DAACI-VoDAn: Improving Vocal Detection with New Data and Methods

Helena Cuesta
Daaci Ltd.
London, UK
helena@daaci.com

Nadine Kroher, Aggelos Pikrakis
Time Machine Capital 2 Ltd.
London, UK
{nadine, aggelos}@tmc2.ai

Stojan Djordjevic
Daaci Ltd.
London, UK
stojan@daaci.com

***Abstract*—Vocal detection (VD) algorithms aim to detect the presence of vocals in music recordings and are an essential pre-processing step for other tasks, including singer identification and lyrics transcription. However, the lack of large-scale annotated datasets has slowed down research in the field, in particular w.r.t. the application of modern deep learning methods. This paper introduces DAACI-VoDAn, a novel dataset for VD that contains 706 full-length music tracks and vocal segment annotations. In addition, we propose a new method for the task that outperforms state of the art methods on DAACI-VoDAn as well as on an existing VD dataset. Our approach combines a convolutional head, that is pre-trained on large amounts of weakly-labeled data, with a temporal-convolutional architecture which models the occurrence of two-dimensional patterns over time.**

I. Introduction

Singing voice detection or simply vocal detection (VD), is a classification task that aims at detecting segments of a music track that contain singing voice. In the field of Music Information Retrieval (MIR), VD is frequently approached at frame-level granularity, i. e. , a system predicts the presence or absence of vocals in short successive audio frames. The frame length varies depending on the method and ranges from a few milliseconds (i. e. , 5 ms) to 100 ms for coarser predictions. Detecting the presence of vocals in a music track can be an important pre-processing step for other MIR tasks such as lyrics transcription [1], singer identification [2] and singing voice transcription [3], to name but a few. Song-level tags related to the presence of vocals are furthermore commonly used in interfaces for the exploration of production music catalogues and frame-level annotations can be part of music track visualisation tools.

Early VD approaches used standard classifiers like Support Vector Machines (SVM), Random Forests (RF) and Hidden Markov Models (HMM) that operate on hand-crafted low-level descriptors [4] or features derived from neural networks trained on natural speech [5]. For a detailed review of systems that follow this approach, the reader is refered to [6]. In an effort to overcome the limited availability of data, the work in [7] used dictionary learning to detect vocal segments in folk music in an unsupervised manner.

More recent methods have relied on Deep Learning (DL) architectures and in particular on Convolutional Neural Networks (CNN) and temporal architectures, such as Recurrent Neural Networks (RNN) and Long Short-Term Memory networks (LSTM). While CNNs are a powerful tool for learning two-dimensional patterns (i. e. , in spectrograms), the motivation to employ recurrent architectures is based on their capability of modelling the temporal evolution of a music signal (temporal context) rather than evaluating short frames in isolation.

In [8], an LSTM was shown to outperform RFs when operating on hand-crafted features, and the work in [9] obtained similar results when employing a bi-directional LSTM on mel-spectrograms after a harmonic-percussive source separation (HPSS) pre-processing stage. Similarly, [10] and [11] used long-term recurrent convolutional networks (LRCN) with source separation as a pre-processing stage and hand-crafted audio features as input to train genre-specific methods. The work in [12] obtained slightly better results using a CNN architecture on mel-spectrograms and concluded that pitch shifting as a data augmentation method yields a small performance improvement. In an extension to their work, the same authors proposed a regularisation mechanism to early layers in their network to compensate for the loudness-dependency of their CNN-based system [13]. The architectures presented in [12] and [9] were later explored within a knowledge distillation framework via a teacher-student paradigm [14]. Another recent deep learning approach is a joint method for singing melody detection and extraction based on a Convolutional RNN (CRNN) [15]. A comparative study [16] involving several network architectures and input representations showed that, due to the lack of large data volumes, data-demanding end-to-end architectures (that take raw samples as input) are outperformed by spectrogram-based methods.

Despite the paradigm shift towards the use of DL for most MIR tasks, there have been relatively few DL methods for the VD task. A possible explanation is the lack of large volumes of representative, high quality data. The Jamendo corpus [17], which is widely used in the context of VD, only contains 93 tracks (443 minutes of audio in total). In addition to the provided recordings, which were collected from the Creative Commons music platform Jamendo[1], the corpus contains segment-based labels and pre-computed data splits. The segments were annotated by a single annotator without a documented annotation strategy. DALI [18] contains 5358 tracks with time-aligned vocal melody notes and lyrics (sourced from karaoke platforms). The alignment is done automatically following a teacher-student paradigm to iteratively refine the results.

[1]https://www.jamendo.com

In addition to Jamendo and DALI, some other music datasets that have been extended, adapted, or re-purposed to be used in the context of VD. These include MedleyDB [19], for which frame-level vocal annotations can been derived from instrument activations [20], [21], or MIR-1K [22] and iKala [23], two collections of Chinese pop music originally developed for vocal separation where vocal segments can be inferred from fundamental frequency (F0) annotations. The RWC Pop dataset [24], which at the time of writing this paper was unavailable, contains 100 pop music tracks, for which singing voice annotations were provided in [25]. Furthermore, the work in [16] created a dataset of binary labels by randomly sampling a set of tracks from the FMA dataset [26] and manually annotating the presence or absence of vocals for 2-sec excerpts. This dataset contains slightly over 16000 excerpts that are split into training (75%) and testing (25%) data partitions.[2] In an effort to provide vocal segment annotations on a larger scale, the authors in [6] annotate a corpus of music tracks from various sources (including Jamendo, the RWC Pop dataset and YouTube). However, the dataset is not available at the time of writing.

Overall, available data for the VD task is rather limited in volume and diversity. Despite its small size and an experiment in [16] that revealed its limited generalisation capabilities to other music styles, the Jamendo corpus remains the default dataset for benchmarking VD systems.

The contributions of this paper are three-fold. First, we release DAACI-VoDAn (Vocal Detection ANnotations), a new annotated dataset for VD that contains 706 songs of diverse genres, artists, and cultures, together with manually created vocal segment annotations that follow a well-defined annotation strategy. With this dataset, we attempt to alleviate the problem of data scarcity for vocal detection and hope it will foster further research on the topic. Second, we benchmark two existing DL-based VD systems on DAACI-VoDAn and compare to their performance on the Jamendo corpus. Third, we propose a novel VD method that outperforms the state of the art on both on datasets. Our approach combines a convolutional head, pre-trained on large amounts of weakly-labeled data, to model relevant 2D spectral patterns, with a Temporal-Convolutional layer (TCN) to model their evolution of time.

II. DAACI-VoDAn Dataset

The dataset consists of 706 music tracks which were selected to cover a broad variety w.r.t. genre, era, and instrumentation, and are intended to be representative of the music universe available through commercial catalogues and streaming services. We provide segment-level vocal annotations for each track in the dataset, which have been created manually according to a well-defined annotation strategy described in more detail in Section II-A.

DAACI-VoDAn is based on full-length tracks with an average song duration of 234 seconds (ca. 3.8 minutes), covering 448 unique artists. The total amount of audio is 45 hours and 53 minutes, which represents more than six times the size of the Jamendo corpus. Overall, vocals are present in roughly 70% of audio frames. 25 out of the 706 tracks are purely instrumental and do not contain any vocals.

All data is available for research purposes under a non-commercial use license.[3] While copyright issues prevent us from sharing the audio files directly, we provide a metadata file which, in addition to artist and song title, links each song via a unique ID to a YouTube URL. Even though providing YouTube URLs has a few shortcomings, it is common practice since it enables the release of other valuable data, i. e. , vocal annotations.

A. Annotation methodology

The annotations were created manually by a team of four annotators using Sonic Visualiser [27] and professional-grade headphones. All annotators have formal music education and ample experience in annotating music for research purposes. The annotation effort was conducted in a peer-reviewing system where each annotation was created by one annotator and reviewed by another. The tracks were evenly distributed among the four annotators.

While the task of annotating vocal segments might appear straightforward at first glance, there can be numerous ambiguous scenarios. To provide consistent and extendable annotations, we defined an annotation strategy in the form of a set of rules:

1) We defined a minimum inter-region distance of 200 ms. Two vocal regions which are at least 200 ms apart are annotated as separate regions. Shorter vocal rests are not considered and a single long region is annotated instead.
2) We annotated heavily processed (e. g. , chorus or distortion effects) singing voice sections as vocals, as long as they are still recognizable as originating from a human voice.
3) We extended vocal sections to include reverb and delay "tails" as long as they are audible. Similarly, we considered inhalations or "breathing" sounds, e. g. , at the beginning of a vocal phrase, as part of vocal section.
4) We considered segments containing speech as vocal sections but excluded whistling. Vocal ensembles and background vocals are also labelled as vocals.

While the Jamendo dataset does not disclose any annotation strategy, an in-depth inspection showed that annotated vocal segments appear to only cover clear and unprocessed vocals and do not seem to include reverb tails or inhalations. Consequently, detecting vocals in DAACI-VoDAn can be considered a more challenging and granular task.

B. Data splits

In order to enable a direct comparison among methods, we provide pre-computed training (80%, 565 tracks), validation (10%, 71 tracks) and test (10%, 70 tracks) splits for the dataset. Since our collection contains multiple songs by the same artist, the splits do not overlap w.r.t. artist (all songs by the same artist belong to the same partition) in order to ensure a fair evaluation strategy and exclude any influence of the so-called *album effect*.

[2]For the sake of comparison, when all Jamendo tracks are split into non-overlapping 2-sec segments, a dataset of roughly 19000 excerpts is created.

[3]https://zenodo.org/record/7991496

Moreover, the dataset contains a few songs performed by two artists. In these cases, if both artists have at least one other song in the dataset separately, we ensure all tracks by both artists, as well as the joint track, are assigned to the same partition.

III. Experimental Setup

A. Comparative evaluation

We conduct a set of experiments to evaluate the performances of existing methods and the novel approach on both DAACI-VoDAn and Jamendo. More specifically, we select two existing deep learning-based systems: the CNN from [12] (`SCH-CNN`) and the RNN from [9] (`LEG-RNN`). Then, we propose a novel architecture for the task: a convolutional TCN (`CNN-TCN-TL`) which is described in detail in the next sections.

B. Baseline systems

While there do exist a few other more recent systems that achieved similar results (e. g. , [10], [11]), we select `SCH-CNN` and `LEG-RNN` as a reference, since, given their performance, they are still considered state of the art [20]. Furthermore, they use the same input representation (mel-spectrograms) as our proposed method, which enables a direct comparison. For both methods we follow the most recent version of the network architecture and the feature extraction stages, described in [20] and available online[4]. Based on the experiments reported in [13], we modify the original CNN from [12] and add zero-mean convolutional filters in the first layer of the network. For the `LEG-RNN`, we implement the best-performing network with three hidden layers of size 30, 20, and 40, respectively.

C. Proposed method

We propose a network with a convolutional front-end followed by a temporal convolutional (TC) layer [28]. The motivation behind this design is that the front-end is capable of learning 2D patterns in the input space and the TC layer can model the occurrence of these patterns over time. TC layers make use of dilated convolutions to increase the receptive field, enabling long temporal relationships to be captured efficiently. We refer to this architecture as `CNN-TCN-TL`.

Pre-processing. Similar to [10] and [11], we first run the source separation system Spleeter [29] on the music tracks as a pre-processing step to isolate the vocal component. Future work will explore the more recent approach Demucs [30]. Note that the baseline system in `LEG-RNN` follows a similar approach by employing HPSS as a pre-processing stage. After discarding the accompaniment, we compute the mel-spectrogram of the vocal track. All audio tracks are resampled to $44\,100$ Hz before computing the mel-spectrograms using librosa [31] with a hop size of 5.8 ms, and 64 mel bands. Note that our feature extraction parameters differ from those in `SCH-CNN` and `LEG-RNN`.

Network architecture. The proposed network, depicted in Figure 1, takes mel-spectrogram patches of size 344×64 as input, which covers roughly 2-sec of audio across 64 Mel bands. The front-end CNN has four layers with $16\,(3 \times 3)$[5], $32\,(3 \times 3)$, $32\,(3 \times 3)$, $32\,(3 \times 3)$, respectively, followed by ReLU activation and batch normalisation. After the second, third, and fourth convolutional layers, the frequency dimension is halved by using max-pooling layers. This convolutional head is followed by a fully-connected layer with 32 neurons and its output passes through a TC layer with $16\,(3 \times 3)$ filters and seven dilation levels exponentially increasing from 1 to 64. The output of the network is a dense layer with one neuron of sigmoid activation, wrapped into a time distributed layer to produce one prediction per input time frame.

Pre-training on weak labels. An advantage of this two-stage architecture is that the front-end can be pre-trained separately to enable transfer learning (TL). To this end, we use weakly-labeled data to pre-train the convolutional head of `CNN-TCN-TL` on global labels referring to larger segments. We then train its temporal component only on DAACI-VoDAn. Weak labels can be obtained automatically in large volumes and, despite their noisy nature, can increase the effective size of the training dataset for tasks where manual annotations are sparse and costly. Similar strategies have recently been successfully employed in several machine learning disciplines [32], [33], including the related task of audio event detection [34], [35]. Weakly-labeled data was also used in [36] with saliency maps to locate vocals in a music track with higher granularity.

Here, we obtain weak labels via track-level tags related to the presence of vocals. We use the LastFM API[6] to collect $8\,583$ tracks which contain the tag "instrumental", and $8\,807$ tracks with at least one tag of either "vocals", "male vocals", or "female vocals". We extract 60-sec from the middle of the song and, while we cannot ensure that vocals will be present within this segment and some noise will inevitably be introduced, we assign a global label to the excerpt based on the corresponding song-level tag.

With this dataset, we train `CNN-TCN-Pool`, a network that has the same convolutional head as `CNN-TCN-TL` but instead of a TCN layer, it applies a combination of global pooling operations (max- and average- pooling) to the CNN outputs, to summarise their content along the time dimension. After training, we freeze the weights of the convolutional head from `CNN-TCN-Pool` and use it as a front-end in `CNN-TCN-TL`. In this way, only the TC component is trained on the smaller target dataset.

Training setup. After the pre-training stage, we train `CNN-TCN-TL` for 100 epochs using the Adam optimizer [37] with an initial learning rate of 10^{-4}. We additionally implement an early stopping mechanism with a patience of 10 epochs also based on the validation loss. To account for class imbalance in the dataset, we employ class weights when training on DAACI-VoDAn.

Post-processing. All networks go through the same post-processing stage to produce the final binary predictions. Specifically, we first apply a median filter of fixed length (390 ms) to the raw outputs of the networks to smooth the predictions along time. The filter length was chosen based

[4]https://github.com/kyungyunlee/ismir2018-revisiting-svd

[5]The number of filters is written first (16) and their shape follows in parentheses (3×3).

[6]https://www.last.fm

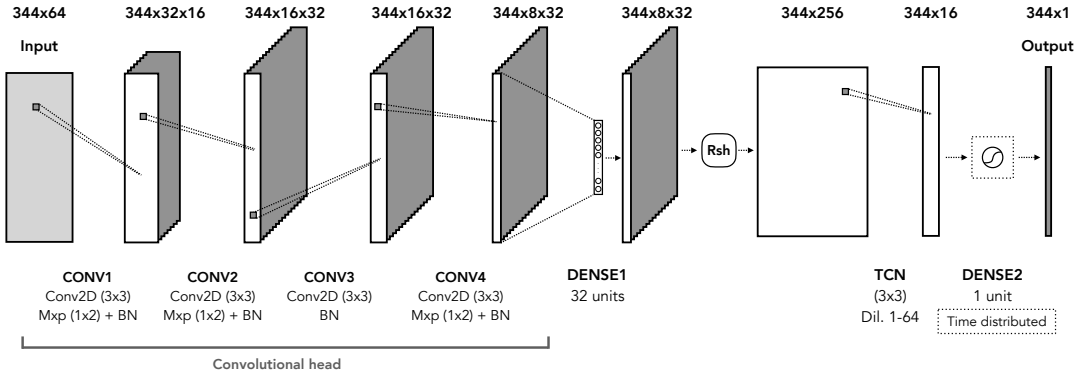

Fig. 1. CNN−TCN−TL architecture diagram. *Mxp* and *BN* refer to max-pooling operations and batch normalisation, respectively. Feature map dimensions are indicated above, and filter shapes for each convolutional layer are shown in parentheses under each layer name. *Rsh* is a reshape layer, *Dil.* stands for "dilation", and TCN is the temporal convolutional layer. The dotted rectangle indicates a time distributed layer, which has one unit with sigmoid activation.

TABLE I
COMPARATIVE EVALUATION RESULTS. SUFFIXES -J (JAMENDO) AND -D (DAACI-VODAN) INDICATE THE DATASET USED FOR TRAINING AND EVALUATION. *F1-Score, Precision, Recall* REFER TO THE POSITIVE (VOCALS) CLASS. THE BEST RESULTS FOR EACH DATASET ARE HIGHLIGHTED.

Model	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)
SCH-CNN-J	86.68	86.72	86.00	87.45
LEG-RNN-J	84.74	87.07	84.48	89.82
CNN-TCN-TL-J	**93.35**	**93.45**	**91.59**	**95.38**
SCH-CNN-D	85.87	90.28	85.94	94.02
LEG-RNN-D	86.94	90.72	89.10	92.11
CNN-TCN-TL-D	**90.16**	**92.93**	**92.22**	**93.67**

on the inter-region interval defined by the vocal annotation strategy. Then, a classification threshold is applied to binarise the predictions. This threshold is optimised independently for each model on the validation set.

IV. RESULTS

A. Comparative evaluation

Table I summarises the results comparing the different methods with respect to accuracy, precision, recall, and F1-Score, considering the presence of vocals as the positive class. All metrics are calculated by aggregating all frames in the test set (as opposed to track-based averaging).

Analysing the results obtained on the Jamendo corpus, we observe that, despite the limited amount of data, the proposed method outperforms the two baselines for all metrics, reducing the false positives and negatives by a large margin. Similarly, focusing on DAACI-VoDAn (lower part of Table I), we observe that CNN−TCN−TL yields the best results across metrics.

When comparing the performance of CNN−TCN−TL on both datasets, we observe a performance drop from Jamendo to DAACI-VoDAn. However, SCH−CNN and LEG−RNN experience the opposite. This behaviour can be explained by the complexity of the annotations and the effect of the source separation step. We mentioned above that DAACI-VoDAn annotations capture a set of subtleties that are more challenging to recognise. An inspection of the predictions revealed that the two baselines do not capture them. Hence, they benefit from the increased amount of data but have a poorer performance when compared to CNN−TCN−TL. On the other hand, the predictions from the proposed model capture part of these more granular information, showing that the system is better suited for the task than the baselines. However, more data examples covering these more ambiguous cases are necessary for the model to learn to distinguish them correctly. Moreover, an investigation of the Spleeter outputs showed that it eliminates part of these effects, e. g. , breaths before a sung phrase. Hence, in some cases, the model is unable to recognise them because they are not present in the signal after pre-processing.

TABLE II
ABLATION STUDY RESULTS ON DAACI-VODAN.

Model	Accuracy (%)	F1-Score (%)
CNN-TCN-Mix	87.89	91.33
CNN-TCN	89.88	92.69
CNN-TCN-TL	**90.16**	**92.93**

B. Ablation study

We conduct an ablation study to assess the effect of (a) source separation as a pre-processing step, and (b) the transfer learning from weak labels. To this end, we evaluate the performance of two variants of the proposed method on DAACI-VoDAn. CNN−TCN does not leverage TL and is trained on DAACI-VoDAn end-to-end instead. CNN−TCN−Mix furthermore skips the source separation step and is trained end-to-end on the original signal. Table II summarises the ablation study results. We observe that each step towards the proposed method improves the performance: CNN−TCN−Mix yields better results (cf. Table I) than SCH−CNN, which also operates on the mixture, but with a significantly smaller network (1.4M parameters for SCH−CNN vs. 51k parameters for CNN−TCN−Mix). Moreover, we find that source separation as a pre-processing stage (CNN−TCN) contributes to a performance improvement (from 87 to 89% accuracy), compared to CNN−TCN−Mix. Finally, when we leverage TL on weakly-labeled data (CNN−TCN−TL), we further improve the model's performance, showing that the integration of these steps into our proposed method yields the best results in the studied scenarios.

V. CONCLUSIONS

In this paper, we introduced DAACI-VoDAn, a novel dataset for vocal detection that contains 706 full-length music

tracks and manually-annotated vocal segments. We proposed a new deep learning-based method for VD (`CNN-TCN-TL`) that uses source separation as a pre-processing stage and combines a convolutional head, pre-trained on weakly-labeled data, with a temporal-convolutional architecture. We evaluated `CNN-TCN-TL` and two baseline systems on the new dataset and on the Jamendo corpus and found that the proposed method outperformed the baselines in all studied scenarios. Furthermore, an ablation study showed that both source separation and pre-training on large amounts of weakly-labeled data boost the performance of the model. However, we found that the source separation step eliminates some parts of the signal that the model requires to recognise the subtleties present in DAACI-VoDAn. In this direction, future work will explore a double-branch architecture that combines pre-processed and raw inputs to account for potential mistakes in the separation.

VI. Acknowledgements

We want to thank Haydn Leech, Alex Harley, and Dan Peeke for creating the dataset annotations.

References

[1] E. Demirel, S. Ahlbäck, and S. Dixon, "MSTRE-Net: Multistreaming acoustic modeling for automatic lyrics transcription," in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 151–158.

[2] W. Cai, Q. Li, and X. Guan, "Automatic singer identification based on auditory features," in *Proc. of the Intl. Conference on Natural Computation (ICNC)*, 2011, pp. 1624–1628.

[3] N. Kroher and E. Gómez, "Improving accompanied flamenco singing voice transcription by combining vocal detection and predominant melody extraction." in *Proc. of the Intl. Computer Music Conference (ICMC)*, Athens, Greece, 2014, pp. 1051–1056.

[4] B. Lehner, G. Widmer, and R. Sonnleitner, "On the reduction of false positives in singing voice detection," in *Proc. of the Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7480–7484.

[5] A. L. Berenzweig and D. P. Ellis, "Locating singing voice segments within music signals," in *Proc. of the Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2001, pp. 119–122.

[6] B. Lehner, J. Schlüter, and G. Widmer, "Online, loudness-invariant vocal detection in mixed music signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1369–1380, 2018.

[7] A. Pikrakis *et al.*, "Unsupervised singing voice detection using dictionary learning," in *Proc. of the European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1212–1216.

[8] B. Lehner, G. Widmer, and S. Bock, "A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks," in *Proc. of the European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, 2015, pp. 21–25.

[9] S. Leglaive, R. Hennequin, and R. Badeau, "Singing voice detection with deep recurrent neural networks," in *Proc. of the Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 121–125.

[10] R. Romero-Arenas, A. Gómez-Espinosa, and B. Valdés-Aguirre, "Singing voice detection in electronic music with a long-term recurrent convolutional network," *Applied Sciences*, vol. 12, no. 15, 2022.

[11] Y. Sun *et al.*, "Investigation of singing voice separation for singing voice detection in polyphonic music," in *Proc. of the Conference on Sound and Music Technology (CSMT)*. Singapore: Springer Nature Singapore, 2023, pp. 79–90.

[12] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015, pp. 121–126.

[13] J. Schlüter and B. Lehner, "Zero-mean convolutions for level-invariant singing voice detection," in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 321–326.

[14] S. Paul *et al.*, "Knowledge distillation for singing voice detection," in *Proc. of Interspeech*, 2021, pp. 4159–4163.

[15] S. Kum and J. Nam, "Joint detection and classification of singing voice melody using convolutional recurrent neural networks," *Applied Sciences*, vol. 9, no. 7, p. 1324, 2019.

[16] S. D. You, C.-H. Liu, and W.-K. Chen, "Comparative study of singing voice detection based on deep neural networks and ensemble learning," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–18, 2018.

[17] M. Ramona, G. Richard, and B. David, "Vocal detection in music with support vector machines," in *Proc. of the Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 1885–1888.

[18] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Dali: a large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm." in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, September 2018.

[19] R. Bittner *et al.*, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, vol. 14, 2014, pp. 155–160.

[20] K. Lee, K. Choi, and J. Nam, "Revisiting singing voice detection: a quantitative review and the future outlook," in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 506–513.

[21] R. Monir, D. Kostrzewa, and D. Mrozek, "Singing voice detection: A survey," *Entropy*, vol. 24, no. 1, p. 114, 2022.

[22] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2009.

[23] T.-S. Chan *et al.*, "Vocal activity informed singing voice separation with the iKala dataset," in *Proc. of the Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 718–722.

[24] M. Goto *et al.*, "RWC Music Database: Popular, Classical and Jazz Music Databases." in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2002, pp. 287–288.

[25] M. Mauch *et al.*, "Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music." in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 233–238.

[26] M. Defferrard *et al.*, "FMA: A dataset for music analysis," in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[27] C. Cannam, C. Landone, and M. Sandler, "Sonic Visualiser: an open source application for viewing, analysing, and annotating music audio files," in *Proc. of the ACM Multimedia Intl. Conference*, Firenze, Italy, October 2010, pp. 1467–1468.

[28] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[29] R. Hennequin *et al.*, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020.

[30] A. Défossez *et al.*, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[31] B. McFee *et al.*, "librosa: Audio and music signal analysis in python," in *Proc. of the Python in Science Conference*, vol. 8, 2015, pp. 18–25.

[32] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7598–7604, 2019.

[33] S. Rajaraman and S. Antani, "Weakly labeled data augmentation for deep learning: a study on COVID-19 detection in chest X-rays," *Diagnostics*, vol. 10, no. 6, p. 358, 2020.

[34] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proc. of the ACM Intl. Conference on Multimedia*, 2016, pp. 1038–1047.

[35] N. Turpault *et al.*, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proc. of the DCASE Workshop*. New York University, 2019.

[36] J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples," in *Proc. of the Intl. Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2017, pp. 44–50.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.