

2021-05-07

# Artificially Synthesising Data for Audio Classification and Segmentation to Improve Speech and Music Detection in Radio Broadcast

Venkatesh, S

<http://hdl.handle.net/10026.1/16930>

---

10.1109/ICASSP39728.2021.9413597

ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

IEEE

---

*All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.*

# ARTIFICIALLY SYNTHESISING DATA FOR AUDIO CLASSIFICATION AND SEGMENTATION TO IMPROVE SPEECH AND MUSIC DETECTION IN RADIO BROADCAST

*Satvik Venkatesh\**, *David Moffat\**, *Alexis Kirke\**, *Gözel Shakeri†*, *Stephen Brewster†*,  
*Jörg Fachner‡*, *Helen Odell-Miller‡*, *Alex Street‡*, *Nicolas Farina††*,  
*Sube Banerjee‡‡*, and *Eduardo Reck Miranda\**

\* Interdisciplinary Centre for Computer Music Research, University of Plymouth, UK

† School of Computing Science, University of Glasgow, UK

‡ Cambridge Institute of Music Therapy Research, Anglia Ruskin University, UK

†† Centre for Dementia Studies, Brighton and Sussex Medical School, UK

‡‡ Faculty of Health, University of Plymouth, UK

## ABSTRACT

Segmenting audio into homogeneous sections such as music and speech helps us understand the content of audio. It is useful as a pre-processing step to index, store, and modify audio recordings, radio broadcasts and TV programmes. Deep learning models for segmentation are generally trained on copyrighted material, which cannot be shared. Annotating these datasets is time-consuming and expensive and therefore, it significantly slows down research progress. In this study, we present a novel procedure that artificially synthesises data that resembles radio signals. We replicate the workflow of a radio DJ in mixing audio and investigate parameters like fade curves and audio ducking. We trained a Convolutional Recurrent Neural Network (CRNN) on this synthesised data and outperformed state-of-the-art algorithms for music-speech detection. This paper demonstrates the data synthesis procedure as a highly effective technique to generate large datasets to train deep neural networks for audio segmentation.

**Index Terms**— Audio Segmentation, Audio Classification, Music-speech Detection, Training Set Synthesis, Deep Learning

## 1. INTRODUCTION

Automatically understanding the content of audio data is useful for indexing audio archives, target-based distribution of media, speech recognition, and intelligent remixing. It includes the task of audio segmentation, which divides an audio signal into homogeneous segments. These segments contain audio classes like music, speech, environmental sounds, and noise, to name but a few. The specificity of audio classes depends on the application. For instance, in radio broadcast, some relevant audio classes include music, speech, noise, and silence [1].

Primarily, there are two approaches to audio segmentation — (1) distance-based segmentation and (2) segmentation-by-classification [2]. In the former, boundaries of acoustic events are directly detected. This is done by calculating a distance metric, such as Euclidean distance, Bayesian information criterion (BIC) [3], or generalized likelihood ratio (GLR). For a given audio, a distance curve

is plotted. The peaks on this distance curve are associated with the boundaries of audio events because they comprise high acoustic changes. The advantage of this technique is that it is generally unsupervised and does not require knowledge of the individual audio classes. However, the disadvantage is that it is more sensitive to dissimilarities within each audio class.

In segmentation-by-classification, as the name suggests, the audio is divided into individual frames, typically in the range of 10 to 25 ms. These frames are independently classified and eventually the boundaries of audio events are detected. Traditionally, this was performed through algorithms like Gaussian mixture model (GMM) [4], support vector machine (SVM), and factor analysis (FA) [5]. In recent years, due to the advances in deep learning, segmentation-by-classification has gained more popularity through neural network architectures like bidirectional long short-term memory (B-LSTM) [6], Convolutional Recurrent Neural Network (CRNN) [7], and Temporal Convolutional Neural Network (TCN) [8].

Machine learning models are generally trained using proprietary audio such as television and radio broadcast. This imposes a serious hindrance in the repeatability of research because this audio cannot be shared across different research groups. Annotating these datasets is a time-consuming and expensive task. For example, the study by Schlüter et al. [9] annotated 42 hours of radio broadcast with the help of paid students. Moreover, a dataset called Open Broadcast Media Audio from TV (OpenBMAT) [10] was cross-annotated by three different annotators and each of them spent approximately 130 hours to annotate 27.4 hours of audio. As the labels in these datasets need to be precise enough for the models to train, the annotations in such datasets are generally verified by at least one other person. These factors impose many challenges for a researcher who wants to freshly explore audio segmentation.

The literature comprises many datasets that contain individual files of music and speech. However, these files are different from broadcast audio because broadcast audio is well-mixed. To our knowledge, the only openly available annotated database for this task is the MuSpeak dataset [11], which contains approx. 5 hours of audio. Moreover, OpenBMAT [10] focused on estimating the relative loudness of music, but not speech and music detection.

In this paper, we present a novel approach to artificially synthesise audio that resembles a radio broadcast. We replicate the process of a radio DJ in mixing audio content. This was done by in-

---

This study was supported by Engineering and Physical Sciences Research Council (EPSRC) grants EP/S026991/1, EP/S027491/1, EP/S027203/1, and EP/S026959/1.

investigating fade curves, audio ducking, fade durations, and silences. The artificially mixed audio only uses openly available music-speech datasets that contain individual files of music and speech. Using this data synthesis procedure, large amounts of training data can be generated to train deep neural networks. The trained models are useful for real-world applications and achieve state-of-the-art performance on human-labelled datasets. The implementation, code, and pre-trained models associated with this study are openly available in this GitHub repository<sup>1</sup>.

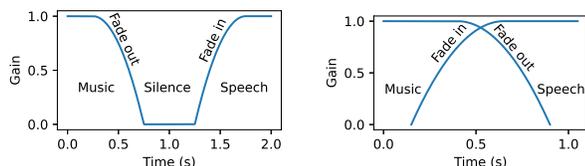
## 2. DATA SYNTHESIS

### 2.1. Datasets

In this study, we used datasets that contain audio files labelled as either music or speech. We did not use data that contains mixed audio. Instead, the radio content was synthesised through combining and mixing the music and speech data together. We used the MUSAN corpus [12], GTZAN music and speech detection dataset [13], and the Scheirer & Slaney dataset [14]. These are the commonly used datasets in music-speech detection studies [8]. When we conducted initial tests with our neural network, we observed that there were confusions between wind instruments like flute and speech. Additionally, some vocal sections without accompaniment were confused with speech. Therefore, we extended our data repository to using the Instrument Recognition in Musical Audio Signals (IRMAS) dataset that includes many examples of wind instruments [15], GTZAN genre recognition [16] for additional music examples, Singing Voice Audio Dataset which contains unaccompanied vocals [17], and a section of the LibriSpeech corpus [18] for more speech examples. We also considered noise examples from the MUSAN corpus to enable the neural network to detect task-irrelevant examples. These are sounds that cannot be labelled as either music or speech. For instance, environmental sounds, babble noise, unintelligible speech, footsteps, and so on. The total number of audio files for music, speech, and noise was 6876, 6885, and 665 respectively.

### 2.2. Audio Transitions

In radio programmes, shifts between music and speech and vice versa are generally smoothed through transitions. We broadly observed two types of transitions, which we have termed as normal transition and cross-fade transition. In a normal transition, an audio event is faded out, followed by a short period of silence, and then a new audio event is faded in. An example can be found in figure 1a. In a cross-fade transition, as the name suggests, the two audio signals are overlapping. While one is fading out, the other is fading in, as shown in figure 1b.



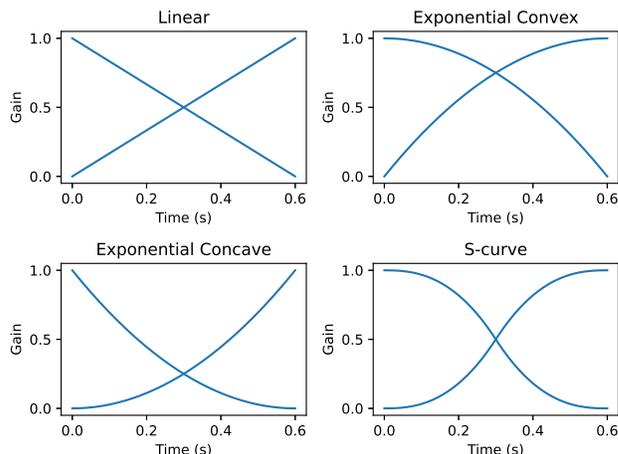
(a) Normal fade transition

(b) Cross-fade transition

**Fig. 1.** Two types of audio transitions

### 2.3. Fade Curves

During audio mixing, engineers use different fade curves depending on the context. We have considered four popular curves that are commonly used in mixing [19] — linear, exponential convex, exponential concave, and s-curve. Figure 2 illustrates the types of fade curves.



**Fig. 2.** Four types of fade curves

### 2.4. Parameters of Transitions

Each audio example that was synthesised in this study was 8 s long. We felt that 8 s was a long enough duration to capture the entire fade curve. The three audio classes — speech, music, and noise were stored in different directories. Each time an audio class was chosen for the data synthesis, a random file was selected from the entire list of files and then a random segment was extracted.

For an audio transition, the two audio events and a time stamp of the transition are randomly chosen. For example, music to speech, speech to noise, speech to speech, etc., transitioning at a specific time. Note that we also allowed repetition of the same audio class, such as speech to speech because this would suggest cases like interviews.

To cover a wide range of possibilities that can occur while mixing radio programmes, we randomised the various parameters of audio transitions. Each time an audio example was synthesised, a random fade curve was chosen. Subsequently, a random fade duration was chosen from a uniform distribution ranging from 0 s to the maximum possible duration. For a normal fade transition, the gap of silence between audio events was randomised.

### 2.5. Background Music

In radio programmes, it is very common to have background music playing alongside foreground speech. Audio ducking is the process of reducing the volume of background music. It is generally performed to make speech intelligible. Many radio broadcasters have their own guidelines to audio ducking [20]. Therefore, in order to artificially synthesise audio examples with background music, it is important for us to consider these guidelines.

We adopted the integrated loudness metric by ITU BS.1770-4 [21] to calculate the loudness of audio. This is measured in loud-

<sup>1</sup><https://github.com/satvik-venkatesh/audio-seg-data-synth/>

ness units (LU). There is no ideal loudness difference (LD) between speech and background music because it is highly subjective. Commonly, the literature recommends a minimum LD of 7 to 10 LU [20]. Moreover, in cases of very quiet background music, the LD can be as high as 23 LU.

In order to implement our data synthesis procedure, we require a minimum and maximum LD to choose random values from a uniform distribution. We empirically observed the average performance of the network over multiple training cycles on different LDs and also manually listened to synthesised audio examples. We set the LD range to be between 7 and 18 LU.

In radio programmes, audio ducking can be performed through either volume automation or side-chain compression. We chose the former technique because it was relatively straightforward to achieve accurate LDs during data synthesis.

## 2.6. Overview

There are different combinations of audio classes that occur in the synthesised examples — music, speech, noise/silence, and speech over background music. Each example can either have no transitions (that is purely a single audio class) or one transition (that is two audio classes connected through fade curves) with a probability of 0.5. An overview of the data synthesis procedure can be found in figure 3.

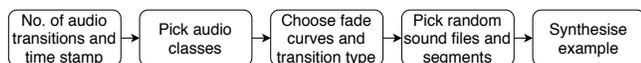


Fig. 3. An overview of the data synthesis procedure.

## 3. EXPERIMENTS

### 3.1. Pre-processing and Feature Extraction

All audio files used in this study were resampled to 22.05 kHz mono signals. Silences in the audio files were removed/shortened by using Sound eXchange (SoX). For our data synthesis to work smoothly, we required the audio files to have a minimum duration of 8 s. Some datasets such as the IRMAS have only 3 s audio files. To address this, we looped the audio to obtain the required duration.

Mel spectrograms have been commonly adopted by audio segmentation studies to extract features [6, 8]. We set the hop size to 220 (10 ms) and FFT size to 1024 (46 ms). The selected audio segments were peak-normalised before synthesising an example. After synthesis, the whole example was peak-normalised again. We extracted 80 log-scale-Mel bands from 64 Hz to 8 kHz.

### 3.2. Validation and Test Sets

We are using artificially synthesised data for training. As these are not real-world examples, we cannot use synthesised data for validation and testing. We incorporated the MuSpeak dataset, which contains 5 h 14 m 14 s of audio. We also collected 9 h of broadcast audio from BBC Radio Devon, which was manually annotated by the authors. The audio in the BBC dataset were split into files of 1 h. Three hours of our annotations were verified by an external audio mixing engineer who was not involved in the research and paid for his time. Additionally, a random section of 15 minutes was blind-annotated by him independently. We found an agreement of 99.49% with our annotations by using 10 ms segment verifications. This was

done to ensure that audio events were similarly perceived by different people. In order to explore the robustness of data synthesis, we did not use any of this data for training. The data from MuSpeak and BBC dataset was shuffled and used as validation and test sets as a 50-50% split.

In order to compare our model with other state-of-the-art algorithms, we also evaluated it on dataset number 1 of the Music Information Retrieval Evaluation eXchange (MIREX) 2018 music and speech detection competition<sup>2</sup>. This dataset contains 27 hours of audio from various TV programmes. Although our data synthesis was designed for radio programmes, this dataset would provide us with a good evaluation of our model.

### 3.3. Network Architecture

For this study, we adopted a CRNN, which is a state-of-the-art architecture for audio classification and segmentation tasks [7, 22]. The input shape of the network was  $802 \times 80 \times 1$ , equivalent to 802 time steps and 80 Mel bins. The output of the network comprised  $802 \times 2$  neurons with sigmoid activations, where two neurons perform a binary classification for music and speech at every time step. The network performs multi-output detection, independently detecting the regions of music and speech. This is important for models working with radio data because music and speech can occur simultaneously. Binary cross-entropy was used as the loss function.

We used the Adam optimizer with a constant learning rate of 0.001 and batch size of 128. The first two layers of the network were 2D convolutional layers with a kernel size of 7 and a stride of 1. The input was padded with zeros such that ‘same’ convolutions were performed to ensure that the time resolution remains the same. The next two layers were bidirectional gated recurrent units (B-GRU) with 80 units each.

In this study, we evaluated the model using different training sets, as explained in section 3.4. Hence, a model architecture was finalised by optimising the performance across different training datasets. For regularisation, we implemented early-stopping and used batch normalisation after all the layers. Max pooling along the dimension of Mel bins was performed after the convolutional layers. A dropout of 0.2 was added only after the convolutional layers because we observed that it was not effective for the B-GRU layers.

### 3.4. Training Datasets

In order to evaluate the effectiveness of our data synthesis algorithm, we constructed 4 training datasets. All datasets contain 40960 examples of 8 s audio (which is approximately 91 h of audio). Initial tests conveyed this was an adequate number of examples to train the network.

1. Dataset-only files (d-OF): This dataset contains audio segments of only speech, music, or noise. There was no mixing of audio events within each example. 40960 examples were randomly sampled from our data repository. We did not include the whole corpus because of computational limitations and to manage redundancy.
2. Dataset-only files and background music (d-OFB): In addition to d-OF, this dataset contains examples of speech over background music. The volume of background music was normalised according to the method explained in section 2.5. However, this dataset did not contain any audio transitions.

<sup>2</sup>[https://music-ir.org/mirex/wiki/2018:Music\\_and/or\\_Speech\\_Detection](https://music-ir.org/mirex/wiki/2018:Music_and/or_Speech_Detection)

- Dataset-no normalisation (d-NN): In this dataset, the data synthesis was performed as explained in section 2, except for the loudness normalisation of background music according to loudness of foreground speech. However, all examples of speech, music, and noise were peak-normalised before synthesis.
- Dataset-data synthesis (d-DS): In this dataset, the data synthesis was performed exactly as explained in section 2.

### 3.5. Post-processing

A threshold of 0.5 was used to make binary classifications on the output layer. The length of each file in the test set was approximately 1 h. We traversed the audio file with a window size of 8 s and hop size of 6 s. We discarded the predictions made on the first and last second of each audio example because they might be unreliable. This technique was adopted from the study by Gimeno et al. [6].

In the audio segmentation pipeline, predictions made by the model are generally sent through a post-processing phase to remove spurious transitions and events. This is done through either median filtering [6, 9] or setting thresholds for minimum durations of audio events [8]. We adopted the latter approach and set thresholds for minimum speech duration, minimum music duration, maximum silence between speech, and maximum silence between music. These values were obtained from the study by Lemaire et al. [8] and set to 1.3 s, 3.4 s, 0.4 s, and 0.6 s respectively.

## 4. RESULTS

To evaluate the models, we adopted metrics implemented in the `sed_eval` toolbox [23], which has been widely adopted by audio event detection studies [8, 22, 24]. The segment-level evaluation was performed with a segment size of 10 ms. Table 1 presents the model’s performance on different datasets. The highest overall F-measure was obtained by d-DS, which implemented the entire data synthesis procedure. The F-measures of d-OF and d-OFB were at least 3% lower than d-DS because their datasets did not contain audio transitions. This demonstrates that modelling radio DJ-like transitions is an effective technique. Additionally, there is a marginal difference between d-OF and d-OFB, which explains that adding background music to speech in the training examples is not sufficient, but there needs to be audio transitions.

The dataset d-NN contained background music that was peak-normalised, but not normalised with respect to loudness of foreground speech. Therefore, music F-measure of d-DS surpasses the value of d-NN by more than 2%. This proves that randomising the loudness of background music with respect to foreground speech within a LD of 7 to 18 was an effective method. Speech F-measure for d-NN was slightly greater than d-DS. However, this might be because the background music in d-NN was at a relatively constant volume, which improves speech detection but compromises music detection.

Dataset	F <sub>overall</sub>	F <sub>s</sub>	F <sub>m</sub>
d-OF	93.54	94.58	92.99
d-OFB	93.68	94.95	92.99
d-NN	95.33	<b>96.44</b>	94.73
d-DS	<b>96.69</b>	96.17	<b>96.97</b>

**Table 1.** The F-measure of our CRNN model trained on different datasets.

Table 2 shows the segment-level evaluation of our d-DS model on the MIREX speech and music detection dataset. The evaluations of other submissions were obtained from the MIREX website. Our model significantly outperforms the other models for F-measures of music. This is attributed to the presence of audio transitions and loudness normalisation of background music in the synthesised dataset. Our model also obtains the highest F-measure for speech detection.

All the other submissions in the competition used real-world data [25, 26]. Therefore, these results demonstrate that our data synthesis is a highly effective approach for audio segmentation. Moreover, there was another task in MIREX 2018 that was solely for music detection. Our model places second in this task, preceded by the submission by Meléndez-Catalán et al. [27]. Their model was trained on 30 hours of TV programmes, which comes from the same data distribution. It is important to note that the MIREX evaluation dataset can contain background music over foreground speech, audience noises, sound effects, everyday-life sounds, sounds of the city, and so on. As our data synthesis procedure only considered foreground speech, it explains the poor precision for music in table 2. Our model predicted many of the sound effects as music. The performance of our model over TV programmes can be improved by considering these factors in the data synthesis.

Algo.	F <sub>m</sub>	P <sub>m</sub>	R <sub>m</sub>	F <sub>s</sub>	P <sub>s</sub>	R <sub>s</sub>
[25]	49.36	62.4	40.82	77.18	<b>96.83</b>	64.15
[26, a]	38.99	80.72	25.7	91.15	87.95	94.6
[26, b]	54.78	85.7	40.26	90.9	89.45	92.41
[26, c]	31.24	<b>98.73</b>	18.56	90.86	83.83	<b>99.17</b>
d-DS	<b>85.76</b>	79.37	<b>93.27</b>	<b>92.21</b>	89.71	94.85

**Table 2.** F-measure, precision, and recall of our CRNN model trained on ‘d-DS’ and other algorithms evaluated on dataset number 1 of MIREX 2018 speech and music detection competition.

## 5. CONCLUDING DISCUSSIONS

In this study, only artificially synthesised data was used to train a model for audio segmentation and classification. We adopted a training dataset belonging to a different distribution from the validation and test sets. Despite this, we obtained a high F-measure on our local test set. Furthermore, we obtained state-of-the-art performance for speech and music detection on the MIREX 2018 competition dataset.

There were noticeable differences between the BBC Radio Devon recordings and the data repository we have used for data synthesis. The BBC recordings have greater dynamic range compression, cleaner speech, and generally use side-chain compression for audio ducking. Therefore, including a small number of radio recordings in the training dataset might improve the model’s performance. Additionally, incorporating audio effects like dynamic range compression in the data synthesis pipeline might improve the model’s performance.

Many studies have suggested end-to-end deep learning to be a potential pathway for future audio classification and segmentation research [8, 28]. However, it requires much more data than using Mel spectrograms as features. As labelling large amounts of data is an expensive and time-consuming task, our data synthesis procedure serves as a potential solution to generate large amounts of training data and advance the state-of-the-art in audio segmentation and classification systems.

## 6. REFERENCES

- [1] Theodoros Theodorou, Iosif Mporas, and Nikos Fakotakis, “An overview of automatic audio segmentation,” *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 6, no. 11, pp. 1, 2014.
- [2] Taras Butko and Climent Nadeu, “Audio segmentation of broadcast news in the albayzin-2010 evaluation: overview, results, and discussion,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2011, no. 1, pp. 1, 2011.
- [3] Hao Xue, HaiFeng Li, Chang Gao, and ZiQiang Shi, “Computationally efficient audio segmentation through a multi-stage BIC approach,” in *3rd International Congress on Image and Signal Processing*. IEEE, 2010, vol. 8, pp. 3774–3777.
- [4] Marko Kos, Matej Grasic, Damjan Vlaj, and Zdravko Kacic, “On-line speech/music segmentation for broadcast news domain,” in *2009 16th International Conference on Systems, Signals and Image Processing*. IEEE, 2009, pp. 1–4.
- [5] Diego Castán, Alfonso Ortega, Antonio Miguel, and Eduardo Lleida, “Audio segmentation-by-classification approach based on factor analysis in broadcast news domain,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, pp. 34, 2014.
- [6] Pablo Gimeno, Ignacio Viñals, Alfonso Ortega, Antonio Miguel, and Eduardo Lleida, “Multiclass audio segmentation based on recurrent neural networks for broadcast domain data,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2020, no. 1, pp. 1–19, 2020.
- [7] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho, “Convolutional recurrent neural networks for music classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.
- [8] Quentin Lemaire and Andre Holzapfel, “Temporal convolutional networks for speech and music detection in radio broadcast,” in *20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.
- [9] 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)*, 2012.
- [10] Blai Meléndez-Catalán, Emilio Molina, and Emilia Gómez, “Open broadcast media audio from TV: A dataset of tv broadcast audio with relative music loudness annotations,” *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, 2019.
- [11] MuSpeak Team, “MIREX muspeak sample dataset,” 2015, <http://mirg.city.ac.uk/datasets/muspeak/> [Last accessed on 13-10-2020].
- [12] David Snyder, Guoguo Chen, and Daniel Povey, “Musan: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.
- [13] George Tzanetakis and Perry Cook, “Marsyas: A framework for audio analysis,” *Organised sound*, vol. 4, no. 3, pp. 169–175, 2000.
- [14] Eric Scheirer and Malcolm Slaney, “Construction and evaluation of a robust multifeature speech/music discriminator,” in *IEEE International Conference on Acoustics, Speech, and Signal processing (ICASSP)*, 1997, vol. 2, pp. 1331–1334.
- [15] Juan J Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera, “A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals,” in *13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 559–564.
- [16] George Tzanetakis and Perry Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [17] Dawn A. A. Black, Ma Li, and Mi Tian, “Automatic identification of emotional cues in chinese opera singing,” in *13th International Conference on Music Perception and Cognition and the 5th Conference for the Asian-Pacific Society for Cognitive Sciences of Music*, 2014.
- [18] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “LibriSpeech: an ASR corpus based on public domain audio books,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.
- [19] Eric Tarr, *Hack Audio: An Introduction to Computer Programming and Digital Signal Processing in MATLAB*, Routledge, 2018.
- [20] Matteo Torcoli, Alex Freke-Morin, Jouni Paulus, Christian Simon, and Ben Shirley, “Background ducking to produce esthetically pleasing audio for TV with clear speech,” in *146th Audio Engineering Society Convention*, 2019.
- [21] ITU-R, “ITU-R Rec. BS.1770-4: Algorithms to measure audio programme loudness and true-peak audio level,” 2017.
- [22] Emre Cakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.
- [23] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, pp. 162, 2016.
- [24] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen, “DCASE 2017 challenge setup: Tasks, datasets and baseline system,” in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.
- [25] Minsuk Choi, Jongpil Lee, and Juhan Nam, “Hybrid features for music and speech detection,” *Music Information Retrieval Evaluation eXchange (MIREX)*, 2018.
- [26] Matija Marolt, “Music/speech classification and detection submission for MIREX 2018,” *Music Information Retrieval Evaluation eXchange (MIREX)*, 2018.
- [27] Blai Meléndez-Catalán, E Molina, and E Gomez, “Music and/or speech detection MIREX 2018 submission,” *Music Information Retrieval Evaluation eXchange (MIREX)*, 2018.
- [28] Jongpil Lee, Taejun Kim, Jiyoung Park, and Juhan Nam, “Raw waveform-based audio classification using sample-level CNN architectures,” *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.