others (ot). The class "others" is not evaluated in the final scoring.

In the following lines we describe each of the acoustic classes defined: the class "speech" contains speech

under studio recording conditions using a close microphone. The "music" class contains music understood in a general sense. The "speech over music" class is defined as the overlap of classes "speech" and "music". The "speech over noise" class contains all the speech that is not recorded under studio conditions or overlapped with any kind of noise. Two voices overlapping are also defined as "speech over noise". Finally, "others" contains any audio that does not match the four previously defined classes.

As it can be appreciated, there is a clear unbalance in the class distribution because most of the data contains speech (92% combining speech, speech over noise and speech over music). However, classes that contain music are underrepresented (only 20% combining music and speech over music). The main language of the 3/24 TV channel is Catalan, with the 87% of the speech segments coming from Catalan speakers and the remaining 13% coming from Spanish speakers. Concerning the gender distribution, 63% of speech fragments are from male speakers and 37% are from female speakers.

Additionally, we describe the data released in the Albayzín 2012 audio segmentation evaluation [51] that is used in the final part of our experimentation to check the generalisation capabilities of our proposal. The new audio introduced in this version is taken from Aragón radio archive, separated in 3 different subsets: two developments sets of 2 h each (dev1 and dev2), and a test set consisting of 18 h. All the audio is sampled at 16 kHz. The use of previous data released in the 2010 version was allowed in order to train the segmentation systems for the 2012 Albayzín evaluation.

As in the 2010 Albayzín evaluation, the main goal is segmenting an audio document indicating where speech, music and/or noise is present. However, in the 2012 version, no prior classes are defined and a multiple layer labelling is proposed, allowing 3 possible overlapped classes, speech, music and noise, to be present at any time in the audio document. This format slightly differs with the experimental setup presented for the Albayzín 2010 evaluation, but it is equivalent to a multiclass segmentation task if a set of non-overlapped labels are generated considering the different combinations of speech, music and noise. A more detailed explanation of this process is given in Subsection 5.6.

4.2 Metrics

The main metric used to evaluate our results is the segmentation error rate (SER). This metric is inspired by the diarisation error rate (DER), a metric used in NIST speaker diarisation evaluations [63], and it can be seen as the ratio of the total length of the incorrectly classified audio to the total length of the audio in the reference. Given a dataset to evaluate Ω, each document is divided into continuous segments and the segmentation error time for each segment n is defined as:

$$\Xi(n) = T(n)[\max(N_{\text{ref}(n)}, N_{\text{sys}}(n)) - N_{\text{correct}}(n)] \quad (2)$$

where $T(n)$ is the duration of the segment n, $N_{\text{ref}}(n)$ is the number of reference classes that are present in segment n, $N_{\text{sys}}(n)$ is the number of classes predicted by the system that are present in segment n and $N_{\text{correct}}(n)$ is the number of reference classes that are present in segment n and were correctly assigned by the segmentation system. This way, the SER is computed as follows:

$$\text{SER} = \frac{\sum_{n\in\Omega} \Xi(n)}{\sum_{n\in\Omega}(T(n)N_{\text{ref}}(n))} \quad (3)$$

Additionally, the original metric proposed in the Albayzín 2010 evaluation is considered in our experiments in order to favour the comparison with previous publications. This metric represents the average class error over all the classes. Let C be the set of the five acoustic classes defined in the evaluation, $C = \{\text{mu}, \text{sp}, \text{sm}, \text{sn}, \text{ot}\}$. This way the error metric can be computed according to the following equation:

$$\text{Avg error} = \frac{1}{|C|}\sum_{i\in C}\frac{\text{dur}(\text{miss}_i) + \text{dur}(\text{fa}_i)}{\text{dur}(\text{ref}_i)} \quad (4)$$

where $\text{dur}(\text{miss}_i)$ is the total duration of all miss errors for the ith acoustic class, $\text{dur}(\text{fa}_i)$ is the total duration of all false alarm errors for the ith acoustic class and $\text{dur}(\text{ref}_i)$ is the total duration of the ith acoustic class according to the reference. Using this metric, an incorrectly classified segment computes as a miss error for an acoustic class and a false alarm error for another. Due to the fact that class distribution is clearly unbalanced, errors from different acoustic classes are weighted differently according to the total duration of the class in the database. This metric was originally proposed in the evaluation because, by computing the average of the error over all the acoustic classes, participants are encouraged not to focus only on the best represented classes in the database.

In both metrics, SER and average class error, a collar of $\pm$ 1 second around each reference boundary is not scored in order to avoid uncertainty about when an acoustic class ends or begins, and to consider inconsistent human annotations.

In our final analysis, we also report other metrics traditionally shown for classification tasks such as the overall accuracy, and the precision, recall and F_1 score per class. For the set of classes C defined previously, they can be computed as shown in the following equations:

$$\text{Accuracy} = \frac{1}{|C|}\sum_{i\in C}\frac{\text{tp}_i + \text{tn}_i}{\text{tp}_i + \text{tn}_i + \text{fp}_i + \text{fn}_i} \quad (5)$$

$$\text{Precision}_i = \frac{\text{tp}_i}{\text{tp}_i + \text{fp}_i} \tag{6}$$

$$\text{Recall}_i = \frac{\text{tp}_i}{\text{tp}_i + \text{fn}_i} \tag{7}$$

$$\text{F}_{1i} = 2\frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{8}$$

where tp_i represents the number of true-positive predictions for the class i, tn_i is the number of true-negative predictions for the class i, fp_i is the number of false-positive predictions for the class i, and fn_i is the number of false-negative predictions for the class i.

5 Results

5.1 Feature analysis

Two different kinds of errors can be differentiated in our system: a classification error due to an incorrectly labelled frame, and a segmentation error due to a temporal mismatch between the hypothesis and reference class boundaries. In a first set of experiments only the classification errors are taken into account because ground truth segments are given to the system. A classification label is emitted then as the class maximising the score averaged for the whole ground truth segment. Then, the classification error is computed simply dividing the number of oracle segments incorrectly classified by the total number of oracle segments. In the following set of experiments both segmentation and classification errors are shown as no boundaries are given to the system.

In order to validate experimentally our proposal, different frontend configurations were assessed. Our setup started with a 64 band Mel log filter bank; then, frequency resolution was gradually increased evaluating an RNN classifier with 80 and 96 band Mel log filter bank as input. As explained before, chroma features were also incorporated aiming to discriminate correctly the music classes. Eventually, first- and second-order derivatives were computed to include information about the audio signal dynamics. All these cases use a simple setup consisting of an RNN classifier based on a single BLSTM layer.

In Table 1, we show the classification error obtained with the RNN classifier using oracle boundaries for different frontend configurations.

It can be seen that increasing the frequency resolution leads to a consistent improvement in the classification accuracy while the number of parameters of the model is also increased. However, when incorporating chroma features the improvement obtained is significantly better than the one obtained with a bigger frequency resolution. If we compare the best log Mel filter bank setup (96 bands) with the best one with chroma (80 bands + Chr) it can be seen that, with a similar number of parameters, the error drops significantly. Finally, first- and second-order derivatives are computed, achieving the best result in our experiment at the cost of increasing the number of parameters in the model.

Table 1 Classification error with oracle boundaries for the 1BLSTM RNN classifier on the test partition for different frontend configurations (Chr, chroma; Δ, ΔΔ, 1st and 2nd order derivatives)

Features	Classification error (%)	# Parameters
64 bands	16.22	200K
80 bands	16.20	217K
96 bands	**16.05**	**233K**
64 bands + Chr	**13.40**	**213K**
80 bands + Chr	13.80	229K
96 bands + Chr	14.51	246K
64 bands + Chr + Δ, ΔΔ	13.35	368K
80 bands + Chr + Δ, ΔΔ	**13.34**	**418K**
96 bands + Chr + Δ, ΔΔ	13.37	467K

Additionally, in Table 2, we present the segmentation error rate and the error per class obtained with the RNN classifier. This results now takes into account both classification and segmentation error.

In this case, the best Mel log filterbank configuration is achieved using 80 bands. Increasing the frequency resolution seems to not be so relevant when dealing with the segmentation task compared to the classification one. We can notice that, by incorporating chroma features, the error in the class "Speech over music" and "Speech over noise" decreases significantly when comparing the 64 bands to the same one with chroma, with a relative improvement

Table 2 SER, error per class and average class error for the 1BLSTM RNN classifier on the test partition for different frontend configurations (Chr, chroma; Δ, ΔΔ, 1st and 2nd order derivatives)

Feats	SER	Class error (%)				Avg
		mu	sp	sm	sn	
64 bands	18.18	18.54	32.43	32.48	35.76	29.80
80 bands	**17.70**	**18.19**	**31.33**	**31.41**	**34.91**	**28.96**
96 bands	17.93	20.68	30.84	32.09	34.25	29.46
64 bands + Chr	**16.97**	**18.83**	**30.88**	**29.92**	**32.76**	**28.10**
80 bands + Chr	17.89	19.77	32.23	29.55	33.92	28.87
96 bands + Chr	17.65	19.75	30.68	31.62	33.66	28.93
64 bands + Chr + Δ, ΔΔ	16.61	17.46	29.93	29.26	32.60	27.31
80 bands + Chr + Δ, ΔΔ	**16.25**	**16.82**	**30.00**	**26.75**	**32.07**	**26.41**
96 bands + Chr + Δ, ΔΔ	16.46	17.38	29.92	27.98	32.70	27.00

of 8.55% and 9.15% respectively. This is due to the capabilities of chroma features to capture musical information, which helps our system to discriminate noise and music in a more accurate way. This behaviour is also consistent with the classification accuracy improvement observed when using chroma features in the ground truth boundary experiments. The best result is obtained with the frontend that combines 80 bands, chroma features and the first and second order derivatives with a SER of 16.25%, equivalent to an average class error of 26.41%. Furthermore, it can be observed that including first and second order derivatives shows a greater relative improvement when considering the segmentation errors compared to the case where only classification errors are considered. We can infer then, that the dynamic information incorporated by the 1st and 2nd order derivatives may be more relevant to generate the class boundaries than to the classification task itself.

So far, only an architecture with a single BLSTM layer has been evaluated. In the following experiment our goal is to determine the most appropriate number of BLSTM layers for our system. Choosing the best feature frontend (80 Mel + chroma + 1st and 2nd derivatives), we evaluate now our system stacking two and three BLSTM layers. Results for this experiment are presented in Table 3.

Including 2 stacked BLSTM layers shows a slight relative improvement of around 2.20% compared to the case of using a single BLSTM layer. However, no further improvement is observed when including a third layer. That is why we choose the architecture using 2 BLSTM stacked layers as our baseline in futures experiments. An average class error of 25.84% is obtained, equivalent to a SER of 15.91%. This results are the one we compare against in the following sections to evaluate the different neural architectures proposed. In the following sections, the performance of our full system combining the RNN classifier and the HMM resegmentation module is evaluated with a new set of experiments.

5.2 HMM resegmentation

With the objective of illustrating the influence of the inertia imposed by the resegmentation module on the segmentation system, Fig. 4 shows the scatter plot of the relative improvement in performance using the HMM resegmentation versus the minimum segment length (T_{min}) for different values of the downsampling factor. It can be seen that the best performing configurations have a minimum segment length between 0.5 and 1.5 seconds, values which are in the order of magnitude of the 2 s collar applied in the evaluation. A fast decrease in performance is observed when the minimum segment length is increased for values above 3 s. With such configuration, our system is not able to capture some of the fast transitions happening in the audio; thus, a considerable amount of errors are likely to happen, and the performance is decreased. However, no configuration showed a decrease in performance when compared to the case of not using the HMM resegmentation.

In Table 4, we show the results on the test partition for the full segmentation system that combines the RNN classifier and the HMM resegmentation for the best feature configuration and different values of downsampling factor, L, and minimum segment length, T_{min}.

Compared to the best results in Table 2, it can clearly be seen that the HMM resegmentation reduces significantly the system error by forcing a minimum segment length for the class labels. This error reduction is equivalent to a 21.68% relative improvement in terms of SER for the best configuration. Again, it can be observed that, as long as the T_{min} value stays in the range between 0.5 and 1.5 s, the performance of the system is not highly affected by the variations in the downsampling factor. The SER metric in the four parameters configuration presented in Table 4 varies from 12.46 to 12.57%, an absolute difference of only 0.11% between the best and the worst case. This way, reducing the high frequency transitions through imposing a certain amount of inertia in the neural network output, our segmentation system achieves a SER of 12.46%. This value serves also for comparison in the following lines, where new neural architectures are evaluated.

Table 3 SER, error per class and average class error for the RNN classifier on the test partition for the best frontend configurations and different number of stacked BLSTM layers

Layers	SER	Class error (%)				Avg
		mu	sp	sm	sn	
1 BLSTM	16.25	16.82	30.00	26.75	32.07	26.41
2 BLSTM	**15.91**	**16.28**	**28.82**	**26.32**	**31.94**	**25.84**
3 BLSTM	16.02	15.71	27.46	30.09	30.74	26.00

5.3 Combination and pooling experiments

Our initial experiments using the HMM resegmentation module proved that reducing the temporal resolution of the output is beneficial for the segmentation system. Our goal introducing the "Combination and pooling" block is that this downsampling could be implemented inside the neural network itself.

The temporal pooling layers are configurable via a pooling factor parameter, N, that controls the length of the output sequences compared to the input length. The pooling layers separate an input sequence in N different subsequences with the same length and no overlaping. Then, the output is computed applying a given pooling mechanism for each of these subsequences. In all cases, a single element is returned for every N frames in the input. On the other hand, the 1D convolutional layer is

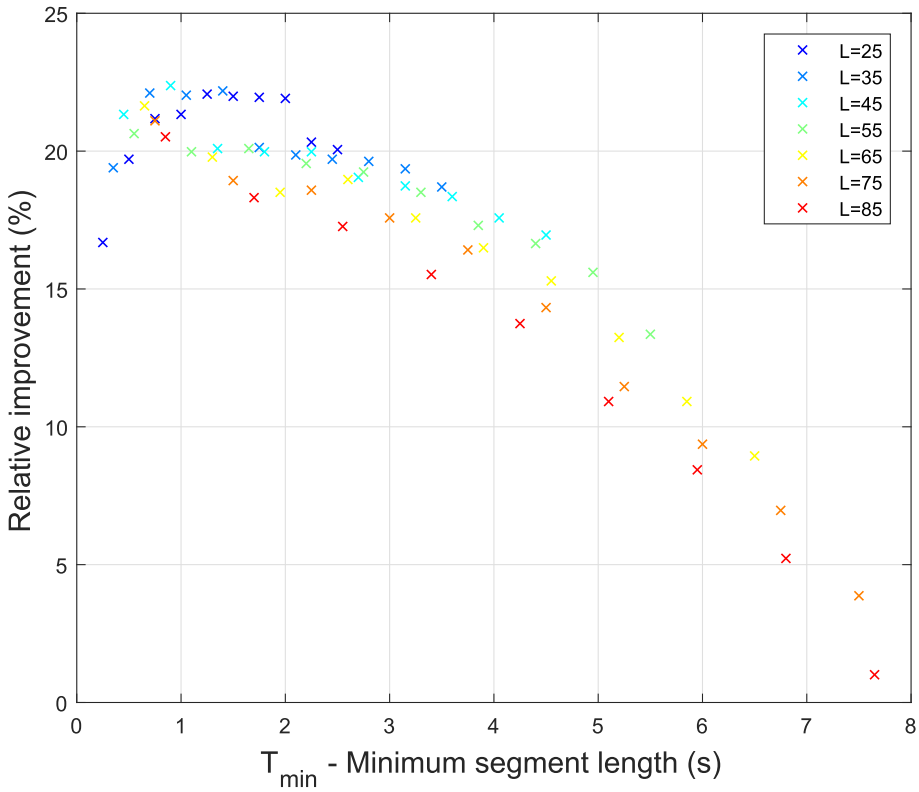

Fig. 4 Relative improvement over the RNN classifier using the HMM resegmentation module for the best feature configuration versus minimum segment length forced by the system

configured to have the same number of input and output channels in all cases, a kernel size of 1 and no padding.

In Table 5, we present the results obtained when using the "Combination and Pooling" block in all its variations before the first BLSTM layer. For this experiment, we consider an average pooling mechanism with a pooling factor of $N = 10$.

Experimental results show that using temporal pooling before the first BLSTM layer strongly degrades the performance of the segmentation system. The degradation is even stronger when first combining the input features using a 1D convolutional layer with a relative degradation of 11.80% compared to the baseline RNN classifier. This may come motivated by an early reduction of the input dimensionality, before the neural network has been able to process any kind of information. Bearing in mind these results we can discard this type of configurations in future experiments.

In the following lines, we present the results for the other two remaining configurations implemented: the one using the "Combination and Pooling" block between the first and second BLSTM layers and the one with the block right after the last BLSTM layer. Figure 5 shows the relative improvement compared to the RNN baseline classifier for the setup using the combination and pooling block between both BLSTM layers and the setup using the combination and pooling block after the last BLSTM layer.

Table 4 SER, error per class and average class error for the RNN classifier combined with the resegmentation module over test partition for the best feature configuration and different values of the down-sampling factor, L, and minimum segment length, T_{min}

L, T_{min}	SER	Class error (%)				Avg
		mu	sp	sm	sn	
25, 1.25 s	12.49	14.55	21.99	19.08	24.88	20.13
35, 1.4 s	12.48	14.31	22.26	18.70	25.10	20.10
45, 0.9 s	**12.46**	**14.19**	**22.14**	**18.82**	**25.04**	**20.05**
55, 0.55 s	12.57	16.12	22.00	18.94	24.95	20.50

Table 5 Average class error and relative improvement over the baseline system for the RNN classifier with the combination and pooling block before the first BLSTM layer on the test partition

Config	Avg class error (%)	Rel. improvement (%)
Conv Pool $\text{BLSTM}_{1,2}$	29.30	− 11.80
Conv $\text{BLSTM}_{1,2}$	**26.26**	**− 1.60**
Pool $\text{BLSTM}_{1,2}$	28.31	− 8.72

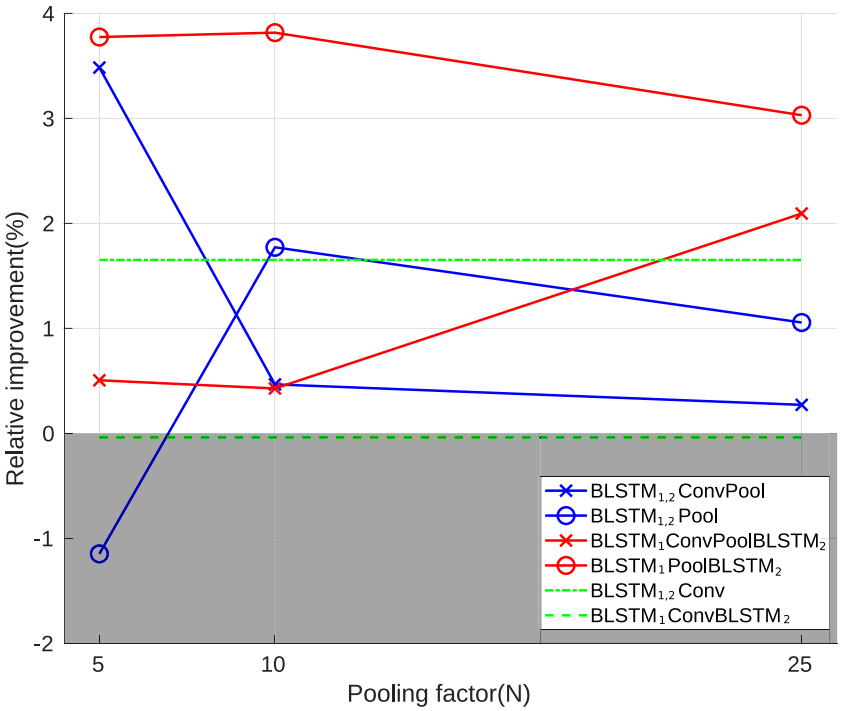

Fig. 5 Relative improvement over the baseline RNN classifier for the setup using the combination and pool block between both BLSTM layers and the setup using the combination and pooling block after the last BLSTM layer

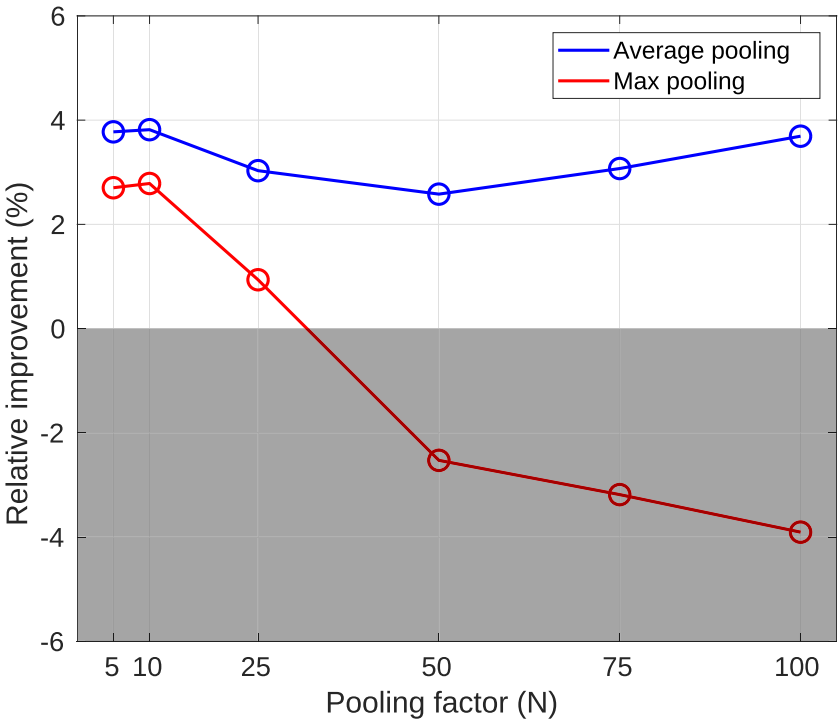

Fig. 6 Relative improvement over the baseline RNN classifier for different pooling factors and pooling techniques using the BLSTM$_1$ *Pool*BLSTM$_2$ architecture

In these architectures, we have experimented with three different values for the pooling factor: 5, 10 and 25, and an average pooling mechanism. We also present in green straight lines the architectures using only a 1D conlutional layer used, with an independent behaviour of the pooling factor.

Concerning the convolutional only architectures (green lines), it can be observed that recombining internal BLSTM representations does not show a significant improvement in performance, with the $\text{BLSTM}_1\text{ConvBLSTM}_2$ system really close to the baseline classifier and the $\text{BLSTM}_{1,2}\text{Conv}$ showing a relative improvement of around 1.8%. The best results are obtained consistently among three evaluated pooling factors for the BLSTM_1 *Pool*BLSTM_2 configuration (red circles), where a temporal pooling layer is used in between the first and the second BLSTM layers. The best case achieves a relative improvement of 3.8% compared to the baseline system. Furthermore, this new approach does not add more parameters to the model and it decreases computational complexity because the second BLSTM layer is working at a smaller sampling rate.

Being proved that the best performing setup is the one using only a pooling layer in between the first and second BLSTM layers, in the following experiment we perform a deeper analysis of this neural architecture, considering now two different pooling mechanism: average pooling and max pooling, and a wider variation range of the pooling factor, from 10 to 100. In Fig. 6 we present the results for all the evaluated configurations in terms of the relative improvement obtained when compared to the baseline RNN classifier output using the best feature configuration (80 bands + chroma + derivatives). Two differentiated behaviours can be observed: the average pooling configurations (blue line) show a general improvement between 3 and 4% without a strong dependence on the pooling factor. However, max pooling (red line) degrades its performance significantly when increasing the pooling factor, even showing worse results than the baseline for pooling factors greater than 25. Bearing this results in mind, only the average pooling configurations are taken into account in the following experiments.

In Table 6 we show the detailed results for the average pooling setup experiments in terms of SER, error per class and average class error.

The best result is obtained for the pooling factor 10, immediately followed by the configurations of 5 and 100. These results show a relative improvement of 3.82%, 3.77% and 3.69% respectively, without increasing the number of parameters in the neural network and reducing the computational load of our system because the second BLSTM layer is working at a smaller sampling rate.

Finally, results of the BLSTM_1*Pool*BLSTM_2 system combined with the HMM resegmentation module are shown in Table 7. Compared to the RNN baseline system using the HMM module, no significant improvement is observed, with the SER decreasing from 20.05 to 19.90% in the $N = 10$ setup. A performance degradation is even observed for bigger pooling factor setup. This could be motivated by the fact that the pooling layer has already performed part of the smoothing that the HMM did in the RNN baseline architecture, so the combination of both pooling layers and HMM module could not lead to a significant improvement. However, this architecture is interesting because this way we can decrease the computational load of the HMM module that now is working at a sampling rate ten times smaller.

Table 6 SER, error per class and average class error for the BLSTM_1*Pool*BLSTM_2 RNN classifier on the test partition for different pooling factors (*N*) and average pooling

N	SER	Class error (%)				Avg
		mu	sp	sm	sn	
5	15.49	15.57	28.71	24.63	30.71	24.90
10	15.47	**15.55**	**29.16**	**24.34**	**30.51**	**24.89**
25	15.51	16.85	27.17	26.23	30.06	25.08
50	15.53	16.40	28.60	24.92	30.85	25.19
75	15.54	16.87	27.97	25.55	29.88	25.07
100	15.49	18.22	26.77	24.91	29.80	24.92
No pool	15.91	16.28	28.82	26.32	31.94	25.84

5.4 Mixup data augmentation

Mixup is a data-agnostic data augmentation routine [55] that generates new virtual training examples. These virtual examples are generated according to the following equations:

$$\begin{cases} \tilde{\mathbf{x}} = \lambda \mathbf{x_i} + (1-\lambda)\mathbf{x_j} \\ \tilde{\mathbf{y}} = \lambda \mathbf{y_i} + (1-\lambda)\mathbf{y_j} \end{cases} \tag{9}$$

where $(\mathbf{x_i}, \mathbf{x_j})$ are two feature vectors randomly drawn from the training dataset and $(\mathbf{y_i}, \mathbf{y_j})$ are their corresponding one hot encoding labels, and $\lambda \in [0, 1]$. In the practical implementation $\lambda \sim \text{Beta}(\alpha, \alpha)$, with α being the mixup

Table 7 SER, error per class and average class error for the BLSTM_1*Pool*BLSTM_2 RNN classifier combined with the HHM resegmentation on the test partition for different pooling factors (N) and average pooling

N	SER	Class error (%)				Avg
		mu	sp	sm	sn	
5	12.48	14.31	22.16	18.69	25.12	20.07
10	**12.41**	**12.87**	**22.58**	**19.16**	**24.99**	**19.90**
100	13.47	22.85	24.32	19.31	26.20	23.17
No pool	12.46	14.19	22.14	18.52	25.04	20.05

hyperparameter that controls the strength of the interpolation for the pairs of examples. Furthermore, this technique is simple to implement and it does not require a high computational overhead. It is important to note that the use of mixup augmentation leads to no addition of more external datasets, so the proposed system is still under the conditions imposed by the Albayzín 2010 evaluation.

Mixup augmentation has been shown to improve model generalisation capabilities in different domains, including some audio classification tasks [64]. In our set of experiments, mixup augmentation is applied directly in the feature space.

Table 8 shows the results obtained training the BLSTM_1*Pool*BLSTM_2 architecture using mixup data augmentation for different α values compared to the same system trained without mixup augmentation. It can be seen that the result is not higly dependent on the α hyperparameter. Best result is obtained for α=0.2, however all the evaluated configuration show similar results. In general terms, mixup augmentation is able to achieve a relative improvement of 5% compared to a system not trained using mixup.

It can be noted that mixup augmentation shows a significant improvement on the "Speech over music" class with an absolute improvement of 2.23%. This class is the one defined as a pure combination of other two classes in the dataset. This fact shows that generating new virtual examples as a linear combination in the feature domain can be beneficial for our segmentation system.

As our final experiment, we present in Table 9 the results combining the BLSTM_1*Pool*BLSTM_2 RNN classifier trained with mixup data augmentation and the HMM resegmentation module.

With this setup, we achieve the best performing segmentation system in this work, which is equivalent to a SER of 11.80% and an average class error of 19.25%.

5.5 Discussion

Once our different system proposals have been experimentally evaluated, in this subsection we aim to compare our results with the ones obtained previously in the literature and perform an analysis on the segmentation system performance.

Table 8 SER, error per class and average class error on the test partition for the BLSTM_1*Pool*BLSTM_2 RNN classifier (Avg pooling, $N = 10$) trained using mixup augmentation with hyperparameter α

Mixup	SER	Class error (%)				Avg
		mu	sp	sm	sn	
$\alpha = 0.1$	14.84	15.21	27.99	23.05	29.34	23.90
$\alpha = 0.2$	**14.80**	**14.64**	**28.20**	**22.01**	**29.01**	**23.56**
$\alpha = 0.3$	14.81	16.03	26.32	23.89	28.22	23.62
No mixup	15.47	15.55	29.16	24.34	30.51	24.89

Table 9 SER, error per class and average class error on the test partition for the BLSTM_1*Pool*BLSTM_2 RNN classifier (Avg pooling, $N = 10$) trained with mixup augmentation with hyperparameter α combined with the HHM resegmentation

Mixup	SER	Class error (%)				Avg
		mu	sp	sm	sn	
$\alpha = 0.1$	12.88	14.39	23.87	18.68	25.70	20.66
$\alpha = 0.2$	**11.80**	**12.46**	**22.86**	**17.34**	**24.35**	**19.25**
$\alpha = 0.3$	12.14	14.80	21.71	17.37	23.54	19.36
No mixup	12.41	12.87	22.58	19.16	24.99	19.90

Figure 7 shows the results obtained in the Albayzín 2010 test partition by different systems already presented in the literature. The winner team of the original Albayzín 2010 evaluation proposed a segmentation by classification approach based on a hierarchical GMM/HMM (dark blue) including MFCCs, chroma and spectral entropy as input feature [65]. The best result so far in this database was obtained with a solution based on factor analysis combined with a Gaussian backend (orange) and MFCCs with 1st and 2nd order derivatives as input features [17]. Our three previously explained final results combining the RNN classifier and the HMM resegmentation are also presented: the RNN baseline (purple), the BLSTM_1*Pool*BLSTM_2 RNN approach (green) and the BLSTM_1*Pool*BLSTM_2 RNN trained using mixup augmentation (light blue).

Additionally, in order to compare our results with a DNN-based system, we trained and evaluated a different system using the neural architecture proposed as baseline in the DCASE challenge for sound environment detection [66], a task similar to the one presented in this paper. This approach is based on 3 2D CNN layers with 64 channels each followed by a single GRU cell with 64 hidden units. The input features are 64 dimensional log Mel filter banks. It can be seen that our three systems outperform previous results in this database, with our RNN combined with the pooling setup and trained with mixup augmentation achieving a relative improvement of 19.72% in terms of SER compared to the FA HMM approach. Furthermore, if the comparison is made with the DCASE baseline neural architecture, a DNN-based system, a 22.97% relative improvement is obtained with our best system.

Results presented in Figure 7 are complemented with the ones shown in Table 10 that introduces the average class error and the error per class obtained for the same systems presented before.

A general improvement over all the classes can be observed comparing to the previous approaches to this

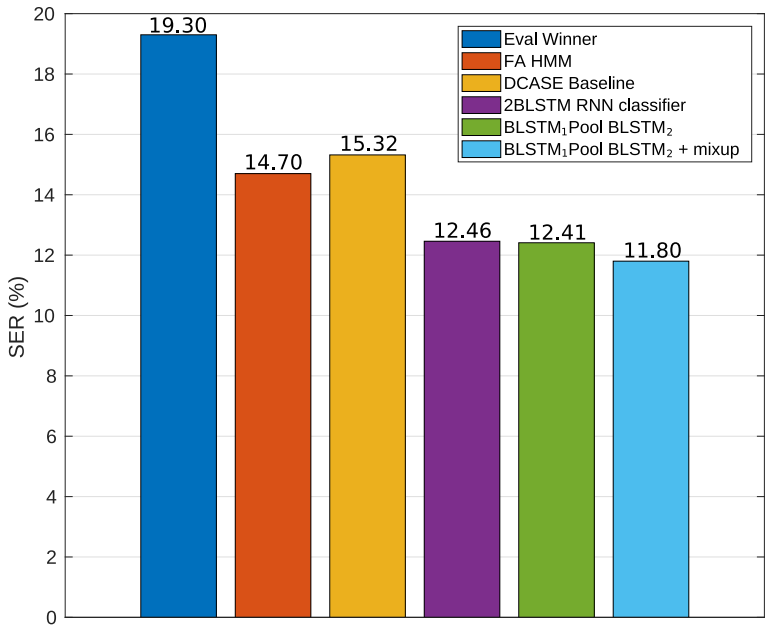

Fig. 7 Results obtained on the Albayzín 2010 test partition for different systems proposed in the literature compared to our proposed RNN approaches in terms of SER

task. It is specially significant the error difference obtained in the classes that contain music (absolute decrease of 6.34% for "Music" and 6.26% for "Speech over music" comparing our best system to the FA HMM system). On the other hand, the class "Speech" obtains really similar results. This difference in performance may be motivated by the introduction of chroma features, helping the adequately representation of music, and the linear combination of classes in training done using mixup augmentation. If the comparison is made with another DNN-based approach like the proposed DCASE baseline, again we can observe a general improvement in performance over all the classes evaluated.

As a different performance measure, Fig. 8 shows the confusion matrix for the best system presented in this work. It can be seen that one of the highest error terms is obtained for the frames predicted as "Speech over noise" but are labelled as "Speech over music", with 12% of the frames from the last class. Something similar happens with 12% of the "Speech over noise" frames incorrectly classified as "Speech". The class "Music" obtains the best classification results despite being significantly underrepresented in the database (only 5% of total). As it was observed when comparing with the other systems in the literature, this fact may come motivated by the use of chroma features, capturing adequately the musical structures and helping discriminate correctly music. The worst classification results are given for the "Others" class, not taken into account for scoring. This one, like music, is also heavily underrepresented in the database (3% of total), but in this case, this class comprises any other signal outside the definition of the other 4 classes what results in a unspecific definition making the classification harder.

Table 10 Average class error and error per class obtained on the Albayzín 2010 test partition for different systems proposed in the literature compared to our proposed RNN approaches

System	Class error (%)				Avg
	mu	sp	sm	sn	
Eval winner [65]	19.20	39.50	25.00	37.20	30.30
FA HMM [17]	18.80	23.70	*23.60	29.10	23.80
DCASE Baseline	19.03	25.58	23.59	29.52	25.18
RNN baseline [52]	14.19	22.14	18.82	25.04	20.05
RNN + Pool	12.87	22.58	19.16	24.99	19.90
RNN + Pool + mixup	**12.46**	**22.86**	**17.34**	**24.35**	**19.25**

Finally, we report the results of our best performing system in some of the traditional classifications metrics shown in the literature. Table 11 shows the overall accuracy and the precision, recall and F1 score, both per class and averaged, at frame level for the Albayzín 2010 test data evaluated using our BLSTM$_1$*Pool*BLSTM$_2$ proposal trained using mixup augmentation.

Our system achieves an overall accuracy of 85%, with a balanced average result in precision and recall.

As we have already explained in this paper, the audio segmentation task is very often a preprocessing step used before other tasks. Therefore, the complexity of the models to be used must take into account a trade-off between

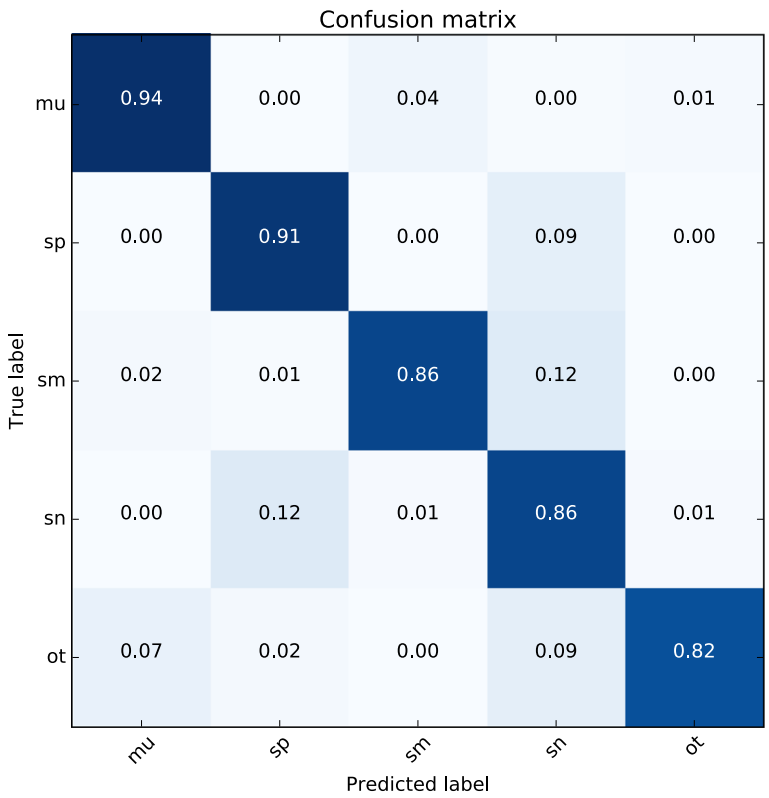

Fig. 8 Confusion matrix for the best parameter configuration evaluated in this paper: $BLSTM_1$ *Poo*/$BLSTM_2$ RNN ($N = 10$) trained with mixup ($\alpha = 0.2$) combined with the HMM resegmentation

computation time and accuracy. In order to asses this trade-off, Table 12 presents the processing time required by our best performing system to process a 1-h-long audio both using a CPU (Intel Xeon E5-220@2 GHz with 64 Gb RAM) and a GPU (GeForce GTX 1060) setup, with a single thread execution being used in all cases. In both cases, the run time is below 2 min, achieving a real-time factor close to 0.03. From our point of view, this is a reasonable processing time for this task. The most time-consuming part is the feature extraction. However, this time is mainly due to the use of a single core execution and, if it were to be implemented in a real product application, it could be reduced using multi-threading strategies and other code optimisation techniques. Additionally, Mel filter banks features are suitable to be reused in other posterior tasks.

Table 11 Accuracy (Acc.) and precision (Prec), recall (Rec) and F1 score (F1) per class and on average for the Albayzín 2010 test data evaluated using the best performing system presented in this paper

Class	Prec	Rec	F1
mu	0.88	0.89	0.88
sm	0.93	0.84	0.89
sn	0.85	0.84	0.84
sp	0.83	0.88	0.85
Avg	0.87	0.86	0.87
Acc.			**0.85**

5.6 Evaluation on a different dataset

In the previous subsection, we have analysed the results achieved with our proposed segmentation system on the Albayzín 2010 test data and we have proven the performance of our method compared to previous results in the literature. In order to evaluate the generalisation capabilities of our proposal, in this subsection we aim to evaluate it on a different dataset, namely the CARTV dataset, proposed in the Albayzín 2012 evaluation. This dataset differs from the data presented in the 2010 version: the overlap of three different classes is allowed (speech, music and noise). In order to match our multiclass classification

Table 12 Processing time required by our best performing system to process a 1-hour long audio using a CPU and GPU bases setup

	Feat extraction	Inference	Total time	RTF
CPU	1min 28s	28s	1min 56s	0.032
GPU		2s	1min 30s	0.025

framework, this format needs to be converted to non-overlapping classes, obtaining 8 different classes: "speech," "music," "noise," "speech and music," "speech and noise," "speech and music and noise," "music and noise" and "silence." Due to the similarity with the class defined as "others" in the Albayzín 2010 evaluation, we decided to combine "music and noise," "noise" and "silence" (they represent only the 3% of the total time in the database) in a single class that we also name as "others." Therefore, we can see this problem in a similar way to the task in the Albayzín 2010 dataset, but including a new class "speech and music and noise" that was not present in the 2010 version of the evaluation.

The evaluation of our system trained on the Albayzín 2010 data on the Albayzín 2012 test data would imply the consistent loss of the "speech and music and noise" class, that is not present in the data seen by the neural network. It also must be noticed that the change of domain from television to radio data could affect the results obtained. Furthermore, the low amount of data available from CARTV dataset in the development subsets (dev1 and dev2 contain only 4 h of audio) suggests that training a new neural network from scratch may not be the most suitable solution.

Taking all this statements into account, we opted for a solution that adapts our best performing model trained on the Albayzín 2010 data to the radio domain using the 4 hours of development data available from the CARTV dataset. This adaptation process is described in the following lines:

- The pretrained model on the Albayzín 2010 data is taken as the training starting point of the neural network. The final classification layer is removed and then a new one with 6 output neurons is randomly initialised.
- The whole neural network (BLSTM layers and final classification layer) is trained with the CARTV dev1 and dev2 data using the same strategies presented in the previous sections (temporal pooling and mixup augmentation). The learning rate used in the BLSTM layers is ten times smaller than the learning rate used for the final classification layer.

The HMM resegmentation is used in the same way as described previously in this paper. Aiming to compare with a different DNN-based architecture, a similar approach is done in the DCASE baseline architecture, removing the last linear layer and randomly initialising a new one with 6 neurons, to then retrain the neural network with development data from CARTV dataset.

Table 13 presents the results on the Albayzín 2012 test data for different systems compared to our proposed RNN-based approach combined with the HMM resegmentation module in terms of SER. In addition to our proposal and the DCASE baseline architecture, we show the results of the two best performing systems in the original Albayzín 2012 evaluation. The first system presented [67] is based on the use of 1024 mixtures GMMs to model each of the possible combinations of acoustic classes. Then, a Viterbi decoding is performed to obtain the segmentation labels. Input features are MFCCs and first and second order derivatives. The second system presented [68] applied an HMM-GMM speech recognition approach in which the vocabulary set is defined by the possible acoustic classes. Input is based on MFCC features, considering first- and second-order derivatives too.

Table 13 SER on the Albayzín 2012 test partition for different systems proposed in the literature compared to our proposed RNN approach

System	SER
RNN proposal (pool + mixup)	**24.93**
DCASE baseline	31.21
GMM + Viterbi decoding [67]	26.34
HMM-GMM [68]	26.53

It can be observed that if we compare the result obtained with a different DNN approach such as the DCASE baseline architecture, our systems achieves a relative improvement of 20.12% in terms of SER. This improvement is in the same order of magnitude as the improvement observed in the 2010 evaluation data, reflecting a consistent behaviour for our proposed neural architecture. If the focus is set on the results achieved in the original Albayzín 2012 evaluation, a 5.35% relative improvement can be observed compared to the evaluation winner. This improvement is significantly smaller than the improvement achieved in the Albayzín 2010 evaluation. This fact may come motivated by the small amount of in-domain data released for the 2012 evaluation. Our DNN approach would be able to profit from a bigger amount of data, whereas more traditional approaches such as GMM-HMM can achieve competitive results with a smaller amount of data.

6 Conclusions

In this paper, we have explored several architectures of RNN-based classifiers for the multiclass audio segmentation task. Our proposal, based on a segmentation-by-classification approach, combines the BLSTM modelling capabilities with an HMM backend to smooth the results. Different front-ends have been evaluated, proving how useful chroma features can be when representing music. Furthermore, the combination of BLSTM and HMM was proved to be appropriate, reducing significantly the

system error by forcing a minimum segment length for the segmentation labels.

We propose the introduction of a "Combination and Pooling" block in the neural architecture in several configurations. We showed that a time pooling architecture then can be used in between two BLSTM layers to get a subsampled output, removing temporal redundant information and achieving a relative improvement of around 5% in the neural network output. This result is still underperforming our proposed HMM resegmentation module, but we believe it is an interesting insight into the introduction of pure DNN smoothing in the audio segmentation tasks. Yet further research is needed on this technique to fill the gap between the HMM and the DNN pooling.

Furthermore, through mixup data augmentation, a data-agnostic data augmentation technique, we introduced another 5% relative improvement on the neural network modelling classes as a linear combination. No additional datasets were include to work under the Albayzín 2010 evaluation conditions.

Competitive results have been obtained with our RNN-based approach, resulting in a relative improvement of 19.72% and 5.35%, respectively, compared to the best result in the literature so far for the Albayzín 2010 and 2012 evaluations.

Abbreviations

Adam: Adaptative moment estimation; ASR: Automatic speech recognition; BIC: Bayesian information criterion; BLSTM: Bi directional long short-term memory; CNN: Convolutional neural networks; CRNN: Convolutional recurrent neural networks; DER: Diarisation error rate; DNN: Deep neural network; FA: Factor analysis; FIR: Finite impulse response; GLR: Generalised likelihood ratio; GMM: Gaussian mixture model; HMM: Hidden Markov model; LSTM: long short-term memory; MFCC: Mel frequency cepstrum coefficient; PLP: Perceptual linear predictive; RNN: Recurrent neural network; SAD: Speech activity detection; SER: Segmentation error rate; SVM: Support vector machine

Acknowledgements

Authors would like to thank the reviewers and editors for their effort in the improvement of this manuscript. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

Authors' contributions

PG performed and designed the set of experiments, also took an important role in the analysis of the results and was the main contributor in writing the manuscript. AO and IV proposed a experimental methodology and guided the analysis of the results. AM proposed the introduction of convolutional layers and mixup techniques and helped in the implementation. All authors participated in reviewing and editing the manuscript. The final manuscript was read and approved by all the authors.

Funding

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, Government of Aragón (Reference Group T36_17R) and co-financed with Feder 2014-2020 "Building Europe from Aragón". Author Pablo Gimeno was supported in part by the Government of Aragón with a grant for predoctoral research contracts (2018–2022) co-funded by the Operative Program FSE Aragón 2014-2020.

Availability of data and materials

The Albayzín 2010 dataset and the Albayzín 2012 dataset are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Received: 1 August 2019 Accepted: 11 February 2020
Published online: 05 March 2020

References

1. T. Theodorou, I. Mporas, N. Fakotakis, An overview of automatic audio segmentation. Int. J. Inf. Technol. Comput. Sci. (IJITCS). **6**(11), 1–9 (2014)
2. P. Dhanalakshmi, S. Palanivel, V. Ramalingam, Classification of audio signals using AANN and GMM. Appl. Soft Comput. **11**(1), 716–723 (2011)
3. M. R. Hasan, M. Jamil, M. Rahman, et al., in *3rd International Conference on Electrical & Computer Engineering (ICECE)*. Speaker Identification Using Mel Frequency Cepstral Coefficients, (2004), pp. 565–568
4. E. Wong, S. Sridharan, in *Proc. IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing*. Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification, (2001), pp. 95–98. https://doi.org/10.1109/isimp.2001.925340
5. H.-Y. Lo, J.-C. Wang, H.-M. Wang, in *IEEE International Conference on Multimedia and Expo (ICME)*. Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval, (2010), pp. 304–309. https://doi.org/10.1109/icme.2010.5583009
6. A. Gallardo-Antolín, J. M. Montero, Histogram equalization-based features for speech, music, and song discrimination. IEEE Sig. Process. Lett. **17**(7), 659–662 (2010)
7. R. Huang, J. H. Hansen, Advances in unsupervised audio classification and segmentation for the broadcast news and NGSW corpora. IEEE Trans. Audio Speech Lang. Process. **14**(3), 907–919 (2006)
8. J. Saunders, in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Real-time discrimination of broadcast speech/music, (1996), pp. 993–996. https://doi.org/10.1109/icassp.1996.543290
9. C.-H. Wu, Y.-H. Chiu, C.-J. Shia, C.-Y. Lin, Automatic segmentation and identification of mixed-language speech using delta-BIC and LSA-based GMMs. IEEE Trans. Audio Speech Lang. Process. **14**(1), 266–276 (2006)
10. M. Kotti, E. Benetos, C. Kotropoulos, Computationally efficient and robust BIC-based speaker segmentation. IEEE Trans. Audio Speech Lang. Process. **16**(5), 920–933 (2008)
11. A. Dessein, A. Cont, An information-geometric approach to real-time audio segmentation. IEEE Sig. Process. Lett. **20**(4), 331–334 (2013)
12. J. Foote, in *IEEE International Conference on Multimedia and Expo (ICME)*. Automatic audio segmentation using a measure of audio novelty, (2000), pp. 452–455. https://doi.org/10.1109/icme.2000.869637
13. R. Yin, H. Bredin, C. Barras, in *Proc. Interspeech 2017*. Speaker change detection in broadcast tv using bidirectional long short-term memory networks, (2017), pp. 3827–3831. https://doi.org/10.21437/interspeech.2017-65
14. A. Misra, in *Proc. Interspeech*. Speech/nonspeech segmentation in web videos, (2012), pp. 1977–1980
15. G. Richard, M. Ramona, S. Essid, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams, (2007), pp. 461–464. https://doi.org/10.1109/icassp.2007.366272
16. Y. Lavner, D. Ruinskiy, A decision-tree-based algorithm for speech/music classification and segmentation. EURASIP J. Audio Speech Music. Process. **2009**, 2 (2009)
17. D. Castán, A. Ortega, A. Miguel, E. Lleida, Audio segmentation-by-classification approach based on factor analysis in broadcast news domain. EURASIP J. Audio Speech Music. Process. **2014**(1), 34 (2014)
18. J. Ajmera, I. McCowan, H. Bourlard, Speech/music segmentation using entropy and dynamism features in a HMM classification framework. Speech Commun. **40**(3), 351–363 (2003)
19. L. Lu, H. Jiang, H. Zhang, in *Proc. 9th ACM International Conference on Multimedia*. A robust audio classification and segmentation method, (2001), pp. 203–211. https://doi.org/10.1145/500141.500173
20. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Sig. Process. Mag. **29**(6), 82–97 (2012)
21. L. Deng, G. Hinton, B. Kingsbury, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New types of deep neural

network learning for speech recognition and related applications: An overview, (2013), pp. 8599–8603. https://doi.org/10.1109/icassp.2013.6639344
22. D. Snyder, D. Garcia-Romero, D. Povey, S. Khudanpur, in *Proc. Interspeech*. Deep neural network embeddings for text-independent speaker verification, (2017), pp. 999–1003. https://doi.org/10.21437/interspeech.2017-620
23. D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, A. McCree, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Speaker diarization using deep neural network embeddings, (2017), pp. 4930–4934. https://doi.org/10.1109/icassp.2017.7953094
24. X. Shao, C. Xu, M. S. Kankanhalli, in *Proc. Joint 4th International Conference on Information, Communications and Signal Processing, and 4th Pacific Rim Conference on Multimedia*. Applying neural network on the content-based audio classification, (2003), pp. 1821–1825. https://doi.org/10.1109/icics.2003.1292781
25. H. Meinedo, J. Neto, in *9th European Conference on Speech Communication and Technology*. A stream-based audio segmentation, classification and clustering pre-processing system for broadcast news using ANN models, (2005)
26. X.-K. Yang, D. Qu, W.-L. Zhang, W.-Q. Zhang, An adapted data selection for deep learning-based audio segmentation in multi-genre broadcast channel. Digit. Sig. Process. (2018). https://doi.org/10.1016/j.dsp.2018.03.004
27. K. J. Piczak, in *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. Environmental sound classification with convolutional neural networks, (2015), pp. 1–6. https://doi.org/10.1109/mlsp.2015.7324337
28. D. Doukhan, J. Carrive, in *9th International Conferences on Advances in Multimedia (MMEDIA)*. Investigating the use of semi-supervised convolutional neural network models for speech/music classification and segmentation, (2017)
29. B.-Y. Jang, W.-H. Heo, J.-H. Kim, O.-W. Kwon, Music detection from broadcast contents using convolutional neural networks with a mel-scale kernel. EURASIP J. Audio Speech Music. Process. **2019**(1), 11 (2019)
30. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
31. F. A. Gers, J. Schmidhuber, F. Cummins, in *9th International Conference on Artificial Neural Networks (ICANN)*. Learning to forget: continual prediction with LSTM, (1999), pp. 850–855. https://doi.org/10.1049/cp:19991218
32. A. Graves, A.-r. Mohamed, G. Hinton, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Speech recognition with deep recurrent neural networks, (2013), pp. 6645–6649. https://doi.org/10.1109/icassp.2013.6638947
33. M. Sundermeyer, R. Schlüter, H. Ney, in *Proc. Interspeech*. LSTM neural networks for language modeling, (2012), pp. 194–197
34. G. Heigold, I. Moreno, S. Bengio, N. Shazeer, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. End-to-end text-dependent speaker verification, (2016), pp. 5115–5119. https://doi.org/10.1109/icassp.2016.7472652
35. F. Eyben, F. Weninger, S. Squartini, B. Schuller, in *IEEE International Conference O nAcoustics, Speech and Signal Processing (ICASSP)*. Real-life voice activity detection with LSTM recurrent neural networks and an application to Hollywood movies, (2013), pp. 483–487. https://doi.org/10.1109/icassp.2013.6637694
36. J. Kim, J. Kim, S. Lee, J. Park, M. Hahn, in *Proc. 8th International Conference on Signal Processing Systems*. Vowel based voice activity detection with LSTM recurrent neural network, (2016), pp. 134–137. https://doi.org/10.1145/3015166.3015207
37. D. de Benito-Gorron, A. Lozano-Diez, D. T. Toledano, J. Gonzalez-Rodriguez, Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset. EURASIP J. Audio Speech Music. Process. **2019**(1), 9 (2019)
38. I. Viñals, P. Gimeno, A. Ortega, A. Miguel, E. Lleida, in *Proc. Interspeech*. Estimation of the number of speakers with variational bayesian PLDA in the DIHARD diarization challenge, (2018), pp. 2803–2807. https://doi.org/10.21437/interspeech.2018-1841
39. I. Viñals, P. Gimeno, A. Ortega, A. Miguel, E. Lleida, *In-domain adaptation solutions for the RTVE 2018 diarization challenge*, pp. 220–223. https://doi.org/10.21437/iberspeech.2018-45
40. N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, M. Liberman, First dihard challenge evaluation plan, 2018. tech. Rep. (2018)
41. E. Lleida, A. Ortega, A. Miguel, V. Bazán-Gil, C. Pérez, M. Gómez, A. de Prada, Albayzin 2018 evaluation: the iberspeech-RTVE challenge on speech technologies for spanish broadcast media. Appl. Sci. **9**(24), 5412 (2019)
42. J. S. Downie, A. F. Ehmann, M. Bay, M. C. Jones, The music information retrieval evaluation exchange: Some observations and insights. Advances in music information retrieval, 93–115 (2010). https://doi.org/10.1007/978-3-642-11674-2_5
43. B. Meléndez-Catalán, E. Molina, E. Gomez, in *Music Information Retrieval Evaluation eX-change (MIREX)*. Music and/or speech detection MIREX 2018 submission, (2018)
44. M. Choi, J. Lee, J. Nam, in *Music Information Retrieval Evaluation eX-change (MIREX)*. Hybrid features for music and speech detection, (2018)
45. M. Mandel, J. Salamon, D. P. W. Ellis, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*. (New York University, New York, 2019)
46. S. Kapka, M. Lewandowski, in *Proc. DCASE2019 Challenge*. Sound source detection, localization and classification using consecutive ensemble of CRNN models, (2019). https://doi.org/10.33682/9f2t-ab23
47. L. Lin, X. Wang, in *Proc. DCASE2019 Challenge*. Guided learning convolution system for DCASE 2019 task 4, (2019). https://doi.org/10.33682/53ed-z889
48. S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, G. Gravier, in *9th European Conference on Speech Communication and Technology*. The ESTER phase II evaluation campaign for the rich transcription of french broadcast news, (2005)
49. J. Žibert, F. Mihelič, J. P. Martens, H. Meinedo, J. P. D. S. Neto, L. Docio, C. G. Garcia-Mateo, P. David, J. Žďánskỳ, M. Pleva, et al., in *9th European Conference on Speech Communication and Technology*. The COST278 broadcast news segmentation and speaker clustering evaluation-overview, methodology, systems, results, (2005)
50. T. Butko, C. Nadeu, Audio segmentation of broadcast news in the Albayzin-2010 evaluation: overview, results, and discussion. EURASIP Journal on Audio, Speech, and Music Processing. **2011**(1), 1 (2011)
51. A. Ortega, D. Castan, A. Miguel, E. Lleida, in *Proc. Iberspeech 2014: VIII Jornadas en Tecnología del Habla and IV Iberian SLTech Workshop*. The Albayzín 2012 audio segmentation evaluation, pp. 283–289
52. P. Gimeno, I. Viñals, A. Ortega, A. Miguel, E. Lleida, in *Proc. Iberspeech 2018*. A recurrent neural network approach to audio segmentation for broadcast domain data, pp. 87–91. https://doi.org/10.21437/iberspeech.2018-19
53. T. Ko, V. Peddinti, D. Povey, S. Khudanpur, in *Proc. Interspeech*. Audio augmentation for speech recognition, (2015), pp. 3586–3589
54. J. Schlüter, T. Grill, in *6th International Society for Music Information Retrieval (ISMIR) Conference*. Exploring data augmentation for improved singing voice detection with neural networks, (2015), pp. 121–126
55. H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, Mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
56. M. A. Bartsch, G. H. Wakefield, in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. To catch a chorus: using chroma-based representations for audio thumbnailing, (2001), pp. 15–18. https://doi.org/10.1109/aspaa.2001.969531
57. N. Jiang, P. Grosche, V. Konz, M. Müller, in *42nd International Conference: Semantic Audio*. Analyzing chroma feature types for automated chord recognition, (2011)
58. H. Papadopoulos, G. Peeters, in *International Workshop on Content-Based Multimedia Indexing (CBMI)*. Large-scale study of chord estimation algorithms based on chroma representation and HMM, (2007), pp. 53–60. https://doi.org/10.1109/cbmi.2007.385392
59. F. Eyben, F. Weninger, F. Gross, B. Schuller, in *Proc. 21st ACM International Conference on Multimedia*. Recent developments in openSMILE, the Munich open-source multimedia feature extractor, (2013), pp. 835–838
60. S. Ruder, An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
61. A. Paszke, G. Chanan, Z. Lin, S. Gross, E. Yang, L. Antiga, Z. Devito, *Automatic differentiation in PyTorch*, vol. 30, (2017), pp. 1–4
62. F. Gustafsson, Determining the initial states in forward-backward filtering. IEEE Trans. Sig. Process. **44**(4), 988–992 (1996)
63. NIST, *The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan*, (Melbourne, 2009)
64. K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, S. Liu, in *19th Pacific Rim Conference on Multimedia*. Mixup-based acoustic scene classification using multi-channel convolutional neural network, (2018), pp. 14–23. https://doi.org/10.1007/978-3-030-00764-5_2
65. A. Gallardo Antolín, R. San Segundo Hernández, UPM-UC3M system for music and speech segmentation (2010)

66. R. Serizel, N. Turpault, H. Eghbal-Zadeh, A. P. Shah, in *Proc. Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*. Large-scale weakly labeled semi-supervised sound event detection in domestic environments, (2018), pp. 19–23. https://hal.inria.fr/hal-01850270
67. D. Tavarez, E. Navas, D. Erro, I. Saratxaga, in *Proc. Iberspeech 2012*. Audio segmentation system by Aholab for Albayzin 2012 evaluation campaign, pp. 577–584
68. S. Cerdà, J. Albert, A. Giménez Pastor, J. Andrés Ferrer, J. Civera Saiz, A. Juan Císcar, *Albayzin evaluation: the PRHLT-UPV audio segmentation system*, pp. 596–600

Publisher's Note