de Benito-Gorron *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2019) 2019:9
https://doi.org/10.1186/s13636-019-0152-1

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset

Diego de Benito-Gorron*, Alicia Lozano-Diez, Doroteo T. Toledano and Joaquin Gonzalez-Rodriguez

Abstract

Audio signals represent a wide diversity of acoustic events, from background environmental noise to spoken communication. Machine learning models such as neural networks have already been proposed for audio signal modeling, where recurrent structures can take advantage of temporal dependencies. This work aims to study the implementation of several neural network-based systems for speech and music event detection over a collection of 77,937 10-second audio segments (216 h), selected from the Google AudioSet dataset. These segments belong to YouTube videos and have been represented as mel-spectrograms. We propose and compare two approaches. The first one is the training of two different neural networks, one for speech detection and another for music detection. The second approach consists on training a single neural network to tackle both tasks at the same time. The studied architectures include fully connected, convolutional and LSTM (long short-term memory) recurrent networks. Comparative results are provided in terms of classification performance and model complexity. We would like to highlight the performance of convolutional architectures, specially in combination with an LSTM stage. The hybrid convolutional-LSTM models achieve the best overall results (85% accuracy) in the three proposed tasks. Furthermore, a distractor analysis of the results has been carried out in order to identify which events in the ontology are the most harmful for the performance of the models, showing some difficult scenarios for the detection of music and speech.

Keywords: Acoustic event detection, Speech activity detection, Music activity detection, Neural networks, Convolutional networks, LSTM

1 Introduction

Recognizing and labeling the events found in audio signals is not a new challenge for machine perception. Such task has already been studied in the literature from several perspectives. While the term acoustic event detection (AED) is used for the recognition of sound events in a wide sense, the set of acoustic events under study is usually defined by the field of application.

Works on the detection of specific events can be also found, such as voice activity detection for recognizing the presence of human speech [1–3] or music activity detection, the analogous detection problem oriented to musical contents [4, 5]. In both cases, the complexity of the problem does not come from the number of different event classes to be detected, but from the high variability of the contents found in speech and music signals. Detecting the presence of speech and music events is particularly useful in speech-processing technologies. On the one hand, a voice activity detection stage allows the system to operate only over the relevant audio segments, namely, those which contains speech. Nevertheless, musical contents, which are very common in real-life recordings and audio broadcasts, are likely to be detected as speech as well, having a negative impact on the system.

Previous research in these fields usually involved rule-based systems and relatively small-sized datasets, being convenient settings for event-specific detection and classification of reduced sets of events [6, 7]. However, large-scale AED systems would need enough data to represent the huge amount of different acoustic events we can hear in the real world as well as the variability of those events.

*Correspondence: diego.benito@uam.es
AUDIAS (Audio, Data Intelligence and Speech) — Universidad Autonoma de Madrid, Madrid, Spain

Guided by the success of ImageNet [8], a large-scale image dataset which has favored the recent development of computer vision and its related fields, Google introduced AudioSet [9] in 2017 as a large-scale dataset consisting of more than two million 10-s audio segments directly extracted from YouTube videos. Each audio segment in AudioSet is weakly labeled (i.e., the temporal location of each audio event along the 10-s length is not available) with the different events contained in it, regardless of the sequential or simultaneous nature of the events. Every label refers to a specific acoustic event class defined in the AudioSet Ontology. This ontology was provided along with the dataset and defines a hierarchical structure of 632 audio event categories (of which 527 are used as labels for the segments in the dataset).

The ontology and the dataset defined in Google AudioSet have already been used to carry out several works and evaluations, such as the last editions of the DCASE challenge [7]. The size of this dataset in both the number of utterances and the diversity of audio events draws a new paradigm for the development of machine learning-based AED systems, where some research has been already performed [10–13].

The primary aim of this work is to study the performance of different neural network-based classifiers in the detection of speech and musical events along the wide variety of audio segments found in AudioSet. In contrast with most of the research conducted on this dataset so far, we employ a standard time-frequency representation of each audio segment (mel-spectrogram) as features, rather than the embeddings computed and provided by Google [9, 10]. Another differential aspect of our approach is the selection of two specific categories of events as targets, speech, and music, which is motivated by the relevance of these events to speech processing applications. Although the labeling of the rest of event classes is ignored during classification, such information has been useful in order to perform an audio event distractor analysis. This analysis is one more contribution of this work that has indicated which acoustic events are the most harmful for speech or music detection.

Through this work, we also compare two different setups for classification: (*a*) training a single classifier for each particular class as a binary problem, and (*b*) training a four-class model that includes each possible combination of both classes: "music and speech," "speech and no-music," "no-speech and music," and finally "no-music and no-speech."

The rest of the paper is organized as follows: Section 2 explains the choice of Deep Neural Networks for the task at hand, briefly presenting the different architectures that will be used. Section 3 describes the data and labels considered for our experiments and defines the parameters used to design the neural network models. Section 4 contains the experimental results for the tasks of speech event detection, music event detection, and simultaneous speech-music event detection, which are then compared and discussed. The distractor analysis is explained and discussed in Section 5, listing the most relevant distractor events in each category. Finally, Section 6 includes the conclusions of this work and highlights its key points.

2 Why DNNs in speech and music detection?

One of the main features of audio signals is their variation with time. Digital audio signals are formed by a stream of samples with a temporal structure. Thus, the content of a given time window is significantly more relevant when future and previous intervals of the signal are considered as context [14]. Such a property can be exploited in machine learning by means of recurrent models, where the input data is treated as a temporal sequence and the context is taken into account by the internal state of the model.

In recurrent neural networks (RNNs) [15], the state information of the current timestep is supplied as feedback when processing the following window. In other words, each neuron or node has an additional input which is computed from the activations of the layer in the previous timestep, providing the model with a memory. However, such memory still fails to model long-term dependencies in the input sequences. As new information flows into the network, the potential influence of previous timesteps in the present output decreases rapidly. This unconvenience of RNN structures is known as the vanishing gradient problem [16].

Long short-term memory (LSTM) [17] recurrent neural networks are composed of recurrent units designed to avoid the vanishing gradient problem. Each LSTM unit takes the input data as temporal sequences. At each time step, they decide whether they store, forget, or output the information they have gathered. This is possible thanks to their input, output and forget gates, which depend on the input data and trainable weights. The output of each LSTM unit is another temporal sequence, allowing a model to stack several consecutive LSTM layers.

The spectrum of audio signals, in particular its temporal structure, is widely used to model such signals in a tractable way, leading to well-known two-dimensional representations of audio such as the spectrogram or the melgram. These representations can be interpreted as single-channel images. Convolutional neural networks, or CNNs, are known for their suitability to image data processing [18], as they are able to take advantage of this kind of representations using convolutional layers. CNNs store learnable filters which are applied by means of a convolution to the input data. The use of convolution operations allows the network to take advantage of the context of each feature (i.e., the adjacent values) in the

two-dimensional input. When processing spectrogram-like representations, that implies being able to learn time-frequency patterns, achieving remarkable performance [10, 11].

Different approaches have been proposed to feed the networks directly with the waveform of the audio signals instead of extracting features from the data as a first step, in order to develop end-to-end systems. The CLDNN [3, 19] (acronym for convolutional LSTM DNN) architecture is specifically designed for such task, which is also referred to as feature learning. Related research in the field includes models like SincNet [20] or Wavenet [21], the latter being mainly proposed as a generative model for audio signals. Through this work, we propose neural networks containing both convolutional and LSTM stages as well. In contrast with the aforementioned CLDNN structures, our convolutional LSTM models are fed with the mel-spectrograms extracted from the audio segments, not with the audio waveforms.

Aside from recurrent and convolutional networks, feed-forward (also called fully connected) neural networks can be applied to audio signals as well. These models treat each value in the input data as time-independent features; hence, they are not optimal to learn and recognize patterns in audio signals. However, temporal context can be introduced by feeding these networks with the information of previous and future timesteps next to the current one (e.g., concatenating feature vectors of consecutive frames).

Although this work focuses on speech and music detection, neural networks have been applied to other audio signal processing tasks as well, some of them aiming to find high-level features of speech signals (e.g., language [22, 23], speaker [20, 24], or speech recognition [25–27]) or music (such as musical genre [28] or key [29]).

3 Experimental framework

3.1 Datasets and labels description

3.1.1 Speech and music labels

We define music and speech classes directly from the weak labels found in AudioSet. Thus, our "music" segments are those which include the music event tag (`/m/04rlf`), meaning that music can be heard at some point in those audio segments. In a similar way, our "speech" segments are chosen as those which include not only the human speech event tag (`/m/09x0r`) among their labels, but also those segments which include any subcategory of speech that directly implies the presence of speech (e.g., "male speech," "female speech," "child speech," or "conversation"). This simple inference mechanism avoids some of the labeling inconsistencies in segments that contain spoken voice, but are not labeled with the speech event tag due to the human-labeling process of AudioSet.

We decided not to expand the music class in a similar way, because the definition of music is considerably more complex. For instance, some random notes played in a piano would make clear the presence of the "piano" acoustic event, but considering those sounds as music would depend on the listener and on a wider cultural context.

3.1.2 Original AudioSet subsets

Google AudioSet dataset is originally divided in three disjoint subsets, named balanced train, evaluation and unbalanced train. Both balanced train (22,160 segments) and evaluation (20,371) subsets are built following a criterion of maximum class balance across every type of event. Audio segments are not provided by Google, each segment is instead identified by its YouTube video ID and its temporal location inside the video. This information is enough to obtain the corresponding audio files using an automated script. Such files have been downloaded in WAV-PCM stereo format, with 16 bits per sample and a sample rate of 16 kHz. Additionally, it is worth mentioning that public access to the videos is not assured, as they are web content that could be deleted by the uploader in any moment or retired by the platform for some reason (e.g., inappropriate content or copyright infringement). This is the reason why our evaluation and balanced train downloads contain less segments than they were supposed to (40,906 instead of 42,351).

Although music and speech events are particularly common in the dataset, the event class balance found in the balanced train and the evaluation subsets does not guarantee the balance between "speech" and "non-speech" or "music" and "non-music" segments. Observing the prior probabilities of music and speech over the downloaded balanced train + evaluation subset (Table 1), we found that 27.81% of the segments include events labeled as music and 26.26% include speech events. Additionally, only 6.83% of the segments include simultaneously speech and music events. On the other hand, 52.75% of the segments do not include speech or music events.

These priors make the balanced train + evaluation subset inconvenient for our proposed task, specially when considering a four-class problem, where the classes would be very unbalanced. Meanwhile, unbalanced train (Table 2) shows a more reasonable balance among the

Table 1 Distribution of speech and music events over the 40,906 downloaded balanced train + evaluation segments

	No-music (%)	Music (%)	Total (%)
No-speech	52.75	20.99	73.74
Speech	19.44	6.83	26.26
Total	72.19	27.81	100

Table 2 Distribution of speech and music events over the unbalanced train segments

	No-music (%)	Music (%)	Total (%)
No-speech	17.30	33.50	50.80
Speech	33.76	15.44	49.20
Total	51.06	48.94	100

studied classes, with 49.20% of its segments containing speech events and 48.94% of segments including music.

3.1.3 AUDIAS-balanced—June 2018 set

We have solved the unbalance problem by adding to our experimental set 37,030 segments from the unbalanced train subset. This subset is big enough to let us select segments from the underrepresented classes and obtain a more balanced set (Table 3), where the four possible classes show priors above 24%. We have named this new set AUDIAS-balanced—June 2018, and the list of segments and their labels are available at the following URL: http://audias.ii.uam.es/Downloads/AUDIAS_Junio18_filelist_sep.txt.

3.2 Feature extraction (mel-spectrograms)

Each audio segment has been represented as a mel-spectrogram or melgram. This representation is a time-frequency matrix where the frequency axis follows the mel-frequency scale [30], a log-based perceptual representation of the spectrum.

The mel-spectrogram transformation is based on the computation of the short-time Fourier transform (STFT) spectrogram. The frequency bins of the STFT are then transformed to the mel scale by means of a mel-filter bank. For this process, we have used Hanning windows of 32 ms with 20 ms shifts and 128 mel-filters.

The modulus M of the obtained mel-spectrograms has been transformed to decibels using the expression in Eq. 1.

$$M_{\mathrm{dB}} = 20\log_{10}(1+M) \tag{1}$$

The result for each audio segment is a 128×500 matrix, with 128 frequency bins and 500 time steps.

The waveform, spectrogram, and mel-spectrogram representations are illustrated in Fig. 1 (speech segment) and Fig. 2 (music segment).

Table 3 Distribution of speech and music events over the 77,936 AUDIAS-balanced—June 2018 segments

	No-music (%)	Music (%)	Total (%)
No-speech	27.69	24.10	51.79
Speech	24.10	24.10	48.21
Total	51.79	48.21	100

3.3 Design and parameterization of the models

In this work, we have evaluated several neural network architectures on the defined classification problems. All these architectures receive the log-compressed mel-spectrogram of the audio segment as an input. As an output, the networks provide an estimation of the posterior probability of the segment belonging to each possible class. For this purpose, we included in every architecture a SoftMax output layer, which is a multidimensional generalization of a logistic function. The dimension of the SoftMax layer depends on the number of considered classes: two nodes for the music and speech binary setups (a) and four in the case of the simultaneous music-speech classification setup (b).

We have designed five different families of architectures: fully connected networks (FConn), also known as feed-forward; convolutional neural networks (CNN), long short-term memory (LSTM) networks, and two hybrid convolutional-LSTM networks, C1-LSTM and C2-LSTM, with one-dimensional and two-dimensional convolutional filters respectively.

In order to tackle the large amount of design possibilities, two integer parameters have been defined, L (related to the number of hidden layers) and N (the number of neural units contained in each hidden layer).

3.3.1 Fully connected models

The proposed fully connected (FConn) models consist on L fully connected hidden layers, each one containing N units (Fig. 3). The first hidden layer of these models takes the whole mel-spectrogram as an input, row-wise. The ReLU activation function is applied to the outputs of every hidden layer. As this architecture is the most simple neural network design, it is considered as a baseline for our experimental set.

The ReLU activation function is a continuous function consisting on two linear segments (Eq. 2). Its derivative is not continuous, but this does not suppose a problem in practice. Actually, as the computation of its derivative is immediate, ReLU activations are widely used and allow training processes to converge considerably faster.

$$\mathrm{ReLU}(x) = \begin{cases} 0, & x \leq 0 \\ x, & \text{otherwise} \end{cases} \tag{2}$$

3.3.2 CNN models

Our CNN models contain L hidden convolutional layers with N filters each. After each convolutional layer, we add a MaxPooling layer with a 2×2 grid, aiming to isolate the relevant information and reduce the size of the feature matrices. Thus, each layer operates in a different scale of the mel-spectrogram, allowing the latter layers to access wider temporal and frequential contexts. The proposed CNN models end with a flatten layer which transforms the features to 1-D vectors, and a fixed-size fully connected

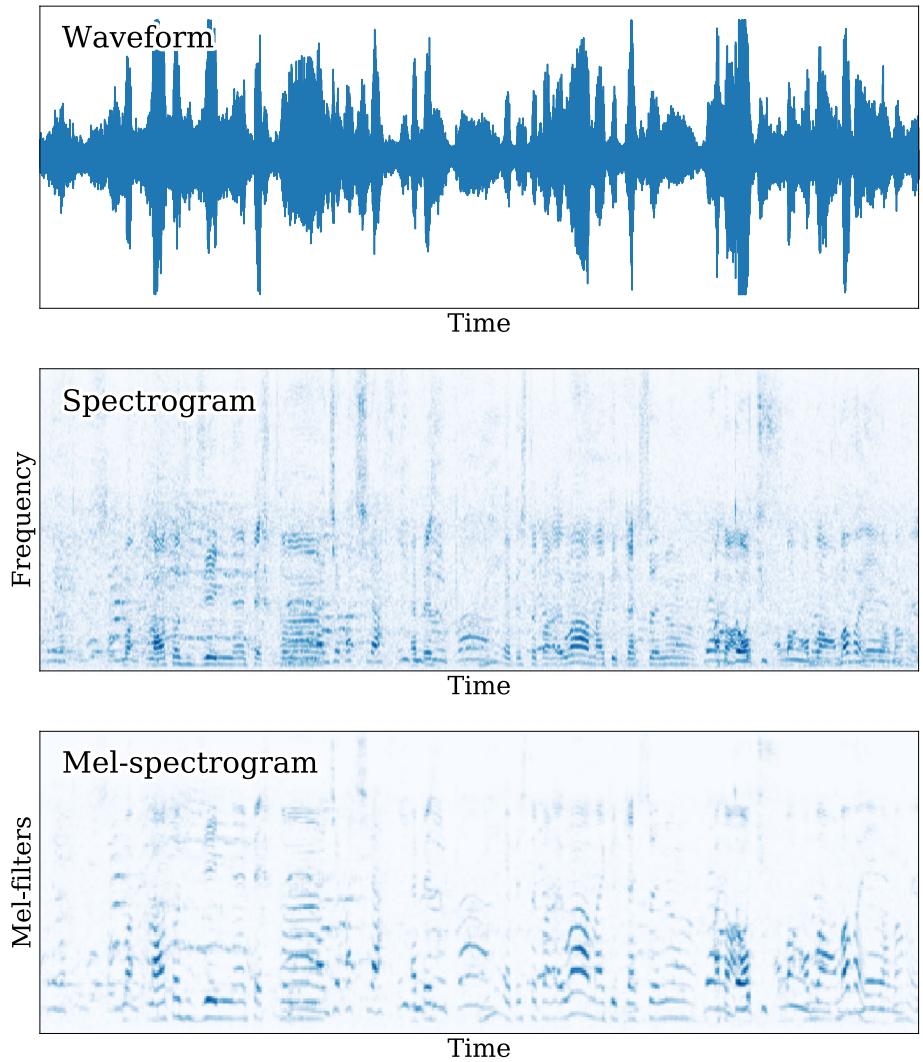

Fig. 1 Waveform, spectrogram, and mel-spectrogram of a 10-s speech segment obtained from Google AudioSet. The mel-spectrogram, based on the auditory-based mel-frequency scale, provides better resolution for lower frequencies than the spectrogram

layer with 512 units right before the output layer. The filter size is fixed in each network to 3×3 (CNN3x3) or 7×7 (CNN7 $\times$ 7).

3.3.3 LSTM models

Our LSTM models consist on L hidden layers, each one with N LSTM units. At the last hidden layer, the last value of each sequence is selected in order to reduce dimensionality before the output layer.

In addition to the already described networks, two hybrid architectures are proposed which feature both convolutional and LSTM stages.

3.3.4 C1-LSTM models

The first hybrid architecture consists on L one-dimensional convolutional layers of N filters, each one followed by a MaxPooling layer of size 2. The length of the filters is 3, and they affect only the frequential dimension. After these convolutional layers, we add L LSTM layers. Thus, the aim of these models is to process the frequential context in the convolutional stage, then the temporal structure in the LSTM layers. These architectures will be referred to as C1-LSTM.

3.3.5 C2-LSTM models

The last architecture contains L two-dimensional convolutional layers with N filters of size 3×3, each one followed by a MaxPooling layer with 2×2 grid. The difference with respect to the CNN3 $\times$ 3 models is the substitution of the flatten layer with a single LSTM layer of N units. The last value of the LSTM output is selected and passed to a fully connected layer of 512 units. This is intended to

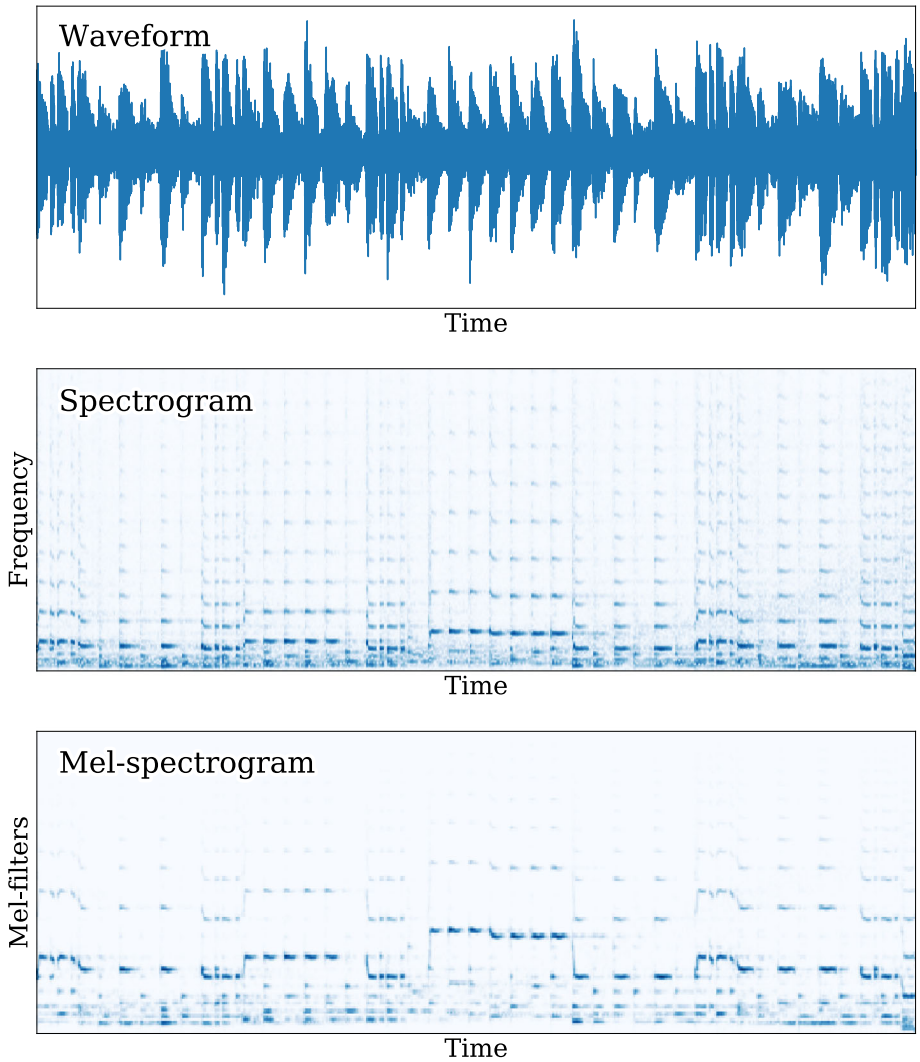

Fig. 2 Waveform, spectrogram, and mel-spectrogram of a 10-s music segment obtained from Google AudioSet. The mel-spectrogram, based on the auditory-based mel-frequency scale, provides better resolution for lower frequencies than the spectrogram

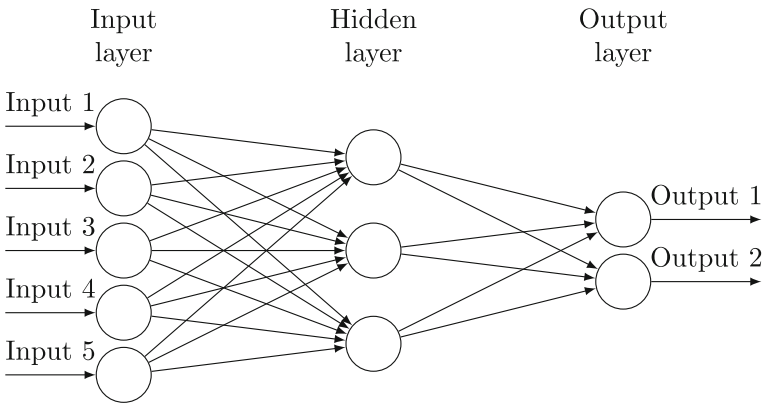

Fig. 3 Basic scheme of a fully connected neural network with five inputs, two outputs, and a single hidden layer

work as a time-aware dimensionality reduction of the processed information. These architectures will be referred to as C2-LSTM.

3.4 Training method and model selection criteria

The subset of Google AudioSet used in this work, defined in Section 3.1 and containing 77,936 audio segments, has been divided into training, validation, and test sets. We have taken apart 30% of the segments for test (23,383), and the other 70% has been used during training, divided into a 80% partition for training (43,643) and 20% for validation (10,910). This separation maintains the class priors in each set.

The loss function chosen to optimize is the empirical cross-entropy, a common loss function for classification tasks. The empirical cross-entropy measures the adequation of the estimated posterior distribution (i.e., the output of the SoftMax layer) to the ground truth labels. The optimizer used is Adam [31], widely used to train neural networks.

The criterion for model selection is the minimization of the validation loss. Thus, the model that obtains a lower cross-entropy over the validation set is the one selected as best model, regardless of its number of parameters, its training time, or any other aspect.

We have implemented, trained, and evaluated the proposed models in Keras [32], using TensorFlow as backend [33] and single GPU acceleration (NVIDIA Ge-Force GTX 1080).

Dropout is a well-known technique to prevent overfitting and improve the classification performance of the models [34]. However, it has not been applied to the networks from the beginning, but only to the best setting found for each classification task (Section 4.5). This way, we have been able to evaluate the effect of dropout while minimizing the additional computational time required.

4 Results and discussion

The proposed models have been trained performing a grid search over discrete values of L and N. Such values depend on the architecture and the classification task and will be detailed in the corresponding sections. We present results for a total of 260 networks, considering 6 architectures or model families (FConn, CNN3 $\times$ 3, CNN7 $\times$ 7, LSTM, C1-LSTM and C2-LSTM) in three different classification tasks (speech detection, music detection, and simultaneous speech-music detection).

As a first step, we have looked for adequate values for the learning rate and the batch size, obtaining the fastest convergence of the fully connected training when using a learning rate of 10^{-4} and a batch size of 128 examples. Although this learning rate allowed the convergence of every architecture, convolutional networks required a reduced batch size of 8 examples due to memory limitations.

4.1 Speech event detection results

We have trained a total number of 100 different neural network architectures for the speech event detection task, defined by the parameterization of the number of hidden layers (L) and the number of units contained in each layer (N) (a more detailed description is provided in Section 3.3).

- FConn: $L = [2, 3, 4, 5, 6]$, $N = [16, 32, 64, 128, 256, 512, 1024, 2048]$ (40 networks)
- CNN3 $\times$ 3: $L = [4, 5, 6, 7]$, $N = [32, 64, 128, 256]$ (16 networks)
- CNN7 $\times$ 7: $L = [6, 7]$, $N = [32, 64, 128, 256]$ (8 networks)
- LSTM: $L = [1, 2, 3]$, $N = [32, 64, 128, 256]$ (12 networks)
- C1-LSTM: $L = [1, 2, 3, 4]$, $N = [32, 64, 128, 256]$ (16 networks)
- C2-LSTM: $L = [6, 7]$, $N = [32, 64, 128, 256]$ (8 networks)

The grid-search performed for each of the six architectures receives different ranges of the parameters L and N. The ranges are selected according to the complexity of the models, their particular features and the observation of previous results. For instance, we have decided the range of the parameter L in CNN7x7 convolutional networks once the results of CNN3 $\times$ 3 were observed. A full-range parameter exploration in each model would have been unfeasible due to training time limitations.

Our fully connected models (FConn) serve as a baseline for the classification task. A total of 40 FConn networks have been trained, ranging from 2 to 6 hidden layers and 16 to 2048 nodes per layer. The best FConn setting in terms of validation cost has 6 hidden layers with 512 nodes each, and yields an accuracy of 76% in the validation and test subsets (Table 4, first row).

Table 4 shows the classification results obtained with the best network found in each architecture in terms of cost (empirical cross-entropy) and accuracy over the train, validation, and test subsets. The best network is selected as the one which yields the minimum cost over the validation set. The number of trainable parameters of each network is included in each row in logarithmic scale ($p = \log_{10}(\text{no. parameters})$).

Each point drawn in the scatter plot in Fig. 4 represents one of the neural networks trained for the speech event detection task (a total of 100 networks), using a different shape for each model family. The position of each point along the vertical axis shows the cross-entropy yielded by the network in speech event detection over the validation

Table 4 Speech event detection results with different network architectures

Model	*L*	*N*	*p*	Train		Validation		Test	
				Cost	Acc.%	Cost	Acc.%	Cost	Acc.%
FConn	6	512	6.23	0.489	77.03	0.510	76.45	0.518	75.58
CNN3 × 3	7	128	6.04	0.322	86.86	0.383	83.65	0.387	83.72
CNN7 × 7	6	64	6.17	0.362	85.02	0.380	84.07	0.390	83.21
LSTM	1	64	4.70	0.547	73.69	0.544	73.51	0.547	73.41
C1-LSTM	3	256	6.40	0.406	82.56	0.436	80.96	0.437	80.80
C2-LSTM	*6*	*256*	*6.59*	*0.377*	*84.30*	*0.375*	*84.34*	*0.382*	*83.99*

The Model column refers to the network architecture, *L* and *N* are the number of hidden layers and nodes in each layer (the detailed function of these parameters in each structure can be found in Section 3.3). *p* is a base-10 logarithmic measure of the number of parameters. The value of the cost or loss function and the clasiffication accuracy is included for the training, validation, and test subsets. The best model in terms of validation cost is highlighted in italics

subset (i.e., the validation cost). Additionally, the complexity of the networks is illustrated by the horizontal axis of the scatter plot, representing the number of trainable parameters of each network in a logarithmic scale.

The network which achieves the lowest validation cost in speech event detection is a C2-LSTM model with $L = 6$ and $N = 256$. Following the description of the C2-LSTM architecture (Section 3.3.5), this network consists on 6 convolutional layers, where each layer contains 256 two-dimensional filters of size 3×3, followed by a single LSTM layer of 256 units and a fully connected layer of 512 units. A 2×2 MaxPooling layer is included after each convolutional layer in order to reduce dimensionality. Such network yields a 84% accuracy in both validation and test subsets. The false negative and false positive rates of this network are detailed in the confusion matrix in Fig. 5.

4.2 Music event detection results

Following the same procedure as for the speech event detection task (Section 4.1), we trained 100 different neural network architectures for music event detection. In this case, the grid-search parameterization is identical to the one described in Section 4.1.

Table 5 summarizes the classification results of the trained networks, including the best one from each architecture. The best network in terms of validation cost is a CNN7x7 with $L = 6$ and $N = 128$ (Section 3.3.2). Such network is composed of six convolutional hidden layers including 128 filters with size 7×7. Each convolutional layer is followed by a MaxPooling 2×2 layer, and a fully-connected layer containing 512 units is found immediately before the output layer.

The scatter plot in Fig. 6 illustrates the validation cost obtained by each network versus the number of trainable parameters, and the confusion matrix obtained by the best network over the test set is represented in Fig. 7.

4.3 Simultaneous speech-music event detection results

We have trained a set of 60 different neural networks to perform both the speech and the music event detection tasks simultaneously. Thus, these networks are tackling

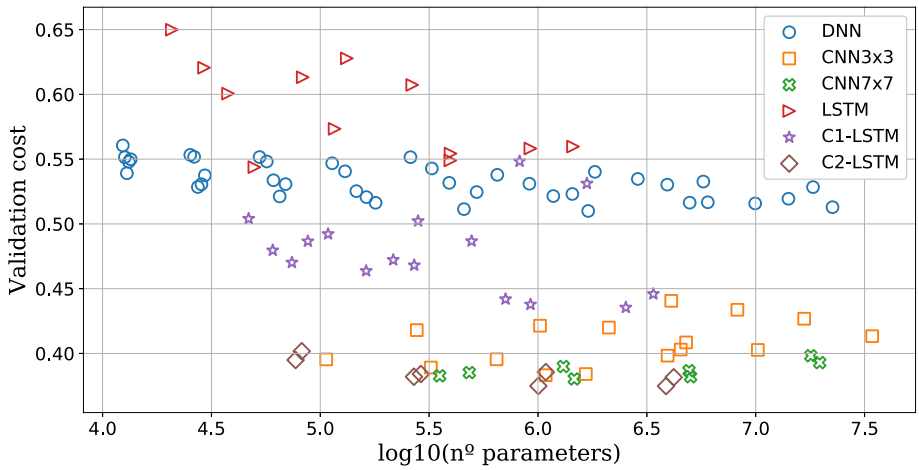

Fig. 4 Speech event detection validation cost across the evaluated models. Each point in the scatter plot represents one of the trained networks, and its position along the horizontal and vertical axes indicates the number of parameters (in a logarithmic scale) and the validation cost obtained, respectively

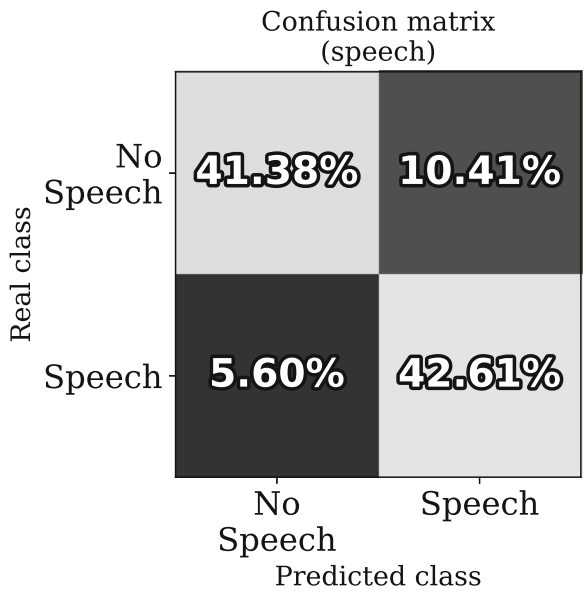

Fig. 5 Confusion matrix of the best network setting for speech event detection (C2-LSTM model with $L = 6$ and $N = 256$). The percentages indicate the total amount of test segments in each possible real class-predicted class combination

a four-class problem where the possible classes for a segment are "no-music and no-speech," "speech and no-music," "no-speech and music," and "music and speech."

There is only one correct class for each segment in this approach, even if the segment contains both speech and music events. Then, the classification accuracy only considers as correct the segments where both detections (speech and music) are right. Such is a more strict measure than the accuracies obtained in Sections 4.1 and 4.2.

Table 6 summarizes the results obtained for the double event detection task. In view of the results obtained in the previous tasks (Sections 4.1 and 4.2), we performed the grid search over the FConn, CNN3 × 3, C1-LSTM, and C2-LSTM structures. In those experiments, LSTM networks did not perform as well as other structures, and the CNN7 × 7 networks showed practically the same performance as the CNN3 × 3 models (Figs. 4 and 6), so we decided not to include these structures in the simultaneous speech-music detection experiments. The grid search over the FConn hyperparameters has also been reduced to the most relevant ranges of

The scatter plot in Fig. 8 illustrates the validation cost obtained by each network in this task versus the number of trainable parameters.

itL and *N*.

The best setting is a C2-LSTM architecture with 6 convolutional layers of 256 nodes, followed by an LSTM layer of 256 blocks and a fully connected layer of 512 nodes. Such network achieves 71% classification accuracy in both the validation and the test sets. The resulting confusion matrix is showed in Fig. 9. However, to better compare these results to those yielded by the models obtained in Sections 4.1 and 4.2, we have divided this 4×4 matrix into two 2×2 matrices (Fig. 10) which represent the performance of the model in speech and music event detection, respectively.

The examination of the 2×2 confusion matrices in Fig. 10 lets us assess the accuracy of the selected model at the speech event detection task, 83.81%, and at the music event detection task, 84.16%.

4.4 Comparison (single-task vs. double-task)

As the results in Sections 4.1, 4.2 and 4.3 show, the classification accuracies of the best settings for speech event detection and music event detection are near 84%, not only when training separate networks for each task, but also when tackling both tasks simultaneously with a single network.

Table 7 shows the different classification accuracies achieved in each task by the selected networks over the test subset, along with the false-positive and false-negative rates obtained. In both speech and music event detection, the dedicated networks yield slightly superior accuracies, albeit the results are practically identical.

Table 5 Music event detection results with different network architectures

Model	*L*	*N*	*p*	Train		Validation		Test	
				Cost	Acc.%	Cost	Acc.%	Cost	Acc.%
FConn	4	2048	7.15	0.518	74.73	0.552	72.50	0.554	72.74
CNN3x3	7	256	6.60	0.362	85.28	0.386	84.14	0.396	83.51
CNN7x7	*6*	*128*	*6.69*	*0.355*	*85.46*	*0.379*	*84.19*	*0.379*	*84.20*
LSTM	3	32	4.57	0.559	72.39	0.553	72.98	0.554	72.65
C1-LSTM	3	256	6.40	0.431	81.08	0.466	79.48	0.460	79.75
C2-LSTM	6	128	6.00	0.333	86.61	0.383	84.34	0.380	84.49

The Model column refers to the network architecture, *L* and *N* are the number of hidden layers and nodes in each layer (the detailed function of these parameters in each structure can be found in Section 3.3). *p* is a base-10 logarithmic measure of the number of parameters. The value of the cost or loss function and the clasiffication accuracy is included for the training, validation and test subsets. The best model in terms of validation cost is highlighted in italics

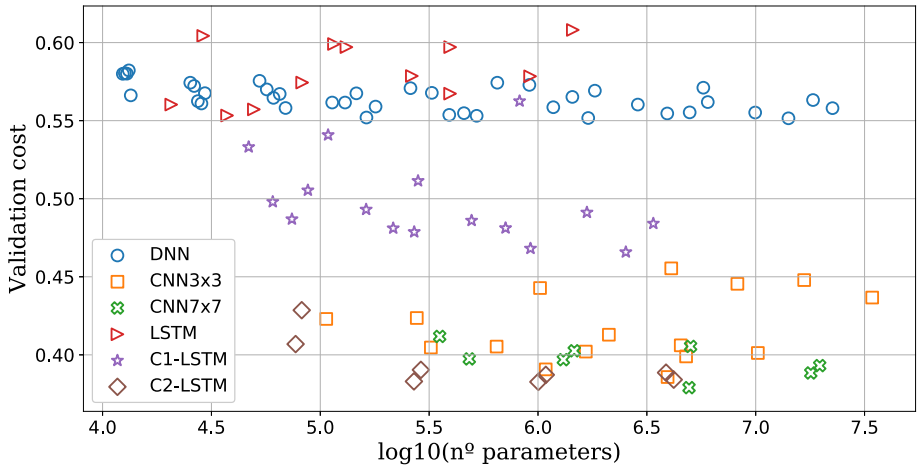

Fig. 6 Music event detection validation cost across the evaluated models. Each point in the scatter plot represents one of the trained networks, and its position along the horizontal and vertical axes indicates the number of parameters (in a logarithmic scale) and the validation cost obtained, respectively

It is important to highlight that using a single network to detect both kinds of events neither has an impact on the classification or implies an increase on the model complexity. While the selected speech detection network uses 3.9×10^6 trainable parameters and the selected music detection network uses 4.9×10^6, the best setting for detecting music and speech with a unique network requires 3.9×10^6 parameters, concluding that a similar amount of trainable weights is sufficient to perform both classification tasks with a single network.

From this point on, further analysis will be carried out using the best double-task network (C2-LSTM structure with $L = 6$ and $N = 256$, with results presented in Table 6).

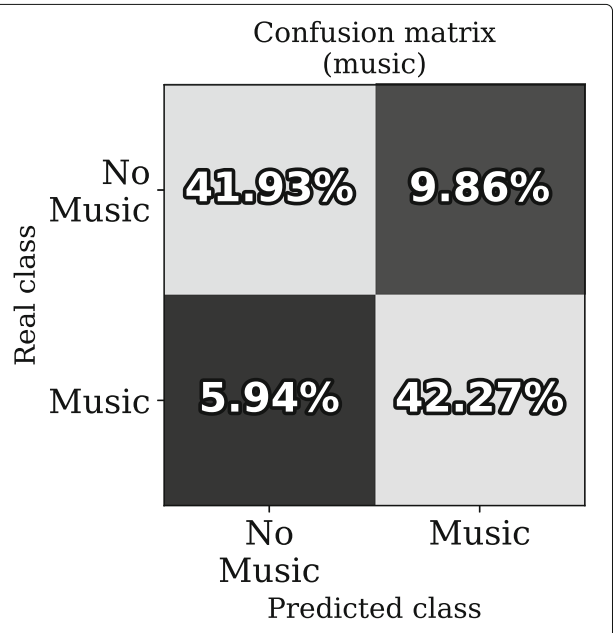

Fig. 7 Confusion matrix of the best network setting for music event detection (CNN7 × 7 with $L = 6$ and $N = 128$). The percentages indicate the total amount of test segments in each possible real class-predicted class combination

4.5 Dropout

Dropout [34] is a commonly used technique to prevent the phenomenon of overfitting in deep neural networks and improve generalization. It is based on a random deactivation of the nodes in a layer during training time (i.e., activations are set to zero with some probability P).

A neural network where dropout is applied has different neurons available in each training update, stopping the network from storing the training data or concentrating all the meaningful information in a few nodes. Dropout leads to longer training times, but often lets the networks reach better results in validation and test.

A full search for the most appropriate configuration of dropout would involve the tuning of the dropout probability of each layer in the network, as we could consider different probabilities in each hidden layer. However, a more simple approach is to tune a unique probability P_{drop} for the whole network, which would only require an additional hyperparameter.

The parameter sweep to find the best P_{drop} is feasible once we have fixed every other design decision, as it only requires training a few extra networks. Our search has considered the following range for the dropout probability:

$$P_{\text{drop}} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6] \quad (3)$$

Table 6 Simultaneous speech-music event detection results with different network architectures

Model	*L*	*N*	*p*	Train		Validation		Test	
				Cost	Acc.%	Cost	Acc.%	Cost	Acc.%
FConn	6	256	5.77	0.977	58.93	1.038	56.19	1.043	55.80
CNN3x3	6	256	6.68	0.726	71.10	0.740	70.39	0.746	70.37
C1-LSTM	4	256	6.53	0.788	67.58	0.877	64.82	0.886	64.04
C2-LSTM	*6*	*256*	*6.59*	*0.651*	*74.43*	*0.726*	*71.48*	*0.733*	*70.98*

The model column refers to the network architecture, *L* and *N* are the number of hidden layers and nodes in each layer (the detailed function of these parameters in each structure can be found in Section 3.3). *p* is a base-10 logarithmic measure of the number of parameters. The value of the cost or loss function and the clasiffication accuracy is included for the training, validation and test subsets. The best model in terms of validation cost is highlighted in italics

Actually, typical dropout probabilities range from 10 to 40%, whereas higher values tend to have a negative impact in the performance of the network. Including 50% and 60% dropout probabilities allows us to check and measure this negative effect.

Our results, shown in Table 8, confirm the theoretical assumptions. The best network in terms of validation loss is obtained with $P_{\text{drop}} = 0.4$ (val. cost = 0.692). Such model reaches an accuracy of 72.44% over the test set, while the best result without dropout was 70.98%. Although such is a slight improvement, it shows that the network actually benefits from the dropout technique even when no clear signs of overfitting were detected.

4.6 Receiver operating characteristic curves and average precision results

Across the previous sections, the performance of the proposed models has been measured in terms of empirical cross-entropy—the cost function optimized during training—and classification accuracy. Validation cost has been used for model selection, whereas accuracy has provided a more interpretable perspective of the performance of the classifiers.

However, additional metrics such as the area under the ROC (receiver operating characteristic) curve or the average precision (AP) per class are common in the field of acoustic event detection and can further describe the performance of the systems. The following sections present the ROC curves and AP yielded by the best obtained model, which is described in Table 8.

4.6.1 ROC curves

The ROC curve is plotted as the true positive rates of a given classifier against its false-positive rates in a range of different decision thresholds. The area under the curve (AUC) is bounded between 0 and 1 and summarizes the performance of the classifier.

ROC curves have been computed for both event categories (speech and music) separately and are shown in Fig. 11. Posterior probabilities for speech and music are obtained from the SoftMax output of the double-task network as the sum of individual class probabilities (i.e.,

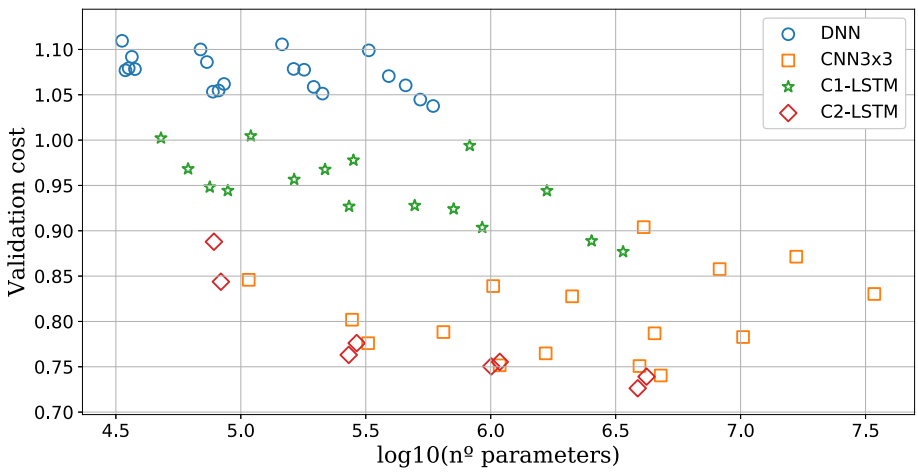

Fig. 8 Speech-music event detection validation cost across the evaluated models. Each point in the scatter plot represents one of the trained networks, and its position along the horizontal and vertical axes indicates the number of parameters (in a logarithmic scale) and the validation cost obtained, respectively

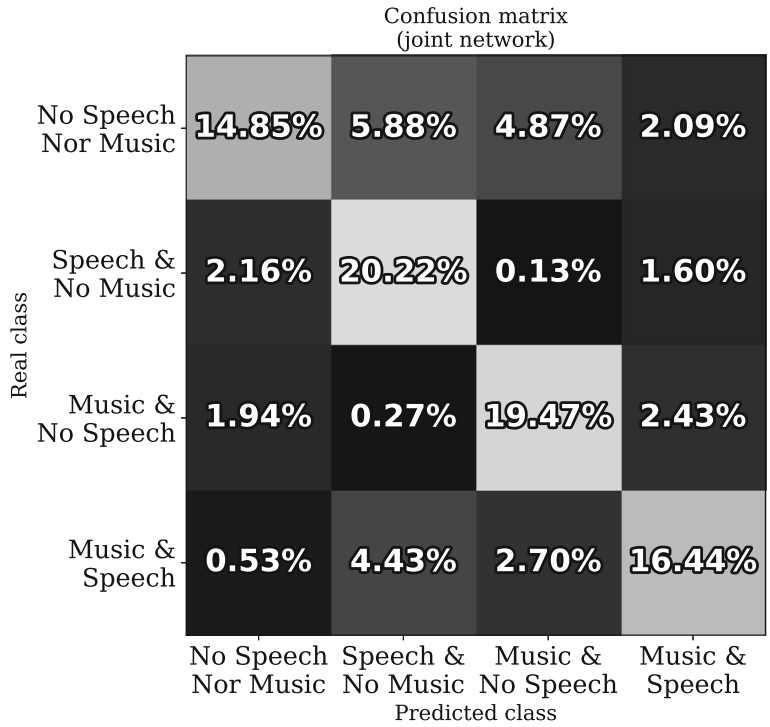

Fig. 9 Four-class confusion matrix of the best network setting for speech and music event detection (C2-LSTM model with $L = 6$ and $N = 256$). The percentages indicate the total amount of test segments in each possible real class-predicted class combination

"speech and no-music" + "speech and music" for speech, "no-speech and music" + "speech and music" for music). AUCs of 0.917 and 0.916 are obtained respectively for the speech and music categories.

4.6.2 Average precision

The performance of an acoustic event detection system across every class can be described computing its mean average precision (mAP), which is obtained as the mean of the average precision of the system for each individual class.

Our best system achieves an AP of 0.904 for speech event detection and 0.898 for music event detection, outperforming the results reported in [9] or [13] for these specific event categories. It should be highlighted that, in contrast with our system, the cited works target every

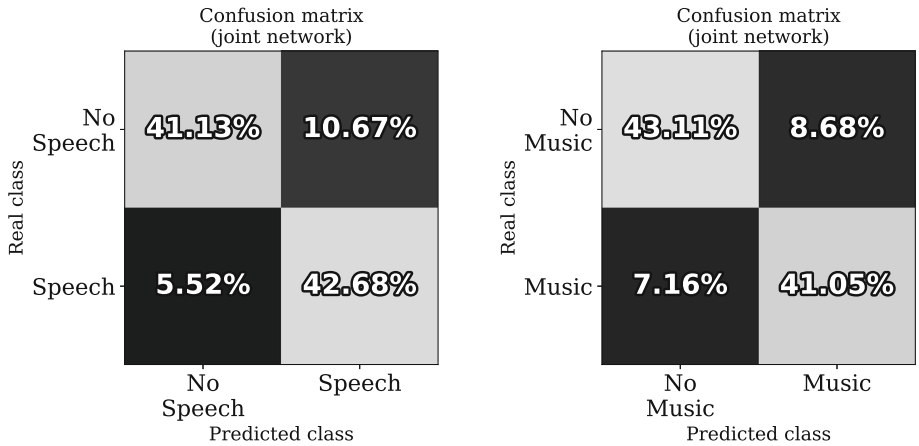

Fig. 10 Two-class confusion matrices of the best network setting for speech and music event detection (C2-LSTM model with $L = 6$ and $N = 256$). The percentages indicate the total amount of test segments in each possible real class-predicted class combination

Table 7 Comparison of the event detection results obtained by the best single-task networks and the best double-task network

		Accuracy (%)	False positive (%)	False negative (%)
Speech	Single-task network	83.99	10.41	5.60
	Double-task network	83.81	10.67	5.52
Music	Single-task network	84.20	9.86	5.94
	Double-task network	84.16	8.68	7.16

event category in the AudioSet ontology and use the whole AudioSet training sets, whereas we only consider two target categories. Additionally, our training set is a much smaller part of AudioSet. For these reasons, the results are not fully equivalent.

Table 9 shows the AP results for both categories and the mAP. However, it is to be noted that mAP should only be compared to other experiments with the same set of events, as a larger, less balanced set of event categories would be likely to lead to a lower mAP.

Thus, the results support the hypothesis that focusing on few event categories which are of special interest (instead of considering the entire set of events described in the AudioSet ontology) favors a better performance of the resulting system for those events, even with less training data.

5 Distractor analysis

So far, we have considered that each audio segment could contain spoken voice, music, both, or none of them. Although appropriate for the proposed task, this is a very limited description of the wide diversity of contents that can be found in AudioSet segments or in other real-life audio signals. Furthermore, those segments without music or voice present a very high variability due to its very own definition.

It is for these reasons that we have found it necessary to perform a posterior analysis of the classification results where we could include information about other event tags found in the AudioSet ontology. The purpose of this study is to give an insight on which events are the most likely to cause a music or speech detection error (i.e., false positive or false negative) when they are found in a segment. These events will be referred to as distractors.

Along Section 4, three different networks have been selected and compared, one trained for speech event detection (4.1), another one trained for music event detection (4.2) and finally a network that performs both detections at the same time (4.3). Given that the perfomances shown by these networks are very similar (Table 7), the analysis has been performed over the results of the combined network, which is able to detect both kinds of events.

In order to carry out the distractor analysis, we defined the following notations for the collection of event tags in AudioSet, T:

$$T = \{t_1, t_2, ..., t_n\} \tag{4}$$

And for the set of audio segment labelings in the data, S:

$$S = \{s_1, s_2, ..., s_m\} \tag{5}$$

Where each segment labeling s_i is a subset of the events in T, representing that a single segment could be labeled with more than one event tag. For the sake of clarity, we can stablish $t_1 = t_{sp}$ (speech label) and $t_2 = t_{mu}$ (music label), as well as represent the relationship between a segment labeling and a tag as

$$\tau_{i,j} = \begin{cases} 1, & t_j \in s_i \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Defining the decisions of a network about a segment as $(y_{i,sp}, y_{i,mu})$, false positives (FP) and negatives (FN) in speech or music can be expressed as

$$FP_{i,sp} \equiv (y_{i,sp} = 1,\ \tau_{i,sp} = 0)$$

Table 8 Results of the P_{drop} sweep using the best setting of the double-task network (C2-LSTM, $L = 6, N = 256$)

P_{drop}	Train		Validation		Test	
	Cost	Acc.%	Cost	Acc.%	Cost	Acc.%
0	0.651	74.43	0.726	71.48	0.733	70.98
0.1	0.634	75.02	0.736	71.25	0.745	70.25
0.2	0.669	73.69	0.708	72.19	0.721	71.25
0.3	0.601	76.10	0.704	73.01	0.721	72.10
0.4	*0.668*	*73.37*	*0.692*	*73.43*	*0.701*	*72.44*
0.5	0.702	72.12	0.726	71.51	0.741	70.91
0.6	0.734	71.06	0.723	71.78	0.735	70.91

The best setting in terms of validation cost is highlighted in italics

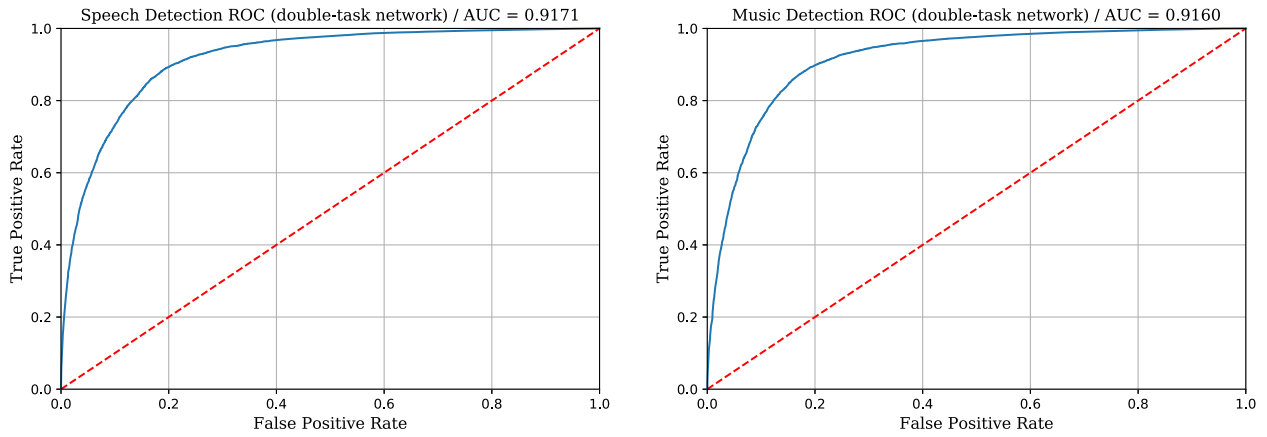

Fig. 11 ROC curves (solid blue lines) of the best obtained system for speech and music event detection. The AUC (area under curve) is 0.9171 for speech event detection and 0.9160 for music event detection

$$FN_{i,sp} \equiv (y_{i,sp} = 0,\ \tau_{i,sp} = 1)$$

$$FP_{i,mu} \equiv (y_{i,mu} = 1,\ \tau_{i,mu} = 0)$$

$$FN_{i,mu} \equiv (y_{i,mu} = 0,\ \tau_{i,mu} = 1)$$

An interesting metric to find distractor events is the conditional probability of obtaining a false negative or a false positive (in speech or in music detection) given that a certain event label is present in the audio segment. For the general case of segment labeling s_i and event label t_j, the conditional probability of a false negative in speech would be expressed as follows:

$$P(FN_{i,sp} \mid \tau_{i,j} = 1) = P(y_{i,sp} = 0,\ \tau_{i,sp} = 1 \mid \tau_{i,j} = 1) \tag{7}$$

Nonetheless, this conditional probability can be biased by the probability of finding t_{sp} given that the segment is labeled with the event t_j ($P(\tau_{i,sp} = 1 \mid \tau_{i,j} = 1)$), leading to high probabilities in those events often found together with speech (e.g., its subcategories in the ontology). For this reason, we find the following expression more appropriate to measure the influence of the event label t_j in the false negatives in speech:

$$P(y_{i,sp} = 0 \mid \tau_{i,sp} = 1,\ \tau_{i,j} = 1) \tag{8}$$

In a similar way, we obtain the three remaining combinations:

Table 9 Average precisions per class and mean average precision of the best obtained system

Speech AP	Music AP	mAP
0.904	0.898	0.901

$$P(y_{i,sp} = 1 \mid \tau_{i,sp} = 0,\ \tau_{i,j} = 1) \tag{9}$$

$$P(y_{i,mu} = 0 \mid \tau_{i,mu} = 1,\ \tau_{i,j} = 1) \tag{10}$$

$$P(y_{i,mu} = 1 \mid \tau_{i,mu} = 0,\ \tau_{i,j} = 1) \tag{11}$$

These new conditional probabilities can be approximated in a simple way as the ratios of the number of favorable cases to the number of possible cases. For instance, the probability in Eq. 8 can be represented as the number of segments with $y_{i,sp} = 0$, $\tau_{i,sp} = 1$ and $\tau_{i,j} = 1$ divided by the number of segments with $\tau_{i,sp} = 1$ and $\tau_{i,j} = 1$. Additionally, we would like to give more importance to more frequent events, as their ratios will be more meaningful (i.e., the ratio $\frac{1}{1}$ is greater but less confident than $\frac{299}{300}$).

With these aspects in mind, we have defined the following scoring functions for distractor events:

$$d^{-}_{j,sp} = \frac{N(y_{i,sp} = 0,\ \tau_{i,sp} = 1,\ \tau_{i,j} = 1)}{\mu + N(\tau_{i,sp} = 1,\ \tau_{i,j} = 1)} \tag{12}$$

$$d^{+}_{j,sp} = \frac{N(y_{i,sp} = 1,\ \tau_{i,sp} = 0,\ \tau_{i,j} = 1)}{\mu + N(\tau_{i,sp} = 0,\ \tau_{i,j} = 1)} \tag{13}$$

$$d^{-}_{j,mu} = \frac{N(y_{i,mu} = 0,\ \tau_{i,mu} = 1,\ \tau_{i,j} = 1)}{\mu + N(\tau_{i,mu} = 1,\ \tau_{i,j} = 1)} \tag{14}$$

$$d^{+}_{j,mu} = \frac{N(y_{i,mu} = 1,\ \tau_{i,mu} = 0,\ \tau_{i,j} = 1)}{\mu + N(\tau_{i,mu} = 0,\ \tau_{i,j} = 1)} \tag{15}$$

Where μ is an auxiliar term included to penalize those events where few occurrences are observed. The value of μ has been set to the average number of event labels in the subsets under study, in a similar fashion to the Dirichlet smooting method for language models [35].

An event ranking has been built for each score function. The score functions d^{-} will be maximized by negative

distractors, i.e., events that harm the correct detection of speech or music when they are present in the audio segment, causing false negatives. On the other hand, the scoring functions d^+ will help us find positive distractors, in other words, those events that are able to cause false detections of speech or music in those segments containing them.

Negative distractors are expected to have a noisy nature or be related to environments where the detection of musical or spoken contents would be more difficult, whereas positive distractors can be explained as very similar events to those we are targeting, and they even could uncover flaws in the definition of the target events.

5.1 Negative distractors for speech

The top 10 distractor events according to the d_{sp}^- scoring function are shown in Table 10. These events have been found to harm the detection of spoken voice events. The top score is held by the "whispering" event, which appears in 30 segments that contain spoken voice, but such spoken contents are only detected in 6 of those segments. Whispered voice is sometimes labeled as speech, but its spectral features are very different. Listening to the false-negative speech segments labeled as "whispering" we can assert that this event is not a proper distractor, but a subgroup of the target class with very particular features.

The rest of events found with the proposed score are related to singing voices (both male and female) and music. This fact suggests that background music is not a convenient acoustic environment for speech event detection, but also that singing is difficult to detect as spoken voice when labeled as so.

Table 10 Top 10 negative distractor events for speech (event labels related to false negative decisions of the network about the "speech" class)

Event	Event ID	Ratio	d_{sp}^-
Whispering	/m/02rtxlg	24/30	0.301
Male singing	/t/dd00003	22/52	0.216
Musical instrument	/m/04szw	66/293	0.193
Female singing	/t/dd00004	19/50	0.191
Singing	/m/015lz1	17/45	0.179
Violin, fiddle	/m/07y_7	13/23	0.179
Music	/m/04rlf	810/5636	0.143
Disco	/m/026z9	10/23	0.137
Bass guitar	/m/018vs	10/23	0.137
Guitar	/m/0342h	34/204	0.134

d_{sp}^- score (Eq. 12) is used to rank the events. The ratio column shows the number of false negatives for speech where the distractor event label is found (numerator) and the number of speech segments that contain the distractor event (denominator)

5.2 Positive distractors for speech

Table 11 lists the top 10 positive distractors for speech, ranked by descending d_{sp}^+ score. Some of the events, as "crowd," "cheering," or "children shouting," could be expected to cause false-positive detections of speech events as they can be similar to spoken voice, whereas other distractors do not have such immediate explanation. For example, "sizzle" is present in 40 segments where no spoken contents are labeled, but 35 of those segments are detected as speech, and more than a half of the 137 segments that contain the "water" event but are not labeled as speech lead to false-positive detections. Further examination of these "water" and "sizzle" segments has revealed that, in many cases, the speech event detection errors are due to labeling mistakes and the segments actually contain speech.

5.3 Negative distractors for music

The negative distractor events found for music are shown in Table 12. It contains several events describing environments ("inside, small room," "outside, rural, or natural," "outside, urban, or manmade") that are not ideal to record music because of the presence of reverberation effects and the background noise (as opposed to studio or high quality recording devices).

Meanwhile, other distractors for music are "animal," "speech," "dog," or "vehicle"—these acoustic events tend to be loud and in the foreground, with music playing on the background and being more difficult to detect.

The case of the "flute" event is worth mentioning. There are 17 test segments labeled as both "music" and "flute," but only 4 of them are correctly detected as music. A brief analysis of these segments suggests that some types

Table 11 Top-10 positive distractor events for speech (event labels related to false positive decisions of the network about the "Speech" class)

Event	Event ID	Ratio	d_{sp}^+
Crowd	/m/03qtwd	76/94	0.521
Insect	/m/03vt0	67/111	0.411
Water	/m/0838f	76/137	0.403
Sizzle	/m/07p9k1k	35/40	0.381
Battle cry	/m/04gy_2	36/44	0.376
Fowl	/m/025rv6n	64/119	0.375
Cheering	/m/053hz1	36/45	0.372
Stir	/m/07ptfmf	32/36	0.364
Children shouting	/t/dd00135	33/39	0.363
Mechanisms	/t/dd00077	42/67	0.353

d_{sp}^+ score (Eq. 13) is used to rank the events. The ratio column shows the number of false positives for speech where the distractor event label is found (numerator) and the number of non-speech segments that contain the distractor event (denominator)

Table 12 Top-10 negative distractor events for music (event labels related to false negative decisions of the network about the "Music" class)

Event	Event ID	Ratio	d^{-}_{mu}
Animal	/m/0jbk	32/84	0.230
Speech	/m/09x0r	1266/5617	0.223
Inside, small room	/t/dd00125	49/166	0.221
Vehicle	/m/07yv9	39/135	0.205
Domestic animals, pets	/m/068hy	19/40	0.199
Dog	/m/0bt9lr	16/27	0.194
Outside, rural	/t/dd00129	22/61	0.190
Flute	/m/0l14j_	13/17	0.180
Outside, urban	/t/dd00128	20/63	0.169
Television	/m/07c52	15/40	0.157

d^{-}_{mu} score (Eq. 14) is used to rank the events. The ratio column shows the number of false negatives for music where the distractor event label is found (numerator) and the number of music segments that contain the distractor event (denominator)

of flute might be detected poorly as music when they are played solo because of their high pitch and narrow-band spectral content.

5.4 Positive distractors for music

Table 13 shows the ten events with the highest d^{+}_{mu} scores. In this case, the ratios and scores are considerably higher than in the previous rankings, and all of the events are related to music. These results suggest on the one hand that the definition of music is very subjective—a musical instrument playing a single note might be considered music or not depending on the listener—, and on the other hand that a wider definition of the music class in terms of event labels would be appropriate to train and evaluate the models more consistently.

Table 13 Top 10 positive distractor events for music (event labels related to false positive decisions of the network about the "music" class)

Event	Event ID	Ratio	d^{+}_{mu}
Percussion	/m/0l14md	110/137	0.600
Pizzicato	/m/0d8_n	73/78	0.588
Drum	/m/026t6	82/94	0.585
Organ	/m/013y1f	74/81	0.582
Keyboard (musical)	/m/05148p4	77/87	0.578
Brass instrument	/m/01kcd	85/116	0.524
Singing	/m/0151z1	51/62	0.471
Hammond organ	/m/03gvt	43/46	0.466
Bass drum	/m/0bm02	43/47	0.461
Tabla	/m/01p970	40/41	0.459

d^{+}_{mu} score (Eq. 15) is used to rank the events. The ratio column shows the number of false positives for music where the distractor event label is found (numerator) and the number of non-music segments that contain the distractor event (denominator)

Overall, the distractor analysis has thrown some light on the interactions between the different acoustic events of AudioSet and their impact in the detection of spoken voice and musical contents. The obtained results have highlighted some labeling mistakes in the dataset as well as the convenience of a wider definition of the music class.

6 Conclusions

In this paper, we have presented our work with the novel database Google AudioSet in the fields of speech activity detection and music activity detection. These events, among the variety of acoustic classes labeled in AudioSet, are particularly relevant to speech processing technologies. To accomplish these tasks, we have proposed and evaluated different neural network architectures, including fully connected, convolutional, LSTM, and hybrid convolutional-LSTM networks. We have considered 2-class and 4-class classification approaches.

The networks are fed with the mel-spectrograms of the audio segments, and the best results are obtained by hybrid Convolutional-LSTM structures (C2-LSTM), that count with a two-dimensional convolutional stage prior to an LSTM layer. These models first process the input features with time-frequency filters and then expand the temporal context in the LSTM blocks.

The audio segments found in AudioSet—collected from YouTube videos—show a vast diversity of contents and are weakly labeled. Nonetheless, the classification performances of the proposed models reach 85% accuracy in both speech and music event detection, thus asserting the capability of neural network models for music and speech detection in real life audio signals. The 2-class and 4-class approaches yield very similar results in terms of both accuracy and number of parameters.

A comparison with general-purpose audio event detection works has been possible in terms of average precision. Our best system achieves average precisions near 0.9 for both speech and music classes, a similar performance to systems which use much more training data. This suggests that focusing on fewer event categories means an advantage for the classifiers.

Additionally, the distractor analysis specifically designed for this work has been proven useful to understand the classification results in more depth. We have proposed two different scoring functions for the events in the ontology, which have uncovered some particularities of the tasks, such as the difficulty of detecting "whispering" as speech or a solo "flute" as music. This analysis has as well flagged some labeling mistakes in AudioSet, and its results will allow us to enhance both these labelings and the definitions of the target classes in future work.

Abbreviations
AED: Acoustic event detection; CLDNN: Convolutional long short-term memory deep neural network; CNN: Convolutional neural network; DCASE: Detection and classification of acoustic scenes and events; DNN: Deep neural network; GPU: Graphics processing unit; LSTM: Long short-term memory; PCM: Pulse code modulation; ReLU: Rectified linear unit; RNN: Recurrent neural network; STFT: Short-time Fourier transform

Acknowledgements
Not applicable.

Authors' contributions
DB designed and performed the experiments and the analysis of the results, and was a major contributor in writing the manuscript. AL guided the experimental methodology and the analysis of the results. DT designed the script for the acquisition of the audio files from Google AudioSet and supervised the design of the analysis of the results. JG participated in the conceptualization of the work, guided and supervised the experimental methodology and the analysis of the results. Additionally, all authors participated in investigating, reviewing, and editing, and read and approved the final manuscript.

Funding
This work has been supported by project "DSSL: Redes Profundas y Modelos de Subespacios para Deteccion y Seguimiento de Locutor, Idioma y Enfermedades Degenerativas a partir de la Voz" (TEC2015-68172-C2-1-P), funded by the Ministry of Economy and Competitivity of Spain and FEDER.

Availability of data and materials
The Google AudioSet ontology and segment list are available at the AudioSet homepage https://research.google.com/audioset/. The list of segments and labels used for the experiments described through this work is available at http://audias.ii.uam.es/Downloads/AUDIAS_Junio18_filelist_sep.txt.

Competing interests
The authors declare that they have no competing interests.

Received: 18 January 2019 Accepted: 26 May 2019
Published online: 17 June 2019

References

1. J. Sohn, N. S. Kim, W. Sung, A statistical model-based voice activity detection. IEEE Signal Proc. Lett. **6**(1), 1–3 (1999)
2. X.-L. Zhang, D. Wang, Boosting contextual information for deep neural network based voice activity detection. IEEE/ACM Trans. Audio, Speech Lang. Process. (TASLP). **24**(2), 252–264 (2016)
3. R. Zazo, T. N. Sainath, G. Simko, C. Parada, in *Interspeech 2016*. Feature learning with raw-waveform CLDNNs for Voice Activity Detection, (2016), pp. 3668–3672. https://doi.org/10.21437/Interspeech.2016-268
4. K. Minami, A. Akutsu, H. Hamada, Y. Tonomura, Video handling with music and speech detection. IEEE MultiMedia. **5**(3), 17–25 (1998). https://doi.org/10.1109/93.713301
5. K. Seyerlehner, T. Pohle, M. Schedl, G. Widmer, in *Proc. of the 10th International Conference on Digital Audio Effects (DAFx'07)*. Automatic music detection in television productions (SCRIME / LaBRI, Bordeaux, 2007)
6. A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, M. Omologo, in *Proceedings of the 1st International Evaluation Conference on Classification of Events, Activities and Relationships. CLEAR'06*. CLEAR Evaluation of Acoustic Event Detection and Classification Systems (Springer, Berlin, 2007), pp. 311–322
7. D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, M. D. Plumbley, Detection and Classification of Acoustic Scenes and Events. IEEE Trans. Multimedia. **17**(10), 1733–1746 (2015). https://doi.org/10.1109/TMM.2015.2428998
8. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV). **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
9. J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, M. Ritter, in *Proc. IEEE ICASSP 2017*. Audio Set: An ontology and human-labeled dataset for audio events (IEEE, New Orleans, 2017)
10. S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al., in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017*. CNN architectures for large-scale audio classification (IEEE, New Orleans, 2017), pp. 131–135
11. Y. Xu, Q. Kong, W. Wang, M. D. Plumbley, Large-scale weakly supervised audio classification using gated convolutional neural network. CoRR. **abs/1710.00343** (2017). 1710.00343
12. Q. Kong, Y. Xu, W. Wang, M. D. Plumbley, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Audio set classification with attention model: A probabilistic perspective, (2018), pp. 316–320. https://doi.org/10.1109/ICASSP.2018.8461392
13. Q. Kong, C. Yu, T. Iqbal, Y. Xu, W. Wang, M. D. Plumbley, weakly labelled audioset tagging with attention neural networks (2019). 1903.00765
14. A. Graves. Supervised sequence labelling. Supervised sequence labelling with recurrent neural networks (Springer, Berlin/Heidelberg, 2012), pp. 5–13
15. J. L. Elman, Finding structure in time. Cogn. Sci. **14**(2), 179–211 (1990)
16. Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
17. S. Hochreiter, J. Schmidhuber, Long Short-Term Memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
18. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks. Commun. ACM. **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
19. T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, O. Vinyals, in *Interspeech 2015*. Learning the speech front-end with raw waveform CLDNNs (International Speech Communication Association, Dresden, 2015)
20. M. Ravanelli, Y. Bengio, Speaker Recognition from raw waveform with SincNet (2018). arXiv preprint arXiv:1808.00158
21. A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, in *9th ISCA Speech Synthesis Workshop*. Wavenet: A generative model for raw audio (International Speech Communication Association, Sunnyvale, 2016), pp. 125–125
22. J. Gonzalez-Dominguez, I. Lopez-Moreno, P. J. Moreno, J. Gonzalez-Rodriguez, Frame-by-frame language identification in short utterances using deep neural networks. Neural Netw. **64**, 49–58 (2015). https://doi.org/10.1016/j.neunet.2014.08.006. Special Issue on "Deep Learning of Representations"
23. R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D. T. Toledano, J. Gonzalez-Rodriguez, Language Identification in Short Utterances Using Long Short-Term Memory (LSTM) Recurrent Neural Networks. PLoS ONE. **11**, 1–17 (2016)
24. A. Lozano-Diez, A. Silnova, P. Matejka, O. Glembek, O. Plchot, J. Pesán, L. Burget, J. González-Rodríguez, in *Odyssey*. Analysis and optimization of bottleneck features for speaker recognition (International Speech Communication Association, Bilbao, 2016)
25. Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, A. Courville, in *Interspeech 2016*. Towards end-to-end speech recognition with deep convolutional neural networks, (2016), pp. 410–414. https://doi.org/10.21437/Interspeech.2016-1446
26. A. Graves, N. Jaitly, in *Proceedings of the 31st International Conference on Machine Learning*. Towards end-to-end speech recognition with recurrent neural networks (PMLR, Bejing, 2014), pp. 1764–1772
27. D. T. Toledano, M. P. Fernández-Gallego, A. Lozano-Diez, in *PloS One*. Multi-resolution speech analysis for automatic speech recognition using deep neural networks: Experiments on timit (Public Library of Science, San Francisco, 2018)
28. I.-Y. Jeong, K. Lee, in *ISMIR 2016, 7th International Society for Music Information Retrieval Conference*. Learning temporal features using a deep neural network and its application to music genre classification (ISMIR, New York City, 2016), pp. 434–440
29. F. Korzeniowski, G. Widmer, in *25th European Signal Processing Conference (EUSIPCO-2017)*. End-to-end musical key estimation using a convolutional neural network (EURASIP, Kos island, 2017), pp. 966–970
30. S. S. Stevens, J. Volkmann, E. B. Newman, A scale for the measurement of the psychological magnitude pitch. J. Acoust. Soc. Am. **8**(3), 185–190 (1937). https://doi.org/10.1121/1.1915893
31. D. P. Kingma, J. Ba, in *ICLR 2015, 3rd International Conference for Learning Representations, San Diego, vol. abs/1412.6980*. Adam: A method for stochastic optimization, (2014). http://arxiv.org/abs/1412.6980

32. F. Chollet, et al., Keras (2015). https://keras.io (accessed on 14 Jan 2019)
33. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al*, in *OSDI '16, 12th USENIX Symposium on Operating Systems Design and Implementation*. TensorFlow: A System for Large-Scale Machine Learning (USENIX, Savannah, 2016), pp. 265–283
34. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
35. C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. (TOIS). **22**(2), 179–214 (2004)

Publisher's Note