Dilated Convolutional Model for Melody Extraction

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

29-01-2022 / 03-02-2022

CITATION

Wang, Xian; Liu, Lingqiao; Shi, Javen (2022): Dilated Convolutional Model for Melody Extraction. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.19092005.v1>

DOI

[10.36227/techrxiv.19092005.v1](https://doi.org/10.36227/techrxiv.19092005.v1)

Dilated Convolutional Model for Melody Extraction

Xian Wang, Lingqiao Liu, and Javen Qinfeng Shi

Abstract—Melody extraction is a challenging task in music information retrieval that enables many down-stream applications. In this paper we propose a simple dilated convolutional model for melody extraction. It takes variable-q transforms as inputs. It first uses consecutive layers of convolution to capture local temporal-frequency patterns. Afterward, it relies only a single layer of dilated convolution for capturing global frequency patterns formed by the pitches and harmonics of active notes. This model is effective in that it achieves the-state-of-the-art performance on most datasets, for both general and vocal melody extraction. In addition, it gets the best performance with the least training data.

Index Terms—Melody extraction, variable-q transform, dilated convolution.

I. INTRODUCTION

MELODY extraction is an import, challenging task in music information retrieval. It underpins many other applications, such as query-by-humming, cover song identification, genre classification and singer identification [1]. With the rapid development of deep learning, recent years have seen great progress of deep learning applied to melody extraction. In the literature a majority of existing studies were focused on vocal/singing melody extraction, because singing voice has many unique features that can be exploited for vocal melody extraction, such as formants, vibrato and tremolo [2].

Various aspects of vocal melody extraction have been investigated in the literature. First, different network structures were proposed, including fully connected networks [3], [4], convolutional neural networks (CNNs) [5]–[7], transformer [8], convolutional recurrent neural networks (CRNNs) [1], [9], and encoder-decoder structures [10]–[12]. It is common practice for vocal melody extraction to simultaneously perform voicing detection that detects whether a frame is voiced (i.e., whether a frame has singing voice), and pitch detection that detects the pitch of the melody. Voicing detection is feasible given the unique features of singing voice. Next, a plethora of deep learning techniques have been applied to vocal melody extraction, such as patch processing [5], ensemble learning [3], joint-task learning [1], transfer learning [10], multi-resolution [7], residual connection [12], atrous spatial pyramid pooling (ASPP) [6], deep connection [6], and semi-supervised learning [13].

On the other hand, there are few studies on general melody extraction. In [14] a traditional signal processing-based approach was proposed for general melody extraction. This approach used the weighted sum of harmonics as pitch salience functions, from which pitch contours were constructed. Afterward, melody contours were selected from pitch contours

based on some criteria. Ref. [15] proposed a long short-term memory (LSTM) network for extracting melody and simultaneously detecting regions of melody using a harmonic sum loss. In [16] a fully convolutional structure was proposed for multi-pitch detection and melody extraction. This structure uses harmonic constant-q transforms (HCQTs) as the input representation. Compared with CQTs, HCQTs have an additional dimension, namely, the harmonic dimension, which for each fundamental frequency f_0 includes the energies at $r f_0$ for $r \in \{0.5, 1, 2, 3, 4, 5\}$. This structure was improved in [17] by arranging the output of each hidden layer into something similar to the HCQT. In [18] a CRNN was proposed for general melody extraction. It takes non-negative matrix factorization (NMF)-based features as inputs, and uses a CNN as the acoustic model and an RNN as the language model.

In this paper we propose a dilated convolutional model for melody extraction. Its input representation is the variable-q transform (VQT). This model consists of two parts. In the first part, four convolution layers are used for modeling temporal and local frequency patterns. To cover a large time span that is essential for melody detection, the convolutions in this part gradually increase the dilation rate over the time axis. In the second part, a single convolution layer is devoted to modeling global frequency patterns arising from the interaction among the pitches and harmonics of notes active simultaneously. To this end, this convolution uses dilation and large kernel size over the frequency axis so as to cover a large frequency range and at the same time control overfitting. To further curb potential overfitting due to the large kernel size, L_2 regularization is applied to the kernel of this convolution. In extensive experiments under various settings, this model outperforms most existing models on most datasets, for both general and vocal melody extraction. In addition, it has better generalization capability in that its performance deteriorates the least when tested on music not well represented in the training data. Furthermore, it performs the best with the least training data.

II. MODEL

A. Input Representation

Our model takes VQTs as inputs. We prefer VQT over CQT, because VQT allows more flexible control of the bandwidths for individual filters [19]. This enables us to improve the time resolution for lower frequencies. We open source our GPU implementation of the VQT which can significantly benefit following research. The VQT is configured as follows. The sampling frequency f_s is 44,100 Hz. The frequency resolution B is 60 bins per octave. The minimum frequency $f_{\min}$ is $m2f(C1) \times 2^{-2/B}$, where $m2f(\cdot)$ is a function that converts

X. Wang, L. Liu, and J. Shi are with the School of Computer Science, The University of Adelaide, Adelaide, South Australia 5005, Australia (e-mail: {xian.wang01, lingqiao.liu, javen.shi}@adelaide.edu.au).

music notes to frequencies. The maximum frequency $f_{\max}$ is $m2f(B9) \times 2^{2/B}$. The bandwidth of filter k is configured as

$$\Omega_k = \max(f_k/Q, 14) \text{ Hz},$$

where f_k is the center frequency of this filter, and Q is the quality factor expressed as

$$Q \triangleq 1 / (2^{1/B} - 1).$$

The hop size h is 256 samples. Except Ω_k , the configuration of the VQT is the same as the configuration used in [16] for the CQT. Consequently, the VQT has 540 frequency bins in total. We further cut the VQT into snippets of 1200 frames and feed them into our neural network as mini examples. Thus, each mini example has size 1200×540 .

B. Model for General Melody Extraction

Fig. 1 gives the structure of the proposed dilated convolutional model for general melody extraction. The legend for this figure is as follows.

- rectangle: a computational block
- $C@T \times F$ above a block: the output of the block has size $T \times F \times C$ which is in order frames $\times$ frequency bins $\times$ channels.
- name under a block: the block is given a name name.
- conv: $c@k_t(d_t) \times k_f(d_f)$: a convolution with c filters, kernel size $k_t \times k_f$ which is in order frames $\times$ frequency bins, and dilation rate $d_t \times d_f$. When d_t or d_f is missing, it is equal to one.
- bn: batch normalization.
- dropout: dropping with probability 0.2.

The inputs to this model are VQT snippets of size 1200×540 . Along the frequency dimension, we assume that only the first 360 frequency bins can be candidates for melody pitches. This range covers notes C1 to B6. `local-0` to `local-3` are four convolution layers used to capture local time-frequency patterns. The inputs to these blocks are properly padded to keep the time and frequency dimensions unchanged. `local-0` uses kernel size 1×5 , and the other three use kernel size 3×5 . For melody extraction, it is necessary to consider a large time span. To this end and inspired by [17], we use non-uniform time-dimension dilation rates 2, 4 and 8 for `local-1`, `local-2`, and `local-3`, respectively. Consequently, after `local-3` we achieve an overall receptive field of 29×17 . This time span is the same or approximately the same as those used in [16], [17].

In polyphonic music, notes active simultaneously can overlap in their pitches and harmonics. This means that to single out the strongest note as the potential melody note, it is essential to aggregate information over a large frequency context. In the literature, various approaches were applied for this purpose, with most of them drawing inspiration from image semantic segmentation. Some approaches are as follows.

- Large kernel size over the frequency dimension [16], [18].
- Max-pooling to gradually shrink the frequency dimension [1], [10], [11].
- Dense connection or similar effects achieved through RNN layers [1], [15].

- ASPP [6].
- Transformer [8].

Our model uses a single layer of dilated convolution to capture global frequency patterns. In particular, block `global` employs a convolution with kernel size 1×97 and dilation rate 1×5 . The resulting receptive field is 4 octaves under and 4 octaves above each pitch. Compared with the above existing approaches, ours is the simplest. Here the use of dilated convolution is feasible, because the receptive field over the frequency dimension is 17 after `local-3` so that even after using dilated convolution the frequency range can still be covered seamlessly.

Melody extraction is sparsely labeled in that at a time point only the melody pitch is known even if multiple pitches could be active simultaneously. On the other hand, the kernel of the dilated convolution in `global` is still quite large in that it has about 0.2 million parameters, accounting for about 91% of the total parameters of the model. For a better generalization performance, we apply L_2 regularization to the kernel of this convolution. The coefficient of the regularization is 10^{-4} .

Before feeding the output of `local-3` into `global`, we need to pad it manually along the frequency dimension. The number of paddings is 240 before frequency bin 0 and 80 after frequency bin 539. This is done in `padding`. The output of `global` has 360 frequency bins, all of which are candidates for the melody pitch. After `global`, in order to increase the nonlinearity in `fusion` we use a dense layer to project the number of feature maps to a smaller number. Afterward, in output we use a dense layer with one unit and sigmoid activation to get the probability of being the melody pitch for each pitch.

In [20] we applied a similar model to multi-pitch detection (MPE). The model proposed here differs from the model in [20] in the following aspects.

- Despite being related in that they both need to detect pitches, MPE and melody extraction differ in that the former is densely labeled, where as the latter is only sparsely labeled. In MPE for each time point the pitches of all the active notes are known.
- Given the above difference, for each layer of the proposed model we used a smaller number of feature maps, because there are not enough positive labels to support the large number of feature maps used in [20].
- To determine if a pitch is the melody pitch, we need to consider a larger time span. Therefore, we used dilated convolution along the time dimension in `local-1` to `local-3`.
- To further alleviate the problem of sparse labeling, we proposed to use L_2 regularization in `global`.

C. Model for Vocal Melody Extraction

The proposed model itself can be used unmodified for vocal melody extraction. In this case we need to modify the configuration of the VQT to reflect the fact that vocal pitches usually have a narrower frequency range than instrumental pitches. Specifically, we lower $f_{\max}$ to $m2f(G9) \times 2^{2/B}$ and keep all the other parameters unchanged. Consequently, now

Fig. 1: Dilated convolutional model.

the VQT has 520 frequency bins. We limit the pitches of vocal melody to be within the first 320 frequency bins, or equivalently, in the range of C1 to D#6.

D. Loss Function

We formulate melody detection as a multi-labeling problem and use binary cross entropy as the loss function. Specifically, for a pitch p the loss is calculated as

$$l(p) = -\Pr_g(p) \log \Pr_d(p) - [1 - \Pr_g(p)] \log [1 - \Pr_d(p)],$$

where $\Pr_g(p)$ and $\Pr_d(p)$ are the ground-truth and the detected probability of pitch p , respectively. For each frame the pitch with the maximum probability is selected as a candidate for melody pitches. To decide whether a frame is voiced or not, we select a threshold that maximizes the voicing accuracy on the validation split as the voicing threshold. Candidate pitches with probabilities larger than the voicing threshold are then selected as melody pitches.

III. EXPERIMENTS

In this section we evaluate the performance of the proposed model for general and vocal melody extraction. In both cases we use only the MedleyDB dataset ([21]) for training and validation, as it is the only dataset that is large scale, of professional quality, has enough genre variety, and was accurately annotated [21]. We use overall accuracy as the sole performance measure. It is calculated with `mir_eval` [22]. We use Adam optimizer ([23]) and set the learning rate to be 10^{-4} . The batch size is one. The checkpoint that yields the best validation performance is selected for testing. Training stops if the validation performance does not improve in 10 epochs. The code is available at <https://shorturl.at/svUX8>.

A. Performance for General Melody Extraction

For the performance of general melody extraction, we will compare the proposed model with [11], [14], [16]–[18]. Ref. [14] is an approach based on signal processing so that it does not need a dataset for training. All the other models are deep learning-based approaches and used MedleyDB for training and validation. MedleyDB has 108 recordings that

TABLE I: Performance of General Melody Extraction Tested on Multiple Datasets

	ADC04	MDB-S	MIREX05	MDB	ORCHSET	WJD
Salamon [14]	69.03	49.17	71.67	50.41	22.56	64.23
Bittner [16]	73.77	62.65	72.91	61.75	42.61	71.57
Basaran [18]	66.88	68.29	73.42	63.61	48.27	73.84
Balhar [17]	74.55	63.08	70.42	64.44	52.52	73.02
Proposed	75.75	65.01	73.64	65.27	57.78	74.67

(a) Under Partition (67, 15, 26)

	ADC04	MDB-S	MIREX05	MDB	ORCHSET	WJD
Hsieh [11]	78.44	66.39	70.50	64.31	53.88	73.47
Proposed	73.47	66.33	71.57	65.01	56.80	74.39

(b) Under Partition (66, 15, 27)

have melody. Ref. [16] used a partition of (71, 9, 27), which is in the order of training split, validation split, and test split. Refs. [17], [18] used a partition of (67, 15, 27). Ref. [11] used a partition of (66, 15, 27). We observe the following about these partitions.

- All the partitions have the same test split.
- Partition (71, 9, 27) has 107 recordings in total.
- Partition (67, 15, 27) has 109 recordings in total, because the training and the test split share one recording.
- Partition (66, 15, 27) was obtained by removing the shared recording from the training split of partition (67, 15, 27).

To compare different models, we must make sure that the same partition is used and that there is no intersection between any pair of splits in this partition. Under this principle, we decide to train the proposed model under two partitions, namely, (67, 15, 26) and (66, 15, 27). They are obtained by starting from partition (67, 15, 27) and removing the intersection recording, respectively, from the test and the training split. Below listed datasets are used for testing.

- MDB: the test split of MedleyDB.
- ADC04: 20 excerpts of roughly 20 s¹.
- MIREX05: MIREX05 training data that has 13 excerpts of 10 to 40 s¹.

¹available at <https://labrosa.ee.columbia.edu/projects/melody/>

- ORCHSET: the Orchset dataset that has 64 excerpts of 10 to 32 s [24].
- MDB-S: The MDB-melody-synth dataset ([25]) has a total of 65 recordings, of which we select those that also appear in the test split of MedleyDB for testing. There are 13 such recordings under partition (67, 15, 26) and 14 under partition (66, 15, 27).
- WJD: a subset of the WJazzD dataset [26]. We use the same subset as the one used in [17]. This subset has 74 recordings.

Tables I(a) and I(b) compare the proposed model with the existing models under each partition, respectively. The performance measure is the overall accuracy in percentage. The highest result on a dataset is highlighted in bold. The results in Table I(a) were obtained in the following ways.

- Salamon [14]: melodies were extracted with the library of Essentia².
- Bittner [16]: We retrained the model.
- Basaran [18] and Balhar [17]: We used the melodies extracted in [17].

In Table I(b), the results for Hsieh [11] were obtained by running the trained model accompanying [11]. We observe the following in the two tables.

- Under partition (67, 15, 26) the proposed model was the best on all the datasets except MDB-S, on which it was the runner-up after Basaran [18].
- Under the other partition, the proposed model and Hsieh [11] achieved the-state-of-the-art performance on four and two datasets, respectively.
- Among all the datasets, ORCHSET was the most challenging for all the models, with their performance plummeting on this dataset. The proposed model was stabler than the others in that its performance deteriorated the least on this dataset. Two factors contribute to the current situation. First, the content of ORCHSET is not well represented in MedleyDB. Second, nearly 94% of the frames in ORCHSET are voiced, far higher than the situation in MedleyDB.
- Among all the datasets, ADC04 and WJD were the easiest for all the models. ADC04 is tiny and has no enough variety in genres, whereas WJD is large-scale but consists only of jazz music.

B. Performance for Vocal Melody Extraction

To evaluate the performance of the proposed model for vocal melody extraction, we will compare it with [1], [6], [11], [12]. We use only MedleyDB for training and validation. The partition is (35, 13, 12), the same as the one used in [11]. We have to retrain these opponent models, because they were trained originally on different datasets and some even adopted dataset augmentation. Moreover, the iKala dataset ([27]) has been unavailable since 2017. The following datasets are used for testing.

- MDB: the test split of MedleyDB.
- ADC04: 12 excerpts of ADC04 with vocal melody.

TABLE II: Performance of Vocal Melody Extraction

	ADC04	MIREX05	MDB	MIR-1K	RWC
Hsieh [11]	84.23	79.53	79.10	60.61	72.57
Gao [6]	72.52	79.76	74.09	61.36	66.42
Kum [1]	67.93	75.59	77.84	60.91	71.03
Yu [12]	67.30	72.70	72.26	59.25	70.67
Proposed	75.71	81.52	80.88	66.84	73.93

(a) Trained on MedleyDB and Tested on Multiple Datasets

model	trained on	OA
Hsieh [11]	MIR-1K, MedleyDB	79.7
Lu [8]	same as above	78.4
Yu [12]	same as above	79.4 [8]
	same as above	77.0 [8]
kum [1]	RWC, MedleyDB, iKala	78.1 [13]
	RWC, MedleyDB, iKala, w/ data aug.	80.5
Kum [13]	RWC, MedleyDB, iKala, unlabeled data	79.2
Yu [9]	RWC, MedleyDB	68.1
Proposed	MedleyDB	80.9

(b) Trained on Different Datasets and Tested on MedleyDB

- MIREX05: 9 excerpts of MIREX05 with vocal melody.
- MIR-1K: 1000 Chinese karaoke clips [28].
- RWC: the popular music collection of the RWC database [29], consisting of 100 popular songs.

Table II(a) shows the overall accuracies in percentage of different models when trained only on MedleyDB under partition (35, 13, 12). Except ADC04 on which it was the second after Hsieh [11], the proposed model was the best on all the other datasets, in particular with a larger margin on MIR-1K over all the other models.

From another perspective Table II(b) compares the performance for vocal melody extraction of some existing models with the proposed. In their original papers, all these existing models were trained on different datasets, but were tested at least on the test split of MedleyDB under partition (35, 13, 12). Thus, we can compare them with the proposed in terms of the testing performance on MedleyDB. In this case we need not retrain the existing models, but instead cite the results published in the literature. Table II(b) presents the results under the this setting. The performance measure is the overall accuracy (OA) in percentage. Regarding the origins of the results in Table II(b), if there is a reference cited after a result, then the result is from the reference. Otherwise, the result is from the original paper where the model was proposed. It is amazing that the proposed model performed the best with the least training data.

IV. CONCLUSIONS

In this paper we proposed a dilated convolutional model for melody extraction. At the heart of this model is the use of dilated convolution to capture large scale global frequency patterns. This model had exceptional performance on most of the datasets for both general and vocal melody extraction. It also achieved the best performance with the least training data. One aspect may hinder the adoption of this model, namely, the large amount of computation brought about by the the dilated convolution, because of the large kernel size used. We are working to tackle this problem.

²<https://essentia.upf.edu/>

REFERENCES

- [1] S. Kum and J. Nam, “Joint detection and classification of singing voice melody using convolutional recurrent neural networks,” *Appl. Sci.*, vol. 9, no. 7, 2019.
- [2] L. Regnier and G. Peeters, “Singing voice detection in music tracks using direct voice vibrato detection,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 1685–1688.
- [3] S. Kum, C. Oh, and J. Nam, “Melody extraction on vocal segments using multi-column deep neural networks,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 819–825.
- [4] F. Rigaud and M. Radenien, “Singing voice melody transcription using deep neural networks,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 737–743.
- [5] L. Su, “Vocal melody extraction using patch-based CNN,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 371–375.
- [6] P. Gao, C. You, and T. Chi, “A multi-scale fully convolutional network for singing melody extraction,” in *Asia-Pac. Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2019, pp. 1288–1293.
- [7] —, “A multi-dilation and multi-resolution fully convolutional network for singing melody extraction,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 551–555.
- [8] W. T. Lu, J. Wang, M. Won, K. Choi, and X. Song, “SpecTNT: A time-frequency transformer for music audio,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 396–403.
- [9] S. Yu, Y. Yu, X. Chen, and W. Li, “HANME: Hierarchical attention network for singing melody extraction,” *IEEE Signal Process. Lett.*, vol. 28, pp. 1006–1010, 2021.
- [10] W. T. Lu and L. Su, “Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 521–528.
- [11] T. Hsieh, L. Su, and Y. Yang, “A streamlined encoder/decoder architecture for melody extraction,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 156–160.
- [12] S. Yu, X. Sun, Y. Yu, and W. Li, “Frequency-temporal attention network for singing melody extraction,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 251–255.
- [13] S. Kum, J. Lin, L. Su, and J. Nam, “Semi-supervised learning using teacher-student models for vocal melody extraction,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2020, pp. 93–100.
- [14] J. Salamon and E. Gómez, “Melody extraction from polyphonic music signals using pitch contour characteristics,” *IEEE Trans. Speech Audio Process.*, vol. 20, no. 6, pp. 1759–1770, 2012.
- [15] H. Park and C. D. Yoo, “Melody extraction and detection through LSTM-RNN with harmonic sum loss,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 2766–2770.
- [16] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, “Deep salience representations for F0 estimation in polyphonic music,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 63–70.
- [17] J. Balhar and J. Hajič jr, “Melody extraction using a harmonic convolutional neural network,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2019.
- [18] D. Basaran, S. Essid, and G. Peeters, “Main melody estimation with source-filter NMF and CRNN,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 82–89.
- [19] C. Schölkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, “A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution,” in *AES Int. Conf. Semantic Audio*, 2014.
- [20] X. Wang, L. Liu, and Q. Shi, “Enhancing piano transcription by dilated convolution,” in *Proc. IEEE Int. Conf. Mach. Learn. App.*, 2020, pp. 1446–1453.
- [21] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, “MedleyDB: A multitrack dataset for annotation-intensive MIR research,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 155–160.
- [22] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “mir_eval: A transparent implementation of common MIR metrics,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014.
- [23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, vol. 9, 2015.
- [24] J. J. Bosch, R. Marxer, and E. Gómez, “Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music,” *J. New Music Res.*, vol. 45, pp. 101–117, 2016.
- [25] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez, and J. P. Bello, “An analysis/synthesis framework for automatic F0 annotation of multitrack datasets,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 71–78.
- [26] M. Pfeleiderer, K. Frieler, J. Abeßer, W.-G. Zaddach, and B. Burkhart, Eds., *Inside the Jazzomat - New Perspectives for Jazz Research*. Schott Campus, 2017.
- [27] T. Chan, T. Yeh, Z. Fan, H. Chen, L. Su, Y. Yang, and J. R. Jang, “Vocal activity informed singing voice separation with the ikala dataset,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2015, pp. 718–722.
- [28] C. Hsu and J. R. Jang, “On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset,” *IEEE Trans. Speech Audio Process.*, vol. 18, no. 2, pp. 310–319, 2010.
- [29] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical and jazz music databases,” in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2002, pp. 287–288.