TEMPORAL CONVOLUTIONAL NETWORKS FOR SPEECH AND MUSIC DETECTION IN RADIO BROADCAST

Quentin Lemaire
KTH Royal Institute of Technology
qle@kth.se

Andre Holzapfel
KTH Royal Institute of Technology
holzap@kth.se

ABSTRACT

The task of speech and music detection aims at the automatic annotation of potentially overlapping speech and music segments in audio recordings. This metadata extraction process finds important applications in royalty collection for broadcast audio. This study focuses on deep neural network architectures made to process sequential data, and a series of recent architectures that have not yet been applied for this task are evaluated, extended and compared with a state-of-the-art architecture. Moreover, different training strategies are evaluated, and we demonstrate the advantages of a pre-training procedure with low-quality data that facilitates the combination of heterogeneous datasets. The study shows that Temporal Convolution Network (TCN) architectures can outperform state-of-the-art architectures. In specific, the novel non-causal TCN extension introduced in this paper leads to a significant improvement of the accuracy.

1. INTRODUCTION

The location of speech and music segments in large amounts of audio recordings is an important metadata information especially in the context of royalty collection in broadcasting. The task of speech and music detection is a multi-label problem, each frame can be labeled as music, speech, both, or neither of both. Assuming potential overlaps between the classes makes speech and music detection a complex and unsolved task, as opposed to a simple discrimination of segments into either speech or music [12]. Apart from royalty collection, a speech and music detection system can be useful to extract the relevant parts of the audio on which to apply other meta-data extraction such as speech-to-text, genre classification or music fingerprinting.

The first approaches to speech/music detection – discussed in the work of Carrey et al. [5] – focused on manually designed features and subsequent classification. More recently, features are automatically learned from spectrogram images using deep neural networks for various audio tasks [4, 7, 14, 22, 27, 33, 37]. End-to-end learning systems with waveform-audio input have been compared with spectrogram-based learning, with better results of the latter approach [16]. This may however be due to the lack of sufficient data in the particular case, as discussed by Pons et al. [24].

Recently, the Temporal Convolutional Network (TCN) [2] showed promising results on tasks involving sequential data. No study to date has compared the TCN architecture with other deep learning architectures such as Recurrent Neural Networks (RNN) for the task of speech and music detection. A key contribution of this paper is the investigation of a novel non-causal TCN architecture. This is of special interest to various applications where real-time analysis is not a constraint. The results of this study demonstrate that TCN architectures can outperform RNN. The final system is compared with the state of the art on the MIREX dataset [1] with results that further document the high performance of the non-causal TCN.

Another contribution of this paper is the use of an efficient pre-training procedure with low-quality data that facilitates the combination of heterogeneous datasets. We compiled the most extensive data resource available until now for the task of speech and music detection, and trained and tested networks using this heterogeneous data resource. Our results document the advantage of the pre-training procedure, and we provide the code of this study as a toolbox for the systematic comparison of system architectures for the speech and music detection task.

The following section is a review of existing work with neural networks on speech and music detection and tasks that employ similar methods. Section 3 explains the method that will be applied in this study. Section 4 presents the results that are discussed in Section 5. Finally, Section 6 draws various conclusions about this work.

2. BACKGROUND

The speech and music detection problem is a sound event detection problem. It applies to an audio stream containing temporal segments of speech and/music in arbitrary positions. Segments of speech and audio may overlap. Furthermore, some parts of the audio might contain neither music nor speech, but task-irrelevant content such as environmental sounds, footsteps or keyboard typing sounds.

[1] https://www.music-ir.org/mirex/wiki/2018:Music_and/or_Speech_Detection

Figure 1 shows an overview of the processing chain typically applied in speech/music detection and other related tasks, such as acoustic event detection. The following subsections provide an overview of examples of these steps from the literature.

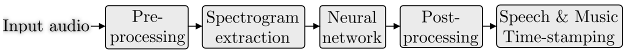

Figure 1: Overview of the processing chain

2.1 Pre-Processing & Spectrogram Extraction

The most commonly used features in the literature are the Mel-scaled log-magnitude spectrograms [14,18,22,27,33]. To obtain them, a Short-Time Fourier Transform (STFT) is computed from the audio sample containing the discrete-time signal. Only the magnitudes are kept and a Mel-scaled Filterbank is applied. Finally, the obtained coefficients are put on a log magnitude scale and normalized.

Data augmentation is a pre-processing solution that artificially creates new data based on the available ones by applying various manipulations either in the time-domain [27] or on the spectrograms [29].

2.2 Network Architectures

In the context of deep neural networks, there are two different ways described in the literature to handle the speech and music detection task (or related tasks). The first possibility is to treat the audio as non-sequential data by working on small excerpts independently which is mainly used for classification tasks. The state-of-the-art architectures are based on the Convolutional Neural Networks (CNN) [14, 19, 27, 33].

The second possibility is to use a sequence model for the audio. An audio sample is injected into the network and each frame will be classified as music, speech, both music and speech or neither of both. The state-of-the-art architectures are mainly based on the Bidirectional Long Short-Term Memory (B-LSTM) [18, 22], which are a type of RNN.

Since the systems have been evaluated on differing test sets, no conclusion regarding the best performing architecture can be obtained from the above cited work. Moreover, music and speech have different temporal and spectral properties than environmental sounds, and it is therefore not clear if results from other tasks apply to speech and music detection.

2.2.1 Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks (CLDNN)

Information from the long-term context in past and future can be very relevant for the classification and especially for borderline cases. It motivates the use of sequential models for the detection tasks. However, CNNs have obtained state-of-the-art results in image and audio feature extraction, which motivated the combination of RNN and CNN in the CLDNN architecture [26] to use the strength of both. CLDNN are divided into three sections: first, the input goes through several convolutional layers. Then, the result goes through a classic LSTM network and finally, several fully connected (FC) layers are applied. This architecture was shown to outperform CNN, RNN or DNN based approaches on various datasets [26]. This architecture has also been reused in several subsequent audio and music related studies [4, 7, 10], but has so far not been applied to speech and music detection.

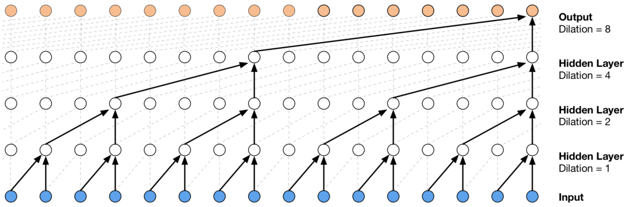

Figure 2: Visualization of a stack of dilated causal convolutional layers. Source: [35]

2.2.2 Temporal Convolutional Network (TCN)

Recently, the Temporal Convolution Network (TCN) was introduced [2] as a simple and flexible architecture using CNN for sequential learning. This architecture is based on the previous work trying to use only CNNs for sequential learning [13, 35] but it remains much simpler. A TCN cell is based on causal and dilated convolutions, and on residual blocks [15]. Figure 2 illustrates a dilated causal convolutional layer. Dilated convolutions permit to retrieve information from far in the past without extensive computation.

TCN architectures have not been used for speech and music detection yet. Therefore, within this paper, we compare four architectures: the standard TCN, a novel non-causal extension of the TCN, the BLSTM and the CLDNN architectures. This non-causal TCN architecture is designed to combine the strengths of both BLSTM and CNN architectures: the bidirectional long-term memory of the BLSTM architecture and the performance and parallelizability of the CNN architecture.

2.3 Post-processing and Evaluation

The network output vector contains two coefficients between 0 and 1, one for speech and one for music, and commonly 0.5 is used as a threshold for the labeling decision [18, 30]. A probabilistic model has been used by [11] to smooth the output and reduce the noise.

Two evaluation methods for speech and music detection – segment-level evaluation and event-level evaluation [21] – are applied within MIREX[1] and in most related studies. Segment-level evaluation compares the labels of segments of fixed size given by the algorithm with the ground-truth labels. Event-based evaluation takes the whole ground-truth events into account and a time-window tolerance is allowed. This evaluation is either made on the on-set of the events or on both the on-set and the off-set.

3. METHODOLOGY

3.1 Datasets

A larger amount of previously compiled datasets have been obtained and combined for this study (*MUSAN Corpus* [31] (MUSAN), *GTZAN Speech and Music Dataset* [34] (GTZAN), *Scheirer & Slaney Music Speech Corpus* [28] (SSMSC), *MuSpeak Speech and Music Detection Dataset* [2] (MuSpeak), *OFAI Speech and Music Detection Dataset* [30] (OFAI), *ESC-50: Dataset for Environmental Sound Classification* [23] (ESC)). In addition, a newly compiled dataset (Sveriges Radio dataset) was included as well, which is composed of songs from different genres that are played on the radio (42h 57mn) and of speech files (17h 24mn) that are extracted from radio programs.

Table 1: Categorization of the datasets according to the precision of their labels.

Low-quality datasets (LQ)	High-quality datasets (HQ)
MUSAN, GTZAN SSMSC, ESC Sveriges Radio dataset	OFAI MuSpeak

The datasets can be separated into two families regarding the precision of their labels, "low-quality datasets" (LQ) and "high-quality datasets" (HQ) as presented in Table 1. The HQ datasets are labeled at the frame-level, which precisely corresponds to the goal of the speech and music detection task to classify each frame of the audio. On the other hand, the LQ datasets are labeled at the file-level, which assumes that all the frames in the audio recording have the same label. It corresponds to a low-precision version of the targeted system (*e.g.* pauses in speech are not annotated).

The datasets were split into training, validation, and test set (if not already done in the source material). HQ datasets were split into 70% training set, 20% validation set and 10% test set, and LQ datasets into 80% training set and 20% validation set, because labels are not precise enough to be used for testing. The resulting data collection contains about 78h15mn / 86h54mn / 15mn / 8h49mn of speech, music, both speech and music, and task-irrelevant data in the LQ data, and 47h12mn of audio combining all the previous categories in the HQ data, making it the most extended data resource compiled so far for the task of speech and music detection.

3.2 Architectures

Three previous sequential architectures and one novel extension will be compared in this paper. All four architectures are followed by a dense layer that reduces the number of coefficients to 2, one for music and one for speech, and by a sigmoid activation to have coefficients between 0 and 1. The range of allowed values for hyper-parameter were chosen to cover the ranges previously presented in literature, and are listed in the following tables for each architecture. In each case, a dropout randomly selected from 0.05 and 0.5 was added.

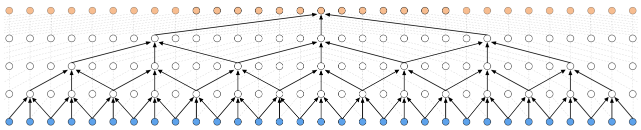

Figure 3: Visualization of a stack of dilated non-causal convolutional layers. The architecture presented in Figure 2 was made non-causal to take both past and future into account for the prediction. The kernel size had to be increased by 1.

The TCN architecture as introduced in [2] is causal. In this paper, we propose a novel extension of the TCN that takes future data into account, the non-causal TCN (ncTCN). To this end, the dilated convolutions were made non-causal, as shown in Figure 3. The use of non-causal dilated convolutions was previously shown to be successful for image processing with the Dilated Temporal Fully-Convolutional Neural Network (DTFCN) architecture [6].

Table 2: Hyperparameters for the four architectures.

Architecture 1: B-LSTM	
Num. of layers	1, 2, 3, 4
Units by layer	25, 50, 75, ... 250
Architecture 2: CLDNN	
Num. of layers	1, 2, 3
Kernel size (conv. layers)	3, 5 or 9
Number of LSTM layers	1, 2, 3
Units per LSTM layer	25, 50, 75, ... 150
Num. of fully-connected layers	1, 2, 3
Units per fully-connected layer	25, 50, 75, ... 150
Architectures 3 & 4: TCN & non-causal TCN (ncTCN)	
Num. of layers	1, 2, 3, 4
Num. of stacks	3, 4, 5, ... 10
Kernel size	3, 5, 7, ... 19
Skip some connections	true/false
Dilatations	$[2^0, 2^1, ..., 2^{N_D}]$ $N_D = 3, 4, ..., 8$
Num. of filters by layer	8, 16, 32

3.3 Comparison methodology

The study in this paper is composed of two phases: The first phase conducts a comparison of four sequential neural network architectures (B-LSTM, CLDNN, TCN, ncTCN) for the speech and music detection task. In order to facilitate the comparison in a reasonable time, two assumptions were made. First, the neural network architecture achieving the best performances on a sub-training set is likely to achieve the best performances on the total training set. And, second, the neural network architecture achieving the best performance with a restricted number of pa-

[2] http://mirg.city.ac.uk - visited 09/11/2018

rameters is likely to perform best without this restriction. Therefore, the comparison was done with a limited number of parameters and on a sub-dataset. A Bayesian hyper-parameter optimization with a Tree of Parzen Estimators (TPE) surrogate [3] was performed for each architecture on the *OFAI* and the *MuSpeak* datasets, with the number of hyper-parameters restricted to 1 million. The best set of hyper-parameters was then used to train each architecture over more epochs to get a final validation loss, and the architecture achieving the lowest validation loss was kept for the second phase.

In the second phase, the best-performing architecture from the first phase is further optimized without limiting the number of hyper-parameters. This hyper-parameter optimization was done on *OFAI*, *MuSpeak*, and *ESC* to have a more balanced dataset between music, speech, and task-irrelevant content. Then the architecture was trained on more training data and evaluated on the test set. However, due to the heterogeneity in the quality of the labels, explained in Section 3.1, only the high-quality datasets represent the target of the network. Therefore, four strategies to take advantage of both HQ and LQ data were compared. The four strategies were to (1) train the network on the high-quality datasets, (2) to train the network on the low-quality datasets, (3) to train the network on both the low and the high-quality datasets at the same time, and (4) to pre-train the network on the low-quality datasets and then train on the high-quality datasets to fine-tune the parameters.

The final system was evaluated on two different test sets. The first test set is the in-house test set described in subsection 3.1 and it shows the generalization of the system on similar data. In order to compare the system with several state-of-the-art algorithms, and to assess the generalization of the system on data different than the one used for the training, the dataset number 2 of the 2018 MIREX Competition [1] [11] was used as a second test set.

The evaluation methods and parameters from MIREX 2018 were applied for comparisons, using the implementation of sed_eval [21]. The segment-level evaluation was conducted with segments of 10ms. The event-level evaluation was performed with a tolerance time-window size of 500ms on on-set only and both on-set and off-set. Precision, Recall, and F-measure were computed for segment level (P_s, R_s, F_s), and event-level (P_e, R_e, F_e).

3.4 Pre-processing

Before the training, the audio samples were re-sampled to 22.05 kHz mono audio samples split into files of 90 s. Then a Short Time Fourier Transform (STFT) with a Hann window, a frame length of 1024 and a hop size of 512 samples was computed. Only the squared magnitude (the power spectrum) was kept and saved for the training. During the training, data augmentation was applied to the saved spectrograms and then, a Mel-filterbank with 80 triangular filters between 27.5 Hz and 8 kHz was applied. Finally, the data were put on a logarithmic scale and normalized to a zero mean and unit variance over the training set.

The data augmentation pipeline applied to each spectrogram used the implementation and paramatrisation of [29], applying time stretching, pitch shifting, Gaussian filtering, loudness manipulation, and block mixing. Data from the HQ datasets and LQ data labeled as task-irrelevant content was augmented without block mixing. Otherwise, couples of speech and of music spectrograms were created, passed individually in the data augmentation pipeline and then each couple was mixed together with random overlap. It helps to have a more balanced dataset by artificially creating overlaps between speech and music.

Broadcast audio is characterized by overlaps between speech and music, for instance in jingles, commercial ads, and transitions between pieces of music. Data augmentation helps to obtain larger overlaps that resemble this characteristics of broadcast audio.

3.5 Training

To allow parallel computation and to speed up the backward pass, mini-batches are used with a sequence length of 270 and a batch size of 32. Binary cross-entropy was minimized during training, and stochastic gradient descent with momentum $m = 0.9$ [25] was used. When the validation loss did not improve after three epochs, the learning rate was divided by 10. Dropout [32] was used and the training was stopped whenever the validation loss had not improved in 5 consecutive epochs.

3.6 Post-processing

A threshold of 0.5 was applied to the output of the networks. A simple strategy was applied to smooth the output and delete spurious breaks or events. To this end, thresholds for the minimal duration of speech (Dur_{sp}) and music (Dur_{mus}) events were defined, respectively. Furthermore, thresholds for the minimal duration of a break in speech (Brk_{sp}) and music (Brk_{mus}) were defined. In order to specify values for these four duration thresholds, the training set was analyzed to obtain statistics on the lengths of the events and the breaks. To choose the best values between the relevant values found with the analysis, each set of values was evaluated on the validation set and the set achieving the best performances was selected. The effect of post-processing will be analyzed separately in the results.

3.7 Implementation

The implementation is done with *Keras* [9] using TensorFlow as a backend [1] and the library *keras-tcn* [3] is used for the TCN implementation. The code is provided as a new framework for speech and music detection that allows comparison between configurations with different architectures, datasets and hyper-parameters. The implementation has been made available on GitHub. [4]

[3] https://github.com/philipperemy/keras-tcn

[4] https://bit.ly/2XcuzsJ

4. RESULTS

4.1 Comparison

After each hyper-parameter optimization, the best configuration has been trained until the early-stopping. Figure 4 shows the micro-averaged [36] ROC curve of the 4 different architectures. The architecture that achieved the best performances under the constraints of the experiment is the non-causal TCN. All three architectures that have not yet been applied to this task (ncTCN, TCN, CLDNN) outperform the BLSTM architecture.

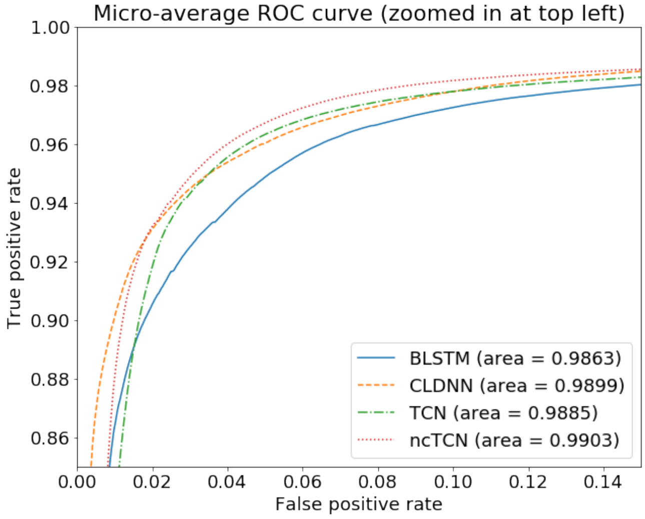

Figure 4: ROC curve micro-averaged over speech and music.

4.2 Dataset strategies

A new hyper-parameter optimization allowing for a bigger range of hyper-parameters was performed for the best-performing architecture (ncTCN). The batch size was reduced to 16 to allow the GPU to work with bigger architectures. The resulting architecture was trained with the four different strategies explained in Section 3.3. The strategy that achieved the lowest validation loss (Table 3) on the targeted high-quality dataset (HQ loss) is pre-training on the low-quality dataset, and subsequent training on the high-quality dataset (Strategy 4).

Table 3: Validation loss of the different strategies to use on high and low quality datasets.

Strategy	LQ loss	LQ/HQ loss	HQ loss
(1) LQ	**0.097**	0.125	0.232
(2) HQ	0.365	0.323	0.096
(3) LQ/HQ	0.098	**0.101**	0.136
(4) Pre-train	0.222	0.181	**0.070**

4.3 Post-processing (PP)

The high-quality training set was analyzed to set the four duration thresholds for the post-processing (see Section 3.6). The 1st, 5th and 10th percentiles were considered to obtain threshold values. For instance, in the case of the 5th percentile for the music event duration, it means that 95 % of the music events in the training set have a length exceeding the threshold. Table 4 presents the evaluation of the system with several post-processing methods based on the values from the different percentiles.

Table 4: F-measures for segment-level evaluation (F_s) and event-level evaluation (F_e) on the high-quality validation set by percentile. Underlined values denote statistically significant differences to the value one row above (paired-sample t-test, $p < 0.05$).

PP	F_s (segment)			F_e (event)		
	All	Mus.	Sp.	All	Mus.	Sp.
None	**0.973**	**0.982**	**0.951**	0.182	0.247	0.129
1st	**0.973**	**0.982**	0.950	$\underline{0.510}$	$\underline{0.589}$	$\underline{0.417}$
5th	**0.973**	**0.982**	0.950	$\underline{0.544}$	0.615	0.454
10th	$\underline{0.971}$	0.981	0.949	**0.547**	**0.617**	**0.459**

The different impact of the post-processing on the segment and on the event-level evaluation is caused by the fact that a small modification at the frame level has a limited impact on the segment-level evaluation, but it can have a significant impact on the event-level evaluation. The set of values from the 5th percentile was selected for the rest of the evaluation since it represents a good compromise between a decrease in the segment-level evaluation and an increase in the event-level evaluation.

4.4 Evaluation of the ncTCN

Table 5: Segment-level evaluation of the final system

	All	Music	Speech
F_s	0.968	0.971	0.957
P_s	0.963	0.969	0.944
R_s	0.973	0.973	0.971

The results of the segment-level and event-level evaluations on the test set 1 are presented in Table 5 and Table 6, respectively. For the segment-level evaluation, the proposed non-causal TCN system obtains an F-measure of 0.971 for the music and of 0.957 for the speech. Due to the marginal effect of post-processing on the segment level, only results with post-processing are depicted. For the event-level evaluation, the system obtains clearly superior results with the post-processing. For example, the overall F-measure on both onset and offset increases from 0.151 without post-processing to 0.417 with post-processing.

Finally, the non-causal TCN was evaluated on the dataset 2 of the 2018 MIREX competition. The results are presented in Table 7 and Table 8 and the results of the other systems are taken from the MIREX website[1]. For the segment-level evaluation, the system obtains the best F-measure (0.946) on speech. On music, the system obtains an F-measure of 0.879 and the best system of the

Table 6: Event-level evaluation of the final system. Underlined values denote statistically significant difference to the value without pre-processing (paired-sample t-test, $p < 0.05$).

		Ons.			On/Offs.		
PP		All	Mus.	Sp.	All	Mus.	Sp.
No	F_e	0.248	0.248	0.248	0.151	0.177	0.115
	P_e	0.153	0.146	0.165	0.930	0.104	0.760
	R_e	**0.650**	**0.828**	**0.499**	**0.394**	0.587	**0.231**
Yes	F_e	$\underline{\mathbf{0.653}}$	$\underline{\mathbf{0.782}}$	$\underline{\mathbf{0.519}}$	$\underline{\mathbf{0.417}}$	$\underline{\mathbf{0.590}}$	$\underline{\mathbf{0.236}}$
	P_e	$\underline{\mathbf{0.733}}$	$\underline{\mathbf{0.778}}$	$\underline{\mathbf{0.671}}$	$\underline{\mathbf{0.447}}$	$\underline{\mathbf{0.587}}$	$\underline{\mathbf{0.305}}$
	R_e	$\underline{0.589}$	$\underline{0.787}$	$\underline{0.422}$	0.376	$\underline{\mathbf{0.593}}$	0.192

competition obtains 0.923. For the event-level evaluation, the system obtains the best F-measure of 0.162 for the onset evaluation of the music. For the other cases, the system does not come first but achieves good results and comes second in 2 of the 3 remaining cases.

Table 7: Comparison with other algorithms on test set 2 of MIREX 2018 for segment-level evaluation. Architectures [8](a, b, c) use a Multi-layer Perceptron to classify both Mel-Frequency Cepstral Coefficient and features extracted by a SampleCNN [17] architecture. Architectures [20](a) are based on a logistic regression classifier and architectures [20](b, c) are based on a Deep Residual Network [15].

	Music			Speech		
Algo.	F_s	P_s	R_s	F_s	P_s	R_s
[8, a]	0.786	0.813	0.760	0.846	0.967	0.751
[8, b]	0.759	0.768	0.750	0.789	**0.975**	0.663
[8, c]	0.797	0.797	0.797	0.823	0.964	0.718
[20, a]	**0.923**	0.977	0.875	0.933	0.913	0.953
[20, b]	0.916	0.925	0.907	0.914	0.933	0.896
[20, c]	0.879	**0.979**	0.797	0.897	0.829	**0.978**
ncTCN	0.879	0.790	**0.990**	**0.946**	0.949	0.943

Table 8: F-measure comparison with other algorithms on the test set 2 of MIREX 2018 for the event-level evaluation and a tolerance time-window of 500ms.

Algorithm	Music		Speech	
	Ons.	On/Offs.	Ons.	On/Offs.
[8, a]	0.087	0.023	**0.223**	**0.077**
[8, b]	0.073	0.020	0.192	0.051
[8, c]	0.068	0.015	0.206	0.052
[20, a]	0.141	0.016	0.063	0.002
[20, b]	0.154	**0.031**	0.116	0.021
[20, c]	0.152	0.022	0.080	0.015
ncTCN	**0.162**	0.0169	0.216	0.070

5. DISCUSSION

Based on the curves from Figure 4, our experiments suggest that the TCN-based architectures are outperforming the RNN-based architectures. Moreover, the TCN-based architectures train faster than the RNN-based architectures at similar sizes (around 80s/epoch for the TCN-based architectures and 340s/epoch for the RNN-bases architectures), results that corroborate conclusions [2] in different task contexts. The proposed non-causal TCN achieves better results than the causal TCN.

We also demonstrated that pre-training the neural network on a low-quality dataset prior to training it on the high-quality dataset improves the validation loss. Table 3 highlights the difference between the different strategies and it shows that the HQ loss function goes down from 0.096 to 0.070 by pre-training the network on low-quality datasets, it represents a relative improvement of 27 %. The analysis of the training set provides relevant thresholds for the post-processing method. Those values allow an improvement for the event-level evaluation on the validation set without harming the segment-level evaluation. Results on the test set (Table 5 and 6) confirm this analysis made on the validation set. Instead of applying some thresholds, more sophisticated probabilistic models may further improve the post-processing.

For the event-level evaluation, the overall results of all algorithms are much lower compared to the segment-level results. It shows one of the limits of the event-level evaluation: it is difficult to standardize precise boundaries for the events and especially the speech events. The results of the event-level evaluation highly depend on the rules that have been chosen during the annotation of the training set. Therefore, the segment-level evaluation might be more relevant to compare different algorithms on a test set with unknown annotation rules.

End-to-end learning may be considered an important area for future research. Pre-training an end-to-end learning solution on low-quality datasets and fine-tuning it on high-quality datasets might be a viable solution to overcome the expensive price of labeling data. This method was shown to be successful in this study and it might be suitable for other tasks.

6. CONCLUSION

In this paper, various architectures have been compared in a speech and music detection task. The findings are consistent with previous studies, demonstrating that convolutional architectures can yield better performance and are faster to train than RNN-based architectures for sequence modeling. Furthermore, the novel non-causal TCN can improve performance when real-time computation is not a constraint. Low-quality data was successfully used to improve the system performance on high-quality data. It provided a better starting point for the learning phase and it converged faster towards a lower minimum. Through the MIREX evaluation, the final system has demonstrated to perform well in relation to the state of the art. This encourages further exploration of TCN and non-causal TCN architectures for sequence modeling tasks.

7. ACKNOWLEDGEMENTS

This work was supported by the NordForsk's Nordic University Hub "Nordic Sound and Music Computing Network - NordicSMC" (proj. nr. 86892), and conducted in collaboration with Swedish Radio.

Special thanks to Christofer Bustad, Paul Nygren, and Ragnar Schön from Sveriges Radio for their help and advice during this project.

8. REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, Berkeley, CA, USA, 2016. USENIX Association.

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.

[3] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 2546–2554, USA, 2011. Curran Associates Inc.

[4] Emre Cakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, Tuomas Virtanen, Emre Cakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 25(6):1291–1303, June 2017.

[5] Michael J Carey, Eluned S Parris, and Harvey Lloyd-Thomas. A comparison of features for speech, music discrimination. In *IEEE International Conference on Acoustics, Speech, and Signal Processing.*, volume 1, pages 149–152, 1999.

[6] Noshaba Cheema, Somayeh Hosseini, Janis Sprenger, Erik Herrmann, Han Du, Klaus Fischer, and Philipp Slusallek. Dilated temporal fully-convolutional network for semantic segmentation of motion capture data. *CoRR*, abs/1806.09174, 2018.

[7] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE, 2017.

[8] Minsuk Choi, Jongpil Lee, and Juhan Nam. Hybrid features for music and speech detection. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2018.

[9] François Chollet et al. Keras. `https://keras.io`, 2015.

[10] Heinrich Dinkel, Yanmin Qian, and Kai Yu. Investigating raw wave deep neural networks for end-to-end speaker spoofing detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2002–2014, 2018.

[11] David Doukhan, Eliott Lechapt, Marc Evrard, and Jean Carrive. Ina's MIREX 2018 music and speech detection system. In *Music Information Retrieval Evaluation eXchange (MIREX 2018)*, 2018.

[12] Lars Ericsson. *Automatic Speech/music Discrimination in Audio Files*. M.Sc. thesis, KTH Royal Institute of Technology, 2010.

[13] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[14] T. Grill and J. Schlüter. Two convolutional neural networks for bird detection in audio signals. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1764–1768, Aug 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] Lars Hertel, Huy Phan, and Alfred Mertins. Comparing time and frequency domain for audio event recognition using deep learning. In *Neural Networks (IJCNN), 2016 International Joint Conference*, pages 3407–3411. IEEE, 2016.

[17] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *CoRR*, abs/1703.01789, 2017.

[18] S. Leglaive, R. Hennequin, and R. Badeau. Singing voice detection with deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, April 2015.

[19] Thomas Lidy and Alexander Schindler. CQT-based convolutional neural networks for audio scene classification. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, pages 60–64, September 2016.

[20] Matija Marolt. Music/speech classification and detection submission for MIREX 2018. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2018.

[21] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162, 2016.

[22] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Recurrent neural networks for polyphonic sound event detection in real life recordings. *CoRR*, abs/1604.00861, 2016.

[23] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[24] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik M Schmidt, Andreas F Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018.

[25] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[26] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference*, pages 4580–4584. IEEE, 2015.

[27] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, March 2017.

[28] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1331–1334 vol.2, April 1997.

[29] Jan Schlüter and Thomas Grill. Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015.

[30] Jan Schlüter and Reinhard Sonnleitner. Unsupervised feature learning for speech and music detection in radio broadcasts. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*, pages 369–376, York, UK, September 2012.

[31] David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A Music, Speech, and Noise Corpus, 2015. arXiv:1510.08484v1.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[33] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool. Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Detection. *ArXiv e-prints*, April 2016.

[34] George Tzanetakis and Perry Cook. Marsyas: A framework for audio analysis. *Org. Sound*, 4(3):169–175, December 1999.

[35] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, page 125, 2016.

[36] Yiming Yang. An evaluation of statistical approaches to text categorization. *Inf. Retr.*, 1(1-2):69–90, May 1999.

[37] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4845–4849. IEEE, 2017.