Hung *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2022) 2022:21
https://doi.org/10.1186/s13636-022-00253-8

EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH Open Access

A large TV dataset for speech and music activity detection

Yun-Ning Hung[1*], Chih-Wei Wu[2], Iroro Orife[2], Aaron Hipple[2], William Wolcott[2] and Alexander Lerch[1]

Abstract

Automatic speech and music activity detection (SMAD) is an enabling task that can help segment, index, and pre-process audio content in radio broadcast and TV programs. However, due to copyright concerns and the cost of manual annotation, the limited availability of diverse and sizeable datasets hinders the progress of state-of-the-art (SOTA) data-driven approaches. We address this challenge by presenting a large-scale dataset containing Mel spectrogram, VGGish, and MFCCs features extracted from around 1600 h of professionally produced audio tracks and their corresponding noisy labels indicating the approximate location of speech and music segments. The labels are several sources such as subtitles and cuesheet. A test set curated by human annotators is also included as a subset for evaluation. To validate the generalizability of the proposed dataset, we conduct several experiments comparing various model architectures and their variants under different conditions. The results suggest that our proposed dataset is able to serve as a reliable training resource and leads to SOTA performances on various public datasets. To the best of our knowledge, this dataset is the first large-scale, open-sourced dataset that contains features extracted from professionally produced audio tracks and their corresponding frame-level speech and music annotations.

Keywords: Speech and music activation detection, Dataset, Production TV audio

1 Introduction

Speech and music activity detection (SMAD) has been a long-studied problem for researchers in music information retrieval due to its wide range of applications. SMAD aims to identify the temporal locations of speech, music, and their corresponding activity levels within a polyphonic mixture of audio signals. A reliable SMAD system can be used to extract relevant parts of audio signals in preparation for other speech or music focused tasks such as spoken language identification [1, 2], speech recognition [3] and detection [4], speaker diarization, and singer identification [5]. For radio broadcasters and television services, by providing timing metadata about music and speech portion of the broadcasted content, SMAD can also help with a variety of tasks, such as data procurement for royalty payments and dialog loudness measurement.

Although the application of deep learning methods has improved SMAD systems in recent years, this data-driven approach requires large amounts of audio data with the corresponding speech and music activity labels. The collection of such datasets, however, has faced multiple challenges. First, labeling the data is costly and labor-intensive. Meléndez-Catalán et al. [6] report that annotating the 27.4 h of Open Broadcast Media Audio from TV (OpenBMAT) dataset took three annotators 130 h each for the music labels. Other radio broadcast datasets were created by hired annotators with the associated cost [7, 8]. Second, the audio content often cannot be easily shared due to the copyright limitations. Although existing radio broadcast data have been successfully used to train SMAD systems [7–9], the restricted access to this data impedes the reproducibility and validity of the research.

The publicly available datasets also suffer from drawbacks. Two datasets of considerable size, OpenBMAT [6] and AVASpeech [10], for example, contain only either music or speech labels, respectively (see Table 1). This

This work was performed while Yun-Ning Hung was an intern at Netflix.
*Correspondence: yhung33@gatech.edu
[1]Music Informatics Group, Georgia Institute of Technology, Atlanta, USA
Full list of author information is available at the end of the article

Table 1 Dataset statistics on three subsets of the proposed TVSM dataset and open-sourced datasets with frame-level annotations. Note that % of music/speech is estimated based on the duration labeled as music or speech and the total duration of the audio content

	% of music	% of speech	% of overlap	Label quality	# of instances	Duration (h)	Usage
TVSM-cuesheet	63%	64%	0.39%	Noisy	656	54.6	Training
TVSM-pseudo	61%	57%	0.33%	Noisy	2563	1538.5	Training
TVSM-test	43%	43%	0.32%	Clean	20	15	Test
OpenBMAT	50%	N/A	N/A	Clean	1647	27.5	Test
AVAspeech	N/A	52%	N/A	Clean	160	45	Test
ORF TV	42%	N/A	N/A	Clean	13	9	Test
Muspeak	76%	24%	N/A	Clean	214	5	Test

means they can be used only for either speech *or* music detection, but not both. The GTZAN Speech and Music dataset [11], Scheirer & Slaney Music Speech [12], MUSAN [13], and Muspeak [14] datasets contain only short segments and non-overlapping speech or music labels. Thus, these datasets can only be used for a simplified music and speech segmentation task, where the audio segments can only be classified into either speech, music, or noise without any overlap. In reality, the nature of television and radio production is such that music and speech co-occur regularly throughout a program. As a result, SMAD is an ongoing and active field of study [15–20], and the need for better open-sourced datasets remains to be addressed.

To solve the dataset limitation hurdle, an alternative research direction is to use synthetic data for training. A recent work by Venkatesh et al. explores existing music and speech datasets to synthetically create training material that resembles real-world radio signals [8]. Their synthesis procedure approximates the audio production workflow for radio such as transition duration, fade curves [21], and audio ducking [22], by randomizing the choices of parameters. For example, the audio transition between music-only, speech-only, and speech with background music might be accomplished by a fade curve with a linear, exponential, or an S-curve shape. This approach allows the generation of overlapping speech and music segments with frame-level annotations. The source datasets used to synthesize training examples consist of audio from BBC Radio Devon and LibriVox, a repository of user-contributed audiobook recordings.

We recognize that only modeling parameters of audio ducking and fade curves may not generate signals that are representative of contemporary, professionally produced audio. Furthermore, data augmentation methods designed for radio broadcasts may not consider the different audio mixing and recording techniques and esthetics used to produce audio-visual media like TV shows and movies. For example, TV/film production and post-production differ from live radio production in the use of dialog re-recording, foley, and sound effects.

In this work, we address the data challenge from a different angle. Specifically, we show how leveraging a large-scale dataset with noisy labels can improve SMAD results. The presented TV Speech and Music (TVSM) dataset is derived from around 1600 h of professionally recorded and produced audio for TV shows. The noisy labels are derived from different sources such as subtitles, scripted musical cue sheets, or pre-trained model's predictions (see Section 3 for details). Due to copyright limitations, the audio is processed into Mel spectrogram representations which are also used to train the proposed benchmark models. Two additional features, VGGish [23] and Mel-frequency cepstral coefficients (MFCCs) with 20 coefficients, are also provided in the released dataset due to their popularity in audio-related tasks [24, 25]. We study the generalizability of our proposed dataset by training various SOTA models and evaluating on a variety of datasets across different domains, such as broadcast, YouTube video, and TV shows. In the following sections, we first review existing approaches on SMAD in Section 2. Section 3 provides the statistics of the dataset and a detailed description on the data collection process. Section 4 gives an overview of our proposed benchmark and third-party methods, and Section 5 presents the detailed evaluation results.

2 Related work

Extracting representative features from the audio that have discriminative power for music and speech is one of the key components of a SMAD system and has been investigated in several prior studies. Early systems adopt a more generic approach of using low-level features such as amplitude, cepstra, pitch, zero-crossings, line spectral frequencies, and RMS, followed by a simple classifier, such as Gaussian mixture model or nearest neighbors [26–29]. Later on, more targeted features have been investigated. For instance, Wieser et al. proposed the continuous frequency activation (CFA) feature to detect music in TV productions or to detect speech and music in radio broadcasts [30]. Other input feature representations include the transient activation (TAC) feature to model

transient/percussive activities in an audio signal [31] or the utilization of multiple self-similarity matrices generated from different audio features to detect transition points [17]. By exploring a variety of input features and machine learning approaches, Khan and Al-Khatib conclude that the range of zero-crossings, the variance of the Haar discrete wavelet transform, the root mean square of a lowpass signal, the spectral flux, the linear predictive coefficients, and the variance of four Mel frequency cepstral coefficients in combination with a multi-layer perceptron (MLP) classifier can produce the best result in their experimental setting [32]. Pinquier et al. proposed entropy modulation, stationary segment duration, and number of segments as the main features for speech/music classification [33]. Instead of using hand-crafted features, Ajmera et al. extracted entropy and dynamism features through neural networks, and trained an HMM model with heuristic information for speech/music classification [34]. Although input features are important to assist machine learning classifiers for SMAD, different use cases might require different input features. In contrast, data-driven approaches can learn feature representations directly from training data and ease the labor of designing hand-crafted features.

As a result, the research focus in recent years has gradually shifted towards machine-learned features/models as opposed to hand-crafted features. For example, Papakostas and Giannakopoulos [19] detected music and speech using a convolutional neural network (CNN) with a spectrogram input. Similarly, Doukhan et al. [18] presented an open-source speech and music segmentation system based on the log Mel spectrogram and a CNN architecture. Jang et al. used a trainable Mel-kernel to extract the features for SMAD [35]. Multiple works explore the impact of different model architectures. De Benito-Gorron et al. compared several neural network architectures, including a fully connected (FC) neural network, a CNN, and a recurrent neural network (RNN), and concluded that the recurrent architecture is most suitable for the task [36]. Lemaire and Holzapfel leveraged the success of temporal convolutional networks (TCN) in modeling sequential data and trained a SMAD system on both in-house and openly available datasets [9]. To fully utilize the existing data resources, they trained the model on clip-level datasets and fine-tuned it on frame-level datasets. However, due to copyright restrictions, most of these methods are trained on either private or datasets with limited size, hindering the progress of data-driven approaches. As an alternative approach, transfer learning has been studied by Choi et al. as a way of leveraging feature representations learned from different domains [37]. Venkatesh et al. also explored a data augmentation technique for training with a synthetic dataset remixed from existing datasets [8].

3 Dataset

Inspired by the abovementioned studies, we continue the investigation of effective SMAD models with an alternative solution to limited data. Instead of augmenting or synthesizing data, we explore the possibility of using large-scale data with noisy labels. In contrast to clean labels, which indicate accurate start and end times for each speech/music region, noisy labels only provide approximate timing, which may impact SMAD classification performance. Nevertheless, these noisy labels allow us to increase the scale of the dataset with minimal manual efforts.

3.1 Overview

The presented TVSM dataset has a total number of 1608.1 h of professionally recorded and produced audio for TV shows. Unlike the majority of the pre-existing datasets shown in Table 1, TVSM is significantly larger in size and contains both speech and music labels at the frame level. TVSM also contains overlapping music and speech labels, and both classes have a similar total duration.

The dataset contains three subsets based on their annotation types and intended functionality. In particular, *TVSM-cuesheet* and *TVSM-pseudo* are used for training, and *TVSM-test* is used for testing. Additionally, these subsets are distinguishable by their label creation process, which will be elaborated on in the later sections. For TVSM-cuesheet, each instance is a 5-min excerpt from a TV show, whereas for *TVSM-pseudo* and *TVSM-test*, each instance is a complete episode. *TVSM-pseudo* is also larger in size than TVSM-cuesheet. The total instances of *TVSM-pseudo* is four times more than *TVSM-cuesheet* while the total duration is twenty-eight times longer than *TVSM-cuesheet* (see Table 1). To facilitate the comparison across different methods and ensure the correctness of the benchmarks on our proposed dataset, *TVSM-test* is manually annotated and does not rely on noisy labels.

The content of the proposed dataset comes from a large proprietary database of TV shows for online streaming services and has been authorized for processing and distribution in a feature representation. All audio tracks are legally accessed and processed into pre-extracted feature representations. The instances from this dataset are sampled between 2016 and 2019. The contents of the dataset come from 13 countries with approximately 60% originating from the USA. The length of the content ranges from a few minutes to over 1 h across various genres as listed in Fig. 1. The audio is originally delivered from the recording studios in a standard 5.1 surround format and a 48-kHz sampling rate. We downmix the multichannel files to stereo and then mono via the standard left-only/right-only (Lo/Ro) downmix formula [38] and downsample the audio to a sampling rate of 16 kHz. The

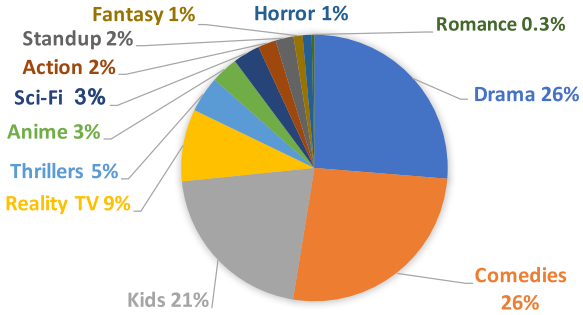

Fig. 1 Genre distribution in the TVSM dataset

proposed dataset contains audio tracks in three different languages, namely English, Spanish, and Japanese. All the original audio signals are normalized to an average loudness being -27 LKFS $\pm$ 2 LU dialog-gated via the Dolby Professional Loudness Metering (DPLM) tool. The language distribution is shown in Fig. 2. The name of the episode/TV show for each sample remains unpublished. However, each sample has both a show ID and a season ID to help users identify the connection between the samples. For instance, two samples from different seasons of the same show would share the same show ID and have different season IDs.

3.2 Training set

3.2.1 Speech labels

Both *TVSM-pseudo* and *TVSM-cuesheet* contain speech labels derived from subtitle information. Since each TV program is typically delivered in a package of video, audio, and subtitle files, pairs of audio and subtitles are readily available. Thus, subtitle timestamps are a reliable source of approximate start and end times of speech utterances. In addition to expected deviations in precise timing information, another possible source of label noise is the minimal temporal duration. The minimal duration of each subtitle is set to one second in order to allow viewers enough time to easily read. The minimum duration is a delivery specification from the online streaming service. Nevertheless, the time stamps extracted from the subtitles seem to consistently cover the speech regions based on our preliminary examination. Closed captions (CC) are excluded because—unlike subtitles which transcribe *only* speech—CCs describe all important audio events including environmental noises, sound effects, and speaker identities, making them a much noisier source for speech labels. In addition to the typical transcripts of conversational speech, the lyrics from singing voices are also included in the subtitle files. Since singing is composed of words and semantic meaning like speech, we also include sung regions in the speech labels.

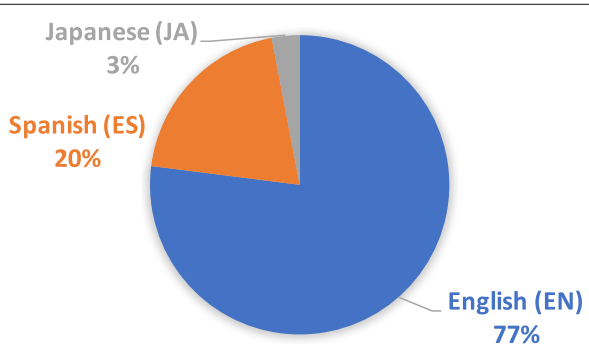

Fig. 2 Language distribution in the TVSM dataset

3.2.2 Music labels

The noisy music labels of *TVSM-pseudo* and *TVSM-cuesheet* come from two different sources. For *TVSM-cuesheet*, the music labels are based on the cue sheets in our proprietary sources. Cue sheets provide additional metadata that document the appearance of music in TV shows and films, including the start and end times. They are originally intended to keep a record of the music usage and entitled parties within an audio/visual program. The cue sheets are stored in a structured manner to facilitate search and combinations with other data sources. We matched 656 cue sheets from our internal database to our TVSM dataset, leading toward the creation of *TVSM-cuesheet*.

Cue sheet data is trustworthy, but the music labels created from cue sheets are not necessarily as accurate as one would expect. Two issues can be observed. First, although cue sheets mark the start and end times of the music regions, audio mixing techniques such as fade-in and fade-out might offset the times by when the music is perceptible, resulting in potentially inaccurate music labels. Second, although the time stamps from cue sheets are generally correct, issues such as incorrect time offsets occur. An analysis using a rule-based method was implemented to detect such errors. Approximately 20% of the files were identified and corrected. Overall, cue sheets can be assumed to be accurate within a tolerance window of around 5 s.

For audio content without a matching cue sheet, our pre-trained model is used to generate pseudo-labels. Details on the pre-trained model and the pseudo-label generation will be given in Section 5. Since the labels are predicted by a pre-trained model, they can be particularly noisy in hard-to-detect regions due to incorrect predictions. We refer to this subset as *TVSM-pseudo*. The length distribution of this subset can be seen in Fig. 3.

3.3 Test set

To evaluate and benchmark our proposed dataset, we manually labeled 20 audio tracks from various TV shows

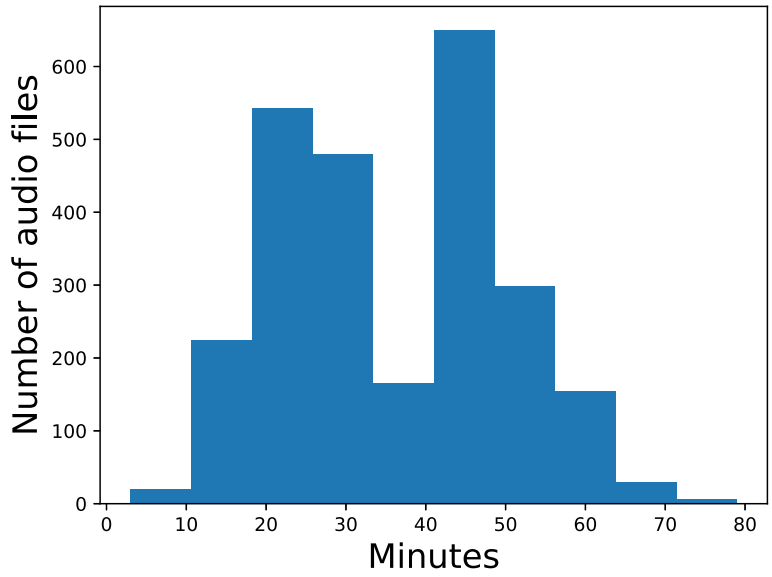

Fig. 3 The length distribution (min) of the TVSM-pseudo dataset

which do not overlap with our training data (i.e., *TVSM-cuesheet* and *TVSM-pseudo*), and we refer to this subset as *TVSM-test*. Audio files in *TVSM-test* are chosen based on the genre distribution in TVSM dataset. We do not exclude audio files that come from the same TV show in either *TVSM-cuesheet* or *TVSM-pseudo* since the content from the same TV show can be quite different. Moreover, we provide metadata (e.g., show ID and season ID) of each sample to identify samples that come from the same TV show. To create this test set, four volunteers manually annotated the speech and music ground truth labels. Volunteers were all amateur musicians who used headphones and the Sonic Visualiser application[1] to annotate active regions. One of the fundamental issues encountered during the annotation process is the definition of music and speech. Specifically, the heavy usage of ambient sound and sound effects in various TV shows makes it difficult to determine the active music regions. Similarly, switches between conversational speech and singing voices in certain TV genres also present unique challenges for the annotators. To ensure the consistency of the labels and avoid ambiguity, the following guidelines were agreed upon among the annotators for differentiating music and speech:

- Any music that is perceivable by the annotator at a comfortable playback volume should be annotated.
- Since singing lyrics are included in the subtitles, human singing voices should all be annotated as both speech and music.
- Ambient sound or sound effects without apparent melodic contours should not be annotated as music.
- Traditional phone bell, ringing, or buzzing without apparent melodic contours should not be annotated as music.
- Filled pauses (uh, um, ah, er), backchannels (mhm, uh-huh), sighing, or screaming should not be annotated as speech.

3.4 Processed features

Due to copyright limitations, the audio data is made available as Mel spectrogram representations. We use the PyTorch package Torchaudio[2] to extract the Mel spectrograms. After resampling the input audio to 16 kHz, 128 Mel bands are extracted with a hop size and window length of 512 and 1024, respectively.

In addition to the Mel spectrograms, we also provide MFCCs and VGGish features. VGGish features are extracted by the pre-trained VGGish model [23] as 128-dimensional vector with 0.96 s time resolution (no overlap). The features are PCA transformed (with whitening) and quantized to 8-bits. We use PyTorch's implementation of VGGish model[3] to extract the features. MFCC features are extracted by using the `Librosa`[4] python package, with 20 coefficients and the same window length, hop size, and sampling rate as the Mel spectrogram.

[1] https://www.sonicvisualiser.org/, last accessed on 04/10/2022

[2] https://pytorch.org/audio/stable/index.html, last accessed on 01/05/2022

[3] https://github.com/harritaylor/torchvggish

[4] https://librosa.org/doc/main/index.html

4 Methods

4.1 Third-party methods

For comparison with our benchmark methods, we included two third-party methods, which represent state of the art, in our evaluation. The first one is the InaSpeechSegmenter (referred to as T1) presented by [18]. InaSpeechSegmenter is a CNN-based audio segmentation toolkit to split audio signals into homogeneous zones of speech, music, and noise. The model used in this method is trained on the Muspeak dataset.

The second method is the CRNN model proposed by [8] (referred to as T2). The CRNN model used in this method is one of the templates for the model in our benchmark method (see Section 4.2.2); the main difference between this method and our benchmark method is the training data. Venkatesh et al. [8] trained their model on the combination of synthetic data and real-world radio examples. The source materials for their synthetic dataset come from a variety of sources which contain audio files labeled as either music, speech, or noise. To mimic the transition between music and speech in real-world radio programs, the authors incorporated mixing techniques such as fade curves and audio ducking during the synthesis process. Based on the reported results, this method is able to achieve the SOTA performance when trained with hybrid synthetic & real-world training data.

4.2 Benchmark method

In order to explore the interaction of large-scale data, architectures, and hyperparameters, we investigate several input representations and deep-learning architectures in our benchmark method.

4.2.1 Input features

Log-Mel spectrograms are used as the input representation for the benchmark methods. In addition to the commonly used log-Mel spectrogram, per-channel energy normalization (PCEN) is also explored as an input normalization method on Mel spectrogram instead of using log. PCEN was originally proposed by [39] for keyword spotting and has also been reported useful in sound event detection (SED) [40]. However, PCEN has rarely been studied in the music domain despite the task similarities between SED and SMAD. PCEN has trainable parameters for dynamic gain control, temporal integration, and range compression. In contrast to static compression (such as logarithmic transformation), PCEN has been shown as a viable alternative aiming at improving robustness to channel distortion [41].

4.2.2 Model architecture

The architectural choices in this work are informed by the results of prior studies. To this end, we use the best architectures found in the literature, namely the convolutional recurrent neural network (CRNN) and TCN, as our starting point.

The CRNN model in this work is adopted from an architecture proposed by [8] with minor adjustments to accommodate our input/output requirements. This CRNN architecture consists of three convolutional layers, followed by two bi-directional recurrent layers and one fully connected layer. The only difference between our proposed CRNN and the one used by [8] is the number of filters in the convolutional layers. We found that by using half of the filters for the first two layers, the model performs better when training on TVSM-cuesheet. The reason might be that TVSM-cuesheet has limited training data, compared to the large amount of synthesized training data used by [8]. So, we reduce the complexity of the model to avoid over-fitting.

The TCN model in this work also follows a similar configuration as described in [8] with the exception of a smaller kernel size due to the lower temporal resolution of our input representation. This modification stabilizes the optimization process in training. The proposed TCN architecture consists of three repeated layers with each layer having different stacks of dilated 1-D convolutional layers. Table 2 lists the detailed parameters for both architectures.

A linear layer is attached to the end of each model to generate a matrix $A \in R^{C \times T}$ where T represents the number of time frame. For each frame, the output consists of two continuous values ($C = 2$) ranging from 0 to 1, representing the probability of speech and music respectively. Finally, a max-pooling layer is used to generate frame-level predictions with a temporal resolution of 5 frames in each second. The TCN and CRNN models have a total number of 258,834 and 831,890 parameters, respectively.

4.2.3 Training setup

We apply a random sampling strategy during training time, where each training sample is a 20-s segment chunked by randomly selecting an audio file and the starting time of the audio on the fly. The models in this work are trained by minimizing binary cross-entropy (BCE) loss.

Table 2 Parameters for TCN and CRNN model architecture

Model Arch.	Parameters	Values
TCN	Kernel size	{3, 5, 5}
	No. filters	{32, 16, 32}
	No. stacks	{9, 5, 2}
	No. dilations	{3, 7, 2}
	Use skip connections	{False, true, true}
CRNN	Kernel size	{3, 11, 11}
	No. filters	{64, 64, 16}
	No. GRU units	{80, 40}

TVSM-cuesheet/TVSM-pseudo is divided into 90% and 10% for training and validation purposes, respectively. The validation set is randomly chosen from the training set. Similar to the test set, we do not exclude episodes from the same TV show. We use the Adam optimizer [42] with 0.001 learning rate and 0.0001 weight decay to optimize the models. Early stopping is applied if the validation loss does not change for 10 epochs. The learning rate decreases with a scaling factor of 0.3, if the validation loss does not change for 2 epochs.

5 Experiment

5.1 Experimental setup

To understand the influence of different variables in our experimental setup (i.e., model architecture, training data, PCEN), we include the following variants of our adopted models in the ablation study:

- TCN-Cue: TCN trained on TVSM-cuesheet data
- TCN-P-Cue: TCN + PCEN trained on TVSM-cuesheet data
- CRNN-Cue: CRNN trained on TVSM-cuesheet data
- CRNN-P-Cue: CRNN + PCEN trained on TVSM-cuesheet data
- TCN-P-Pseu: TCN + PCEN trained on TVSM-pseudo data
- CRNN-P-Pseu: CRNN + PCEN trained on TVSM-pseudo data

In brief, CRNN-P-Cue uses CRNN while TCN-P-Cue uses TCN as the model architecture and is trained on *TVSM-cuesheet*. The pre-trained model of CRNN-P-Cue is used to predict the labels of the audio in *TVSM-pseudo* since it achieves better performance than TCN-P-Cue (see Tables 4 and 5). The labels predicted by CRNN-P-Cue are then used to train the same CRNN architecture (CRNN-P-Pseu) and TCN architecture (TCN-P-Pseu).

The `sed_eval` toolbox [43] is used to perform segment-level evaluation as used in the MIREX 2018 competition[5] to be comparable and consistent with prior work [8, 9]. For each method, we report the class-wise F-score and error rate with a segment size of 10 ms. The error rate is the summation of deletion rate (false negative) and insertion rate (false positive), as defined in the toolbox. Since a binary decision must be attained for music and speech to calculate the F-score, a threshold of 0.5 is used to quantize the continuous output of speech and music activity functions. No post-processing is applied. Furthermore, a two-tailed paired T-test is conducted to compare the statistical significance between two variants. We report the p-value between the F-score of each sample from two different setups.

[5] https://www.music-ir.org/mirex/wiki/2018:Music_and/or_Speech_Detection, last accessed on 04/30/2021

The investigated methods are evaluated on *TVSM-test* and four additional openly available datasets to represent a variety of audio content. The statistics of these datasets are listed in Table 1. The OpenBMAT [6] and ORF TV dataset [44] are collected from TV programs, and AVASpeech [10] comprises audio from YouTube, while Muspeak [14] has a variety of content such as concert, radio broadcast, and low-fidelity folk music.

5.2 Results

Tables 4 and 5 show the evaluation results for speech and music, respectively. These results are discussed in the following sections.

5.2.1 Third-party methods

It can be observed that T2 outperforms T1 on music detection. This gap in performance is largely due to an inherent limitation of T1. Since T1 is only capable of predicting either speech or music, it struggles when the data contains both speech and music simultaneously. This can be seen from Tables 4 and 5 that the F1-score for speech is similar to T2 while the music has a large difference. Moreover, take TVSM-test for example, as shown in Table 3, T1 has deletion rate 0.68 for music while 0.18 for speech. The insertion rate relates to false positive while deletion rate relates to false negative. Both are the lower the better. As a result, the high deletion on music indicates that when speech and music appear together, the model is biased towards predicting speech, causing a significant drop in the performance of music detection. This highlights the importance of building a system that supports the detection of overlapping speech and music. The differences between T1 and T2 are statistically significant ($p < 0.05$) for music but not for speech.

5.2.2 CRNN vs. TCN

Comparing the performance of TCN-Cue and CRNN-Cue, it can be found that the CRNN outperforms the TCN on all test sets, especially for music. This result is generally consistent with the findings of [8]. While they hypothesized that the use of the TCN might be advantageous for handling longer sequences due to its longer effective memory, our results show that even with a longer input sequence (20 s as opposed to 8 s in [8]), the performance difference between CRNN and TCN architectures

Table 3 Error, deletion, and insertion rate evaluated on TVSM-test dataset by T1 methods. The metrics are calculated via `sed_eval` toolbox

	Error rate	Deletion rate	Insertion rate
Music	0.70	0.68	0.02
Speech	0.33	0.18	0.15

remains substantial. The differences between TCN-Cue and CRNN-Cue are statistically significant ($p < 0.05$) for both music and speech.

5.2.3 PCEN

The usefulness of PCEN can be determined by comparing CRNN-Cue/TCN-Cue and CRNN-P-Cue/TCN-P-Cue. The results show that PCEN is helpful on all the music datasets except for TVSM-test. One possible explanation is the presence of ambient sound (e.g., low-frequency tone without constantly changing pitch) in this test set. TVSM-test shares considerable similarities with the training dataset (TVSM-cuesheet and TVSM-pseudo). The level difference between music and ambient sound can sometimes be very subtle in the training set. As a result, PCEN could accidentally increase the volume of ambient sound effect, leading to more false positives in the output. To verify this assumption, we examined several instances from TVSM-test where ambient sounds are presented, and the similar false-positive behaviors can be observed. Figure 4 shows one of these samples, which is a snippet of a TV show from TVSM-test where a woman is talking and an ambient sound is playing at the same time. We can see that with PCEN, most of the ambient sound is predicted as music. Moreover, compared to CRNN, TCN seems to benefit more from the addition of PCEN. This could imply that for the less efficient model, input features play a more important role in determining the overall performances. Compared to music, speech prediction does not improve by PCEN. Since all the audio tracks in our proposed dataset are normalized to a consistent speech-gated loudness level, the advantage of dynamic gain control from PCEN could be less effective in this case. The differences between CRNN-Cue/TCN-Cue and CRNN-P-Cue/TCN-P-Cue are statistically significant ($p < 0.05$) for both music and speech.

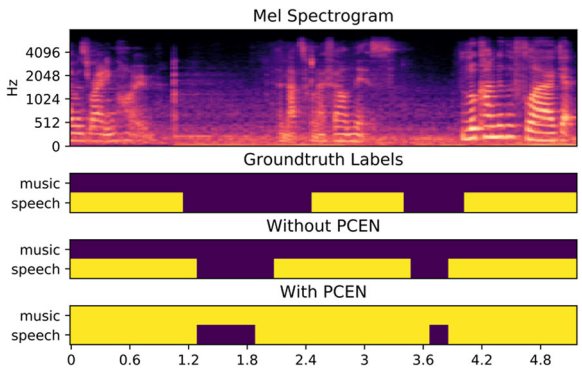

Fig. 4 An example selected from TVSM-test where PCEN fails. The figure shows the detection of background ambient sound with or without PCEN in a 5-s segment. Yellow regions indicate active labels while purple regions stand for inactive labels

5.2.4 Impact of training data

The differences between CRNN-P-Cue/TCN-P-Cue and CRNN-P-Pseu/TCN-P-Pseu outline the influence of different training subsets. In sum, both TCN and CRNN models trained on TVSM-pseudo can achieve better performance on speech and music prediction. The result suggests that large amount of pseudo labels can be used as a valuable source for training. Other training strategies such as noisy student training ([45]) can also be explored in the future to benefit from pseudo labels.

Surprisingly, for the CRNN model, the improvement from using a larger subset (TVSM-pseudo) versus a smaller subset (TVSM-cuesheet) for training is only marginal; this is especially the case for music. Since the music labels in TVSM-pseudo are generated via pseudo-labeling using our pretrained model, the same types of prediction errors are likely to propagate to the larger set. As a result, the quality of these pseudo-labels could potentially limit the results and reduce the effect of increasing the size.

The limitation can be seen from Fig. 5, especially on the ORF TV and OpenBMAT datasets. We discover that CRNN-P-Pseu is more sensitive to background noise, such as low-frequency ambient sound or background actor talking, when training on TVSM-pseudo. The misclassification of noise to music results in a higher false-positive rate.

The differences between CRNN-P-Cue/TCN-P-Cue and CRNN-P-Pseu/TCN-P-Pseu are statistically significant ($p < 0.05$) for both music and speech.

5.2.5 Comparison with the state of the art

By comparing our benchmark method (CRNN-Cue) with the best SOTA method (T2), as shown in Tables 4 and 5, CRNN-Cue outperforms T2 on most of the test sets for both speech and music. Since both CRNN-Cue and T2 have a similar model architecture, the primary difference is the training material. The improvement is more obvious when compared to our best benchmark method (CRNN-P-Pseu). This result supports our assumption that a large but noisy-labeled real-world dataset can serve as a viable solution to the data challenge in SMAD and can lead to improvements over training with synthesized data. The only exception is the speech detection performance on the Muspeak dataset. There are two potential reasons for this discrepancy. First, the labels of human singing voice in Muspeak follow a different definition, as singing is labeled as music only in Muspeak, whereas in TVSM all singing with recognizable lyrics is also labeled as speech via the subtitle metadata. Second, Muspeak is partly included in the training data of T2 according to [8], which decreases the confidence in the validity of a direct comparison.

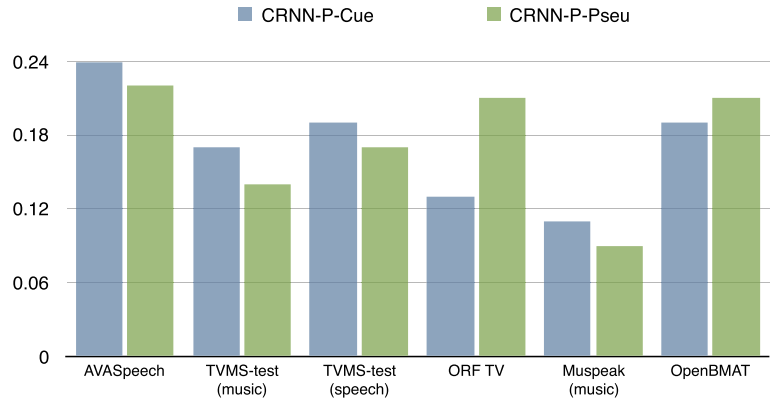

Fig. 5 The mean error rate (the lower the better) across all datasets as described in the `sed_eval` toolbox. The models CRNN-P-Cue and CRNN-P-Pseu are selected for comparison. TVSM-test (music) and Muspeak (music) represent the music evaluation while TVSM-test (speech) represents the speech evaluation. The other test datasets only contain either speech or music labels as described in Section 3

Table 4 *F*-measures for segment-level evaluation on speech detection

	Model Arch.	Training data	PCEN	Muspeak	AVASpeech	TVSM-test
Third-party method (T1)	CNN			0.94	0.79	0.84
Third-party method (T2)	CRNN			**0.97**	0.77	0.81
TCN-Cue	TCN	TVSM-cuesheet		0.60	0.86	0.90
TCN-P-Cue	TCN	TVSM-cuesheet	✓	0.61	0.86	0.89
TCN-P-Pseu	TCN	TVSM-pseudo	✓	0.60	**0.88**	**0.91**
CRNN-Cue	CRNN	TVSM-cuesheet		0.63	0.86	**0.91**
CRNN-P-Cue	CRNN	TVSM-cuesheet	✓	0.63	0.86	**0.91**
CRNN-P-Pseu	CRNN	TVSM-pseudo	✓	0.67	**0.88**	**0.91**

The Highest result of each evaluation dataset is marked as boldface

Table 5 *F*-measures for segment-level evaluation on music detection

	Model Arch.	Training data	PCEN	ORF TV	Muspeak	OpenBMAT	TVSM-test
Third-party method (T1)	CNN			0.60	0.93	0.47	0.48
Third-party method (T2)	CRNN			0.85	**0.99**	0.85	0.88
TCN-Cue	TCN	TVSM-cuesheet		0.79	0.86	0.82	0.88
TCN-P-Cue	TCN	TVSM-cuesheet	✓	0.86	0.93	0.84	0.90
TCN-P-Pseu	TCN	TVSM-pseudo	✓	0.87	0.97	0.87	0.93
CRNN-Cue	CRNN	TVSM-cuesheet		0.89	0.93	0.88	0.93
CRNN-P-Cue	CRNN	TVSM-cuesheet	✓	**0.92**	0.94	0.90	0.91
CRNN-P-Pseu	CRNN	TVSM-pseudo	✓	**0.92**	0.95	**0.91**	**0.94**

The Highest result of each evaluation dataset is marked as boldface

5.3 Dataset deliverables

This work includes a GitHub repository for python module delivery and a Zenodo entry for dataset delivery[6].

The GitHub repository includes:

- Python code for data pre-processing, including scripts for 5.1 downmixing, Mel spectrogram generation, MFCCs generation, VGGish features generation, and the PCEN implementation
- Python code for the experiment, including scripts of data loaders, model implementations, training pipeline, and evaluation pipeline
- Pre-trained models for each conducted experiment
- Prediction output of each audio in the evaluation datasets
- `README.txt` file that documents the usage of the code for reproducibility

We use Zenodo to enable version control of the proposed dataset. The Zenodo entry includes:

- An agreement form for anyone who is interested in using this dataset for research purposes.
- Mel spectrogram, VGGish, and MFCCs features of the proposed dataset, which are stored in NumPy format with unique IDs assigned for each file.
- Speech and music labels stored in csv format; each row in the csv file has a start time, end time, and class labels to describe each speech and music region. The csv file has the corresponding IDs as the pre-extracted features.
- Metadata for each instance stored in csv format; each row in the csv file containing each file's name and the corresponding show ID, season ID, release year, genre, length, original country, and language.

6 Conclusion

We presented TVSM, a large-scale TV show dataset with noisy labels for the task of SMAD. The dataset contains two training subsets with noisy labels generated by different strategies and a test subset with clean, manual-created annotations. Compared to other publicly available datasets, our TVSM is larger in size and is sampled from real-world professionally produced audio that is diverse in both genres and languages. We investigated the effectiveness of each subset for various model architectures and third-party methods. Compared to two third-party methods trained with synthetic and small-scale data, our proposed benchmark methods were able to generalize better and outperform state-of-the-art results on several existing datasets, in spite of training on noisy labels. Our evaluation results suggest that, while it is possible to leverage large data with noisy labels (e.g., TVSM-pseudo) for training SMAD models, the quality of the labels is still crucial for further improvements. Future directions include:

- A detailed investigation of the impact of the constituent languages in the training materials in order to achieve a better generalization for speech detection
- Experimenting with alternative pseudo-labeling methods such as teacher-student learning [46] and other model architectures such as forms of attention mechanism in order to minimize the deleterious impact of the noisy labels
- Providing other metadata or labels for the TVSM dataset, such as loudness level of music and speech
- Adding new labels to this dataset, such as human non-speech vocalizations [47]
- Exploring data augmentation techniques to improve the model's robustness against ambient sound

[6] https://github.com/biboamy/TVSM-dataset

Abbreviations

SMAD: Speech and music activity detection; SOTA: State-of-the-art; OpenBMAT: Open Broadcast Media Audio from TV; TVSM: TV speech and music; MFCCs: Mel-frequency cepstral coefficients; CFA: Continuous frequency activation; TAC: Transient activation; MLP: Multi-layer perceptron; CNN: Convolutional neural network; FC: Fully connected; RNN: Recurrent neural network; TCN: Temporal convolutional networks; Lo/Ro: Left-only/right-only; DPLM: Dolby Professional Loudness Metering

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13636-022-00253-8.

Additional file 1: Supplementary materials.

Acknowledgements

Not applicable.

Authors' contributions

Y.H. conducted the experiments and was a major contributor of this manuscript. Y.H., C.W., I.O., and A.H. prepared/analyzed the data and contributed to the writing of the manuscript. W.W. and A.L. supervised this work and assisted with the editing of the manuscript. All authors read and approved the final manuscript.

Funding

This research was supported by Netflix as an internship project.

Availability of data and materials

The datasets generated and/or analyzed during the current study are available in the Zenodo repository (link will be added if the manuscript is accepted). The code supporting the conclusions of this article is available in the public GitHub repository (Link will be added if the manuscript is accepted). The dataset content is also cleared by the rights holder to be published.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Music Informatics Group, Georgia Institute of Technology, Atlanta, USA. [2] Netflix, Inc., Los Gatos, USA.

Received: 16 April 2022 Accepted: 25 July 2022
Published online: 03 September 2022

References

1. J. Valk, T. Alumäe, in *IEEE Spoken Language Technology Workshop (SLT)*. Voxlingua107: a dataset for spoken language recognition (IEEE, 2021), pp. 652–658
2. A. S. Ba Wazir, H. A. Karim, M. H. L. Abdullah, N. AlDahoul, S. Mansor, M. F. A. Fauzi, J. See, A. S. Naim, Design and implementation of fast spoken foul language recognition with different end-to-end deep neural network architectures. Sensors. **21**(3), 710 (2021)
3. H. Nguyen, Y. Estève, L. Besacier, in *Proc. Interspeech 2021*. Impact of encoding and segmentation strategies on end-to-end simultaneous speech translation, (2021), pp. 2371–2375. https://doi.org/10.21437/Interspeech.2021-608
4. F. Albu, D. Hagiescu, M. Puica, L. Vladutu, in *Proceedings of the International Technology, Education and Development Conference*. Intelligent tutor for first grade children's handwriting application (IATED, Valencia, 2015), pp. 3708–3717
5. T. Theodorou, I. Mporas, N. Fakotakis, Int. J. Inf. Technol. Comput. Sci. (IJITCS). **6**(11), 1 (2014)
6. B. Meléndez-Catalán, E. Molina, E. Gómez, Open broadcast media audio from tv: a dataset of tv broadcast audio with relative music loudness annotations. Trans. Int. Soc. Music Inf. Retr. **2**(1), 43–51 (2019)
7. J. Schlüter, R. Sonnleitner, in *Proceedings of International Conference on Digital Audio Effects*. Unsupervised feature learning for speech and music detection in radio broadcasts (DAFx, York, 2012)
8. S. Venkatesh, D. Moffat, E. R. Miranda, Investigating the effects of training set synthesis for audio segmentation of radio broadcast. Electronics. **10**(7), 827 (2021)
9. Q. Lemaire, A. Holzapfel, in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*. Temporal convolutional networks for speech and music detection in radio broadcast (ISMIR, Delft, 2019), pp. 229–236
10. S. Chaudhuri, J. Roth, D. P. Ellis, A. Gallagher, L. Kaver, R. Marvin, C. Pantofaru, N. Reale, L. G. Reid, K. Wilson, et al., in *Proceedings of ISCA Interspeech*. Ava-speech: a densely labeled dataset of speech activity in movies (ISCA, Hyderabad, 2018)
11. G. Tzanetakis, P. Cook, Marsyas: a framework for audio analysis. Organised Sound. **4**(3), 169–175 (2000)
12. E. Scheirer, M. Slaney, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. Construction and evaluation of a robust multifeature speech/music discriminator, vol. 2, (1997), pp. 1331–1334
13. D. Snyder, G. Chen, D. Povey, Musan: a music, speech, and noise corpus. arXiv preprint arXiv:1510.08484 (2015)
14. D. Wolff, T. Weyde, E. Benetos, D. Tidhar, MIREX muspeak sample dataset (2015). http://mirg.city.ac.uk/datasets/muspeak/. Accessed 30 Sept 2020
15. R. Huang, J. H. Hansen, Advances in unsupervised audio classification and segmentation for the broadcast news and NGSW corpora. IEEE Trans. Audio Speech Lang. Process. **14**(3), 907–919 (2006)
16. D. Wang, R. Vogt, M. Mason, S. Sridharan, in *International Conference on Signal Processing and Communication Systems*. Automatic audio segmentation using the generalized likelihood ratio (IEEE, Gold Coast, 2008), pp. 1–5
17. N. Tsipas, L. Vrysis, C. Dimoulas, G. Papanikolaou, Efficient audio-driven multimedia indexing through similarity-based speech/music discrimination. Multimedia Tools Appl. **76**(24), 25603–25621 (2017)
18. D. Doukhan, E. Lechapt, M. Evrard, J. Carrive, in *Music Information Retrieval Evaluation eXchange*. Ina's mirex 2018 music and speech detection system (ISMIR, Paris, 2018)
19. M. Papakostas, T. Giannakopoulos, Speech-music discrimination using deep visual feature extractors. Expert Syst. Appl. **114**, 334–344 (2018)
20. P. Gimeno, I. Viñals, A. Ortega, A. Miguel, E. Lleida, Multiclass audio segmentation based on recurrent neural networks for broadcast domain data. J. Audio Speech Music Process. **2020**(1), 1–19 (2020)
21. E. Tarr, *Hack audio: an introduction to computer programming and digital signal processing in MATLAB®*. (Routledge, USA, 2018)
22. M. Torcoli, A. Freke-Morin, J. Paulus, C. Simon, B. Shirley, Preferred levels for background ducking to produce esthetically pleasing audio for TV with clear speech. J. Audio Eng. Soc. **67**(12), 1003–1011 (2019)
23. 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 *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Cnn architectures for large-scale audio classification (IEEE, New Orleans, 2017), pp. 131–135
24. G. Kour, N. Mehan, Music genre classification using MFCC, SVM and BPNN. Int. J. Comput. Appl. **112**(6), 43–47 (2015)
25. K. Koutini, H. Eghbal-zadeh, G. Widmer, Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks. IEEE/ACM Trans. Audio Speech Lang. Proc. **29**, 1987–2000 (2021). https://doi.org/10.1109/TASLP.2021.3082307
26. M. J. Carey, E. S. Parris, H. Lloyd-Thomas, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1*. A comparison of features for speech, music discrimination (IEEE, Phoenix, 1999), pp. 149–152
27. K. El-Maleh, M. Klein, G. Petrucci, P. Kabal, in *IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4*. Speech/music discrimination for multimedia applications (IEEE, Istanbul, 2000), pp. 2445–2448
28. E. D. Scheirer, M. Slaney, *Multi-feature speech/music discrimination system*. (Google Patents, USA, 2003)
29. C. Panagiotakis, G. Tziritas, A speech/music discriminator based on RMS and zero-crossings. IEEE Trans. Multimed. **7**(1), 155–166 (2005)
30. E. Wieser, M. Husinsky, M. Seidl, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Speech/music discrimination in a large database of radio broadcasts from the wild (IEEE, Florence, 2014), pp. 2134–2138
31. J. Han, B. Coover, in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. Leveraging structural information in music-speech dectection (IEEE, San Jose, 2013), pp. 1–6
32. M. K. S. Khan, W. G. Al-Khatib, Machine-learning based classification of speech and music. Multimed. Syst. **12**(1), 55–67 (2006)
33. J. Pinquier, J.-L. Rouas, R. A. E-OBRECHT, Robust speech/music classification in audio documents. Entropy. **1**(2), 3 (2002)
34. J. Ajmera, I. McCowan, H. Bourlard, Speech/music segmentation using entropy and dynamism features in a HMM classification framework. Speech Comm. **40**(3), 351–363 (2003)
35. B.-Y. Jang, W.-H. Heo, J.-H. Kim, O.-W. Kwon, Music detection from broadcast contents using convolutional neural networks with a Mel-scale kernel. J. Audio Speech Music Process. **2019**(1), 1–12 (2019)
36. D. de Benito-Gorron, A. Lozano-Diez, D. T. Toledano, J. Gonzalez-Rodriguez, Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset. J. Audio Speech Music Process. **2019**(1), 1–18 (2019)
37. K. Choi, G. Fazekas, M. B. Sandler, K. Cho, in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*. Transfer learning for music classification and regression tasks, (ISMIR, Suzhou, 2017), pp. 141–149
38. A. Standard, A52/A: digital audio Compression Standard (AC-3, E-AC-3), Revision B, Adv. TV Syst. Comm. 78–79 (2005)
39. Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, R. A. Saurous, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Trainable frontend for robust and far-field keyword spotting (IEEE, New Orleans, 2017), pp. 5670–5674
40. V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, J. P. Bello, Robust sound event detection in bioacoustic sensor networks. PloS ONE. **14**(10), 0214168 (2019)
41. V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, J. P. Bello, Per-channel energy normalization: why and how. IEEE Signal Process. Lett. **26**(1), 39–43 (2019)
42. D. P. Kingma, J. Ba, in *3rd International Conference on Learning Representations*. Adam: a method for stochastic optimization, (2015)
43. A. Mesaros, T. Heittola, T. Virtanen, Metrics for polyphonic sound event detection. Appl. Sci. **6**(6), 162 (2016)
44. K. Seyerlehner, T. Pohle, M. Schedl, G. Widmer, in *Proceedings of of the 10th International Conference on Digital Audio Effects (DAFx)*. Automatic music detection in television productions (DAFx, Bordeaux, 2007)
45. M. Won, K. Choi, X. Serra, in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*. Semi-supervised music tagging transformer (ISMIR, 2021)
46. S. Kum, J.-H. Lin, L. Su, J. Nam, in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*. Semi-supervised learning using teacher-student models for vocal melody extraction (ISMIR, 2020), pp. 93–100

47. Y. Gong, J. Yu, J. Glass, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vocalsound: a dataset for improving human vocal sounds recognition (IEEE, ICASSP, 2022), pp. 151–155

Publisher's Note