CROSS-VERSION SINGING VOICE DETECTION IN CLASSICAL OPERA RECORDINGS

Christian Dittmar[1] **Bernhard Lehner**[2] **Thomas Prätzlich**[1]
Meinard Müller[1] **Gerhard Widmer**[2]
[1] International Audio Laboratories, Erlangen, Germany
[2] Johannes Kepler University, Linz, Austria
christian.dittmar@audiolabs-erlangen.de, bernhard.lehner@jku.at

ABSTRACT

In the field of Music Information Retrieval (MIR), the automated detection of the singing voice within a given music recording constitutes a challenging and important research problem. In this study, our goal is to find those segments within a classical opera recording, where one or several singers are active. As our main contributions, we first propose a novel audio feature that extends a state-of-the-art feature set that has previously been applied to singing voice detection in popular music recordings. Second, we describe a simple bootstrapping procedure that helps to improve the results in the case that the test data is not reflected well by the training data. Third, we show that a cross-version approach can help to stabilize the results even further.

1 Introduction

In classical opera, singing voice is considered to be one of the most important musical aspects. Locating vocal segments in an opera recording is an important prerequisite for applications such as singing voice separation or music structure analysis. The task of singing voice detection (also known as vocal detection) comprises automatic segmentation of a music recording into vocal (one or more singers) and non-vocal (accompaniment or silence) parts. A typical example of such a temporal segmentation is shown in Figure 1, where the black rectangles below each plot are ground truth segments and the red rectangles show automatically detected segments. The main challenge in automatic vocal detection comes both from the huge variety of singing voice characteristics as well as the simultaneous presence of other pitched musical instruments in the accompaniment. Especially in opera, the singers are often accompanied by instruments playing the same sequence of notes. Since the singers voice should dominate over the accompaniment, expressive techniques such as pronounced vibrato and the so called singer's formant [18] are often used. Moreover, the pitch and dynamic range of professional opera singers goes well beyond singing voices in popular music.

There has been quite some research on the problem of singing voice detection. The majority of previous contributions employ some sort of machine learning approach in combination with the extraction of audio features (see Section 2). When using machine learning, two major aspects need to be considered. First, appropriate audio features have to be designed that are suitable for the singing voice detection task. A delicate trade-off between elaborate, but error-prone extraction steps on the one hand, and undirected low-level features on the other hand has to be made. In this context, we introduce a novel extension to a previously proposed feature set and show that it is appropriate for singing voice detection.

Second, a supervised machine-learning algorithm usually learns from training data. It is well known that the performance of an optimized classifier can drop significantly if the "closed world" of the training data does not match the "open world" of the target data. A typical example is found in speech processing where systems trained with clean speech usually fail under noisy or reverberant conditions. One possibility to approach this challenge is so-called bootstrapping [14, 19]. As a second main contribution, we show how bootstrapping can help to improve singing voice detection by adapting classifiers to the specific recording under analysis. Furthermore, we describe a cross-version fusion approach [8] that can improve the results in case several versions of a music piece are available, which is a realistic assumption for opera and classical music in general.

2 Related Work

Although singing voice detection seems to be a task that is not so hard for human listeners, automatic singing voice detection remains difficult due to expressive characteristics of the singing voice and the diversity of accompaniment music playing simultaneously. These specific challenges have already been brought up in early works on the topic [2]. Given an unknown music recording, automatic singing voice detection is usually performed as a frame-wise estimation of singing voice activity. Even though this poses a binary classification problem with just two classes, the acoustical variance within each class is so large that it is necessary to train the classifier with a wide range of training data.

Bootstrapping, i.e., the idea of using training data taken from the target recording itself, was proposed before as unsupervised [14] and user assisted [19] strategy for improving classification performance. One of the first attempts to separate the singing voice from the accompaniment prior to the feature extraction stage was described in [20]. Postprocessing of the so-called posterior probabilities obtained during classification was described in [12].

A large set of low-level features was used in conjunction with a Support Vector Machine (SVM) classifier in [15]. Furthermore, the authors published singing voice annotations for training, validation and test subsets of the JAMENDO corpus, enabling reproducible comparisons between different methods (see Section 5.2). The same test corpus was used for evaluation in [16], where the feature extraction focused on vibrato and tremolo properties. A study on the effect of accompaniment music in singing vs. rap discrimination was presented in [6]. Very promising results in singing voice detection and related tasks were reported in [13]. However, the proposed signal processing chain was quite elaborate and involved an estimation of the predominant pitch, which can lead to substantial error propagation to all the feature extractors depending on it.

Lehner et al. [10] focused on achieving comparable results using a light-weight approach. In a follow-up work, they improved the achievable precision by introducing novel audio features tailored to the singing voice detection scenario [11]. A recent paper [4] showed that two cross-version post-processing strategies can improve the singing voice detection performance achievable with the light-weight feature set of [10, 11].

So far, the best classification performance on the JAMENDO data set was reported in [9], using a Bidirectional Long Short-Term Memory Recurrent Neural Network as machine learning scheme that inherently takes the temporal context of low-level feature sequences into account. However, it reads as if the authors selected the optimal network architecture according to the best results obtained w.r.t. the test set instead of the validation set. Thus, we think that their results might be overly optimistic.

3 Baseline Singing Voice Detection

Our baseline system for singing voice detection closely follows the approach proposed in [10, 11]. The extraction of descriptive audio features is performed by splitting the audio signals into frames and transforming each frame to the spectral domain. Low-level and mid-level audio features are computed from each resulting spectral frame, forming a feature vector by concatenation. Supervised machine learning is employed to train a classifier for discriminating the feature vector assigned to each frame into the two classes vocal and non-vocal. Note that the vocal class usually comprises singing voice plus accompaniment, which makes the task more intricate.

Feature name and reference	Abbrev.	Dim.
Mel-frequency Cepstral Coefficients [10]	MFCC	30
Vocal Variance [11]	VOCVAR	5
Fluctogram Variance [11]	FLUCT	17
Spectral Contraction Variance [11]	NSD	17
Spectral Flatness Mean [11]	FLAT	17
Polynomial Shape Spectral Contrast [1, 7]	PSSC	24

Table 1. Feature names, abbreviations, and dimensionality of the low-level and mid-level audio features used.

3.1 Feature Extraction and Processing

Table 1 lists the complete set of features that is used in our approach. Since most of our descriptors are wellknown in the MIR literature, we only highlight a few aspects here.

Mel-Frequency Cepstral Coefficients (MFCC) are one of the most common audio features widely used in diverse audio classification tasks. They are designed to capture the spectral envelope of an audio signal using only a few coefficients in the so-called Cepstral domain. As described in [10], we use an optimized parametrization with a different time-frequency resolution and a higher number of coefficients than usual. A strongly related feature is the **Vocal Variance**, which basically captures the variance in the first 5 MFCCs across a number of consecutive frames.

The mid-level features **Fluctogram, Spectral Contraction, and Spectral Flatness** are the most important contributions from [11]. All three are extracted in 17 overlapping frequency bands, where each band covers two octaves and neighboring bands are spaced three semitones apart. The Fluctogram encodes the relative frequency fluctuation of salient tonal components in each band, without the need for an actual estimation of a predominant pitch. Spectral Contraction and Flatness are designed to complement the Fluctogram, encoding whether there are reliable harmonic components with clear sinusoidal peaks or rather a noise-like distribution of the spectrum within the current band boundaries.

Spectral Contrast encodes the relation of peaks to valleys of the spectral magnitude in several sub bands. The band boundaries have been specified for the Octave-Based Spectral Contrast (OBSC) [7] and the Shape-Based Spectral Contrast (SBSC) [1]. In general, both variants can be interpreted as harmonicity or tonality descriptor. We suggest a modification of the already existing methods, both of which were successfully used for music genre classification tasks. In the previous approaches, the spectral magnitude values in each sub band are sorted and the relation between the lowest and highest fraction is encoded via statistical measures. In our modification, we propose to fit a third-order polynomial to the ordered magnitude values and store the three polynomial coefficients together with the offset as descriptors. Therefore, we refer to this feature as **Polynomial Shape Spectral Contrast (PSSC)**. It is

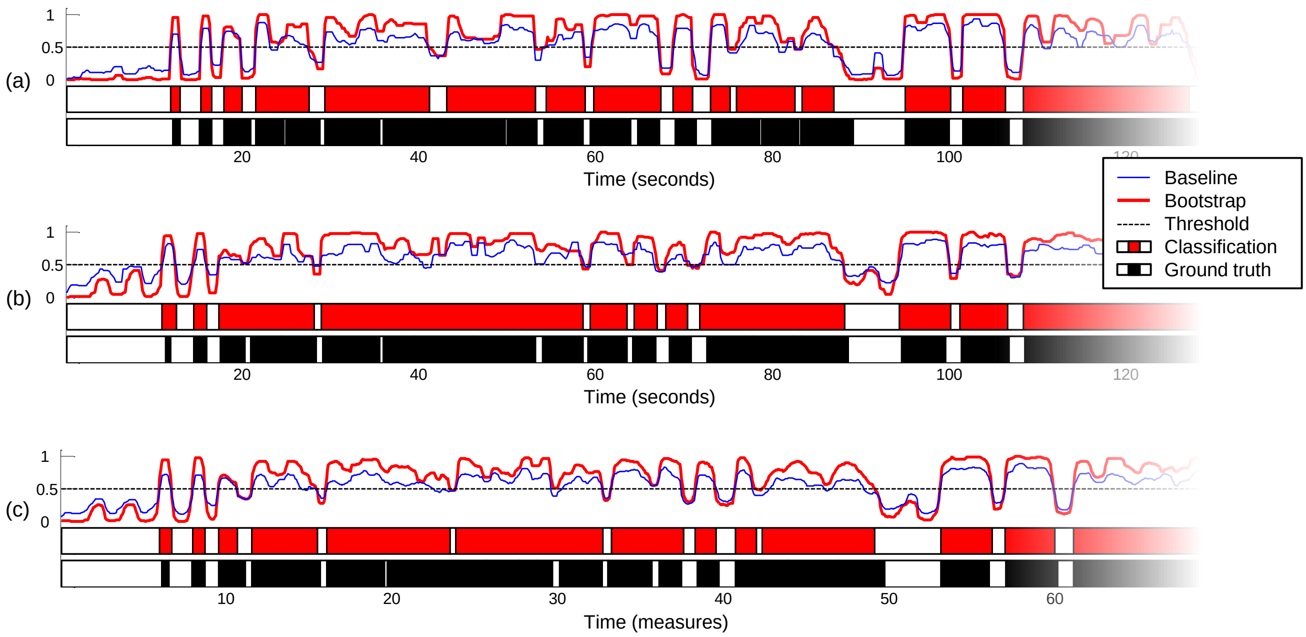

Figure 1. Illustration of the cross-version post-processing strategies as described in Section 4.1 and Section 4.2. The curves and annotations are based on an excerpt corresponding to the first 80 measures of the duet No. 6 (Agathe and Ännchen): "Schelm! halt fest" from the opera "Der Freischütz" by Carl Maria von Weber. For each case, the decision functions of the baseline (blue thin curve) and bootstrap (red bold curve) classifier are shown. The colored time-lines below the decision curves show the automatically detected singing voice activity (red segments, derived from bootstrap decision) vs. the ground truth (black segments). **(a):** Recording of the performance conducted by Karl-Heinz Bloemecke (2013). **(b):** Recording of the performance conducted by Carlos Kleiber (1973). **(c):** Cross-version results based on three performances (including Bloemecke and Kleiber) after temporal alignment to a common, measure-based time axis and subsequent averaging across the individual decision functions.

computed for each of the 6 sub bands (0-200 Hz, 200-400 Hz, 400-800 Hz, 800-1600 Hz, 1600-3200 Hz, and 3200-8000 Hz), yielding a feature vector with 24 attributes. In contrast to the procedure in [1, 7], we do not apply any decorrelation procedure to the raw features, hence reducing the computational complexity. Compared to the before mentioned versions of spectral contrast, our modification resulted in better accuracy on our internal data set (PSSC: 80.2%, OBSC: 73.4%, and SBSC: 72.3%).

In total, the concatenation of all features listed in Table 1 results into a 110-dimensional feature vector per spectral frame. The set of all feature vectors makes up our feature matrix which is split into appropriate training and test sets and used for machine learning in the following.

3.2 Classification and Decision Function

Again following [10,11], we employ Random Forests (RF) [3] as classification scheme. RF are an instance of the so-called Bootstrap Aggregation (Bagging) concept applied to Classification and Decision Trees (CART) [21] classifiers. This machine learning ensemble meta algorithm was designed to improve the stability and accuracy by averaging over a set of weak classifiers trained from random subspaces of the complete feature matrix. In RF, random sets of CARTs are trained by introducing randomness at 2 levels: in the subset of features as well as in the subset of training data [3]. The generalization error of RF depends on the classification strength of the individual CARTs as well as their mutual correlation. As changes in the feature selection cause drastic changes in the tree structure, the individual trees are expected to be uncorrelated. Averaging their individual decisions in the RF leads to decreased variance of the classifier model, which is in general a desirable property.

RFs deliver a frame-wise score value per class that can be interpreted as confidence measure for the classifier decision. In our binary classification scenario, the two score functions are inversely proportional. We pick the one corresponding to our target vocal class and refer to it as decision function in the following. A decision function value close to 1 indicates a very reliable assignment to the vocal class, whereas a value close to 0 points to the non-vocal class. In order to binarize the decision function, we compare it to a threshold. Only frames where the decision function value exceeds the threshold will be classified as vocal. Prior to that, the decision function is smoothed using a median filter. The filter width given in seconds is an important parameter. Median filtering of the decision function is justified by the observation that singing voice activity usually exhibits a certain continuity. So this step helps to stabilize the detection result and to prevent unreasonably short gaps in the decision function, where the classification rapidly flips from vocal to non-vocal or vice versa.

4 Post-processing of Singing Voice Detection

In this section, we describe two approaches suitable for post-processing of intermediate singing voice detection results. First, we describe our approach to unsupervised bootstrap training of a classifier adapted to the recording under analysis. Second, we describe how to perform a late fusion of decision functions by means of time alignment between different versions.

4.1 Bootstrap Training

Inspired by the ideas in [14, 19], we propose to perform a second, specialized RF classification subsequent to the initial singing voice detection stage. The rationale is to remedy the “closed world” vs. “open world” training problem discussed before (see Section 1). We do so by creating an adapted classifier model that is trained with feature vectors exclusively taken from the current recording under analysis. However, this recording does usually not come together with an annotation of its frames to the two classes. So how to assign the feature vectors automatically to the training sets of the vocal respective non-vocal class?

Our idea is to base this assignment on the shape of the decision function generated by the initial RF classifier. Looking at the course of this decision function, we see some extreme values for frames, where the observed feature vectors match very well to either the vocal or non-vocal class of the initial classifier model. However, many values reside in the middle of the range of decision function values, where an assignment to either side is questionable. If we now select two subsets of the feature vectors, each corresponding to an upper and lower fraction (e.g., 20%) of the range of decision function values, we can use these to train a small RF classifier that is adapted to the feature space spanned by the recording under analysis. Before we do so, we stratify the training set, meaning that we randomly select the same number of feature vectors for each class from the subset corresponding to the upper and lower decision values.

In Figure 1, we observe that the new decision functions (red curve) generated by classifying the current song with the adapted RF classifier exhibits a more desirable shape than the decision function generated by the initial RF classifier (blue curve). In Figure 1(a), it can be seen, that the bootstrap decision function can close small gaps, where the initial decision function dipped below the decision threshold (e.g., at around 80 s).

4.2 Cross-Version Fusion

In [8], Konz et al. introduced the intuitive yet effective idea to exploit the availability of different recordings of the same piece of music for stabilizing automatic chord recognition results. We pursue the same idea here in order to perform a late fusion of decision functions obtained from the initial singing voice detection. This is achieved by warping the individual decision functions obtained for different versions of the same piece to a version-independent representation with a musical time axis given in measures (respective sub-divisions thereof) instead of seconds. For the moment, we assume that the required temporal position of measure boundaries is given. In Section 5.3, we sketch how to retrieve the measure boundaries automatically.

Authors and Reference	Accuracy	F-measure
Biased Guess (all frames vocal)	46.3	0.64
Vembu and Baumann 2005 [20]	77.4	0.77
Ramona et al. 2008 [15]	82.2	0.84
Regnier and Peeters 2009 [16]	—	0.77
Lehner et al. 2013 [10]	84.8	0.85
Lehner et al. 2014 [11]	88.2	0.87
Leglaive et al. 2015 [9]	91.5	0.91
Proposed feature set	88.2	0.87

Table 2. Singing voice detection results achievable with our novel feature set in comparison to other authors. The basis of all measurements is a publicly available subset of the JAMENDO corpus [15].

In general, the procedure described above yields a set of time-aligned decision functions that we use to derive a fused, overall decision function. To this end, we use the most straightforward approach and just take the arithmetic mean of the decision values of all aligned decision functions. The averaging is intended to compensate for noise and artifacts that might occur in the individual decision functions. Figure 1(c) presents the resulting decision function on the measure-related time axis. We show the fused decision function derived from baseline singing voice detection (thin blue curve) overlayed with the fused decision function derived from bootstrap training (bold red curve). It can be seen that the averaging leads to a slightly more stable decision function. Comparison of the fused bootstrap decision function against the decision threshold (dashed black line) yields our estimated singing voice segments (black rectangles). In general, the estimated segments exhibit improved agreement to the ground truth segmentation in comparison to Figure 1(a) and 1(b).

5 Evaluation

In this section, we assess the performance of our proposed methods. First, we validate our novel feature set on a public benchmark data set. Second, we show that bootstrapping and cross-version fusion can help to improve the results for classical opera recordings.

5.1 Experimental Settings

For our experiments, we are going to fix the following parameters: For the majority of features in Table 1, the hopsize between consecutive analysis frames is 200 ms (feature rate of 5 Hz), the analysis windows have a length of 800 ms. The raw fluctogram, flatness and contraction features are extracted on a finer temporal level, with a hopsize of 20 ms and a window size of 100 ms. We aggregate 40 consecutive frames of these raw features and use their variance as descriptor for fluctogram and contraction, and their means as descriptor for flatness. In the RF classifier, we use 128 individual CART classifiers, each trained with a randomly selected subset of 5 feature dimensions, from the originally 110-dimensional feature space. For post-processing of the decision functions, we employ a median filter with a width of 1.4 s. The decision function threshold is set to 0.5. In the next sections, we keep these settings fixed for the evaluation of our baseline system as well as our proposed post-processing strategies.

5.2 Performance on a Common Benchmark

In order to benchmark our novel feature set against the state-of-the-art, we used a subset of the publicly available JAMENDO music corpus [15]. Each recording in that data set was manually annotated into vocal and non-vocal sections by the original author. Since human annotators can have difficulties in determining singing segment boundaries, the segmentation allowed some uncertainty, i.e., very short instrumental breaks were not labeled as such. The exact split into training, validation and test set is specified in [15]. Table 2 lists our results in comparison to previously published works. The used metrics are the frame-wise F-measure and the accuracy which are computed by evaluating all frames across the 16 test songs. According to the ground truth annotation, the majority of frames belongs to the non-vocal class. We also report the **Biased Guess**, where all frames of a test item are assigned to the vocal class, because in classical opera, the vocal class usually occurs more often. As can be seen, the performance of our proposed feature set is on par with the state-of-the-art. Only the accuracy and F-measure reported in [9] surpass our results, but the comparison might not be entirely fair as discussed in Section 2.

5.3 Opera Case-Study

The opera “Der Freischütz” by Carl Maria von Weber, a work of high relevance for opera studies, was chosen for the further evaluation. For this opera, there exists a large number of historical sources, including a multitude of audio recordings. In the project “Der Freischütz Digital” [1], musicologists and computer scientists cooperate to explore opportunities for new and digital ways of research, analysis and presentation of music related data in critical editions [17].

From the corpus used in the project, we had three different versions of this opera available for the purpose of cross-version singing voice detection. The respective conductors and recording years are shown in Table 3. All numbers in the three versions have orchestral accompaniment and varying number of soloist singers. We picked the numbers 6, 8, and 9 as test cases of different musical complexity, a duet, a solo aria and a trio, respectively.

Opera	Conductor	Year
“Carmen”	Lorin Maazel	1984
“Die Zauberflöte”	Nikolaus Harnoncourt	1988
“Pelleas et Melisande”	Claudio Abbado	1992
“La Cenerentola”	Riccardo Chailly	1993
“La Traviata”	Carlo Rizzi	2005
“Tristan und Isolde”	Daniel Barenboim	1995
“Der Freischütz”	Karl Elmendorff	1944
“Der Freischütz”	Carlos Kleiber	1973
“Der Freischütz”	Karl-Heinz Bloemecke	2013

Table 3. Overview over the used opera recordings. The upper half specifies the operas available as training set, the lower half gives the operas used as test set.

For evaluation purposes, we first had to generate reference annotations of the singing voice activity in these pieces. This was achieved semi-automatically by means of aligning a MIDI version of each piece to the recording and taking the note onsets and offsets of the singing voice as reference. Details about this procedure can be found in [5].

Furthermore, each recording had its measures (i.e., the beginning of each bar) manually annotated to facilitate the alignment between corresponding versions of the same number. The manually annotated bar positions are used to warp the individual decision functions to a common time axis regardless of their original tempo and variations thereof.

5.4 Results and Discussion

The diagrams in Figure 2 illustrate the benefit of applying bootstrap training (see Section 4.1), cross-version fusion (see Section 4.2), as well as a combination of both in two different training scenarios. The bar plots in both (a) and (b) show the F-measures obtained per test item as well as the average F-measure value. The following singing voice detection and post-processing strategies were tested. **Random Guess** refers to randomly assigning the frames of our test data to either the vocal or non-vocal class with equal probability. Since the vocal class occurs more frequently in our test data, the resulting F-Measure is slightly above chance. **Biased Guess** refers to assigning the singing voice class to each frame of a test recording. It can be seen that the resulting F-measure is already quite high, again a consequence of the dominance of the vocal class in our test set. **Baseline Detection** refers to the results obtained by the baseline singing voice detection system as described in Section 3. **Bootstrap Detection** refers to the results obtained by a second classification run with an adapted RF classifier using the bootstrapping strategy as described in Section 4.1. **Cross-version Fusion** refers to the results of

[1] www.freischuetz-digital.de

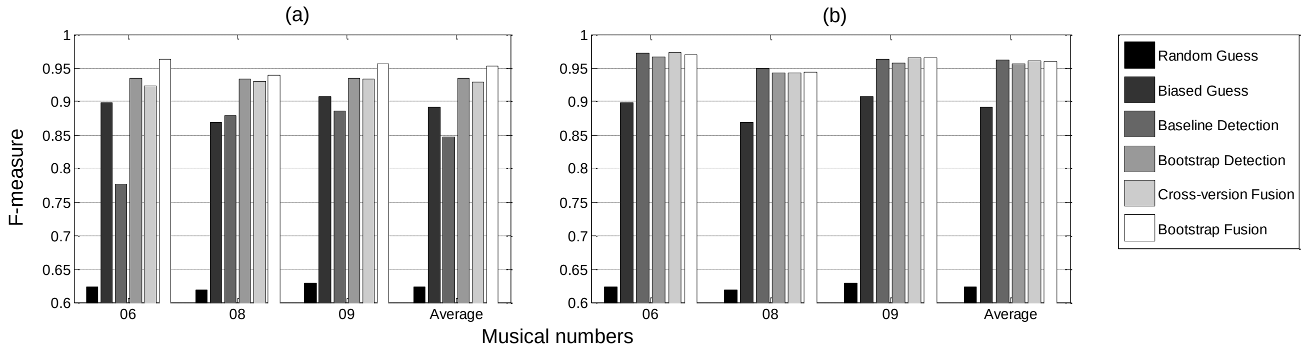

Figure 2. The average F-measures obtained in two different training scenarios and four post-processing strategies. The test set consisted of three versions of the numbers 6, 8, and 9 from the opera "Der Freischütz." (a): Results obtained by training the initial RF classifier with popular music recordings from the RWC and JAMENDO data sets. (b): Results obtained by training the initial RF classifier with classical opera recordings not including "Der Freischütz."

fusing the initial RF decision functions of all available versions of each test recording as described in Section 4.2. Finally, **Bootstrap Fusion** refers to the results obtained by combining both the bootstrap training and the cross-version fusion.

The results in Figure 2(a) were obtained by training the initial RF classifier with a combined data set comprising both the JAMENDO [15] and RWC [13] subsets that are annotated for singing voice. Both corpora are dominated by recordings of popular music. Obviously, this kind of training material differs from the music content in the test set. The average singing voice detection performance stays even below the biased guess. However, this rather poor initial estimate for the vocal frames can be used for bootstrap training. Consequently, the bootstrap training leads to a substantial performance gain, surpassing the bias results.

Cross-version fusion of the imperfect initial decision functions leads to similar improvements as the bootstrap training. The combination of both bootstrap training and cross-version fusion of decision functions delivers the best results in this training scenario.

The results in Figure 2(b) were obtained when training the initial RF classifier with recordings of classical opera. Specifically, we used the operas listed in the upper half of Table 3. In total, the playtime of our training material amounts to approximately 4 h. As can be seen from the F-measure of the baseline RF classifier, this kind of training data gives a considerable performance boost. This is not surprising, since the orchestral timbre as well as the pronounced use of vibrato singing in these opera recordings is very similar to our test items. The remaining F-measures show that the proposed post-processing strategies at best lead to marginal improvements since the performance is already saturated.

From our comparison, we infer that bootstrap training could be recommended as standard post-processing strategy for singing voice detection in classical opera recordings. This is especially true if the initial classification delivers reasonable results that can be surpassed if more appropriate training data would be available. However, bootstrap training does not seem to help much if there exists no combination of feature set, training set, and classifier that can obtain good singing voice detection for the recording under analysis. Moreover, bootstrap training has the drawback that it will likely produce erroneous decision functions when there is no singing voice activity at all throughout a recording. If these cases can not be ruled out from bootstrap training, singing voice detection results could even deteriorate in comparison to the baseline system.

6 Conclusions and Future Work

In this paper, we made two contributions to advancing the state-of-the-art in automatic singing voice detection. First, we proposed a novel extension to a state-of-the-art audio feature set for singing voice detection and validated it on a public benchmark set. Second, we proposed bootstrap training and cross-version fusion as post-processing strategies applicable to intermediate results from a machine learning system. In our case study, involving multiple recordings of Carl Maria von Webers opera "Der Freischütz," we have shown that a combination of bootstrap training and cross-version fusion can help to improve the classification performance if the training data is very different from the test data. While bootstrap fusion might be applicable to improve singing voice detection in various music genres, cross-version fusion can only help if we have multiple, sufficiently similar versions of the same piece of music available. Future work will be directed towards further refinements and applications of these techniques for various kinds of music genres.

7 Acknowledgments

This work has been supported by the BMBF project Freischütz Digital (Funding Code 01UG1239A to C), and by the Austrian Science Fund (FWF) under grants TRP307-N23 and Z159. The International Audio Laboratories Erlangen (AudioLabs) is a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS.

8 References

[1] Vincent Akkermans and Joan Serrá. Shape-based spectral contrast descriptor. In *Proc. of the Sound and Music Computing Conf. (SMC)*, pages 143–148, Porto, Portugal, July 2009.

[2] Adam L. Berenzweig and Daniel P. W. Ellis. Locating singing voice segments within music signals. In *Proc. of the IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 119–122, New Paltz, New York, USA, October 2001.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Christian Dittmar, Thomas Prätzlich, and Meinard Müller. Towards cross-version singing voice detection. In *Proc. of the Jahrestagung für Akustik (DAGA)*, Nuremberg, Germany, March 2015.

[5] Sebastian Ewert, Meinard Müller, and Peter Grosche. High resolution audio synchronization using chroma onset features. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1869–1872, Taipei, Taiwan, April 2009.

[6] Daniel Gärtner and Christian Dittmar. Vocal characteristics classification of audio segments: An investigation of the influence of accompaniment music on low-level features. In *Proc. of the Int. Conf. on Machine Learning and Applications (ICMLA)*, pages 583–589, Miami, Florida, USA, December 2009.

[7] Daning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, volume 1, pages 113–116, Lausanne, Switzerland, August 2002.

[8] Verena Konz, Meinard Müller, and Rainer Kleinertz. A cross-version chord labelling approach for exploring harmonic structures—a case study on Beethoven's Appassionata. *Journal of New Music Research*, 42(1):1–17, January 2013.

[9] Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, Brisbane, Australia, April 2015.

[10] Bernhard Lehner, Reinhard Sonnleitner, and Gerhard Widmer. Towards lightweight, real-time-capable singing voice detection. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 53–58, Curitiba, Brazil, November 2013.

[11] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7480–7484, Florence, Italy, May 2014.

[12] Hanna Lukashevich and Christian Dittmar. Effective singing voice detection in popular music using arma filtering. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pages 165–168, Bordeaux, France, September 2007.

[13] Matthias Mauch, Hiromasa Fujihara, Kazuyoshii Yoshii, and Masataka Goto. Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 233–238, Miami, Florida, USA, October 2011.

[14] Tin Lay Nwe and Ye Wang. Automatic detection of vocal segments in popular songs. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 138–144, Barcelona, Spain, October 2004.

[15] Mathieu Ramona, Gäel Richard, and Bertrand David. Vocal detection in music with support vector machines. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1885–1888, Las Vegas, Nevada, USA, March 2008.

[16] Lise Regnier and Geoffroy Peeters. Singing voice detection in music tracks using direct voice vibrato detection. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1685–1688, Taipei, Taiwan, April 2009.

[17] Daniel Röwenstrunk, Thomas Prätzlich, Thomas Betzwieser, Meinard Müller, Gerd Szwillus, and Joachim Veit. Das Gesamtkunstwerk Oper aus Datensicht - Aspekte des Umgangs mit einer heterogenen Datenlage im BMBF-Projekt "Freischütz Digital". *Datenbank-Spektrum*, 15(1):65–72, 2015.

[18] Zheng Tang and Dawn A. A. Black. Melody extraction from polyphonic audio of western opera: A method based on detection of the singer's formant. In *Proc. of the Int. Society for Music Information Retrieval Conf. (ISMIR)*, pages 161–166, Taipei, Taiwan, October 2014.

[19] George Tzanetakis. Song-specific bootstrapping of singing voice structure. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, volume 3, pages 2027–2030, Taipei, Taiwan, June 2004.

[20] Shankar Vembu and Stefan Baumann. Separation of vocals from polyphonic audio recordings. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 337–344, London, UK, September 2005.

[21] Kristopher West and Stephen Cox. Features and classifiers for the automatic classification of musical audio signals. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.