REVISITING SINGING VOICE DETECTION: A QUANTITATIVE REVIEW AND THE FUTURE OUTLOOK

Kyungyun Lee[1] **Keunwoo Choi**[2] **Juhan Nam**[3]
[1] School of Computing, KAIST
[2] Spotify Inc., USA
[3] Graduate School of Culture Technology, KAIST
kyungyun.lee@kaist.ac.kr, keunwooc@spotify.com, juhannam@kaist.ac.kr

ABSTRACT

Since the vocal component plays a crucial role in popular music, singing voice detection has been an active research topic in music information retrieval. Although several proposed algorithms have shown high performances, we argue that there is still room for improving the singing voice detection system. In order to identify the area of improvement, we first perform an error analysis on three recent singing voice detection systems. Based on the analysis, we design novel methods to test the systems on multiple sets of internally curated and generated data to further examine the pitfalls, which are not clearly revealed with the currently available datasets. From the experiment results, we also propose several directions towards building a more robust singing voice detector.

1. INTRODUCTION

Singing voice detection (or VD, vocal detection) is a music information retrieval (MIR) task to identify vocal segments in a song. The length of each segment is typically at a frame level, for example, 100 ms. Since singing voice is one of the key components in popular music, VD can be applied to music discovery and recommendation as well as various MIR tasks such as melody extraction [7], audio-lyrics alignment [31], and artist recognition [2].

Existing VD methods can be categorized into three different classes. *First*, the early approaches focused on the acoustic similarity between singing voice and speech, utilizing cepstral coefficients [1] and linear predictive coding [10]. The *second* class would be the majority of existing methods, where the systems take advantages of machine learning classifiers such as support vector machines or hidden Markov models, combined with large sets of audio descriptors (e.g., spectral flatness) as well as dedicated new features such as the Fluctogram [14]. *Lastly*, there is a recent trend towards feature learning using deep neural networks, with which the VD systems learn optimized features for the task using a convolutional neural network (CNN) [27] and a recurrent neural network (RNN) [11]. They have achieved state-of-the-art performances on commonly used datasets with over 90% of the true positive rate (recall) and accuracy.

We hypothesize that there are common problems in existing VD methods in spite of such well-performing metrics that have been reported. Our scope primarily includes methods in the second and third classes since they significantly outperform those in the first class. Our hypothesis was inspired by inspecting the assumptions in the existing algorithms. The most common one, for example, has been made on the spectro-temporal characteristics of singing voices; that they include frequency modulation (or vibrato) [15, 24], which leads to our analysis on whether there are any problems by pursuing to be a vibrato detector. We can also raise similar questions on the behavior of the systems in the third class, the deep learning-based systems, by examining on their assumptions and results. Based on the analysis, we invent a set of empirical analysis methods and use them to reveal the exact types of problems in the current VD systems.

Our contributions are as follows :

- A quantitative analysis to clarify and classify common errors of three recent VD systems (Section 4)
- An analysis using curated and generated audio contents that exploit the discovered weakness of the systems (Section 5)
- Suggestions on future research directions (Section 6)

In addition, we review previous VD systems in Section 3 and summarize the paper in Section 7.

2. BACKGROUND

2.1 Problem definition

Singing voice detection is usually defined as a binary classification task about whether a short audio segment input includes singing voice. However, the details have been rather empirically decided. By 'short', the segment length for prediction is often 100 ms or 200 ms. 'Audio' can be provided as stereo, although they are frequently downmixed to mono. More importantly, singing voice is not clearly defined, for example, leaving the question that

	Size	Annotations	Past VD papers	Notes
Jamendo Corpus	93 tracks (443 mins)	Vocal activation	[11], [24], [12], [13], [27], [26]	Train/valid/test split from [22]
RWC Popular Music	100 tracks (407 mins)	Vocal activation, instrument annotation	[26], [27], [14] [13]	VD annotation by [16]
MIR-1K	100 short clips (113 mins)	Vocal activation, pitch contours	[9]	Regular speech files provided
MedleyDB	122 tracks (437 mins)	Melody annotation, pitch annotation	[26]	Multitrack

Table 1: A summary of public datasets relevant to singing voice detection

background vocals should be regarded as singing voice or not. In previous works, this problem has been neglected since the majority of songs in datasets do not include background vocals that are independent of the main vocals. These will be further discussed in Section 6.

2.2 Public Datasets

In Table 1, four public datasets for evaluating VD systems are summarized. Three of them are well described by Lehner et al. [12]: Jamendo Corpus [22], RWC Popular Music Database [4] and MIR-1K Corpus [8]. In addition, we add MedleyDB [3], which is a multitrack dataset, composed of raw mono recordings for each instrument as well as processed stereo mix tracks. Although it does not provide annotations for vocal/non-vocal segments, we utilize the annotations for the instrument activation, which considers vocals as one of the instruments. There can be more benefits by using the multitrack dataset for VD research, which will be discussed in Section 6.

2.3 Audio Representation

In this section, we present the properties as well as the underlying assumptions of various audio representations in the context of VD. Previous works have used a combination of numerous audio features, seeking easier ways for the algorithm to detect the singing voice. They range from audio representations such as short-time Fourier transform (STFT) to high-level features such as onsets and pitch estimations.

- **STFT** provides a 2-dimensional representation of audio, decomposing the frequency components. STFT is probably the most basic (or 'raw') representation in VD, based on which some other representations are either designed and computed, or learned using deep learning methods.
- **Mel-spectrogram** is a mel-scaled frequency representation and usually more compressive than STFTs and originally inspired by the human perception of speech. Being closely related to speech provides a good motivation to be used in VD, therefore mel-spectrogram has been actively used as an input representation of CNNs [27] and RNNs [11]. When deep learning methods are used, mel-spectrogram is often preferred due to its efficiency compared to STFT.
- **Spectral Features** such as spectral centroid and spectral roll-off are statistics of a spectral distribution of a single frame of time-frequency representations (e.g., STFT). A particular and most noteworthy example is **Mel-Frequency Cepstral Coefficients** (MFCCs). MFCCs have originally been designed for automatic speech recognition and take advantages of mel-scale and Fourier analysis for providing approximately pitch-invariant timbre-related information. They are often (assumed to be) relevant to MIR tasks including VD [12, 25]. Spectral features, in general, are not robust to additive noise, which means that they would be heavily affected by the instrumental part of the music when used for VD.

3. MODELS

In this section, we introduce three recent and distinctive VD systems that have improved the state-of-the-art performances along with the details of our re-implementation of them. [1] They are briefly illustrated in Figure 1, where x and y indicate the input audio signal and the output prediction, respectively.

3.1 Lehner et al. [14] (`FE-VD`)

This feature engineering (FE) method, `FE-VD` is based on the Fluctogram, spectral flatness, vocal variance and other hand-engineered audio features. We select this model for its rich and task-specific feature extraction process to compare with the other models. Although the features are ultimately computed frame-wise, context from the adjacent frames are taken into account, supposedly enabling the system to use dynamic aspect of the features. The features are aimed to reduce the false positive rate caused by the confusion between singing voice and pitch-varying instruments such as woodwinds and strings. Random forest classifier was adopted as a classifier, achieving an accuracy of 88.2% on the Jamendo dataset. While their methods have shown reduction in the false positive rates on strings, Lehner et al. mentions woodwinds such as pan flutes and saxophones still show high error rate.

Following [14], we extract 6 different audio features (the Fluctogram, spectral flatness, spectral contraction, vocal variances, MFCCs and delta MFCCs), resulting in 116-dimensional features per frame. We use input size of

[1] http://github.com/kyungyunlee/ismir2018-revisiting-svd

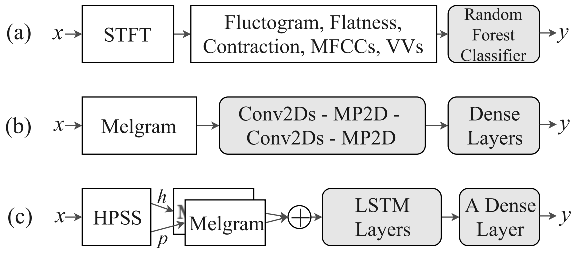

Figure 1: Block diagrams for three VD systems – (a) FE-VD [14], (b) CNN-VD [27], and (c) RNN-VD [11]. *x* and *y* for input audio signal and output prediction (probability of singing voice). Rounded and gray blocks are trainable classifiers or layers. The details of the features in (a) are explained in [14]. In (c), '+' indicates frequency-axis concatenation and 'h' and 'p' are the separated harmonic/percussive components.

1.1 seconds as the input to the random forest classifier, where we perform grid search to find optimal parameters. As a post-processing step, we apply the median filter of 800 ms on the output predictions.

3.2 Schlüter et al. [27] (CNN-VD)

Recently, VD systems using deep learning models have shown the state-of-the-art results [11, 26, 27]. These systems often use basic audio representations such as STFT as an input to the models such as CNN and RNN, expecting the relevant features are learned by the model. We first introduce a CNN-based system [27].

Schlüter et al. suggested a deep CNN architecture with 4 3-by-3 2D convolution layers. We name the CNN model CNN-VD. As a result, the system extracts ~~trained,~~ relevant *local* time-frequency patterns from its input, a mel-spectrogram. During training, they apply data augmentation such as pitch shifting and time stretching on the audio representation. They reported that it reduces the error rate from 9.4% to 7.7% on the Jamendo dataset.

Our CNN architecture is identical to the original one and uses an input size of 115 frames (1.6 sec). However, we do not perform data augmentation or threshold optimization for a fair comparison with other models. Thus, we use 0.5 as the threshold value for the prediction. Here, we also apply median filter of 800 ms for smoothing.

3.3 Leglaive et al. [11] (RNN-VD)

As another deep learning-based system, Leglaive et al. [11] proposed a recurrent neural network with bi-directional long short-term memory units (Bi-LSTMs) [6], with an assumption that temporal information of music can provide valuable information for detecting vocal segments. We name this system RNN-VD. For the classifier input, the system performs double-stage harmonic-percussion source separation (HPSS) [20] on the audio signal to extract signals relevant to the singing voice. For each frame, Mel-spectrograms of the obtained harmonic and percussive components are concatenated as an input for the classifier. Several recurrent layers followed by a shared densely-connected layer (also known as time-distributed dense layer) yield the output predictions for each input frame. This model achieves the state-of-the-art result without data augmentation, showing accuracy of 91.5% on the Jamendo dataset. From this result, although the contributions from additional preprocessing vs. recurrent layers may be combined, we can assume that past and future temporal context help to identify vocal segments.

	FE-VD	CNN-VD	RNN-VD
Acc.(%)	87.9	86.8	87.5
Recall(%)	91.7	89.1	87.2
Precision(%)	83.8	83.7	86.1
F-measure(%)	87.6	86.3	86.6
FPR(%)	15.3	15.1	12.2
FNR(%)	8.3	10.9	12.8

Table 2: Results of our implementations on the Jamendo test set. FPR and FNR refer to false positive rate and false negative rate, respectively.

For our RNN architecture, we use the best performing model from the original article [11], one with three hidden layers of size 30, 20 and 40. The input to the model is 218 frames (3.5 seconds) and the threshold value of 0.5 is used to predict the presence of singing voice as done in [11].

4. EXPERIMENT I: ERROR CATEGORIZATION

The purpose of this experiment is to identify common errors in the VD systems through our implementation of models from Section 3. The results and observations lead to the motivation of experiments in Section 5. Librosa [18] is used in audio processing and feature extraction stages.

4.1 Data and Methods

Three systems (FE-VD, CNN-VD, and RNN-VD) are trained on the Jamendo dataset with the suggested split of 61, 16 and 16 for training, validation and test sets [22], respectively. They are primarily tested on the Jamendo test set. For qualitative analysis, we also utilize MedleyDB.

4.2 Results

The test results of our implementation are shown in Table 2. We did not focus on fine-tuning individual models because three systems altogether are used as a tool to get a generalized view of the recent VD systems, thus showing slightly lower performances compared to the results in original papers. Overall, FE-VD, CNN-VD and RNN-VD show a negligible difference on the test scores. We observe trends that are similar to the original papers in terms of performance and the precision/recall ratio.

Upon listening to the misclassified segments, we categorize the source of errors into three classes – pitch-fluctuating instruments, low signal-to-noise ratio of the singing voice, and non-melodic sounds.

Song Title	Confusing inst	FE-VD	CNN-VD	RNN-VD
LIrlandaise	Woodwind, Synth	46.6	29.5	22.0
Castaway	Elec. Guitar	62.5	56.5	24.2
Say me Good Bye	N/A	2.8	3.0	2.5
Inside	N/A	5.9	6.7	5.0

Table 3: False positive rate (%) of each system for 4 songs from the Jamendo test set. The top 2 songs are the ones ranked within the top 5 lowest accuracy and the bottom 2 songs are the ones ranked within the top 5 highest accuracies at song level across all three systems.

4.2.1 *Pitch-fluctuating instruments*

Classes of instruments such as strings, woodwinds and brass exhibit similar characteristics as the singing voice, which we refer to as being ‘voice-like’ [28]. By ‘voice-like’, we consider three aspects of the signal, namely, pitch range, harmonic structure, and temporal dynamics (vibrato). Especially, we find temporal dynamics as important attributes that are recognized by the VD systems to identify vocal segments.

Frequency modulation, also known as vibrato, resembles the modulation created from the vowel component of singing voice. This is illustrated in Figure 2, where mel-spectrograms of both female vocalist and an electric guitar show curved lines. We observe that this similarity causes further confusion in the system.

In Table 3, we list two songs found among the top 5 least/most accurately predicted songs in the test set of all three systems. The woodwind in ‘05 - Llrlandaise’ causes high false positives, which may be due to the presence of vibrato and the similarity in pitch range to that of soprano singers (above 220 Hz). FE-VD and CNN-VD show poor performance on woodwinds, probably because the Fluctogram of FE-VD and small 2D convolution kernels of CNN-VD are specifically designed to detect vibrato as one of the features for identifying singing voice. In the same song, all three systems show confusion with the synthesizer. Synthesizers mimicking pitch-fluctuating instruments are particularly challenging as it is difficult to characterize them as a specific instrument type.

In addition, electric guitars are one of the most frequently found sources of false positives, as can be seen from ‘03 - castaway’, mostly caused by the recognizable vibrato patterns. We find the confusion worse when the guitar is played with effects such as wah-wah, which imitates the vowel sound of the human. Lastly, we note that some of the other problematic instruments in our test sets include saxophones, trombones and cellos, which are well-known ‘voice-like’ instruments.

This observation, regarding the system pitfalls on vibrato patterns, is further investigated in Section 5.1.

4.2.2 *Signal-to-noise ratio and the performance*

Lastly, we note that all the three systems are affected by the signal-to-noise ratio (SNR), or the relative gain of vocal component, as one can easily expect. All of the three systems exhibit high false negative rate when the vocal signal is relatively at a low level.

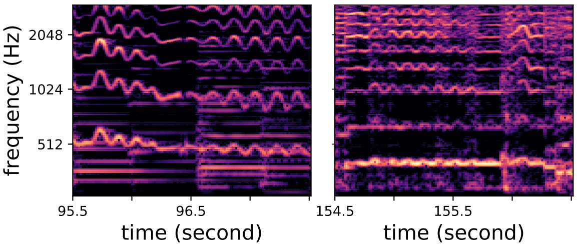

Figure 2: Excerpts of mel-spectrograms from MedleyDB: ‘Handel_TornamiAVagheggiar’ with female vocalist (left) and ‘PurlingHiss_Lolita’ with electric guitar (right) (see Section 4.2.1.)

In systems such as FE-VD, where audio features such as MFCCs or spectral flatness are used, the performance varies by SNR because the features are statistics of the whole bandwidth which includes not only the target signal (vocal) but also additive noise (instrumental). VD systems with deep neural networks are also not free from this issue since the low-level operation in the layers of deep neural networks may end up being a simple pattern matching by computing correlation.

This is a common phenomenon in other tasks as well, e.g., speech recognition, and we continue the discussion to a follow-up experiment in Section 5.2 and finally a suggestion on the problem definition and dataset composition in Section 6.

4.2.3 *Non-melodic Sources*

Although the interest of most VD systems appears to lie mainly in the melodic component of the song, we expected the system to learn percussive nature of the singing voice as well, which is exhibited by consonants from the singers. Therefore, our hypothesis is whether the system is confused by the consonants of singing voice and percussive instruments, resulting in either *i)* missing consonant parts (false negative) or *ii)* mis-classifying percussive instruments (false positive).

From our test results, we encounter false positive segments containing snare drums and hi-hats, but the exact cause of this misclassification is unclear. We further tested the system with drum set solos for potential false positives and with a collection of consonant sounds such as plosives and fricatives from the human voice for potential false negatives, but we did not observe a clear pattern in misclassification. Although we do not conduct further experiment on this, it suggests a deeper analysis, which may also lead to a clear understanding of preprocessing strategies including HPSS.

5. EXPERIMENT II: STRESS TESTING

5.1 Testing with artificial vibrato

Based on the confusion between ‘voice-like’ instruments and singing voice, we hypothesize that the current VD systems use vibrato patterns as one of the main tools for vocal segment detection. We explore the degree of confusion for each VD system by testing them on synthetic vibratos with varying rate, extent and formant frequencies.

5.1.1 *Data Preparation*

We create a set of synthetic vibratos with low pass-filtered sawtooth waveforms with f_0=220 Hz. We vary the modulation rate and frequency deviation ($f_\triangle$) to investigate their effects. Furthermore, we apply 5 bi-quad filters at the corresponding formant frequencies (3 for each) to synthesize so that they would sound like the basic vowel sounds, 'a', 'e', 'i', 'o', 'u' [29]. The modulation rate ranges in $\{0.5, 1, 2, 4, 6, 8, 10$ Hz$\}$ and the frequency deviation ranges in $\{0.01, 0.1, 0.3, 0.6, 1, 2, 4, 8$ semitones$\}$ with respect to its f_0). As a result, the set consists of 7 (rates) $\times 8$ ($f_\triangle$'s) $\times 6$ (5 formants + 1 unfiltered) $= 336$ variations.

5.1.2 *Results*

Figure 3 shows the result of the prediction by the three VD systems on the synthetic vibratos. The accuracy of 1.0 indicates that the system does not confuse the artificial vibratos with singing voice. Here, we observe the performance difference of each model, which were not visible from looking at the scores in Table 2. In general, confusion areas tend to be concentrated on the bottom left to the center area of the graph. The extent and rate of the artificial tones that are highly misclassified seem to be around the range of vibratos of singers, which is said to be around 0.6 to 2 semitone with rate around 5.5 to 8 Hz [30]. We also observe a within-system difference, i.e., the presence and the type of formants affect the models. For instance, vibratos mimicking the vowel 'a' cause higher misclassification in all three models.

`FE-VD` performs much better than the latter two systems. Note that `FE-VD` is a feature engineering model, where unique features, such as the Fluctogram and vocal variance, are mostly adapted from the ones used in speech recognition task. As these features were intentionally designed to reduce false positives from pitch-varying instruments, it appears to significantly reduce error rate on vibratos with rate and extent that are beyond the range of human singers.

`CNN-VD` confuses slightly wider range of vibratos. This is expected to some extent since the model prominently uses 3×3 filters on mel-spectrogram to detect local features, which can be regarded as a *local* pattern detector. In other words, the locality of CNN results in a system that is easily confused by frequency modulation regardless of the non-singing voice aspects of the signal. This implies that the model may benefit from looking at a varying range of time and frequency to learn vocal-specific characteristics such as timbre [21].

Lastly, `RNN-VD` performs better than the `CNN-VD`, though worse than `FE-VD`. On detecting vocal and non-vocal segments, it seems natural, even for humans, that past and future temporal context help. Also, we presume that the preprocessing of double stage HPSS contributes to the robustness of the system against vibrato. Again, this observation leaves a question of separating the contributions from preprocessing and model structure.

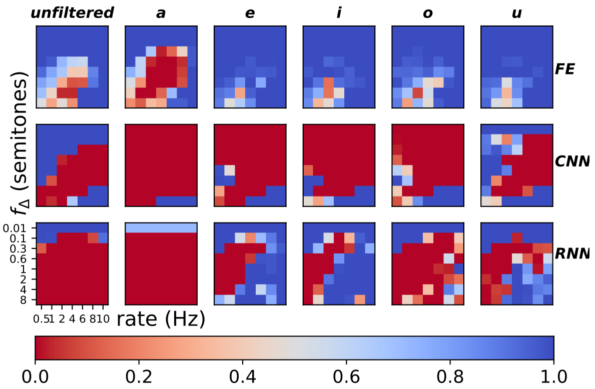

Figure 3: Heat-maps of the accuracies of the vibrato experiment result. Each row corresponds to VD systems (`FE-VD`, `CNN-VD`, `RNN-VD`) and each column corresponds to the formant (unfiltered, 'a', 'e', 'i', 'o', 'u'). Within each heat map, x- and y-axes correspond to the vibrato rate and frequency deviation as annotated on the lower-left subplot (see Section 5.1)

5.2 Testing with SNR

In this experiment, VD systems are tested with vocal gain adjusted tracks to further explore the behavior of the systems on various scenarios, which can reflect the real-world audio settings of live recordings and radios, for example.

5.2.1 *Data preparation*

We create a modified test set using 61 vocal-containing tracks provided by MedleyDB. We use the first 30 seconds of the songs to build a pair of (vocal, instrumental) tracks. Vocal tracks are modified with SNR of {+12 dB, -12 dB, +6 dB, -6 dB, 0 dB}.

5.2.2 *Results*

The results of the energy level robustness test are presented in Figure 4 with false positive rate, false negative rate, and overall error rate. We see a consistent trend across the performance of all three VD systems, which is once again an expected pattern as aforementioned in Section 4.2.2 – that increasing SNR helps to reduce false negatives. Overall error rate also exhibits a noticeable decrease in common with higher SNRs. In practice, one could take advantage of data augmentation with changing SNR to build a more robust system. More importantly, it can be part of the evaluation procedure for VD, as we discuss in Section 6.

While the VD systems behave similarly on all test cases, we note that `FE-VD`, owing to its additional features, shows lowest variance and lowest value for the false positive rate. Also, our assumption that the double-stage HPSS, which filters out vocal-related signals, would make `RNN-VD` more robust against SNR is observed to be not necessarily true as we clearly see performance differences across the varying SNR test cases.

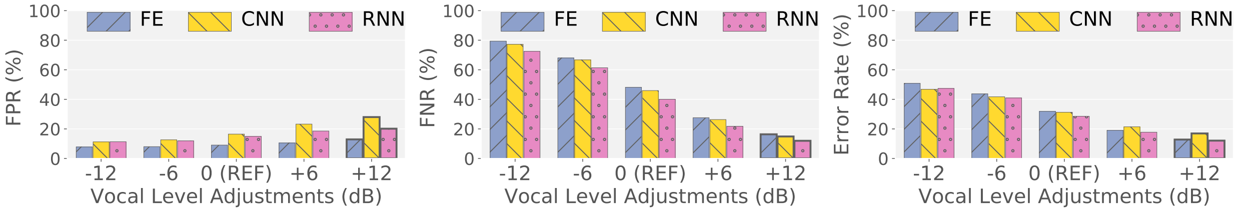

Figure 4: False positive rates, false negative rates, and overall error rates for the three systems in the stress testing with controlling SNR (see Section 5.2).

6. DIRECTIONS TO IMPROVE

6.1 Defining the problem and the datasets

6.1.1 Defining singing voice

By using the annotations in datasets such as Jamendo, many VD systems implicitly assume that the target 'singing voice' is defined as vocal components that correspond to the *main melody*. Other voice-related components such as backing vocal, narration, humming, and breathing are not clearly defined to be singing voice or not.

In some applications, however, they can be of interest. For example, a system may want to find purely instrumental tracks, avoiding tracks with backing vocal. In this case, the method should consider backing vocal as singing voice. However, for Karaoke applications, only the singing voice of the main melody would matter.

Therefore, an improvement can be made on defining the VD problem and creating datasets. For the annotation, a hierarchy among the voice-related components can be useful for both structured training and evaluation of a system [17, 23]. For the audio input, we see a great benefit of multitracks, where main vocal melody, backing vocal, and other components are provided separately.

6.1.2 Varying-SNR scenarios

For a long while, varying SNR had been one of the common ways to evaluate speech recognition or enhancement using dataset such as Aurora [5]. As observed in Section 4.2.2, it can be used as a 'test-set augmentation' to measure the performance of a system more precisely. Also, it can be an additional data augmentation method along with the ones in [27] to build a VD system more robust to various audio settings, such as audios from user generated videos. These can both be easily achieved with a multitrack dataset in practice.

6.1.3 Measuring dataset noise

Human annotators are neither perfect or identical, thus causing annotation noise and disagreement. Since VD is a binary classification problem, we may remain optimistic by assuming that the annotation noise is a matter of temporal precision, which is arbitrary and not agreed among many datasets so far. For example, in RWC Popular Music [16], "short background segments of less than 0.5-second duration were merged with the preceding region" and the annotations have 8 decimal digits (in second), while in Jamendo, they are 3 decimal digits. The optimal precision may depend on human perception of sound which is often said around 10 ms in general [19]. Although it would require a deeper investigation, the current temporal precision may be too high, leading to evaluate the systems with an overly precise annotation.

6.2 Learning from human perception

The characteristic of voice was the main motivation in the very early works exploiting speech-related features [1, 10]. Clearly, however, those approaches that solely relied on speech features showed limited performances. While following works has improved the performance, as our experiments have demonstrated through this paper, the systems do not completely take advantage of the cues that human is probably using, e.g., the global formants, linguistic information, musical knowledge, etc.

6.3 Preprocessing

A light-weight VD system was introduced in [12] where only MFCCs were used to achieve an accuracy of 84.8% on the Jamendo dataset. This implies that there is a possibility to achieve better performance by optimizing the preprocessing stage. One of the unanswered questions is the effect of the preprocessing stage in `RNN-VD` [11] as well as whether similar processing could lead to better performance with other systems, e.g., CNN [27].

7. CONCLUSIONS

In this paper, we suggested that there still are several areas to improve for the current singing voice detectors. In the first set of experiments, we identified the common errors through error analysis on three recent systems. Our observations that the main sources of error are pitch-fluctuating instruments and low signal-to-noise ratios of the singing voice motivated us to further perform stress tests. Testing with synthetic vibratos revealed that some systems (`FE-VD`) are more robust to non-vocal vibratos than others (`CNN-VD` and `RNN-VD`). SNR-varying test showed that SNR manipulation greatly affects the current VD systems, thus it can potentially be used to strengthen the VD systems to become invariant to a wider range of audio settings. As we propose several directions for a more robust singing voice detector, we note that defining the VD problem is dependent on the goal of the system, thus using multitrack datasets can be beneficial. Our future interest is to further investigate on SNR to extend VD systems on uncontrolled audio settings and to examine different components of individual systems, including the preprocessing stage.

8. ACKNOWLEDGEMENTS

We thank Bernhard Lehner and Simon Leglaive for active discussion and code, Jeongsoo Park for sharing Ono's code. This work was supported by the National Research Foundation of Korea (Project 2015R1C1A1A02036962).

9. REFERENCES

[1] Adam L Berenzweig and Daniel PW Ellis. Locating singing voice segments within music signals. In *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, pages 119–122. IEEE, 2001.

[2] Adam L Berenzweig, Daniel PW Ellis, and Steve Lawrence. Using voice segments to improve artist classification of music. In *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. Audio Engineering Society, 2002.

[3] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. MedleyDB: A multitrack dataset for annotation-intensive mir research. In *ISMIR*, volume 14, pages 155–160, 2014.

[4] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical and jazz music databases. In *Proc. of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*, volume 2, pages 287–288, 2002.

[5] Hans-Günter Hirsch and David Pearce. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Chao-Ling Hsu, Liang-Yu Chen, Jyh-Shing Roger Jang, and Hsing-Ji Li. Singing pitch extraction from monaural polyphonic songs by contextual audio modeling and singing harmonic enhancement. In *Proc. of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 201–206, 2009.

[8] Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2010.

[9] Chao-Ling Hsu, DeLiang Wang, Jyh-Shing Roger Jang, and Ke Hu. A tandem algorithm for singing pitch extraction and voice separation from music accompaniment. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(5):1482–1491, 2012.

[10] Youngmoo E Kim and Brian Whitman. Singer identification in popular music recordings using voice coding features. In *Proc. of the 3rd International Conference on Music Information Retrieval (ISMIR)*, volume 13, page 17, 2002.

[11] Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 121–125. IEEE, 2015.

[12] Bernhard Lehner, Reinhard Sonnleitner, and Gerhard Widmer. Towards light-weight, real-time-capable singing voice detection. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 53–58, 2013.

[13] Bernhard Lehner, Gerhard Widmer, and Sebastian Böck. A low-latency, real-time-capable singing voice detection method with lstm recurrent neural networks. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 21–25. IEEE, 2015.

[14] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7480–7484. IEEE, 2014.

[15] Maria E Markaki, André Holzapfel, and Yannis Stylianou. Singing voice detection using modulation frequency feature. In *SAPA@ INTERSPEECH*, pages 7–10, 2008.

[16] Matthias Mauch, Hiromasa Fujihara, Kazuyoshi Yoshii, and Masataka Goto. Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 233–238, 2011.

[17] Brian McFee and Juan Pablo Bello. Structured training for large-vocabulary chord recognition. In *Proc. of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[18] Brian McFee, Matt McVicar, Oriol Nieto, Stefan Balke, Carl Thome, Dawen Liang, Eric Battenberg, Josh Moore, Rachel Bittner, Ryuichi Yamamoto, et al. librosa 0.5. 0, 2017.

[19] Brian CJ Moore. *An introduction to the psychology of hearing*. Brill, 2012.

[20] Nobutaka Ono, Kenichi Miyamoto, Jonathan Le Roux, Hirokazu Kameoka, and Shigeki Sagayama. Separation of a monaural audio signal into harmonic/percussive components by complementary dif-

fusion on spectrogram. In *Signal Processing Conference, 2008 16th European*, pages 1–4. IEEE, 2008.

[21] Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 2744–2748. IEEE, 2017.

[22] Mathieu Ramona, Gaël Richard, and Bertrand David. Vocal detection in music with support vector machines. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 1885–1888. IEEE, 2008.

[23] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6517–6525, 2017.

[24] Lise Regnier and Geoffroy Peeters. Singing voice detection in music tracks using direct voice vibrato detection. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1685–1688. IEEE, 2009.

[25] Martın Rocamora and Perfecto Herrera. Comparing audio descriptors for singing voice detection in music audio files. In *Brazilian symposium on computer music, 11th. san pablo, brazil*, volume 26, page 27, 2007.

[26] Jan Schlüter. Learning to pinpoint singing voice from weakly labeled examples. In *Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 44–50, 2016.

[27] Jan Schlüter and Thomas Grill. Exploring data augmentation for improved singing voice detection with neural networks. In *Proc. of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 121–126, 2015.

[28] Emery Schubert and Joe Wolfe. Voicelikeness of musical instruments: A literature review of acoustical, psychological and expressiveness perspectives. *Musicae Scientiae*, 20(2):248–262, 2016.

[29] Julius Orion Smith. *Introduction to digital filters: with audio applications*, volume 2. Julius Smith, 2007.

[30] Renee Timmers and Peter Desain. Vibrato: Questions and answers from musicians and science. In *Proc. Int. Conf. on Music Perception and Cognition*, volume 2, 2000.

[31] Ye Wang, Min-Yen Kan, Tin Lay Nwe, Arun Shenoy, and Jun Yin. Lyrically: automatic synchronization of acoustic musical signals and textual lyrics. In *Proc. of the 12th annual ACM international conference on Multimedia*, pages 212–219. ACM, 2004.