applied sciences

MDPI

Article

Singing Voice Detection in Electronic Music with a Long-Term Recurrent Convolutional Network

Raymundo Romero-Arenas *, Alfonso Gómez-Espinosa * and Benjamín Valdés-Aguirre

Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Ave. Epigmenio González 500, Fracc. San Pablo, Querétaro 76130, Mexico; bvaldesa@itesm.mx

* Correspondence: ray.romero.arenas@gmail.com (R.R.-A.); agomeze@tec.mx (A.G.-E.); Tel.: +52-442-238-3302 (A.G.-E.)

Abstract: Singing Voice Detection (SVD) is a classification task that determines whether there is a singing voice in a given audio segment. While current systems produce high-quality results on this task, the reported experiments are usually limited to popular music. A Long-Term Recurrent Convolutional Network (LRCN) was adapted to detect vocals in a new dataset of electronic music to evaluate its performance in a different music genre and compare its results against those in other state-of-the-art experiments in pop music to prove its effectiveness across a different genre. Experiments on two datasets studied the impacts of different audio features and block size on LRCN temporal relationship learning, and the benefits of preprocessing on performance, and the results generate a benchmark to evaluate electronic music and its intricacies.

Keywords: singing voice detection (SVD); music information retrieval (MIR); electronic music; long-term recurrent convolutional network (LRCN)

Citation: Romero-Arenas, R.; Gómez-Espinosa, A.; Valdés-Aguirre, B. Singing Voice Detection in Electronic Music with a Long-Term Recurrent Convolutional Network. *Appl. Sci.* **2022**, *12*, 7405. https://doi.org/10.3390/app12157405

Academic Editors: Jongweon Kim and Yongseok Lee

Received: 29 June 2022
Accepted: 21 July 2022
Published: 23 July 2022

1. Introduction

The singing voice is one of the essential components of music, as it communicates the emotions and motives of any song through lyrics. Its high level of expressiveness has made the human voice a topic of numerous study fields, one of them being Music Information Retrieval (MIR). In it, Singing Voice Detection (SVD) is the ability to identify segments of singing activity. It is one of the most researched topics due to its contributions as a preprocessing step in the performance of other tasks and applications such as Singing Voice Separation (SVS) [1,2], Vocal Melody Extraction (VME) [3], and Lyric Alignment [4,5], Lyric Transcription [6], among others.

Traditional methods combine attributes of speech with classification algorithms to detect singing voice segments in songs. For example, features such as Mel-Frequency Cepstral Coefficients (MFCC), Perceptual Linear Prediction (PLP), and Linear Prediction Cepstral Coefficients (LPCC), which are obtained by introducing cepstrum coefficients in Linear Predictive Coding (LPC) [7], and classifiers such as Gaussian Mixture Models (GMM), Hidden Markov Models (HMM) and Support Vector Machines (SVM) were introduced into speech recognition systems [8]. Rocamora and Herrera [9] studied the accuracy of SVD of various statistical descriptors and features, concluding that MFCC was the most appropriate feature, and SVMs was the best-performing classifier. However, they have certain limitations for SVD, as speaking and singing voice are different in terms of how singing uses breath and muscle tension to control pitch and duration, therefore giving it a higher average intensity and greater dynamic range compared to speech [10].

On the other hand, Deep Learning (DL) techniques have been recently used in SVD due to its feature representation and time-space modeling capabilities [11]. For example, Schlüter et al. [12] introduced a model for SVD using a Convolutional Neural Network (CNN) that could work with Mel-spectograms through data augmentation. You et al. [13] combined CNNs with MFCCs and Fast Fourier Transform (FFT) features,

whereas Huang et al. [14] did it with Discrete Fourier Transform (DFT) coefficients, obtaining higher detection accuracy than MFCCs. Hughes et al. [15] proved that a Recurrent Neural Network (RNN) outperformed GMM systems on voice detection activities, and Lehner et al. [16] increased its effectiveness with a Long Short-Term Memory (LSTM) network to detect vocal and nonvocal frames.

Further attempts have mostly consisted of combining DL techniques to separate tasks and increase their performance. Leglaive et al. [17] combined a bidirectional RNN with an LSTM (BLSTM-RNNs) and used a temporal context in order to classify the input vector, whereas Cho et al. [18] used a Gated Recurrent Unit (GRU) to allow each unit to adaptively capture dependencies across time scales. Most recently, Zhang et al. [11] achieved state-of-the-art performance using SVS as a preprocessing method for classifying input vocal signals and a Long Short-Term Recurrent Convolutional Network (LRCN) to extract and sequence the vocal background in the time domain for vocal detection.

The majority of SVD systems are designed and evaluated with popular music. However, Scholz et al. [19] showed that these tend to not generalize well to other genres unseen during training, implying that focusing solely on pop music limits their general applicability. There have been attempts to train SVD models with other music genres. Krause et al. [20] proved the generalization capabilities and robustness of two models with opera recordings as a dataset. However, there are still many genres with their own peculiarities that have not been explored yet, such as electronic music.

Electronic music has been studied in music generation. Conklin et al. [21] developed a Digital Audio Workstation (DAW) plug-in to generate chord sequences and define patterns in EDM through machine learning and model sampling. However, as electronic music focuses more on repetition and looping rather than singing, it has not been explored in other study fields such as MIR. Furthermore, it is evolving into various sub-genres such as dubstep, house, techno, glitch, drum and bass, among others, each one of them with their own intricacies and techniques, so it is still a fresh and developing study area to explore.

Particular differences exists between popular and electronic music due to the singing techniques and instrumentation involved. Within electronic music, songs with a heavy emphasis on digital instruments such as synthesizers and samplers, and the usage of tools such as auto-tune, an audio processor that measures and applies subtle corrections to the pitch of vocal and instrument recordings, provide interesting opportunities to test the effectiveness of MIR algorithms in new scenarios. Additionally, previous research has shown that SVD systems may overfit to factors such as loudness [22] and singing style [19], and therefore it is hard to expect them to generalize to unseen genres.

In this paper, a LRCN is evaluated with a new dataset of electronic music to compare its performance against public-domain datasets of pop music so as to prove the effectiveness of the model in a different and unstudied genre. This will provide a baseline result to study the electronic music genre in the SVD field, and it will yield new insights into the aspects of data training when using SVD algorithms to test unexplored music genres.

This paper is organized as follows: Section 2 gives a general introduction to the SVD system and datasets used. Section 3 presents the performed experiments and obtained results on the system. Conclusions are described in Section 4.

2. Materials and Methods

2.1. Architecture Layout

The proposed system has a preprocessing step to filter the audio from the musical accompaniment to increase the performance of the feature extraction and LRCN training with the datasets, thus maximizing the results to perform the comparisons between them. The architecture of the system is shown in Figure 1.

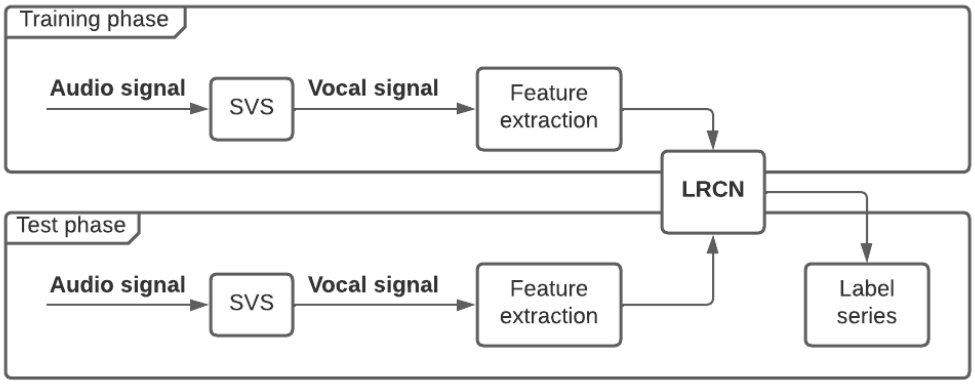

Figure 1. Overview of the proposed SVD system, including preprocessing, feature extraction, and LRCN feeding in the both training and test phases.

2.2. Singing Voice Separation

Singing Voice Separation (SVS) is the task of filtering out musical accompaniment from vocals in a song to obtain a relatively pure human voice. Therefore, it complements SVD in the goal of detecting a singing voice. From all SVS algorithms, Cohen-Hadria et al. [23] concluded that the most effective method is based on U-Net, a deep neural network proposed by Jansson et al. [24] with six convolutional and six deconvolutional layers that takes the spectrum of a singing voice as input. However, if there is a string accompaniment, then the audio cannot be cleanly split, as the separated vocal signal may contain collateral noise as a result.

2.3. Feature Extraction

With the LRCN, audio features can be combined from consecutive audio frames to differentiate between human and nonhuman voices by forming two-dimensional figures in the (x,y) form of (audio frame in time, coefficients of the features). For the identification of vocal signals, the following features are used:

1. ZCR (Zero-Crossing Rate): How many times a signal value changes sign (negative or positive) in a time framE.
2. MFCC (Mel-Spectogram Cepstral Coefficients) [25]: Coefficients that represent the power spectrum speech based on human audition perception. First-order and second-order difference of MFCC are also computed.
3. LPCC (Linear Prediction Cepstral Coefficients) [26]: Cepstrum coefficients in Linear Predicitive Coding (LPC) parameters that represent the nature of the sound based on the vocal tract shape.
4. Chroma [27]: Descriptor that represents the tonal content of an audio signal.
5. SSF (Spectral Statistical Features) [28]: Frequency information of the audio signal (RMSE, centroid, roll-off, flatness, bandwidth, contrast, polly).
6. PLP (Perceptual Linear Prediction) [29]: Alternative to MFCC that wraps the spectra to maintain the content of speech.
7. Spectrum: Representation of sound in terms of frequency vibration. It is calculated to verify the performance of the LRCN and its feature fusion capabilities.

2.4. LRCN Components

The LRCN is a deep model that learns time-series relationships and integrates information in space for deep feature extraction. The CNN layer spatially extracts the audio features for deep feature extraction, while the LSTM layer performs a deep feature encoding of the convolutional layer output to learn the temporal relationship and obtain a final singing or no-singing label.

The LRCN has an input layer with the size of the combined feature vector, an output layer with a sigmoid unit and three hidden layers, and outputs values for each frame block between 0 (no singing voice) and 1 (singing voice). By combining a feature extraction CNN and a LSTM, the model passes the fusion features of successive audio frames into a new vector and then synthesizes its temporal dynamics in a sequential order. The LRCN layers and main functions are shown in Figure 2.

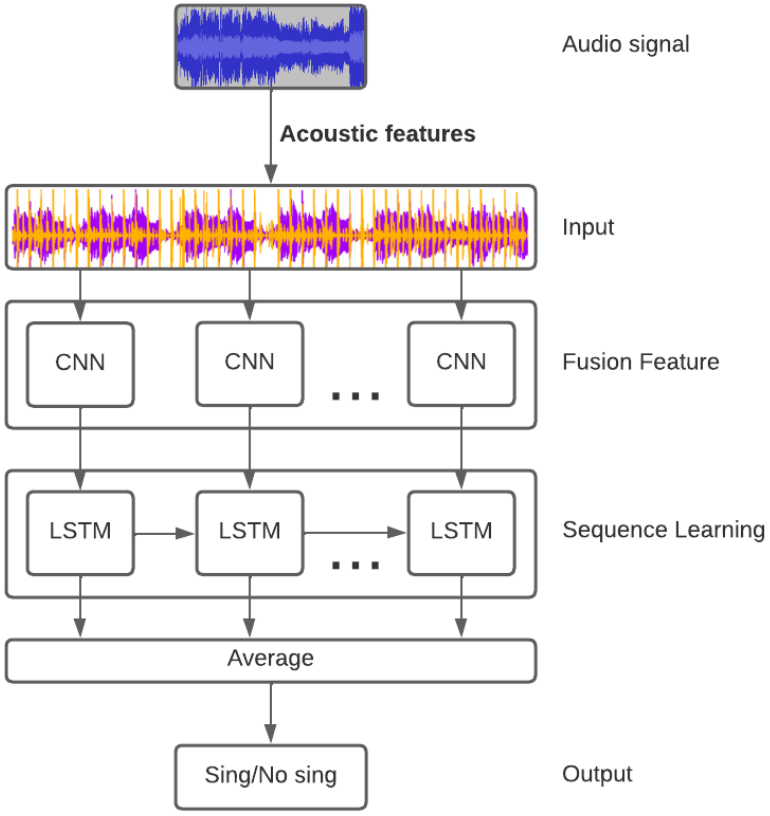

Figure 2. The LRCN has a CNN layer to fuse the audio features into a vector, while the LSTM layer is in order to learn its sequencing to output a sing/nosing label.

The LRCN layer has a stack of one-dimensional inputs and a set of one-dimensional kernels. The max-pooling layer subsamples the inputs by taking the maximum over groups of contiguous frames, and the input layer receives the fusion features of continuous frames in a fixed block size. The parameters for the LRCN layers are shown in Table 1.

Table 1. Parameters of the LRCN layers.

Layer	Output Shape	Parameter
reshape-1 (Reshape)	(None,1,1,20,276)	0
conv_lstm_2d-1 (ConvLSTM2D)	(None,1,17,13)	60,164
dropout-1 (Dropout)	(None,1,17,13)	0
max_pooling_2d-1 (MaxPooling2D)	(None,1,8,13)	0
dropout-2 (Dropout)	(None,1,8,13)	0
flatten-1 (Flatten)	(None,104)	0
dense-1 (Dense)	(None,200)	21,000
dropout-3 (Dropout)	(None,200)	0
dense-2 (Dense)	(None,50)	10,050
dropout-4 (Dropout)	(None,50)	0
dense-3 (Dense)	(None,1)	51

The input data are concatenated on continuous frames. The input vector is mapped to one label based on the ground truth. In the LRCN layer, 256 convolution filters and a convolution kernel operate a one-dimensional convolution along with the features, and it determines the cell state in the input grid and past states of its neighbors through a convolution operator in state-to-state and input-to-state transitions. The structure of the LRCN is shown in Figure 3. The equations are shown in Equations (1)–(5). All inputs, gates, hidden states, and cell outputs of the LRCN are 3D tensors.

$$i(x) = \sigma(W_i * [conv(X(x)), H(x-1), C(x-1)] + b_i) \quad (1)$$

$$o(x) = \sigma(W_o * [conv(X(x)), H(x-1), C(x)] + b_o) \quad (2)$$

$$f(x) = \sigma(W_f * [conv(X(x)), H(x-1), C(x-1)] + b_f) \quad (3)$$

$$C(x) = f(x) * C(x-1) + i(x) * tanh(W_c * [conv(X(x)), H(x-1)] + b_c) \quad (4)$$

$$H(x) = o(x) * tanh(C(x)) \quad (5)$$

where:

$i(x)$ = Input gate;
$o(x)$ = Output gate;
$f(x)$ = Forget gate;
$C(x)$ = Input LRCN cell;
$H(x)$ = Output LRCN cell;
W = Matrix of weights;
b = Vector of bias;
σ = Sigmoid function;
* = Element-wise product;
conv. = Convolution operator.

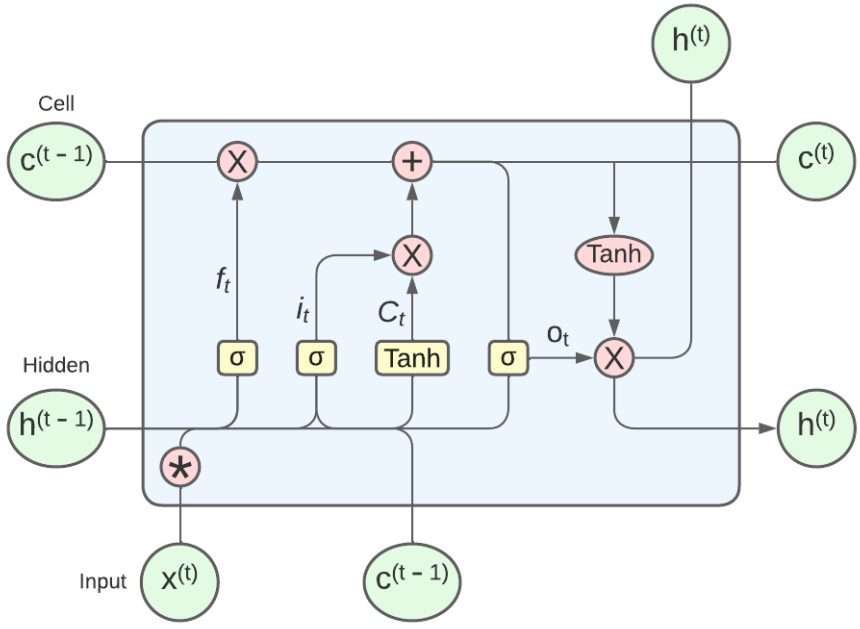

Figure 3. Inner structure of an LRCN Layer, describing input and output cells and layer operations.

2.5. Existing Dataset—Jamendo Corpus

The Jamendo Corpus is a public dataset of 93 copyright-free pop songs introduced by Mathieu Ramona et al. in [30]. Each song is encoded in 44.1 kHZ stereo .ogg or .mp3 format, and has a file with vocal annotations of signing and non-singing parts by the same

person to provide ground truth data. The dataset is divided into a training set of 61 songs, validation set of 16 songs, and test set of 16 songs.

2.6. Proposed Dataset—Electrobyte

Since there are no datasets of electronic music for SVD purposes, a new one was compiled from songs within the genre that have singing or lyrics. In order to avoid using copyrighted music, the songs were extracted from the following sources:

1. **The Arcadium** [31]: Record Label by German artist and producer TheFatRat dedicated to making available copyright-free gaming music.
2. **NCS** [32]: Record Label by British youtuber Billy Woodford dedicated to making available royalty-free electronic dance music for gaming and content creator communities.

The proposed dataset, named **Electrobyte**, consists of 90 electronic songs from various sub-genres such as Glitch Hop, Bass, Drum and Bass, House, Dubstep, Drumstep, Chill, EDM, and Electro Pop, among others, and follows the standards of SVD datasets such as Jamendo Corpus and RWC Popular Music:

- All songs are stored in .mp3 format;
- Voice activation annotations were manually performed and stored for each song in a .lab file;
- The dataset is divided into a training set of 60 songs, validation set of 15 songs, and test set of 15 songs, which were all randomly selected for each set.

2.7. Evaluation Metrics

To evaluate and compare the results of the datasets, model predictions were compared with the ground truth labels to obtain the number of results over all songs in the test set. Accuracy, precision, recall, and F1-measure were calculated to summarize results. The metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

where:

TP = Total of true positive matches;
FP = Total of false positive matches;
TN = Total of true negative matches;
FN = Total of false negative matches;
$F1$ = $F1$-measure.

Accuracy is the ratio of frames classified correctly. Precision is the ratio of frames classified as voiced. Recall is the ratio of frames classified correctly as voiced. $F1$-measure combines precision and recall to measure the accuracy of a model on a dataset. In this case, $F1$-measure is the most important metric, as it implies how correct our model may be when classifying the frame of a song.

3. Experiments

3.1. Technical Requirements

Four experiments were conducted in total. The first involved comparing the performance of the audio features and their combinations as input to the proposed LRCN to find

the optimal feature fusion. In the second experiment, the block size of continuous frames for the input layer was tested. In the third, SVS was evaluated to verify its effectiveness as a preprocessing step. In the last experiment, a comparison between the tested datasets and the state-of-the-art SVD systems was conducted. To replicate them, the proposed LRCN can be found in the following Github repository: https://github.com/Raylogic/SVD-LRCN (accessed on 9 June 2022). The results provided are the typical values obtained after several iterations of the experiments. The general parameters and configuration values used were the following:

- For the training of the LRCN, the following parameters were used across all experiments, based on the model proposed by Zhang et al. [11]:
  - Batch size = 32;
  - Drop-out rate = 0.2;
  - Learning rate = 0.0001;
  - Number of epochs = 10,000;
  - Early stopping = true.
- Feature vectors are calculated by dividing the audio signal into overlapping frames and calculating the FFT (Fast Fourier Transform) of each frame in a Hamming window. After computing and combining the coefficients of MFCC, LPCC, PLP, chroma, and SSF, each feature vector contains 288 coefficients, which then are used to classify the vocal or nonvocal segments with the LRCN model. The frame parameters are based on the values proposed by Zhang et al. [11]:
  - Sampling rate = 16,000 Hz;
  - Low-frequency limit = 0 Hz;
  - High-frequency limit = 800 Hz;
  - Overlap size = 1536 samples (0.96 s);
  - Frame size = 2048 samples (1.28 s).
- For the hardware requirements, the system was operated and trained on a laptop with an Intel(R) Core(TM) i7-9750H CPU @2.60GHz and 16GB of RAM.
- For the software requirements, Python 3.9.5 was used to develop the SVD system. Additionally, the following Python libraries were installed:
  - audioread—2.1.9;
  - chainer—7.8.1;
  - ffmpeg—1.4;
  - h5py—3.6.0;
  - joblib—1.1.0;
  - keras—2.8.0;
  - librosa—0.9.1;
  - matplotlib—3.4.2;
  - numba—0.55.1;
  - numpy—1.20.3;
  - pandas—1.4.1;
  - playsound—1.2.2;
  - pydub—0.25.1;
  - reportlab—3.6.9;
  - scikit-learn—1.0.2;
  - scipy—1.8.0;
  - SoundFile—0.10.3;
  - spectrum—0.8.1;
  - tensorflow—2.8.0;
  - tqdm—4.64.0;
  - wavfile—3.1.1.

To perform the following experiments, preprocessing steps were carried out beforehand on each dataset with the Preprocessing.py script. First, songs were transformed into

.wav format. Then, SVS was performed on all of them through U-Net, generating two audio files per each (raw and vocal audio signal). Feature extraction was then performed on the audio files and stored in a .joblib file, where the feature vectors were generated per block and its associated sing/nosing label was added. Finally, the .joblib files were compiled in two .h5 files (raw and vocal audio signal), where the data were separated into training, valid and test sets. The files were later used to train the LRCN for the SVD task.

In the LRCN.py script, LRCN layers and tasks were configured, which specifically described the necessary functions to extract the data form the .h5 files, train and validate the model, load model weights, perform Early Stopping, predict the presence or absence of singing on the test set songs and obtain the evaluation metrics by calling the Evaluation.py script. In addition, it contained the code to perform the four experiments and store their results on CSV files. The experiments on the datasets were called from the SVD.py script, while the plots from the results were generated on the Box.py script.

3.2. Performance across Features

In the first experiment, features were individually selected to perform the SVD task on the separated vocal signal of the Jamendo and Electrobyte datasets. Based on their *F*1-measure performance, they were selected or dropped for the next phase of the experiment. To determine the best feature combination, based of their performance of a single feature, each was added into the combined feature vector one by one. Once all of the best combinations were evaluated, the one with the best *F*1-measure was selected as the input for the LRCN.

The performance of the features on the separated vocal signal evaluated individually on the Jamendo dataset is shown in Figure 4, and on the Electrobyte dataset in Figure 5.

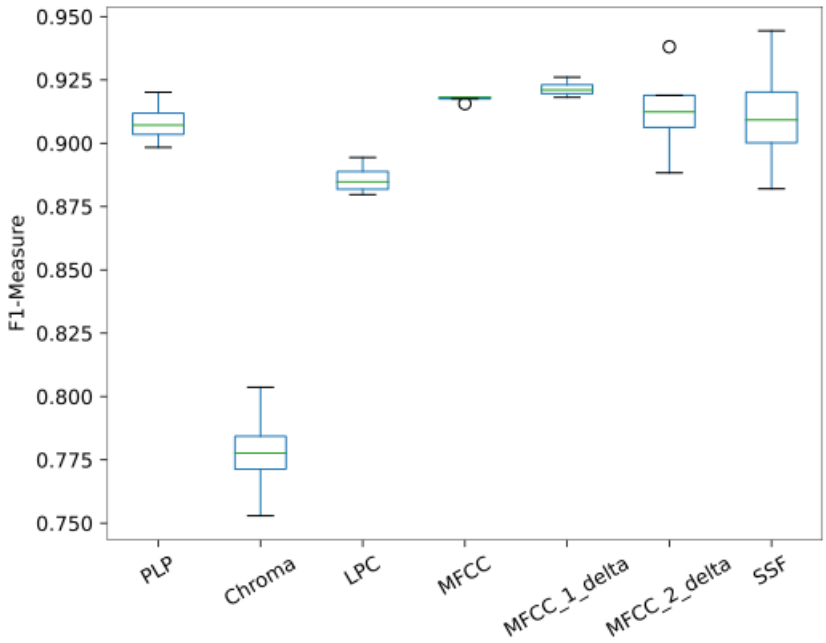

Figure 4. Performance of each audio feature in the Jamendo dataset individually. Circles represent atypical values within the evaluation metrics.

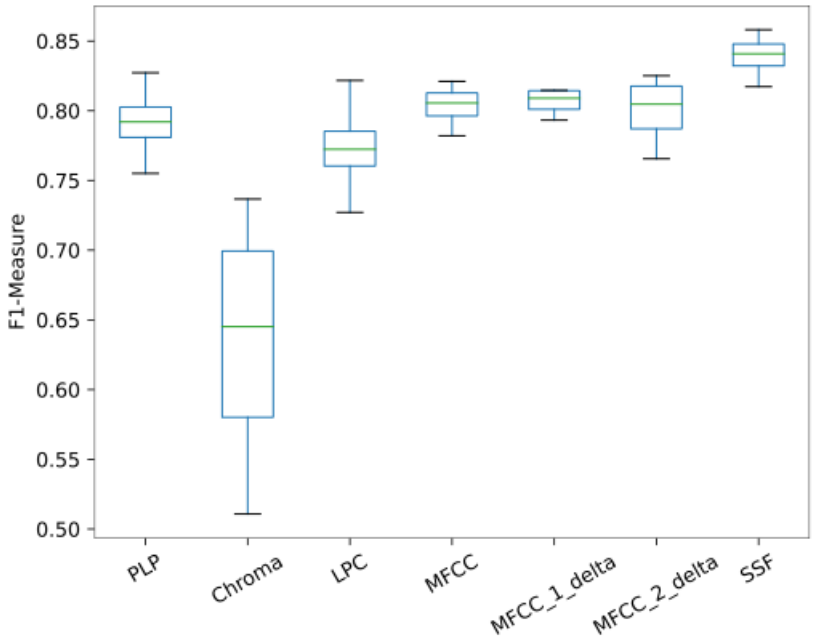

Figure 5. Performance of each audio feature in the Electrobyte dataset individually.

From the results in Figures 4 and 5, the chroma feature exhibited the poorest performance for SVD task, while the SSF achieved the best one. The feature sets were concatenated in each frame and extracted with the convolution layer of the LRCN. For the feature combination phase, the first feature added to the vector was the SSF given its highest *F*1-measure, and the rest were incrementally added to evaluate all possible combinations. The results of the combined features are shown in Figure 6 for the Jamendo dataset and Figure 7 for the Electrobyte dataset.

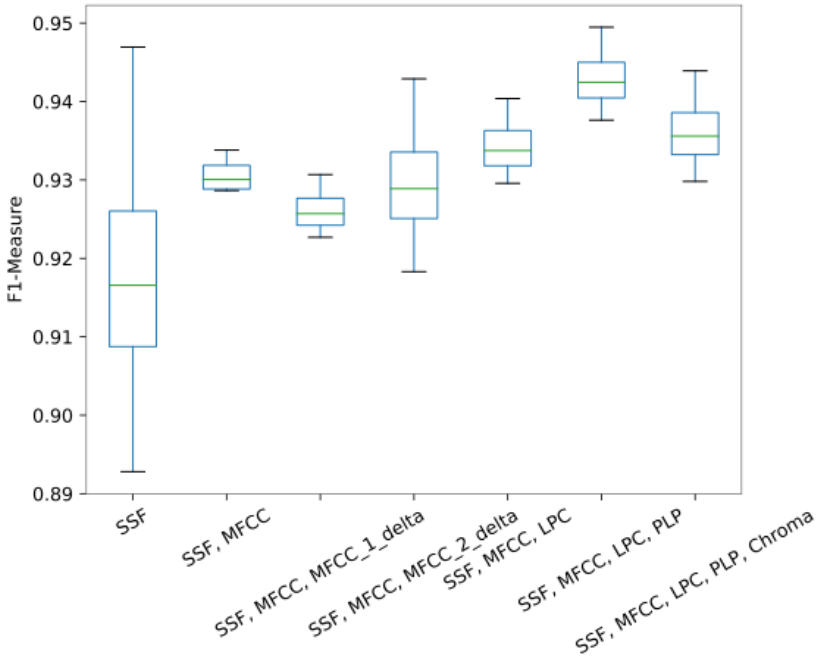

Figure 6. Performance of the best possible feature combinations in the Jamendo dataset.

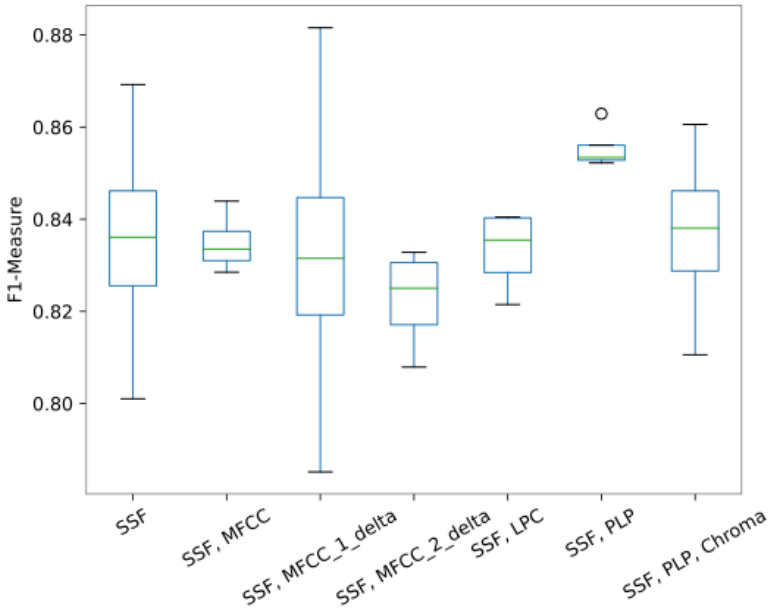

Figure 7. Performance of the best possible feature combinations in the Electrobyte dataset. Circles represent atypical values within the evaluation metrics.

From the results on the Jamendo dataset shown in Figure 6, SSF, MFCC, LPCC, and PLP was the feature combination that achieved the best performance. For the Electrobyte dataset in Figure 7, the best feature combination was SSF and PLP. With the observations from both datasets, the combined features were set as the SSF and PLP.

3.3. Block Size Setting

In the second experiment, the input frame-block size was iterated to find the optimal number of successive frames to feed the LRCN input layer and serve as a control variable for the datasets (Jamendo and Electrobyte). Block sizes from 5 to 29 frames were evaluated to find the amount that gave the best *F*1-measure and most consistent metrics. Figure 8 shows the block sizes performance on the Jamendo dataset, and Figure 9 on the Electrobyte dataset.

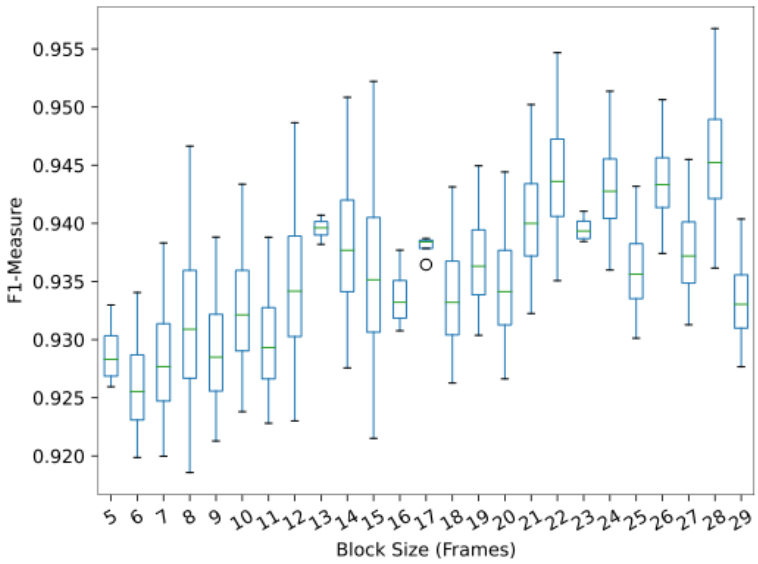

Figure 8. Performance of block sizes (5–29 frames) as input of the LRCN for the Jamendo dataset. Circles represent atypical values within the evaluation metrics.

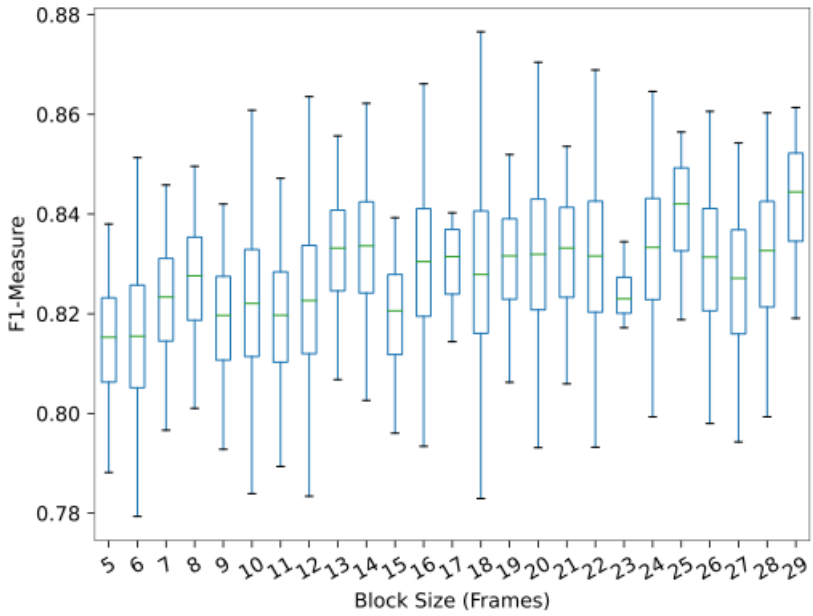

Figure 9. Performance of block sizes (5–29 frames) as input of the LRCN for the Electrobyte dataset.

From the results on the Jamendo dataset, the effect of block size in the LRCN barely fluctuated in terms of *F*1-measure, whereas on the Electrobyte dataset the differences were a little more pronounced. Based on the experiments and the research performed by Zhang et al. [11], the input block size for the LRCN was set to 20 frames.

3.4. Effects of Singing Voice Separation

In this implementation, SVS was adopted as a preprocessing step for SVD through the U-Net network to separate vocals from musical accompaniment and generate two versions of audio signals (raw and audio vocals). The third experiment consisted of extracting the features of both versions in both datasets to train the LRCN and validate the effects of the preprocessing by comparing its F1-measure. The model was set with the block size of 20 frames set on the previous experiment, and a vector of 276 components to remove the feature of chroma (12). The results were compared with both the split vocal and raw audio. Performances are shown in Figure 10.

From the comparison of the classification results from both datasets, the use of separated vocal signals exceeded the raw signals in terms of *F*1-measure, so the application of SVS as a preprocessing step does improved SVD performance. However, it is noticeable that the difference was way more pronounced in Jamendo (more than 20%), whereas in Electrobyte it was minimal (less than 2%). This contrast of results between datasets can be attributed to the unique characteristics of electronic music in terms of voice management. However, further research needs to be performed to pinpoint the exact causes of the result disparities.

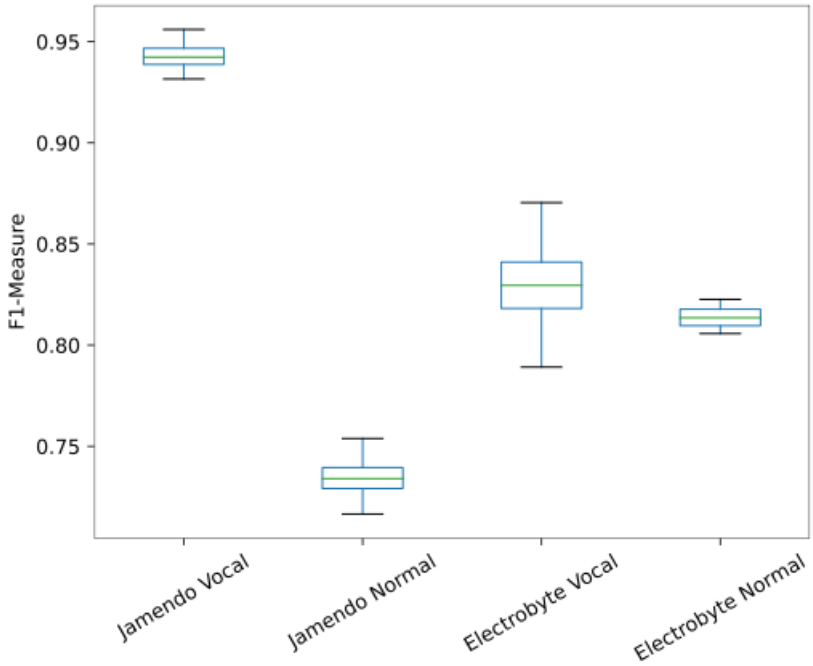

Figure 10. Performance of the LRCN when using and not using SVS as a preprocessing step.

3.5. Comparison with Related Works on Existing Datasets

As the last experiment, to validate the proposed SVD system, both datasets were used to perform 10 prediction results. From all of the evaluation metrics, the *F*1-measure provides the most information about the effectiveness of the model. Furthermore, the standard deviation was calculated to validate the consistency of the results. The model was set with the best parameters obtained on the previous experiments (vector of 276 components, block size of 20 frames, SVS as a preprocessing step). The averaged results are shown in Table 2.

Table 2. Performance of the SVD system on two different datasets.

	Accuracy	Precision	Recall	F1-Measure	Deviation
Jamendo	0.939	0.937	0.945	0.942	0.004
Electrobyte	0.833	0.798	0.861	0.828	0.006

The SVD system performed very well on the Jamendo dataset, with an *F*1-measure of 0.942 and standard deviation of 0.004, whereas on the Electrobyte dataset it achieved an *F*1-measure of 0.828 and standard deviation of 0.006. To validate the robustness of the model, a K-fold cross-validation was performed on both datasets, in which the training, validation, and test sets were resampled seven times by randomly selecting the songs for each. Across all seven samples individually, the standard deviation for the averaged *F*1-measure on Jamendo was lower than 0.004, whereas for Electrobyte it was lower than 0.008. It is worth mentioning that by repeating the experiment several times, one of the samples performed better than the original set on both datasets, with an *F*1-measure of 0.966 on Jamendo and 0.885 on Electrobyte. However, when considering the results of the samples and the original dataset as a whole, the standard deviation for the averaged *F*1-measure on the Jamendo dataset was 0.016, whereas for the Electrobyte dataset it was 0.021. Although the performance of the system in Electrobyte was lower compared to Jamendo across all samples, as this is the first study evaluating SVD in electronic music, these results will serve as a framework for future studies about the intricacies of the genre and how to improve the detection of vocals with more effective models.

Finally, the SVD system was compared with the best value performance from Ramona [30], Schlüter [12], Lehner-1 [33], Lehner-2 [34], Lehner-3 [16], Leglaive [17], and Zhang [11] on the Jamendo dataset. The comparison results are shown in Table 3.

Table 3. Comparison of the SVD system with existing methods on Jamendo.

	Accuracy	Precision	Recall	F-Measure	Deviation
Ramona [30]	0.822	-	-	0.831	-
Schlüter [12]	0.923	-	0.903	-	-
Lehner-1 [33]	0.882	0.880	0.862	0.871	-
Lehner-2 [34]	0.848	-	-	0.846	-
Lehner-3 [16]	0.894	0.898	0.906	0.902	-
Leglaive [17]	0.915	0.895	0.926	0.910	-
Zhang [11]	0.924	0.926	0.924	0.927	-
LRCN	0.939	0.937	0.945	0.942	0.004

Table 3 shows the comparison of the results of the proposed LRCN on the Jamendo dataset against those obtained on Jamendo in related reports under the same conditions. When compared with the classifiers based on LSTM, the proposed LRCN achieved an *F*1-measure of 0.942, which outperforms the BLSTM of Leglaive and LSTM of Lehner-3 by 4%. Furthermore, it shows an improvement over the state-of-the-art method reached by the LRCN of Zhang by 1.5%.

4. Conclusions

In this paper, an SVD system based on LRCN was presented to detect vocals in a new electronic music dataset and evaluated against previous DL models (CNN, LSTM, LRCN) using existing pop music datasets, exhibiting an F1-measure of 0.828 for the electronic music dataset (Electrobyte); however, this was lower compared to its pop music genre counterpart (Jamendo) with an *F*1-measure of 0.942. This provides a benchmark result to further evaluate electronic music in the SVD field with new models and achieve a better performance.

Future work will expand the research on electronic music attributes to find the root causes of the result in terms of vocal signals. Furthermore, new datasets for different genres as popular as pop music (such as rock or hip-hop) will be created to evaluate the LRCN and perform the same experiments. Finally, datasets with combinations of various music genres will be generated and tested to prove the robustness of SVD systems and find patterns or similarities between genres.

Author Contributions: Conceptualization, R.R.-A. and A.G.-E.; methodology, R.R.-A. and A.G.-E.; software, R.R.-A.; validation, R.R.-A.; formal analysis, R.R.-A. and A.G.-E.; investigation, R.R.-A.; resources, R.R.-A.; data curation, R.R.-A.; writing—original draft preparation, R.R.-A.; writing—review and editing, R.R.-A., A.G.-E. and B.V.-A.; visualization, R.R.-A.; supervision, A.G.-E. and B.V.-A.; project administration, R.R.-A., A.G.-E. and B.V.-A.; funding acquisition, B.V.-A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the financial support of Tecnologico de Monterrey in the production of this work.

Data Availability Statement: The Electrobyte dataset is publicly available at (https://zenodo.org/record/6757945) (accessed on 27 June 2022), while the necessary scripts to reproduce these results are available at (https://github.com/Raylogic/SVD-LRCN) (accessed on 9 June 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MIR	Music Information Retrieval
SVD	Singing Voice Detection
SVS	Singing Voice Separation
DL	Deep Learning
MFCC	Mel-Frequency Cepstral Coefficients
LPCC	Linear Prediction Cepstral Coefficients
ZCR	Zero-Crossing Rate
PLP	Perceptual Linear Prediction
SSF	Spectral Statistical Features
LRCN	Long-Term Recurrent Convolutional Network

References

1. Bryan Pardo, Z.R.; Duan, Z. Audio Source Separation in a Musical Context. In *Handbook of Systematic Musicology*; Springer Handbooks; Springer: Berlin/Heidelberg, Germany, 2018; pp. 285–298.
2. Li, Y.; Wang, D. Separation of Singing Voice from Music Accompaniment for Monaural Recordings. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 1475–1487.
3. Rao, V.; Rao, P. Vocal Melody Extraction in the Presence of Pitched Accompaniment in Polyphonic Music. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *18*, 2145–2154.
4. Kan, M.Y.; Wang, Y.; Iskandar, D.; Nwe, T.L.; Shenoy, A. LyricAlly: Automatic Synchronization of Textual Lyrics to Acoustic Music Signals. *IEEE Trans. Audio Speech Lang. Process.* **2008**, *16*, 338–349.
5. Fujihara, H.; Goto, M. Lyrics-to-Audio Alignment and its Application. In *Multimodal Music Processing*; Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik: Dagstuhl, Germany, 2012.
6. Hosoya, T.; Suzuki, M.; Ito, A.; Makino, S. Lyrics Recognition from a Singing Voice Based on Finite State Automaton for Music Information Retrieval. In Proceedings of the 6th International Conference on Music Information Retrieval, London, UK, 11–15 September 2005.
7. Monir, R.; Kostrzewa, D.; Mrozek, D. Singing Voice Detection: A Survey. *Entropy* **2022**, *24*, 114. https://doi.org/10.3390/e24010114.
8. Regnier, L.; Peeters, G. Singing voice detection in music tracks using direct voice vibrato detection. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 1685–1688.
9. Rocamora, M.; Herrera, P. Comparing audio descriptors for singing voice detection in music audio files. In Proceedings of the 11th Brazilian Symposium on Computer Music (SBCM 2007), São Paulo, Brazil, 1–3 September 2007.
10. Vijayan, K.; Li, H.; Toda, T. Speech-to-Singing Voice Conversion: The Challenges and Strategies for Improving Vocal Conversion Processes. *IEEE Signal Process. Mag.* **2019**, *36*, 95–102.
11. Zhang, X.; Yu, Y.; Gao, Y.; Chen, X.; Li, W. Research on Singing Voice Detection Based on a Long-Term Recurrent Convolutional Network with Vocal Separation and Temporal Smoothing. *Electronics* **2020**, *9*, 1458.
12. Schlüter, J.; Grill, T. Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015), Málaga, Spain, 26–30 October 2015.
13. You, S.; Liu, C.H.; Chen, W.K. Comparative study of singing voice detection based on deep neural networks and ensemble learning. *Hum.-Centric Comput. Inf. Sci.* **2018**, *8*, 34.
14. Huang, H.M.; Chen, W.K.; Liu, C.H.; You, S.D. Singing voice detection based on convolutional neural networks. In Proceedings of the 2018 7th International Symposium on Next Generation Electronics (ISNE), Taipei, Taiwan, 7–9 May 2018; pp. 1–4.
15. Hughes, T.; Mierle, K. Recurrent neural networks for voice activity detection. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 7378–7382.
16. Lehner, B.; Widmer, G.; Böck, S. A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks. In Proceedings of the 2015 23rd European Signal Processing Conference (EUSIPCO), Nice, France, 31 August–4 September 2015; pp. 21–25.
17. Leglaive, S.; Hennequin, R.; Badeau, R. Singing voice detection with deep recurrent neural networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 121–125.
18. Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014.
19. Scholz, F.; Vatolkin, I.; Rudolph, G. Singing Voice Detection across Different Music Genres. Semantic Audio. 2017. Available online: https://www.aes.org/e-lib/browse.cfm?elib=18771 (accessed on 25 February 2022).
20. Krause, M.; Müller, M.; Weiß, C. Singing Voice Detection in Opera Recordings: A Case Study on Robustness and Generalization. *Electronics* **2021**, *10*, 1214.
21. Conklin, D.W.W.; Gasser, M.; Oertl, S. Creative Chord Sequence Generation for Electronic Dance Music. *Appl. Sci.* **2018**, *8*, 1704.

22. Schlüter, J.; Lehner, B. Zero-Mean Convolutions for Level-Invariant Singing Voice Detection. In Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France, 23–27 September 2018.
23. Cohen-Hadria, A.; Röbel, A.; Peeters, G. Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), A Coruña, Spain, 2–6 September 2019; pp. 1–5.
24. Jansson, A.; Humphrey, E.J.; Montecchio, N.; Bittner, R.M.; Kumar, A.; Weyde, T. Singing Voice Separation with Deep U-Net Convolutional Networks. In Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 23–27 October 2017.
25. You, S.D.; Wu, Y.C.; Peng, S.H. Comparative Study of Singing Voice Detection Methods. *Multimed. Tools Appl.* **2016**, *75*, 15509–15524.
26. Gupta, H.; Gupta, D. LPC and LPCC method of feature extraction in Speech Recognition System. In Proceedings of the 2016 6th International Conference—Cloud System and Big Data Engineering (Confluence), Noida, India, 14–15 January 2016; pp. 498–502.
27. Ellis, D.P.W.; Poliner, G.E. Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, HI, USA, 15–20 April 2007; Volume 4, pp. IV–1429–IV–1432.
28. McFee, B.; Raffel, C.; Liang, D.; Ellis, D.P.W.; McVicar, M.; Battenberg, E.; Nieto, O. librosa: Audio and Music Signal Analysis in Python. In Proceedings of the 14th Python in Science Conference, Austin, TX, USA, 6–12 July 2015.
29. Hermansky, H.; Morgan, N.; Bayya, A.; Kohn, P. RASTA-PLP speech analysis technique. In Proceedings of the ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, San Francisco, CA, USA, 23–26 March 1992; Volume 1, pp. 121–124.
30. Ramona, M.; Richard, G.; David, B. Vocal detection in music with support vector machines. In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 31 March–4 April 2008; pp. 1885–1888.
31. TheFatRat. The Arcadium. 2016. Available online: https://www.youtube.com/c/TheArcadium (accessed on 20 April 2022).
32. Woodford, B. NCS (No Copytight Sounds)—Free Music for Content Creators. 2011. Available online: https://ncs.io (accessed on 27 April 2022).
33. Lehner, B.; Widmer, G.; Sonnleitner, R. On the reduction of false positives in singing voice detection. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014. https://doi.org/10.1109/icassp.2014.6855054.
34. Lehner, B.; Sonnleitner, R.; Widmer, G. Towards Light-Weight, Real-Time-Capable Singing Voice Detection. In Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR 2013), Curitiba, Brazil, 4–8 November 2013.