MELODY EXTRACTION ON VOCAL SEGMENTS USING MULTI-COLUMN DEEP NEURAL NETWORKS

Sangeun Kum, Changheun Oh, Juhan Nam
Graduate School of Culture Technology
Korea Advanced Institute of Science and Technology
{keums, thecow, juhannam}@kaist.ac.kr

ABSTRACT

Singing melody extraction is a task that tracks pitch contour of singing voice in polyphonic music. While the majority of melody extraction algorithms are based on computing a saliency function of pitch candidates or separating the melody source from the mixture, data-driven approaches based on classification have been rarely explored. In this paper, we present a classification-based approach for melody extraction on vocal segments using multi-column deep neural networks. In the proposed model, each of neural networks is trained to predict a pitch label of singing voice from spectrogram, but their outputs have different pitch resolutions. The final melody contour is inferred by combining the outputs of the networks and post-processing it with a hidden Markov model. In order to take advantage of the data-driven approach, we also augment training data by pitch-shifting the audio content and modifying the pitch label accordingly. We use the RWC dataset and vocal tracks of the MedleyDB dataset for training the model and evaluate it on the ADC 2004, MIREX 2005 and MIR-1k datasets. Through several settings of experiments, we show incremental improvements of the melody prediction. Lastly, we compare our best result to those of previous state-of-the-arts.

1. INTRODUCTION

Melody is a pitch sequence with which one might hum or whistle a piece of polyphonic music in an identifiable manner [10]. Among others, singing voice has been used as a main source of the melody, particularly in popular music. Thus, extracting melodies from singing voice can be used for not only music retrieval, for example, query-by-humming [5] or cover song identification [16] but also voice separation as a guide to inform the voice source.

A number of melody extraction algorithms, which can be applied for singing voice with an additional voice detection step, have been proposed so far and they are well summarized in [13]. The majority of the algorithms are based on computing a saliency function of pitch candidates or separating the melody source from the mixture. They typically return melody as a continuous pitch stream. On the other hand, data-driven approaches based on classification, which categorizes melody into a finite set of pitch labels, have rarely been explored. An early work by Ellis and Poliner used a support vector machine classifier to predict a pitch label from spectrogram [7]. Recently, Bittner et. al. proposed a method using a random forest classifier that predicts a pitch contour from highly hand-crafted features [3]. To the best of our knowledge, no other attempts have been made so far.

This scarcity of classification-based approach might be attributed to the following limitations. First, the extracted melody is supposed to be quantized by the pitch categorization (e.g. semitone unit in [7]). While this discrete outcome may be useful for some applications that require a MIDI-level pitch notation, it loses detailed information about singing styles, for example, vibrato or note-to-note transition patterns. Second, the data-driven approach typically requires sufficient labeled training data to achieve good performance. Finer pitch resolutions may need even more training data and possibly more complicated classifiers that can handle it.

In this paper, we address these limitations of the classification-based approach using multi-column deep neural networks (MCDNN). In the proposed model, each of DNN is trained to predict a pitch label of singing voice with different pitch resolutions. The outputs of the networks are combined and post-processed with a hidden Markov model to produce the final melody contour. Given a single DNN and training data, we observed that performance is inversely proportional to pitch resolutions. By combining the multiple DNNs, we show that the model can achieve higher pitch resolutions and better performance at the same time. In addition, we augment the training data by pitch-shifting the audio content and modifying the pitch label accordingly. We show that this is an effective technique to improve classification performance of the model.

2. RELATED WORK

The MCDNN was originally devised as an ensemble method to improve the performance of DNN for image classification [4]. In this model, each column (or single DNN) share the same network configuration and training

data. However, they are randomly initialized, and the input data may be preprocessed in different ways for each column. The predictions from all columns are averaged to produce the final output. The multi-column approach was applied to image denoising as well [1]. In this approach, each column is trained on a different type of noise, and the outputs are adaptively weighted to handle a variety of noise types. Our proposed model may pose half-way between these two approaches. Each column is trained to conduct a different role, having a different number of outputs. However, we combine the outputs with even weights as they are the same pitch quantity with different resolutions.

As aforementioned, classification-based melody extraction is rarely attempted. Among them, our proposed model is similar to the SVM approach by Ellis and Poliner [7] in that both of them predict a pitch label from spectrogram using a classifier and a hidden Markov model for post-processing. However, our model produces a finer pitch resolution. Also, we take advantage of deep neural networks, which recently has proved to be capable of having great performance with sufficient labeled data and computing power.

3. PROPOSED METHODS

3.1 Multi-Column Deep Neural Networks

Our architecture of the MCDNN is illustrated in Figure 1. Each of the DNN columns takes an odd-numbered spectrogram frames as input to capture contextual information from neighboring frames and predicts a pitch label at the center position of the context window. The DNNs are configured with three hidden layers and ReLUs for the non-linear function in common, but the output layers predict a pitch label with different resolutions. The lowest resolution is semitone, corresponding to the leftmost one. The next ones progressively have higher resolutions by two times (e.g. 0.5 semitones, 0.25 semitones, ...), thereby having as much pitch labels as the increased resolutions. Given the outputs of the columns, we compute the combined posterior as follows:

$$y^N_{MCDNN} = \prod_{i=1}^{N} (y^i_{DNN} + \epsilon) \quad (1)$$

where y^i_{DNN} corresponds to the prediction from i^{th} column DNN, and N corresponds to the number of total columns. We use multiplication in a maximum-likelihood sense, assuming that the column DNNs are independent. We add a small value, ϵ to prevent numerical underflow. Note that, before combining the predictions, those with lower resolutions are actually expanded by locally replicating each element so that the output sizes are the same for all columns. For example, the leftmost DNN in Figure 1, which predicts pitch in semitone, expands the output vector by a factor of 4. As a result, the merged posterior maintains the highest pitch resolution.

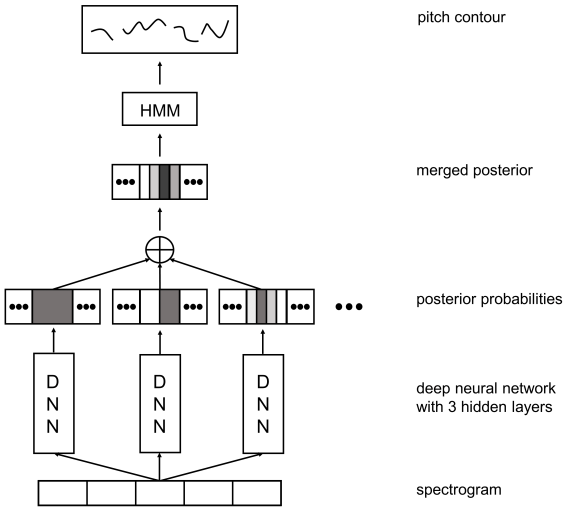


Figure 1: Block diagram of our proposed multi-column deep neural networks for singing melody extraction

3.2 Data Augmentation

Recent advances in deep learning are attributed to the availability of large-scale labeled data among others. Considering that melody-labeled public datasets are not much available, and manual labeling is laborious, it is desirable to augment existing datasets. In our experiments, we augment our training set by changing the global pitch of the audio content. Instead of pitch-shifting by resampling [10], which carries out time-stretching at the same time, we use a phase-vocoder method approach to achieve more natural transposition [9]. Pitch shifting proved to be an effective method of data augmentation for singing voice detection [15]. We will show that it works for singing melody extraction as well. On top of this, we also augment the training data by simply using an extra dataset that covers more music genres, as melody characteristics are quite discriminative over different music genres [14].

3.3 Temporal Smoothing by HMM

Although the MCDNN is trained to capture contextual information by taking multiple frames as input data, this may be limited to learn long-term temporal dependencies that appear on the pitch contours of singing voices. Also, the prediction is performed independently every time step. In order to incorporate the sequential structure further, we conduct temporal smoothing for the combined output of the MCDNN using HMM. We implemented the HMM, following the procedure in [7].

3.4 Singing Voice Detection

The MCDNN is trained with only voiced frames for pitch classification. Therefore, a separate singing voice detection step is necessary for the test phase. However, since singing voice detection itself is a challenging task and not

our main concern in this paper, we evaluate the test data using two scenarios. In the first scenario, we assume that a perfect singing voice detector is available so that we focus on the performance of our model only on voiced frames. In the second scenario, we use a simple energy-based singing voice detector introduced in [7]. The detector sums spectral energy between 200 Hz and 1800 Hz where the singing voice is likely to have a higher level than background music. The sum is normalized by the median energy in the band, and a threshold is used to determine the presence of singing voice. We expect that the performance of our model will range between the results from the two scenarios if a better singing voice detector is available.

4. DATASETS

4.1 Training Datasets

We use the RWC pop music database as our main training set [8]. It contains 100 popular songs with singing voice melody annotations. We divide the database into two splits, 85 songs for training and the remaining 15 songs for validation. In order to avoid bias by gender and the number of singers, we select the songs such that male/female singers and solo/chorus singing are evenly distributed over the training and validation sets. We also prevent the same singer's songs from being split over the two sets so that singer voices in the validation stage are never heard. In order to train the MCDNN more effectively, we augment the training set by applying pitch-shifting by $\pm 1, 2$ semitones. This increases the amount of the training set by five times. Also, we modify the corresponding pitch label accordingly.

Since the RWC database includes only pop music, the model trained on the set may not work well for other genres. We thus increase the size of training set and genre diversity by using 60 vocal tracks of the MedleyDB dataset as an additional training set [3].

4.2 Test Datasets

We examine our proposed model with three publicly available datasets: ADC2004, MIREX05, and MIR1k. Due to the limited accessibility to the datasets [1] and the limitation of our model that can handle singing voice only, we test them with several options. Specifically, the ADC2004 dataset includes some instrumental pieces where the melody is played by saxophones or other musical instruments. The MIREX05 dataset we obtained has only 13 out of the total 25 songs. Furthermore, only 9 of the 13 songs contain singing voice. For these reasons, we evaluate our model on all songs and those with singing voices separately for the two sets.

We report various evaluation metrics for melody extraction, including overall accuracy, raw pitch accuracy, raw chroma accuracy, voicing detection rate and voicing false alarm rate. We compute them using *mir_eval* [11], a Python library designed for objective evaluation in MIR tasks.

[1] We downloaded the ADC2004 and MIREX05 datasets from http://labrosa.ee.columbia.edu/projects/melody/ and the MIR1k dataset from https://sites.google.com/site/unvoicedsoundseparation/mir-1k

4.3 Preprocessing

We resample the audio files to 8 kHz and merge stereo channels into mono. We then compute spectrogram with Hann window of 1024 samples and hop size of 80 samples, and finally compress the magnitude by a log scale. Following the strategy in [7], we use only 256 bins from 0 Hz to 2000 Hz where the human singing voices have a relatively greater level than in other frequency bands with regard to background music.

5. EXPERIMENTS

Given the MCDNN model and training data, we conduct several experiments to figure out the effect of different settings in the model. In the followings, we describe options in training the MCDNN and the experiments.

5.1 DNN Training

We configure the DNN to have three hidden layers, each with 512, 512 and 256 units, and ReLUs for the nonlinear function. For the output layer, we use the sigmoid function instead of the softmax function, which is a typical choice in the categorical classification, because the sigmoid slightly worked better in our experiments. Thus, we use binary cross-entropy between the output layer and the one-hot representation of pitch labels as an objective function to minimize. The pitch labels cover from D2 to F#5 in semitone unit. The label vectors are expanded as pitch resolution increases. We initialize the weights with random values from the uniform distribution and optimize the objective function using RMSprop and 20% dropout for all hidden layers to avoid overfitting to the training set. For fast computing, we run the code using Keras [2], a deep learning library in Python, on a computer with two GPUs.

5.2 Context Size

Our model takes multiple frames of spectrogram as input to take contextual information into account. Our first experiment is to figure out an optimal size of the input for different pitch resolutions. For this experiment, we train a single-column DNN using one million examples from the RWC training set. Every training iteration, we randomly select a subset from the pool. We then verify classification accuracy using only voiced frames on the RWC validation set. Figure 2 shows the classification accuracy for a varying size of the spectrogram input. We experimented with multi-frame as inputs of DNN where the input data were taken from N neighbor spectrogram frames. The accuracy progressively increases up to 7 or 9 frames and then converge to a certain level. This is expected because pitch contours of singing voices usually have continuous curve patterns and this temporal features can be captured better by

[2] https://github.com/fchollet/keras

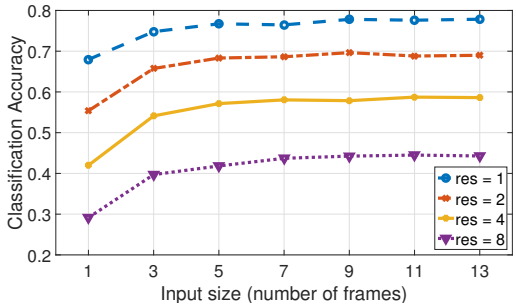


Figure 2: Classification accuracy on the validation set. "res=1" indicates pitch resolution in semitone unit. "res=2", "res=4", and "res=8" indicate progressively higher resolutions than semitone by a factor of 2.

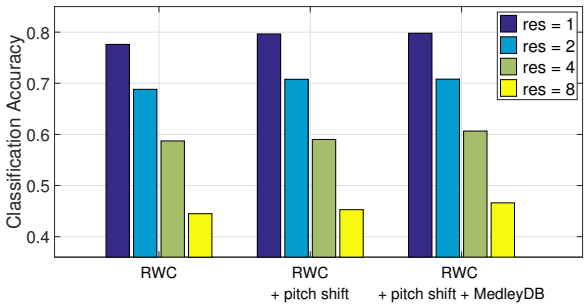


Figure 3: Classification accuracy on the validation set when the pitch-shifted versions of the RWC dataset and 60 vocal songs of the MedleyDB dataset are added in turn to the training set.

taking multiple frames. The result also shows that the validation accuracy is inversely proportional to the pitch resolution. That is, as the resolution increases, the accuracy drops quite significantly. This is also expected because the number of input data per label will decrease given the same training condition and also the accuracy criterion becomes more strict (i.e., slight missing between neighboring pitch labels could have been regarded as a correct prediction). For the following experiments, we fix the input size to 11 frames.

5.3 Data Augmentation

As described in Section 4.1, we augment the training set in two folds. One is by expanding the existing training set using pitch shifting and the other is by making up with another dataset, i.e., 60 songs including singing voices among the MedleyDB dataset. For this experiment, we train a single-column DNN using the increased training pool, specifically, six million examples from the augmented RWC training set and additional 200,000 examples or so from the MedleyDB songs. Again, we verify classification accuracy using only voiced frames on the RWC validation set.

Figure 3 shows the classification accuracy for a varying size of pitch resolution when the pitch-shifted RWC data and MedleyDB data are added to the training data pool in turn. Overall, the accuracy increases by 2 to 3 % with the additional sets. An interesting result is that, with the pitch-shifted data, the accuracy increases more when pitch resolution is low (1 or 2) and, with the additional MedleyDB songs, the accuracy increases more when pitch resolution is high (4 or 8). This is probably because the RWC data is pitch-shifted in semitone units and so technically increases data with low pitch resolutions whereas strong vibrato voices in the opera songs included in the MedleyDB dataset increase data with high pitch resolutions relatively more.

5.4 Single-column vs. Multi-column

As shown in the previous experiments, the classification accuracy is inversely proportional to the pitch resolution in the single-column DNN (SCDNN). That is, as the resolution becomes finer, the classification accuracy decreases, and vice versa. The MCDNN was devised from this empirical result, hoping to achieve both high accuracy and high pitch resolution simultaneously by using the SCDNN with different pitch resolutions together. In this experiment, we validate the idea by comparing the SCDNN and two different combinations of MCDNN. In particular, we evaluate them on the three test sets (ADC2004, MIREX05 and MIR1k), assuming the voiced frames are perfectly detected (the first singing voice detection scenario in Section 3.4).

Figure 4 displays the raw pitch accuracy (RPA) and raw chroma accuracy (RCA). Note that we evaluate the models on the ADC2004 and MIREX05 datasets separately for all songs including instrumental pieces and a subset excluding them (for the latter, the dataset name is suffixed with "vocal"). Overall, the MCDNN improves the melody extraction accuracies. An interesting result is that the MCDNN increases the accuracies on the sets with singing voices quite significantly (about 5 % in RPA and RCA on the MIREX05-vocal) whereas it can be even worse than the SCDNN when instrumental pieces are included. This is actually expected because our model is trained only using voiced frames. This indicates that our model is a specialized melody extraction algorithm that works only on music including singing voices. Comparing the two MCDNN models, there is no significant difference in performance. Thus, the simpler model (the 1-2-4 MCDNN) seems to be a better choice.

5.5 HMM-based Postprocessing

We conduct the Viterbi decoding based on a HMM model for temporal smoothing of the combined prediction. We estimate the prior probabilities and transition matrix from ground-truth of the training set. We then use the prediction of whole tracks as posterior probabilities. Table 1 shows the results as performance increments after applying the Viterbi decoding for the 1-2-4 MCDNN on the test sets.

5.6 A Case Example of Singing Melody Extraction

Our proposed model is capable of predicting temporally smooth pitch contours by using multi-resolution pitch la-

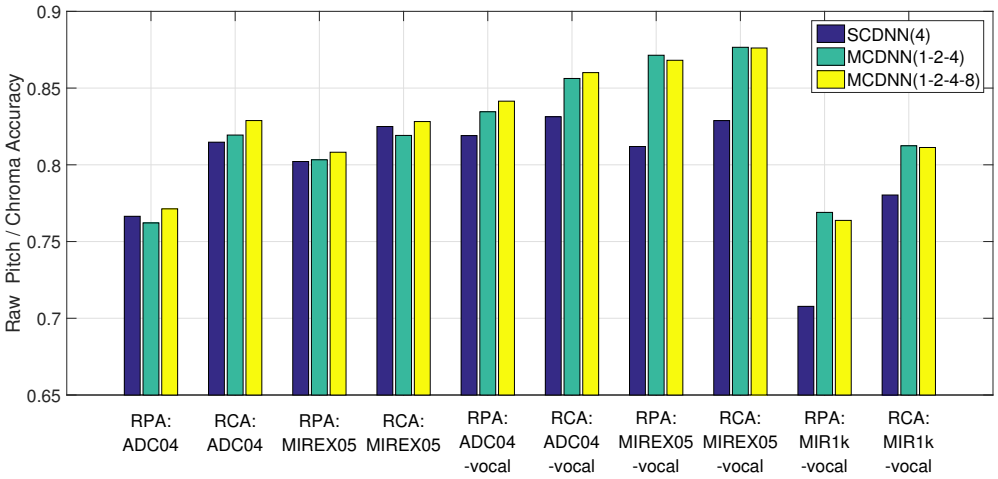


Figure 4: Raw pitch accuracy (RPA) and raw chroma accuracy (RCA) on the ADC2004, MIREX2005 and MIR1K dataset that compare one SCDNN and two different MCDNNs. The "vocal" suffix indicates their subsets that include songs with vocals. Here we assume that we have a perfect voice detector to focus on accuracy on voiced frames.

	without HMM		with HMM	
Dataset	**RPA**	**RCA**	**RPA**	**RCA**
ADC2004	0.749	0.806	0.762	0.816
ADC2004-vocal	0.827	0.852	0.835	0.856
MIREX05	0.801	0.817	0.803	0.817
MIREX05-vocal	0.869	0.875	0.871	0.877
MIR1k	0.766	0.813	0.769	0.813

Table 1: Performance increment by HMM-based smoothing on the 1-2-4 MCDNN.

bels, and the capability is supported more by the augmented datasets. Here we verify it by illustrating an example of singing melody extraction. We selected an opera song from the ADC2004 dataset because the singing voices have dynamic pitch motions such as strong vibrato. Figure 5 shows the results from three different melody extraction models. The left one is from the SCDNN with a pitch resolution of 4 (i.e. 1/4 semitone) and trained only with the RWC dataset. The middle one is from the same SCDNN but trained with additional pitch-shifted RWC dataset and MedleyDB dataset. The right one is from the 1-2-4 MCDNN that has the three pitch resolutions. Comparing the first two models, the additional songs help tracking the vibrato but the second model still misses the whole excursion. With the additional resolutions, the MCDNN makes further improvement, tracking the pitch contours quite precisely.

5.7 Comparison to State-of-the-art Methods

We compare our proposed method with state-of-the-art algorithms on the three test datasets in Table 2. The compared algorithms are all based on pitch saliency [2, 6, 12]. The evaluation metrics include overall accuracy (OA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), voice recall (VR) and voice false alarm (VFA). As mentioned

Algorithm	**OA**	**RPA**	**RCA**	**VR**	**VFA**
Arora [2]	0.690	0.814	0.859	0.765	0.235
Dressler [6]	0.853	0.883	0.889	0.901	0.158
Salamon [12]	0.735	0.763	0.787	0.805	0.151
MCDNN(all)	0.655	0.703	0.759	0.874	0.469
MCDNN(vocal)	0.731	0.758	0.783	0.889	0.412

(a) ADC2004

Algorithm	**OA**	**RPA**	**RCA**	**VR**	**VFA**
Arora [2]	0.634	0.692	0.765	0.810	0.344
Dressler [6]	0.715	0.770	0.806	0.831	0.300
Salamon [12]	0.657	0.676	0.762	0.773	0.263
MCDNN(all)	0.616	0.733	0.752	0.894	0.585
MCDNN(vocal)	0.684	0.776	0.786	0.870	0.490

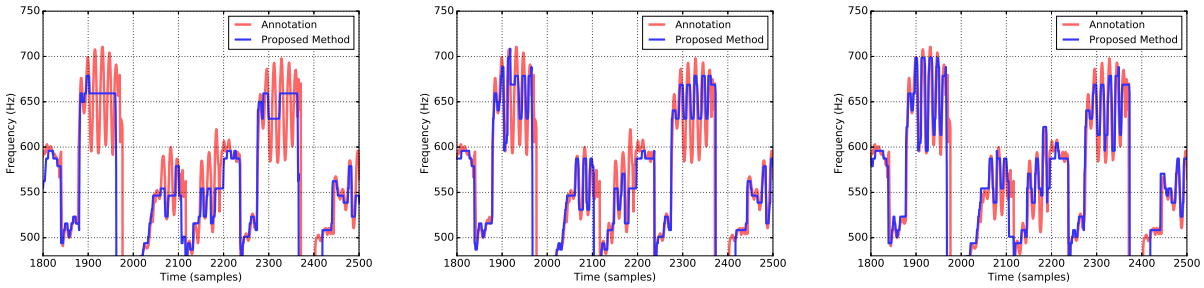
(b) MIREX05

Algorithm	**OA**	**RPA**	**RCA**	**VR**	**VFA**
MCDNN(vocal)	0.613	0.726	0.770	0.934	0.658

(c) MIR-1K

Table 2: Melody extraction results on three test datasets. In this evaluation, we used a simple energy-based voice detector for fair comparison.

in Section 4.2, we have only 13 songs in the MIREX05 dataset. Since we use a simple energy-based voice detector (the second singing voice detection scenario in Section 3.4), the results of our model were not very impressive. However, even with it, the accuracies are quite comparable to some of the algorithms when the test sets include singing vocals. Also, from Figure 4, we can see the RPA and RCA when we have a perfect voice detector. This shows that the accuracies significantly increase, being comparable to the top-notch one.



(a) SCDNN with a pitch resolution of 4 trained with the RWC dataset

(b) SCDNN with a pitch resolution of 4 trained with the additional pitch-shifted RWC dataset and MedleyDB dataset

(c) 1-2-4 MCDNN trained with the additional pitch-shifted RWC dataset and MedleyDB dataset

Figure 5: A case example of melody extraction on an opera song using different models and training data

6. CONCLUSIONS

In this paper, we proposed a novel classification-based melody extraction algorithm on vocal segments using the multi-column deep neural networks. We showed how the data-driven approach can be improved by different settings of the model such as input size, data augmentation, use of multi-column DNN with different pitch resolutions and HMM-based smoothing. The limitation of this model is that it works well only for singing voice because we trained it only with songs where vocals lead the melody. However, this also indicates that our model can be improved to a general melody extractor if a sufficient amount of instrumental pieces are included in the training sets. We compared our model to previous state-of-the-arts. Since we used a simple energy-based singing voice detector, the performance of our model has limitations. However, the results show that, with a better voice detector, our model can be improved further.

7. ACKNOWLEDGMENT

This work was supported by Korea Advanced Institute of Science and Technology (Project No. G04140049), National Research Foundation of Korea (Project No. N01150671) and BK21 Plus Postgraduate Organization for Content Science.

8. REFERENCES

[1] Forest Agostinelli, Michael R Anderson, and Honglak Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *Advances in Neural Information Processing Systems 26*, pages 1493–1501, 2013.

[2] Vipul Arora and Laxmidhar Behera. On-line melody extraction from polyphonic audio using harmonic cluster tracking. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(3):520–530, 2013.

[3] Rachel M Bittner, Justin Salamon, Slim Essid, and Juan P Bello. Melody extraction by contour classification. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, 2015.

[4] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[5] Roger B Dannenberg, William P Birmingham, George Tzanetakis, Colin Meek, Ning Hu, and Bryan Pardo. The musart testbed for query-by-humming evaluation. *Computer Music Journal*, 28(2):34–48, 2004.

[6] Karin Dressler. An auditory streaming approach for melody extraction from polyphonic music. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR*, pages 19–24, 2011.

[7] Daniel P. W. Ellis and Graham E Poliner. Classification-based melody transcription. *Machine Learning*, 65(2):439–456, 2006.

[8] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval, ISMIR*, pages 287–288, 2002.

[9] Jean Laroche. *Applications of Digital Signal Processing to Audio and Acoustics*, chapter Time and Pitch Scale Modification of Audio Signals, pages 279–309. Springer US, Boston, MA, 2002.

[10] Graham E Poliner, Daniel P. W. Ellis, Andreas F Ehmann, Emilia Gómez, Sebastian Streich, and Beesuan Ong. Melody transcription from music audio:

Approaches and evaluation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(4):1247–1256, 2007.

[11] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir eval : A transparent implementation of common mir metrics. In *Proceedings of the 15th International Conference on Music Information Retrieval, ISMIR*, 2014.

[12] Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(6):1759–1770, 2012.

[13] Justin Salamon, Eva Gomez, Daniel P. W. Ellis, and Gael Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *Signal Processing Magazine, IEEE*, 31(2):118–134, 2014.

[14] Justin Salamon, Bruno Rocha, and Emilia Gómez. Musical genre classification using melody features extracted from polyphonic music signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

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

[16] Joan Serra, Emilia Gómez, and Perfecto Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, pages 307–332. Springer, 2010.