electronics

MDPI

Article

Singing Voice Detection in Opera Recordings: A Case Study on Robustness and Generalization

Michael Krause *, Meinard Müller and Christof Weiß

International Audio Laboratories Erlangen, 91058 Erlangen, Germany;
meinard.mueller@audiolabs-erlangen.de (M.M.); christof.weiss@audiolabs-erlangen.de (C.W.)
* Correspondence: michael.krause@audiolabs-erlangen.de

Abstract: Automatically detecting the presence of singing in music audio recordings is a central task within music information retrieval. While modern machine-learning systems produce high-quality results on this task, the reported experiments are usually limited to popular music and the trained systems often overfit to confounding factors. In this paper, we aim to gain a deeper understanding of such machine-learning methods and investigate their robustness in a challenging opera scenario. To this end, we compare two state-of-the-art methods for singing voice detection based on supervised learning: A traditional approach relying on hand-crafted features with a random forest classifier, as well as a deep-learning approach relying on convolutional neural networks. To evaluate these algorithms, we make use of a cross-version dataset comprising 16 recorded performances (versions) of Richard Wagner's four-opera cycle *Der Ring des Nibelungen*. This scenario allows us to systematically investigate generalization to unseen versions, musical works, or both. In particular, we study the trained systems' robustness depending on the acoustic and musical variety, as well as the overall size of the training dataset. Our experiments show that both systems can robustly detect singing voice in opera recordings even when trained on relatively small datasets with little variety.

Keywords: singing voice detection; opera; supervised learning; music processing; music information retrieval

Citation: Krause, M.; Müller, M.; Weiß, C. Singing Voice Detection in Opera Recordings: A Case Study on Robustness and Generalization. *Electronics* **2021**, *10*, 1214. https://doi.org/10.3390/electronics10101214

Academic Editor: Manuel Rosa Zurera

Received: 27 February 2021
Accepted: 17 May 2021
Published: 20 May 2021

1. Introduction

Singing constitutes a central component of many musical traditions. Identifying segments of singing activity—often denoted as *singing voice detection* (SVD)—therefore provides essential information about the content and structure of music recordings and may also serve as a pre-processing step for tasks such as lyrics alignment [1,2] or lyrics transcription [3]. SVD has historically received a lot of attention within the field of music information retrieval (MIR) [4,5]. The majority of SVD systems are designed for and evaluated on popular music [6–10]. However, Scholz et al. [11] showed that data-driven SVD methods usually do not generalize well to other genres not seen during training, implying that the pop-music focus limits the general applicability of SVD systems.

Particular differences exist between popular and classical music, which is due to the distinct singing techniques and instrumentations involved. Within classical music, opera recordings constitute challenging scenarios where singing voices are often embedded in a rich orchestral texture. As a peculiarity of classical music, several recorded performances (*versions*) of a musical work are usually available. Such cross-version scenarios provide great opportunities for testing the robustness of MIR algorithms in different tasks [12,13] and their capability for generalizing across different versions [14,15]. In particular, such cross-version analyses have shown that machine-learning algorithms can overfit to characteristics of certain versions or musical works [15]. In the light of these observations, we want to investigate whether—and to what extent—SVD algorithms suffer from such overfitting effects. While one obtains good evaluation results with state-of-the-art SVD systems, previous investigations have shown that these systems sometimes over-adapt

to confounding factors such as loudness [16] or singing style [11]. Therefore, we cannot generally expect these systems to generalize to unseen musical scenarios. Following these lines, our case study yields insights into the generalization behavior of machine-learning systems, the aspects of training data that are relevant for building robust systems, and the benefits of deep learning against traditional machine-learning approaches, which may be relevant beyond the SVD task. With this, our study may serve as an inspiration for similar research on other data-driven approaches to MIR, audio, and speech processing.

In this paper, we aim at gaining a deeper understanding of two state-of-the-art SVD methods, which represent two commonly used strategies: The traditional strategy based on hand-crafted features [17] and the strategy based on deep learning (DL) [16], respectively (see Section 3 for details). To analyze these systems, we make use of a cross-version scenario comprising the full cycle *Der Ring des Nibelungen* by Richard Wagner (four operas, 11 acts) in 16 different versions, thus leading to a novel dataset spanning more than 200 h of audio in total. In this scenario, different versions vary with regard to singers' timbre and singing style, musical interpretation, and acoustic conditions, whereas different operas vary in singing registers and characters, lyrics, and orchestration. We exploit this scenario in a series of systematic experiments in order to analyze the robustness of the two algorithms depending on the musical and acoustic variety, as well as on the size of the training dataset. Our results indicate that both systems perform comparably well and are capable of generalizing across versions and operas—despite the complexity of the scenario and the variety of the data. This result shows that SVD systems based on traditional techniques may perform on par with DL-based approaches while having practical advantages such as lower computational costs and higher stability against random effects. Moreover, we find a small tendency for both systems to overfit to specific musical material, as well as a tendency for the DL-based system to benefit from large dataset sizes. With these general observations, our experimental results may inform the use of SVD algorithms in other musical scenarios beyond the opera context.

The paper is organized as follows. In Section 2, we discuss related work on singing voice detection in opera settings and SVD algorithms. In Section 3, we describe our re-implementations of the two SVD systems used for our experiments. Section 4 provides an overview of our cross-version dataset. Section 5 contains our experimental results and discusses their implications. Section 6 concludes our paper.

2. Related Work

In this section, we discuss approaches that have been proposed for the task of singing voice detection, as well as related work on SVD for opera recordings.

From a technical perspective, one may distinguish between two general types of SVD approaches. Traditional systems [10,17,18] usually consist of two stages—the extraction of hand-crafted audio features and the supervised training of classifiers such as random forest classifiers (RFC). Often, mel-frequency cepstral coefficients (MFCC) are combined with classifiers, such as support vector machines or decision trees [18,19]. Lehner et al. [20] showed that considering further hand-crafted features together with RFCs can surpass the results obtained with MFCCs in isolation. In particular, they proposed a so-called *fluctogram* representation, which is well-suited for capturing vibrato-like modulations in different frequency bands. They further added features that describe, for each frequency band, the magnitude variance (*spectral flatness*) and concentration (*spectral contraction*), and a feature responding to gradual spectral changes (*vocal variance*). Dittmar et al. [17] adopted this feature set and classifier setup and tested their system within an opera scenario.

More recently, SVD systems relying on deep neural networks (DNNs) have become popular [7,8,21–23], with convolutional neural networks (CNN) being among the most effective types of network [14,16,22]. In particular, Schlüter et al. proposed a state-of-the-art deep-learning system for SVD [16,22]—a CNN architecture inspired by VGGNet [24] consisting of stacked convolutional and max-pooling layers applied to mel-spectrogram excerpts (overlapping excerpts of length 1.64 s). More precisely, their network uses five convo-

lutional layers followed by three fully-connected layers, leading to a total of around 1.6 million parameters. Their approach follows the paradigm of automated feature learning—in contrast to the hand-crafted features described above. Further extensions of this are possible, such as a recently proposed approach by Zhang et al. [25] combining recurrent and convolutional layers.

While most of the mentioned approaches were developed and tested on popular music datasets, there is some previous work focussing on SVD for opera recordings. For example, Dittmar et al. [17] performed SVD experiments within an opera scenario comprising recordings of C. M. von Weber's opera *Der Freischütz*. Using hand-crafted features and RFCs, they showed that bootstrap training [9] helps to overcome genre dependencies. In particular, they report frame-wise F-measures up to 0.95, which still constitutes the state of the art for SVD in opera recordings. They further showed that the existence of different versions can be exploited for improving SVD results by performing late fusion of the individual versions' results. Mimilakis et al. [14] used a cross-version scenario comprising three versions of Richard Wagner's opera *Die Walküre* (first act) for evaluating three SVD models based on deep learning. As one contribution of our paper, we perform experiments using both a traditional and a DL-based system and in both cases outperform the results reported in [14]. Furthermore, we substantially extend the scenario of [14] to the full work cycle *Der Ring des Nibelungen* by adding the other acts and operas of the *Ring* cycle, as well as 13 further versions (see Section 4). We use this extended dataset to perform a series of systematic experiments, analyzing our two systems in depth.

3. Singing Voice Detection Methods

In this paper, we consider two approaches to SVD, one based on traditional machine learning [17] and one based on DL [16]. In our re-implementations of these methods, we aimed to be as faithful to the original publications as possible. A conceptual overview of both methods is given in Figure 1.

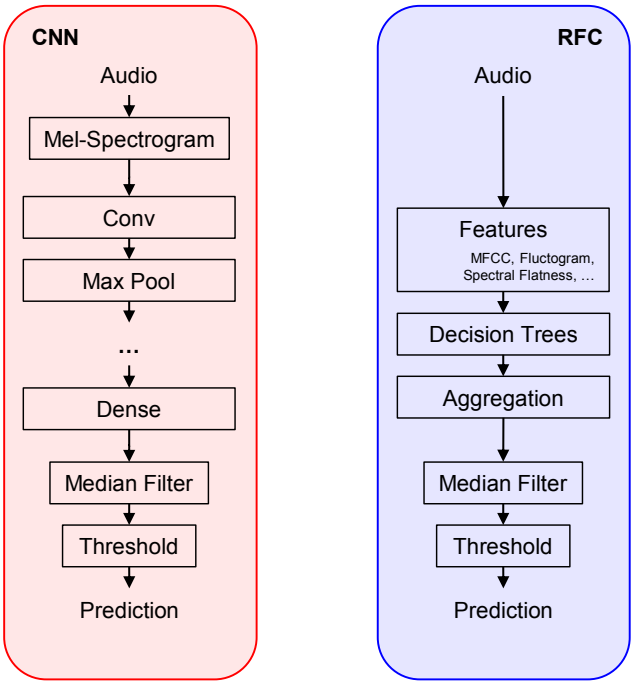

Figure 1. Conceptual overview of the two methods for singing voice detection considered in this paper. While the CNN-based approach operates directly on mel-spectrogram excerpts, the RFC requires an additional feature extraction step. Both approaches output continuous predictions which are post-processed using a median filter and thresholding. The exact architecture of the CNN is omitted for brevity. We refer to the original publications for details.

Closely following [17], we first realize a traditional SVD system relying on hand-crafted features and an RFC classifier. In our re-implementation, we take special care in reproducing the exact feature set, comprising 110 feature dimensions with a feature rate of 5 Hz. Each feature vector incorporates information from 0.8 s of audio (with the exception of the vocal variance feature covering 2.2 s). For the RFC, we use 128 trees per forest as in [17] and use standard settings wherever possible [26,27] (this leads to minor differences in sampling the training data per decision tree and the feature set per decision node compared to [17]). In order to test the validity of our re-implementation, we performed an experiment where we train and test on the public *Jamendo* corpus (https://zenodo.org/record/2585988#.YDNvfmhKhaQ, accessed on 1 February 2021), for which results were also reported in [17]. *Jamendo* is a dataset of over six hours of popular music recordings (published under creative commons licenses), which has been used for experiments on SVD and other tasks [28]. In our experiment, we obtain a frame-wise accuracy of 0.887 and a frame-wise F-measure (with singing as the relevant class) of 0.882. This is close to the accuracy of 0.882 and F-measure of 0.887 as reported in [17] for the same scenario.

For our re-implementation of the DL-based system, we follow the description in [16]. We take special care in reproducing the input representation, the model architecture and the training scheme (since the convolution-activation-batch normalization order is not explicitly stated in [16], we use a potentially different order). To resolve ambiguities in [16], we also consulted a previous publication by the authors [22] and their public source code. As opposed to [16], we do not use any input augmentation in order to ensure comparability with the RFC approach where no such augmentations are used. The results in [16] are reported on an internal dataset. However, for the related method proposed in [22] the authors report an error rate of 9.4% (i.e., an accuracy of 0.906) when training and testing on the *Jamendo* corpus (no data augmentation). Our re-implementation achieves a comparable accuracy of 0.913 for the same scenario.

Both systems output continuous values between 0 and 1 (sigmoid probabilities for the CNN and the fraction of agreeing decision trees for the RFC). Inspired by several SVD approaches [16,17,22], we post-process the results of both the RFC and the CNN using a median filter. As suggested in [17], we use a filter length of 1.4 s and binarize the output with a fixed decision threshold of 0.5. The CNN system outputs predictions at a rate of 70 Hz. To ensure comparability to the RFC-based approach, we, therefore, downsample the CNN predictions to 5 Hz for comparison.

Since neither [17] nor [16] make use of a separate validation set for optimizing hyperparameters, we follow this convention. For the RFC-based system, we try to avoid overfitting by averaging over many trees, each of which is based on a different subsets of the training data. For the CNN-based system, we compute the training loss on mini-epochs of 1000 batches instead of the entire training set. Because of this, we can use early stopping on the training loss to try to prevent overfitting.

Both the traditional and the DL system involve random effects: For the CNN, this includes parameter initialization and random sampling of batches, whereas for the RFC, the choice of features at each split is randomized. Thus, each run of these algorithms produces slightly different results. In our experiments, we compensate for such random effects by averaging all results over multiple runs of the respective algorithm.

The two approaches differ in the computational resources required for training and testing. The computations for individual trees in the RFC can easily be parallelized and run on a standard CPU. The CNN requires a GPU or TPU for efficient training and testing. For example, when training both systems on the same training set of around 200 h of audio (excluding feature computation), the RFC finishes after requiring eight minutes runtime and 3.5 GB of RAM on a desktop computer, while the CNN requires around two hours of training time and 3 GB VRAM on a medium-sized cluster node. These numbers are highly implementation- and hardware-specific, but they demonstrate that the classical system requires less computation time than the deep-learning approach. Inference can be

parallelized for both approaches and takes less than a second for the RFC and about one minute for the CNN on roughly 70 min of test audio.

4. Dataset and Training Scenarios

In this section, we present our cross-version opera dataset, which we use for our systematic experiments in different training–test configurations.

Within Western classical music, Richard Wagner's tetralogy *Der Ring des Nibelungen* WWV 86 constitutes an outstanding work, not least because of its extraordinary length (see Figure 2). Spanning the four operas *Das Rheingold* (WWV 86 A), *Die Walküre* (WWV 86 B), *Siegfried* (WWV 86 C), and *Götterdämmerung* (WWV 86 D), the cycle unfolds an interwoven plot involving many different characters. The characters are represented by different singers with the orchestra adding a rich texture of accompaniment, preludes, and interludes, thus making singing voice detection in recordings of the *Ring* a challenging task. For our experiments, we make use of a cross-version dataset comprising 16 recorded performances—denoted as *versions*—of the *Ring*, each consisting of 13:30 up to 15:30 h of audio data (see Figure 2 and [29] for an overview). All versions are structurally identical, i.e., there are no missing or repeated sections.

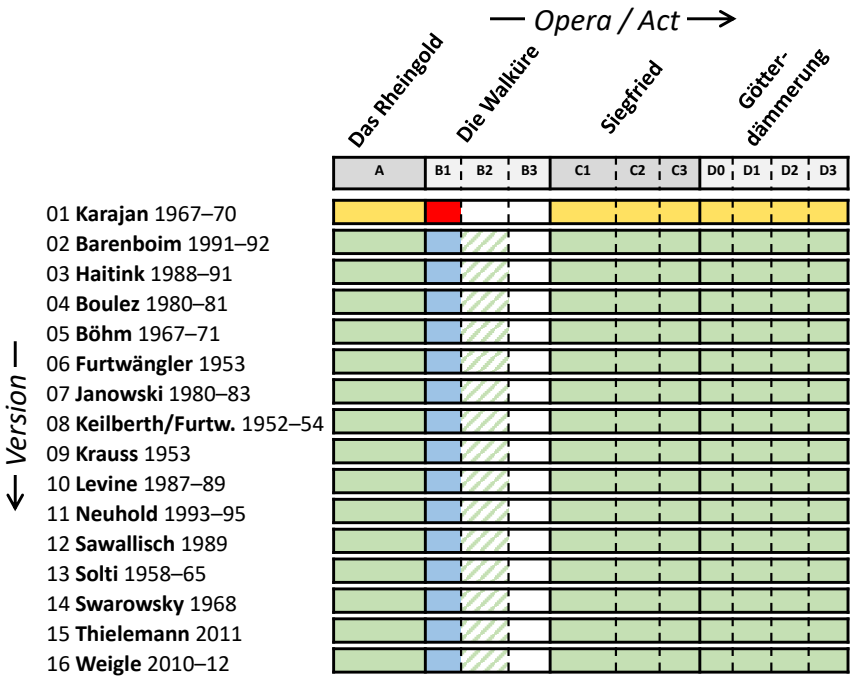

Figure 2. Cross-version dataset comprising 16 versions of Wagner's *Der Ring des Nibelungen* WWV 86. As test data, we use the first act of the second opera *Die Walküre* (WWV 86 B1) in the version conducted by Karajan (red). Training data stems either from the same version but other operas (*opera split*, yellow), from the same act in other versions (*version split*, blue), or from other operas in other versions (*neither split*, green). The hatched green cells (B2) indicate a variant of the *neither split*, where the training data stems from another act of the same opera.

To enable comparability between versions, we produced manual annotations of musical measure positions for versions 01–03 as listed in Figure 2 (see [30] for details). We transferred these measure annotations to versions 04–16 using an automated alignment procedure [31]. We then used the resulting measure positions to generate audio-based singing voice annotations. To this end, we start from the libretto's phrase segments and manually annotate the phrase boundaries as given by the score (in musical measures or beats). To transfer the boundaries to the individual versions, we rely on the measure annotations, refined to the beat level using score-to-audio synchronization [32] within each measure. We use these beat positions to transfer the singing voice segments from the *musi-*

cal time of the libretto to the *physical time* of the performances. The accuracy of the resulting annotations depends, on the one hand, on the accuracy of the measure annotations, which have typical deviations in the order of 100 ms for the manual measure annotations [30] and 200 ms for the transferred measure annotations [31]. On the other hand, score-to-audio synchronization within a measure may introduce further inaccuracies. This is an important consideration for putting any experimental results into context, e.g., for a feature rate of 5 Hz (200 ms) and an average length of a singing voice segment of, say, 4 s, an inaccuracy of one frame already results in a frame-wise error rate of 5%.

For the first act of *Die Walküre* (WWV 86 B1) in version 01 conducted by Karajan (DG 1998), we manually refined the phrase boundaries, thus accounting for both alignment errors and imprecision of singers. We chose this act (B1) since our manual measure annotations are most reliable here, as described in [30]. Moreover, its content is roughly balanced between singing characters, with one female (Sieglinde) and two male singers (Siegmund and Hunding), and with singing activity covering about half its duration (37 of 67 min). In our experiments, we always use this recording and its more accurate annotations for testing (red box in Figure 2).

Inspired by [15], our novel dataset allows us to systematically test the generalization capabilities of our SVD systems in different training–test configurations. To this end, we split our dataset along different axes (Figure 2). In the *opera split*, we train our methods on other operas in the same version and, thus, need to generalize to different musical works (yellow cells in Figure 2). In the *version split*, we use the same act in other versions for training so that the methods need to generalize to a different musical interpretation, different singers, and different acoustic conditions (blue cells). In the *neither split*, neither the test opera nor the test version is seen during training so that the systems have to generalize across both dimensions. In our experiments, we consider different variants of these splits, utilizing, e. g., varying numbers of training versions, operas, or acts. Furthermore, we also exclude in some experiments the second and third act of *Die Walküre* (B2 & B3) since the individual singers (characters) from the first act (B1) re-appear in these acts. When considering all versions or all operas (except *Die Walküre* B1, B2, & B3) for training, we refer to this as a *full split*. Compared to our scenario, Mimilakis et al. [14] used the same test recording (B1 in version 01), but considered only a *version split* with the two versions 02 and 03 (conducted by Barenboim and Haitink, respectively) used for training and validation. We extend this configuration in a systematic fashion in order to study individual aspects of generalization within the opera scenario.

5. Experiments

In the following, we describe our experiments using the systems described in Section 3, taking advantage of the different split possibilities offered by our dataset as described in Section 4. We average all results over five runs in order to balance out effects of randomization during training, as discussed in Section 3. For comparability, we *always* use the first act of *Die Walküre* (B1) in the version by Karajan (01) as our test set as highlighted in Figure 2.

5.1. Training on Different Versions

We begin with a variant of the *version split* as used in [14], which only considers the first act of *Die Walküre*, WWV 86 B1. Here, the training set consists of version 02 (Barenboim) only. On the test set (version 01, Karajan), Mimilakis et al. [14] reported a frame-wise F-measure of 0.80 (we only refer to the results of the zero-mean CNN evaluated in [14], which is most similar to our CNN approach). Using our CNN implementation within the same scenario, we achieve an F-measure of 0.948. The reasons for this substantial improvement remain unclear. With the RFC system, we obtain a comparable result of 0.941, which is similar to the F-measures reported in [17] for the *Freischütz* opera scenario. From this experiment, we conclude that both a traditional and a DL-based system—when

properly implemented and fairly compared—can achieve strong results that are roughly on par with each other.

In the previous experiment, we chose version 02 (Barenboim) as the training version. To investigate the impact of this choice, we repeat the same experiment while changing the training version. Figure 3 shows results for both systems. Dots correspond to mean results averaged over five runs of the same experiment while vertical bars indicate the corresponding standard deviations over those runs. We observe that the choice of training versions has an impact on the test F-measure. The resulting F-measures range from 0.913 for the RFC (version 14) to 0.948 for the CNN (version 02). Furthermore, one can observe that the standard deviations over the runs for individual experiments are higher in the CNN than in the RFC case (see the blue and red vertical bars). In all scenarios, the results are above 0.91 F-measure, which shows that both traditional and deep-learning approaches are capable of generalizing from one version to another version of the same work. Nevertheless, the choice of the training version affects test results, up to around 0.03 F-measure. From a practical point of view, such a difference may seem negligible at first, but when considering a full performance of the *Ring* lasting around 15 h, a difference of 0.03 F-measure can affect roughly 27 min of audio.

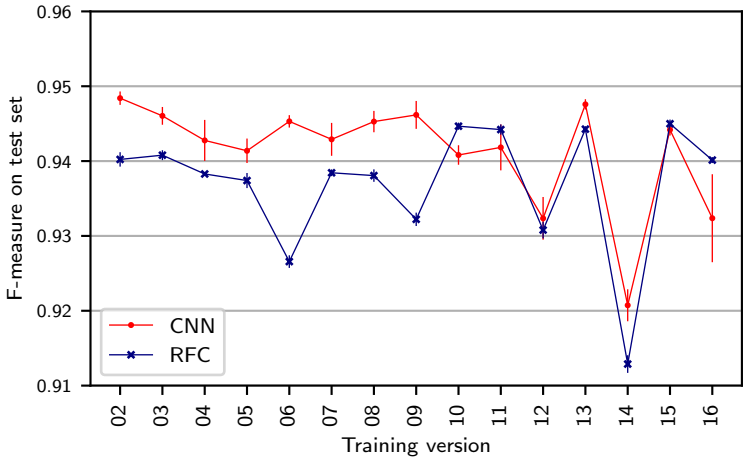

Figure 3. Results for both systems when training on different (individual) versions of the test act (*version split*).

The previous results raise the question whether our systems could benefit from increasing the acoustic and interpretation variety in the training set by training on multiple versions. Figure 4 shows results when systematically increasing the number of training versions of the same act (B1) used in the *version split*. In order to suppress the effect of the particular choice of versions, we repeat each experiment five times and, in each run, randomly sample (without replacement) from all possible versions to create a training set with the specified number of versions. For both classifiers, adding one additional training version leads to improved results. However, adding further training versions does not yield clear improvements. Moreover, these small differences have to be seen in light of the annotation accuracy as discussed in Section 4. Adding one additional training version seems to sufficiently prevent the systems from adapting to the characteristics of individual versions. Additionally, the RFC seems to be more sensitive to the choice of training version: The standard deviation over runs with only one version is larger for the RFC (0.007 percentage points) than for the CNN (0.001), as indicated by the vertical bars around the left-most dots.

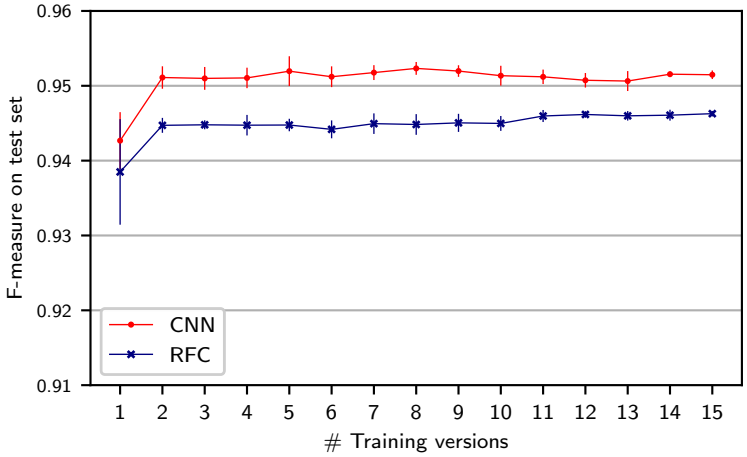

Figure 4. Results for both systems when training on varying numbers of versions of the test act (*version split*).

5.2. Training on Different Musical Material

In the experiments reported so far, we made use of the *version split* where training and test set consist of the same musical material (B1) in different versions. We now want to test generalization to a different musical content and, to this end, use a different act for training (second act of *Die Walküre*, B2) than for testing—a variant of the *neither split* (green hatched cells in Figure 2). As before, we successively increase the number of training versions used. The curves in Figure 5 indicate the results, which are worse in general compared to Figure 4. The RFC system now benefits slightly more from additional training versions, while the CNN seems unaffected by this. Although the CNN yields better results than the RFC, neither system reaches its efficacy on the *version split*. A possible explanation for this may be that both systems might overfit to the musical material in the training act, adapting, e.g., to the instrumentation or the singing characters (high or low register, male or female voice) as appearing in the training data. The small but consistent gap between the curves of the same color in Figures 4 and 5 could be attributed to such work-related overfitting. We understand such small differences to illustrate a trend in the learning behaviors of our methods (though, as mentioned before, these differences may still be of practical relevance when considering an entire performance).

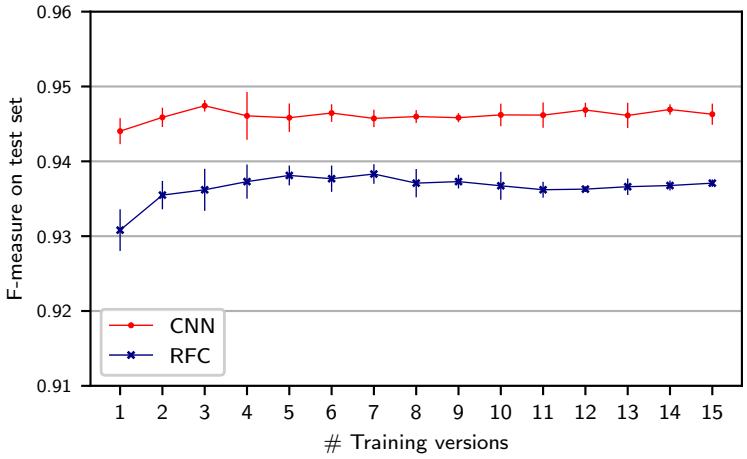

Figure 5. Results for both systems when training on varying numbers of versions of an act (B2) that is different from the test act (*neither split*).

To investigate the impact of such work-related overfitting in more detail, we now examine specific variants of the *opera split* where we use the test version (01, Karajan) for training but take one act from another opera (A, C, or D—excluding B) as the training

act, respectively. Compared to the previous experiment, the musical generalization is now harder (a different opera, rather than different acts from the same opera) but the acoustic generalization is less hard (same version). The results for this experiment (see Figure 6) are generally worse than for training on a different act of the same opera (cf. Figure 5). While the F-measure for the RFC depends only slightly on the particular choice of the training act, this effect is stronger for the CNN. Most prominently, we observe substantial drops when using C1 or C2 (first and second act of *Siegfried*) or D0 (Prologue to *Götterdämmerung*) for training the CNN. This provides insights into specific challenges of generalization: In C1, only male characters (Siegfried, Mime, Wanderer) are singing. In C2, this is similar, except for several short appearances of the character "Waldvogel" (soprano). In D0, in contrast, mainly female singers are singing. All these cases result in a more challenging generalization to B1, where both female and male characters appear. We also observe a drop for the RFC when training on act D2, which does not occur for the CNN. One reason for this difference could be the prominence of the men's choir (Mannen) over large parts of D2, which is mostly absent from the rest of the work cycle. Choir singing could negatively affect, e.g., the fluctogram features used as input to the RFC (which are sensitive to vibrato) but could be accounted for by the automated feature extraction of the CNN. In general, the RFC-based system, which relies on hand-crafted features (capturing vibrato and other singing characteristics), seems to be more robust to different singers and registers in training and test data. The CNN, in contrast, seems to generalize better to other musical content with the same characters and registers (as in Figure 5) and to choir singing.

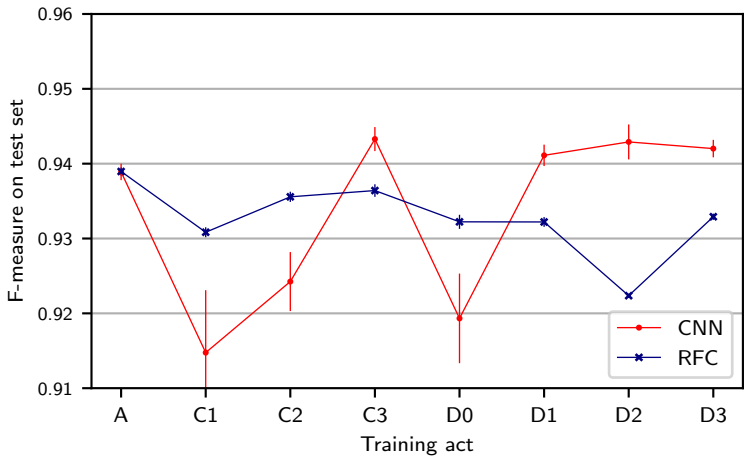

Figure 6. Results for both systems when training on different (individual) acts from the test version (*opera split*).

Furthermore, we can again observe that the standard deviation over CNN runs is higher than for the RFC (see vertical bars). Nevertheless, all results are well above an F-measure of 0.9, meaning that even without training examples for a certain gender of singers, both methods can robustly detect singing in unseen recordings.

5.3. Training on Full Splits

We now extend these experiments to the full splits as shown in Figure 2 (solid cells). Table 1 shows the corresponding results. Let us first discuss the full *opera split*, where we train the systems on all acts from the three other operas (A, C, D) in the test version 01, Karajan (yellow cells in Figure 2). We observe F-measures of 0.948 (CNN) and 0.938 (RFC), respectively. Next, we consider the full *version split*, where we train on all other versions (02–16) for the test act B1 (blue cells in Figure 2). Here, we obtain results of 0.951 for the CNN and 0.946 for the RFC, which are slightly better than for the *opera split*. This confirms our observation that work-related overfitting effects (e.g., to singers' register or gender) help to obtain better results in a *version split* compared to an *opera split*. For the *neither split*, we use all acts of A, C, and D in all other versions (02–16) for training (solid green cells

in Figure 2). In this case, we observe F-measures of 0.952 (CNN) and 0.940 (RFC). Here, interestingly, the CNN performs similar as in the *version split*, though neither test act nor test version are seen during training. In contrast, the RFC yields an F-measure close to its result on the *opera split*. With more versions and operas available, the CNN seems to compensate for the missing test act in the training set. As before, all results are high in general, meaning that both systems work for all considered splits.

Table 1. F-measures on the test set for both systems, using the full variants of the split, respectively.

	Opera Split	*Version Split*	*Neither Split*
CNN	0.948	0.951	0.952
RFC	0.938	0.946	0.940

To better understand the important aspects of the training set—especially in the case that less versions and works are available—we now present an extension of the *neither split* where we successively increase the number of training versions and operas, always using all acts of an opera (Figure 7a). In each of the five experiment runs, we randomly sample among the versions and operas used for the training set. In this visualization, we omit the vertical bars indicating standard deviations for better visibility. For the RFC (blue curves), we observe that the results are almost identical for different numbers of training operas (solid vs. dashed and dotted curves), but slightly improve for higher numbers of training versions. The CNN, in contrast, benefits from using more operas in the case that more training versions are available as well.

Summarizing these results, we find that the CNN has a slightly stronger tendency than the RFC to overfit to the musical material of the training set. We further see that the RFC-based system primarily exploits acoustic variety present in multiple training versions. In comparison, the CNN seems to also exploit variety in musical material and singing characters stemming from different operas. As a consequence, the CNN trained on the full training data of the *neither split* can generalize better to other operas and, thus, better compensate for the missing test act during training. We further find that the results for the CNN system vary more across runs, especially in the case that only few training versions and acts are used. As mentioned above, however, the results for all experiment settings are consistently high, meaning that both methods are feasible for singing voice detection in opera, even if only little training data variety is available.

5.4. Impact of Dataset Size

In our previous experiments, we systematically added training versions and operas, which led to an increase not only of the *variety* of training data but also of the training dataset's *size*. To separately study the two effects, we repeat the experiment from Figure 7a while randomly sub-sampling all training datasets to have equal size (of about as many input patches as obtained from a single act of a single version). Results are shown in Figure 7b. For the RFC (blue), results are almost identical as in Figure 7a. For the CNN, we observe smaller improvements in Figure 7b than in Figure 7a for increasing the number of training versions and operas. Therefore, the improvement seen in Figure 7a appears to stem mainly from the training dataset's size rather than from its variety. This suggests a fundamental difference between the two systems: While the CNN benefits from a larger amount of training data, the RFC is widely unaffected by this. For the RFC, a certain variety in training data seems to be sufficient for reaching an optimal efficacy.

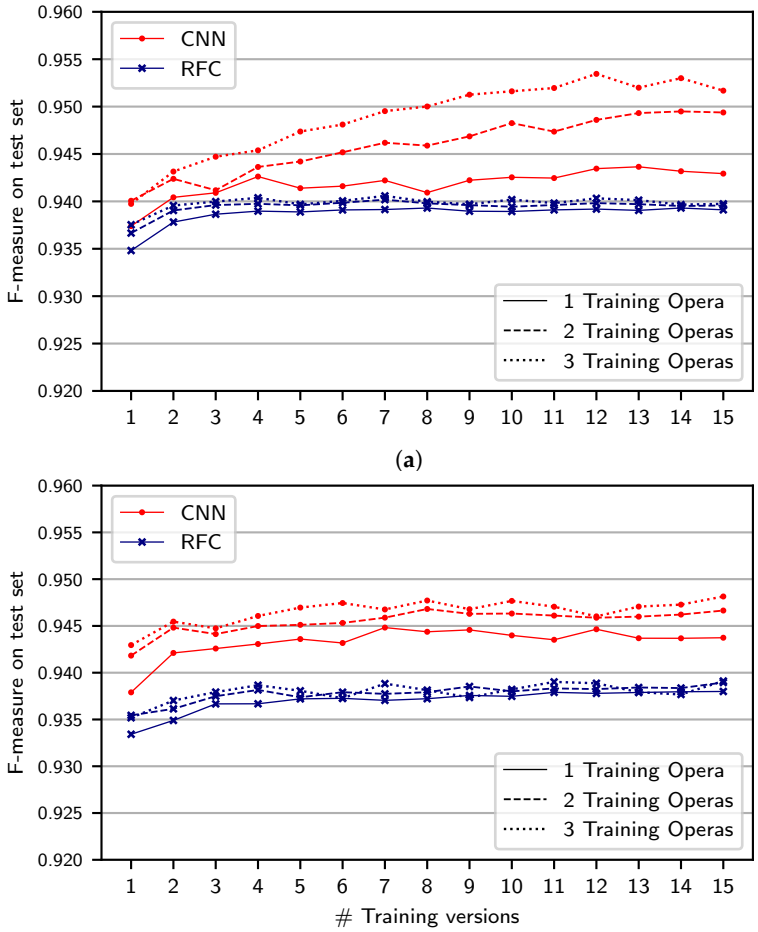

Figure 7. Results for both systems when training on varying numbers of versions and operas (solid, dashed, and dotted curves) that are different from the test version and act (*neither split*). (**a**) Using the full data for training. (**b**) Using sub-sampling of the training datasets to the same size.

5.5. Transfer between Pop and Opera Datasets

We finally compare system results when training on datasets from another genre of music. Specifically, we use the *Jamendo* dataset already discussed in Section 3. We expect generalization between the pop and opera styles to be poor, since SVD methods are known to be heavily genre specific [11] and, in particular, opera singers employ singing techniques that are distinct from those found in Western pop music. Consequently, when training on the *Jamendo* training corpus and evaluating on our own test set, we observe a low F-measure of 0.457 for the RFC (not better than random choice) and a medium result of $F = 0.795$ for the CNN. When using the training set of the full *neither split* of our dataset (see Figure 2) and testing on the *Jamendo* test set instead, we obtain $F = 0.693$ for the RFC and $F = 0.692$ for the CNN. Thus, generalizing from opera to pop works better for the RFC and worse for the CNN, but genre-specific overfitting is evident in both scenarios.

6. Conclusions

Summarizing our experimental findings, we conclude that machine-learning approaches both relying on traditional techniques and on deep learning are useful for building well-performing SVD systems, which are capable of generalizing to unseen musical works, versions, or both at the same time. Both systems achieve F-measures in the order of 0.94 and their results do not drop below 0.91, even when considering training datasets with little musical or acoustic variety. While these are strong results for a challenging SVD scenario, we find a tendency of both systems to overfit to the specific musical material in

the training set. Moreover, we observe that both systems benefit from a certain amount of acoustic variety in the training dataset. Nevertheless, overfitting to musical or acoustic characteristics does not lead to complete degradation of results in our scenario. For practical applications, the traditional approach based on random forest classifiers requires less resources and training time and is more robust to random effects of different training runs. In contrast, the CNN-based method leads to slightly better results in most scenarios, especially when a large training dataset is available.

In a manual analysis, we could trace back most of the remaining errors made by our systems to difficult situations where annotation errors and musical ambiguities play a major role. One source of such ambiguity is the discrepancy between a phrase-based and a note-based consideration of singing voice segments, i.e., the question whether a short musical rest within a singing phrase should be considered as singing. Further ambiguities arise about whether to include breathing as singing, or how to deal with choir passages. In the present study, breathing is mostly excluded from singing since our semi-automatic transfer relies on chroma features. Choir passages are included as singing but only occur in acts D2 and D3. However, as our annotations are based on phrases in the libretto, we could not differentiate, e.g., between silence and singing within sung phrases. Manual annotation could provide more accurate ground truth but is only feasible for smaller datasets. These and other challenges of annotation indicate that both systems are already close to a "glass ceiling" of SVD efficacy, where the definition of the task itself becomes problematic.

Such encouraging results close to the "glass ceiling" cannot generally be expected for other MIR tasks such as, e.g., the recognition of a specific register (soprano, mezzo, alto, tenor, baritone, or bass) or even a specific singer or character (such as Siegfried or Sieglinde). The hand-crafted features used in the RFC-based system have been specifically designed to work well for SVD and may not perform equally well on such more advanced tasks. Thus, deep learning based-approaches may yet outperform classical systems in those contexts. Therefore, it may be promising to extend our studies to such further tasks and to more complex scenarios (including other composers and genres). More generally, the experimental procedures presented in this paper can be applied to various domains of audio and music processing where different data splits are possible. In future work, these procedures may contribute to understanding the benefits of deep learning against traditional machine learning and to identifying the aspects of training data that are relevant for building robust systems.

Author Contributions: M.K. is the primary author and main contributor of this research article. In particular, he implemented the approaches, conducted the experiments, and prepared most parts of this document. All authors substantially contributed to this work, including the development of the ideas, the design of the experiments, the preparation of the dataset, and the writing of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the German Research Foundation (DFG MU 2686/7-2, MU 2686/11-1).

Acknowledgments: The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS. The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE). Moreover, the authors thank Vlora Arifi-Müller, Cäcilia Marxer, and all student assistants involved in the preparation of data and annotations. The authors thank Halil Erdoğan for help with preliminary experiments.

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

References

1. Kruspe, A.M. Application of Automatic Speech Recognition Technologies to Singing. Ph.D. Thesis, Technische Universität Ilmenau, Ilmenau, Germany, 2018.
2. Stoller, D.; Durand, S.; Ewert, S. End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-To-Character Recognition Model. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 181–185.
3. Gupta, C.; Yilmaz, E.; Li, H. Automatic Lyrics Alignment and Transcription in Polyphonic Music: Does Background Music Help? In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 496–500.
4. Humphrey, E.J.; Reddy, S.; Seetharaman, P.; Kumar, A.; Bittner, R.M.; Demetriou, A.; Gulati, S.; Jansson, A.; Jehan, T.; Lehner, B.; et al. An Introduction to Signal Processing for Singing-Voice Analysis: High Notes in the Effort to Automate the Understanding of Vocals in Music. *IEEE Signal Process. Mag.* **2019**, *36*, 82–94. [CrossRef]
5. Berenzweig, A.L.; Ellis, D.P.W. Locating Singing Voice Segments within Music Signals. In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Platz, NY, USA, 24 October 2001; pp. 119–122.
6. Lee, K.; Choi, K.; Nam, J. Revisiting Singing Voice Detection: A Quantitative Review and the Future Outlook. In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 23–27 September 2018; pp. 506–513.
7. Leglaive, S.; Hennequin, R.; Badeau, R. Singing Voice Detection with Deep Recurrent Neural Networks. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 121–125.
8. Lehner, B.; Schlüter, J.; Widmer, G. Online, Loudness-Invariant Vocal Detection in Mixed Music Signals. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1369–1380. [CrossRef]
9. Nwe, T.L.; Wang, Y. Automatic Detection of Vocal Segments in Popular Songs. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Barcelona, Spain, 10–15 October 2004; pp. 138–144.
10. Regnier, L.; Peeters, G. Singing voice detection in music tracks using direct voice vibrato detection. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, 19–24 April 2009; pp. 1685–1688.
11. Scholz, F.; Vatolkin, I.; Rudolph, G. Singing Voice Detection across Different Music Genres. In Proceedings of the AES International Conference on Semantic Audio, Erlangen, Germany, 22–24 June 2017; pp. 140–147.
12. Ewert, S.; Müller, M.; Konz, V.; Müllensiefen, D.; Wiggins, G.A. Towards Cross-Version Harmonic Analysis of Music. *IEEE Trans. Multimed.* **2012**, *14*, 770–782. [CrossRef]
13. Müller, M.; Prätzlich, T.; Driedger, J. A cross-version approach for stabilizing tempo-based novelty detection. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, 8–12 October 2012; pp. 427–432.
14. Mimilakis, S.I.; Weiß, C.; Arifi-Müller, V.; Abeßer, J.; Müller, M. Cross-Version Singing Voice Detection in Opera Recordings: Challenges for Supervised Learning. In *Machine Learning and Knowledge Discovery in Databases, Proceedings of the International Workshops of ECML PKDD 2019, Part II, Würzburg, Germany, 16–20 September* 2019; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2019; Volume 1168, pp. 429–436.
15. Weiß, C.; Schreiber, H.; Müller, M. Local Key Estimation in Music Recordings: A Case Study Across Songs, Versions, and Annotators. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2919–2932. [CrossRef]
16. 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 (ISMIR), Paris, France, 23–27 September 2018; pp. 321–326.
17. Dittmar, C.; Lehner, B.; Prätzlich, T.; Müller, M.; Widmer, G. Cross-Version Singing Voice Detection in Classical Opera Recordings. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Málaga, Spain, 26–30 October 2015; pp. 618–624.
18. Ramona, M.; Richard, G.; David, B. Vocal Detection in Music with Support Vector Machines. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, NV, USA, 31 March–4 April 2008; pp. 1885–1888.
19. Vembu, S.; Baumann, S. Separation of vocals from polyphonic audio recordings. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), London, UK, 11–15 September 2005; pp. 337–344.
20. Lehner, B.; Widmer, G.; Sonnleitner, R. On the reduction of false positives in singing voice detection. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 7480–7484.
21. 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 European Signal Processing Conference (EUSIPCO), Nice, France, 31 August–4 September 2015; pp. 21–25.
22. Schlüter, J.; Grill, T. Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Málaga, Spain, 26–30 October 2015; pp. 121–126.
23. Wang, Y.; Getreuer, P.; Hughes, T.; Lyon, R.F.; Saurous, R.A. Trainable Frontend for Robust and Far-Field Keyword Spotting. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 5670–5674.

24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
25. 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. [CrossRef]
26. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
27. Louppe, G. Understanding Random Forests: From Theory to Practice. Ph.D. Thesis, University of Liege, Liege, Belgium, 2014.
28. Bogdanov, D.; Won, M.; Tovstogan, P.; Porter, A.; Serra, X. The MTG-Jamendo Dataset for Automatic Music Tagging. In Proceedings of the Workshop on Machine Learning for Music Discovery, International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019.
29. Zalkow, F.; Weiß, C.; Müller, M. Exploring Tonal-Dramatic Relationships in Richard Wagner's Ring Cycle. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Suzhou, China, 23–27 October 2017; pp. 642–648.
30. Weiß, C.; Arifi-Müller, V.; Prätzlich, T.; Kleinertz, R.; Müller, M. Analyzing Measure Annotations for Western Classical Music Recordings. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), New York, NY, USA, 7–11 August 2016; pp. 517–523.
31. Zalkow, F.; Weiß, C.; Prätzlich, T.; Arifi-Müller, V.; Müller, M. A Multi-Version Approach for Transferring Measure Annotations Between Music Recordings. In Proceedings of the AES International Conference on Semantic Audio, Erlangen, Germany, 22–24 June 2017; pp. 148–155.
32. Ewert, S.; Müller, M.; Grosche, P. High Resolution Audio Synchronization Using Chroma Onset Features. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, 19–24 April 2009; pp. 1869–1872.