A Multi-task Learning Approach for Melody Extraction

Zhengyu Cao[1], Xiangyi Feng[2], and Wei Li[1,2]

[1] School of Computer Science, Fudan University, Shanghai, China, 201203
[2] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China, 201203

Abstract. Melody extraction aims to produce a sequence of frequency values corresponding to the pitch of the dominant melody from a musical recording, comprising a large variety of algorithms spanning a wide range of techniques. In this paper, a novel DNN-LSTM based architecture is proposed for melody extraction. Melody extraction is regard as a composition of pitch estimation and voicing detection. This paper present a multi-task learning approach so as to perform the two tasks simultaneously, which proves to help the model obtain higher accuracy and better generalization ability. Experiments on public datasets show that the proposed model is capable of modeling temporal dependencies, and have a comparable result to the state-of-the-art methods.

Keywords: Melody extraction, Multi-task learning

1 Introduction

Melody is one of the most important factors in music. In recent years, the extraction of melody has received substantial attention from music information retrieval (MIR) communities, comprising a large variety of algorithms spanning a wide range of techniques, such as query-by-humming (QBH) [1], cover song identification [2], music transcription [3] and music structuring [4], etc. According to Poliner et al. [5], melody is the single (monophonic) pitch sequence that a listener might reproduce when asked to hum or whistle a polyphonic piece of music, and that a listener would recognize as being the essence of that music when heard in comparison. In general, the most dominant pitch sequence is considered as the melody.

Despite the variety of proposed approaches, melody extraction remains highly challenging. Apart from overlapping harmonics and high degrees of polyphony in polyphonic music, one of the complexities and challenges of this task is to determine when the melody is present and when it is not. In the context of melody extraction in polyphonic music, there are primarily two major tasks in the proposed system, i.e., 1) pitch estimation (to estimate the pitch of the melody) and 2) voicing detection (to identify the presence or absence of the melody). Almost all existed melody extraction methods focus on the former task to date. To solve the latter one, generally a subsequent voicing detection step is included.

H. Li et al. (eds.), *Proceedings of the 7th Conference on Sound and Music Technology (CSMT)*, Lecture Notes in Electrical Engineering 635,
https://doi.org/10.1007/978-981-15-2756-2_5

The most common approach is to use static or dynamic thresholds on energy or salience, often involving careful parameter tuning in order to reach peak performance [6–8]. Salamon & Gómez [9] defined a set of contour characteristics and devised rules to distinguish between melodic and non-melodic contours by exploiting their distributions. Bittner et al. [10] perform voicing detection by setting a threshold on the contour probabilities produced by the discriminative model.

Furthermore, since collections of royalty-free music recordings that can be shared for research purposes are relatively scarce, and on the other hand the annotation process for melody is difficult and tedious, there are few public datasets for melody extraction, making this task even more difficult. Most existing deep learning approaches, focusing on pitch estimation, only make use of nearly half samples of the datasets, for which melody is present. Park & Yoo [11] quantize melody pitches such that each pitch corresponds to a pitch class, and assign region without melody to a special pitch class. In this way, voicing detection is performed by comparing the special pitch class with the pitch class, however, such kind of strategy is likely to suffer data imbalance theoretically in the case when not enough training samples are available.

In this paper, a novel yet simple DNN-LSTM based architecture is proposed for melody extraction, which is capable of modeling temporal dependencies and has comparable result to the state-of-the-art methods. Differing from most existing methods, we regard melody extraction as a composition of two subtasks, i.e., pitch estimation and voicing detection. To extract melody for polyphonic music, we present a multi-task learning approach so as to perform the two tasks simultaneously. By jointly training them, we show that the model can obtain higher accuracy and better generalization ability. For reproducibility, we will share the source code at https://github.com/beantowel/Lab_MelExt.

2 Related Works

2.1 Deep Learning for Melody Extraction

More recently, deep learning has shown great advantages and potential, and has been successfully adopted in many areas. To our best knowledge, however, as for melody extraction, there is little work exploring such kind of techniques in the literatures. In these methods, the melody extraction problem is viewed as a classification problem by categorizing melody pitches into a finite set of pitch labels. Kum et al. [6] present a classification-based approach for melody extraction on vocal segments using multi-column deep neural networks, each of the networks is trained to predict a pitch label with different pitch resolutions and their outputs are combined to infer the final melody contour. Bittner et al. [12] use a fully convolutional neural network to learn salience representations from harmonic constant-Q transform representations of music signal and estimate the melody line from those representations by choosing the frequency bin with the maximum salience for each frame. Park & Yoo [11] propose a long short-term memory recurrent neural network (LSTM-RNN) for extracting melody, which

is considered capable of representing the dynamic variations in melody pitch sequence. This paper treats the melody extraction as a frame-level classification problem similar to the above works, by adopting a multi-task approach, the proposed method skipped hard-coded post processing steps widely used in related works to detect voicing frames, replacing it with a deep-learning counterpart.

2.2 Multi-task Learning

In multi-task learning, more than one loss function, corresponding to the 'tasks', is optimized simultaneously. It has been used across lots of applications of machine learning, from natural language processing [13] and speech recognition [14] to computer vision [15]. Multi-task learning can be viewed as a form of inductive transfer, which help improve a model by introducing an inductive bias and thereby cause a model to prefer some hypotheses that explain more than one task. In general, there are two common approaches to perform multi-task learning in the context of deep learning, hard parameter sharing and soft parameter sharing. In hard parameter sharing, some hidden layers are shared among all tasks, while some layers including output layers are task-specific. In soft parameter sharing, different models are trained for different tasks with some constraints introduced, such as the regularized distance between the parameter of the models. For more details, an overview of multi-task learning in deep neural networks has been presented by Ruder [16]. The proposed model was based on hard parameter sharing to get a representation useful for both pitch estimation and voicing detection tasks.

3 Proposed Method

3.1 Architecture

As aforementioned, melody extraction consists of two primary subtasks, pitch estimation and voicing detection. In this paper, we believe that the two tasks are closely related in some way. Therefore, by constructing a joint two-task model, not only can we obtain predictions for both tasks at once, but also they likely benefit each other so that a much more explicit and generalized model is possible.

The proposed method extract melody in a frame-wise manner, by predicting instant frequency every 23 milliseconds. For pitch estimation, we treat it as a classification task, as done by Kum et al. [6], Bittner et al. [12] , Park & Yoo [11], Ellis & Poliner [17]. A pitch range of nearly 5 octaves from 55Hz to 1.76kHz are taken into account, i.e., from A1 to A6, so the model can capture most of the pitches in vocal melody and keep the data distribution of pitch classes roughly balanced. For voicing melody frames, the algorithm is expected to return a frequency value matching the ground truth, which is considered correct if it is within 50 cents (i.e., half a semitone) to the ground truth [9]. Instead of the 50-cents interval used by the evaluation metric, we adopt smaller pitch class interval in the model output, since we focus on vocal melody whose pitch curves

are smooth and continuous due to natural singing styles such as pitch transition patterns or vibrato [18]. An 1/9 semitone resolution is used, which results in 540 pitch classes in total for the pitch estimation output. Given a ground truth instant frequency f Hz, its corresponding pitch class $Clz(f)$ is calculated as:

$$Clz(f) = \lfloor N_{pitchClass} \cdot log_{\frac{f_{high}}{f_{low}}}(\frac{f}{f_{low}}) \rfloor \tag{1}$$

where $N_{pitchClass}$ is the total number of pitch classes, and $[f_{low}, f_{high})$ corresponds to the range of pitch frequency, mapping to $[0 : N_{pitchClass})$, the range of pitch classes. Frequencies out of range were rectified to maximum or minimum values.

The voicing detection determines whether the melody is present or not at some point as a binary classification task. Notice that there are more classification classes for pitch estimation than those for voicing detection, the former task is considered relatively complicated. As a result, pitch estimation is taken as a main task, with voicing detection taken as an auxiliary task. The architecture of multi-task DNN-RNN model and a simpler DNN-only variant is illustrated in Figure 1, based on hard parameter sharing. The model takes several frames from the STFT of the input audio as inputting feature vector, and outputs two labels for each timestamp, corresponding to the predictions of pitch estimation and voicing detection respectively. Then the two labels are merged so as to obtain the melody prediction at the last stage. The lower layers, i.e., layers of deep neural network (DNN), are shared across both tasks, while top layers, i.e., long short term memory networks (LSTMs), are task-specific.

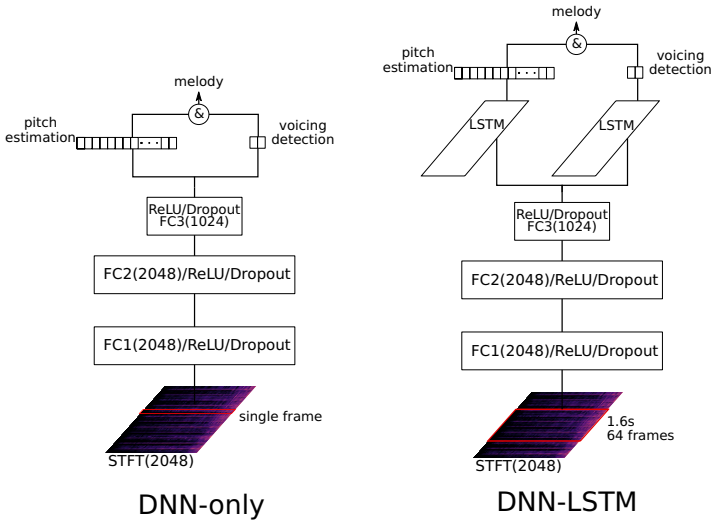

Fig. 1. The architecture of DNN-only and DNN-LSTM model for melody extraction.

In the model, DNN plays a role for representation learning where three hidden layers are used, with 2048, 2048 and 1024 units and ReLUs [19] are adopted as the nonlinear function. While the DNN takes only one frame from the STFT spectrogram for one-off prediction independently, thereby no contextual information is taken into account, which is demonstrated to be important for not only pitch estimation but also voicing detection. To capture temporal dependencies, LSTMs take the output of the last hidden layer of DNN as input, i.e., the representation learnt by DNN, which is supposed to benefit all relevant tasks. Proverbially, Long Short Term Memory networks (LSTMs) [20] are a special kind of RNN, which is capable of learning long-term dependencies and connecting previous information to the present task, and they work tremendously well on a large variety of problems. For each task (pitch estimation and voicing detection), a 2-layer bi-LSTM with 64 units in the cell is trained, thus the model will have an input vector of shape (64, 2048), capturing a $\frac{63\times512+2048}{22050} \approx 1.6$s long audio clip with the sampling rate set to 22050Hz, the STFT window set to 2048 frames long and the hop size set to 512 frames long. The last two fully connected layers will make predictions for pitch estimation and voice detection tasks respectively. To integrate the two predicted results for melody extraction, supposing we have predicted the labels for the two tasks, $label_{pe} = Clz(f) \in [0 : N_{pitchClass})$ for pitch estimation, where function Clz from Equation 1 mapped the frequency to pitch classes and $label_{vd} \in \{0, 1\}$ for voicing detection, where 0 indicates absence of melody and 1 indicates presence of melody, then melody extracted is given by:

$$instantFreq = \begin{cases} Clz^{-1}(label_{pe}), & \text{if } label_{vd} = 1. \\ 0, & \text{if } label_{vd} = 0. \end{cases} \tag{2}$$

3.2 Multi-task Loss

There are two output layers in the proposed system, one for pitch estimation and the other for voicing detection. The output layers produce probabilities over the pitch/voice classes, thus we adopt cross entropy losses by converting ground truth melody label to one-hot vector for each frame. Furthermore, a blurring technique is used as described by Bittner [12], which add a smoothing term to the one-hot vector for pitch estimation making it obey a rectified discrete Gaussian distribution centered at target pitch class with a quarter-semitone deviation σ. For non-voice frames, the target vector for pitch estimation is set to have a small value ϵ on all pitch classes. These method would help in training since the model have a high semitone resolution, nearby frequencies still make a good prediction for the melody extraction task, and they better utilise non-voice data for pitch estimation task. To balance the number of terms in loss functions involved in $loss_{pe}$, loss for pitch estimation, and $loss_{vd}$, loss for voicing detection, we multiply $loss_{vd}$ by a factor of semitone resolution N_{se}. Also, l_2 norm penalty with $\lambda = 10^{-4}$ is included to alleviate over-fitting. Multi-task loss used in the proposed model is shown in (6), where y is the target vector for pitch estimation, y_j corresponds to the component for jth pitch class, x_{pe} is the output for pitch

estimation, z is the one-hot target vector for voice detection, x_{vd} is the output for voice detection and parameter $\alpha \in [0, 1]$ is an auxiliary weight, which controls how cross entropy loss of the auxiliary task (voicing detection) plays a part in the global loss, i.e., 0 means only pitch estimation is concerned while 1 means the two tasks make equivalent effect. Since the losses were computed for each frame, we need to sum it up in the end.

$$y_j = \begin{cases} e^{-\frac{(j-label_{pe})}{2\sigma^2}}, & \text{if } label_{vd} \geq 0 \text{ and } |j - label_{pe}| \leq \frac{N_{se}}{2}. \\ \epsilon, & \text{if } label_{vd} < 0. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

$$\mathbf{loss_{pe}} = -\sum_j \mathbf{y_j} \cdot log(Softmax(\mathbf{x_{pe}})_j) \tag{4}$$

$$\mathbf{loss_{vd}} = -N_{se} \sum_{j=0}^{1} \mathbf{z_j} \cdot log(Softmax(\mathbf{x_{vd}})_j) \tag{5}$$

$$L = sum\{\mathbf{loss_{pe}}\} + \alpha \cdot sum\{\mathbf{loss_{vd}}\} + \lambda \cdot \|\mathbf{w}\|_2 \tag{6}$$

3.3 Training

Since in the experiments training DNN and RNN jointly does not result in good convergence, a strategy of training DNN and RNN separately is adopted. In order to learn a representation with DNN that is helpful for both pitch estimation and voicing detection, we first train a simplified version of the model by removing LSTM layers and directly feed the DNN output to the last fully connected layers, using the same loss as Equation 6. Then we train the full model setting the parameters fixed except for those of LSTMs, i.e., using a pre-trained DNN when training LSTM layers. This can be viewed as a transfer learning technique.

All networks are trained using gradient descent with learning rate 10^{-4} and 70% dropout [21] for all hidden layers to alleviate over-fitting. The model with is implemented using Pytorch [22], an open-source software library for machine learning, and run on a computer with a single GPU with 8GB memory.

3.4 Inferencing

The proposed model takes multiple frames as input to capture temporal dependencies, original STFT spectrogram were segmented into groups of frames in a similar pattern used when slicing samples to frames in STFT. Frames in different groups will overlap in time axis, thus we use a vote mechanism when generating final outputs. As depicted in Figure 2, at timestamp t, if the count of non-voice predictions is greater equal then that of voice, the output will be non-voice, otherwise the median frequency of the predictions will be used as the pitch estimation output.

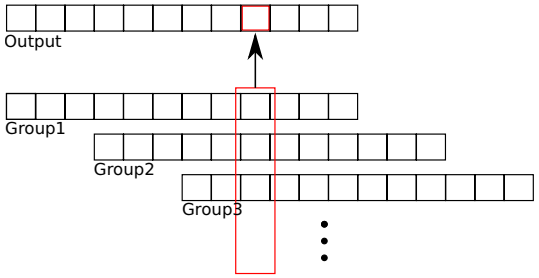

Fig. 2. Overlapping predictions vote for final output.

4 Experiments

4.1 Datasets

Training and Validation Datasets This work use the MedleyDB [23] dataset in the training phase, which contains 108 songs with MELODY2 annotation in total. It covers a wide range music styles including classical, jazz, rock and pop. Among them, 47 songs is instrumental music, the other 61 songs is vocal. Since instrumental music has clearly different characteristics from vocal music and the task of distinguishing the instrumental melody from the accompaniment is more complicated comparing to the vocal music, this work focus on melody extraction for vocal music, where singing voice is regarded as a main source of the melody, and thereby voicing detection here de facto is semantically equivalent to singing voice detection (SVD). The 61 vocal songs from MedleyDB is divided into two sets by a random split, 90% of its songs makes up the training set, the other 10% forms the validation set.

Test Datasets ADC2004[3] and MIREX-05[3] datasets are used for test. The datasets consists of 20 and 13 audio clips across different music styles respectively. Among them, all the 12 and 9 clips carrying vocal melody are used in the experiments.

4.2 Metrics

Following Poliner et al. [5], several evaluation metrics for melody extraction, including voicing recall rate (VR), voicing false alarm rate (VFA), raw pitch accuracy (RPA), raw chroma accuracy (RCA) and overall accuracy (OA) are used as a measure of performance. They are defined as follows.

1. VR: The proportion that a frame which is truly voiced is labeled as voiced.
2. VFA: The proportion that a frame which is not actually voiced is labeled as voiced.

[3]https://labrosa.ee.columbia.edu/projects/melody/

3. RPA: The proportion of voiced frames for which pitches are considered correct.
4. RCA: The proportion of voiced frames for which chroma are considered correct (octave errors are ignored).
5. OA: The proportion of all frames correctly estimated by the algorithm, including both pitch and voicing.

4.3 Effects of Auxiliary Weight for Multi-task Learning

As mentioned before, the auxiliary weight α in this work controls how much the loss of the auxiliary task (voicing detection) contributes to the total loss of the model during training. To test the effects of multi-task approach and figure out an optimal value of auxiliary weight, the experiment verify pitch accuracy (PA)[4], voicing accuracy (VA)[5] and overall accuracy with α range from 0.1 to 0.9 with step size 0.2. In addition, these metrics are computed on the simplified DNN-only model, i.e., no temporal information and LSTM layers is taken into account for faster training. All hyper-parameters except the auxiliary weight remain unchanged. The models with highest OA (i.e., top model for melody extraction) on validation dataset within 10 training epochs is evaluated for each auxiliary weight $\alpha > 0$, since we expect a model for melody extraction ultimately and their PA, VA as well as OA are presented in Table 1. Note that when α is set to 0, only the pitch estimation task is concerned, thus VA and OA do not make sense any more in this case. A DNN model for voicing detection only is also evaluated, by removing the first term in the loss function as Equation 6 while setting α to 1, denoted in the table as N/A. Similarly, in this case, PA and OA should be ignored. Therefore, when α is 0 or N/A, it is actually training a single-task model for either pitch estimation or voicing detection, and when α is larger than 0, a joint two-task model is used.

Table 1. Top models for melody extraction with different auxiliary weights and their accuracies, values labeled with * should be ignored.

α	PA	VA	OA
0	**0.7398**	0.6285*	0.4726*
0.1	0.6288	0.7516	0.7105
0.3	0.6308	0.7529	0.7118
0.5	0.6311	0.7554	**0.7139**
0.7	0.6443	**0.7606**	0.712
0.9	0.6432	0.7597	0.7128
N/A	0.0012*	0.7424	0.2973*

[4]mean raw pitch accuracy of voicing frames
[5]mean voicing accuracy of all frames

In Table 1, the model reaches the best performance in terms of OA when the auxiliary weight is set to 0.5. Comparing to single-task learning (α is 0 or N/A), multi-task learning (α ¿ 0) shows a relatively clear performance boost for voicing detection, though there are no significant differences in the performance for different settings of α when α is greater than 0. As is expected, the integration with the auxiliary task (voicing detection) can benefit the melody extraction task to some extent. Consequently, α is set to 0.5 and the model with the highest OA is used for the following comparison.

4.4 Comparison with State-of-the-Art Methods

The proposed model is trained on the MedleyDB [23] dataset and compared to 4 classical and state-of-the-art methods on 2 test datasets described in Section 4.1. Five evaluation metrics for melody extraction mentioned before are computed using mir_eval [24]. Among the compared algorithms, Salamon [9][6] is based on salience function while Bittner [12][7], Hsieh [25][8] and Lu [26][9] is based on deep neural networks, using a data-driven approach. Proposed model is denoted as MultiDNN and MultiDNN_RNN, the former is the simplified model without LSTM layers, and the latter is the full model.

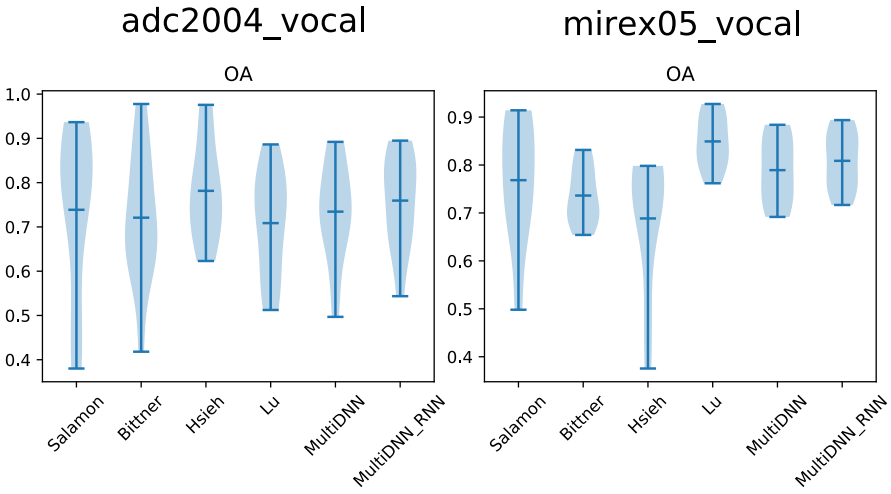

Fig. 3. OA distribution of vocal melody extraction results.

[6]https://github.com/justinsalamon/audio_to_midi_melodia
[7]https://github.com/rabitt/ismir2017-deepsalience
[8]https://github.com/bill317996/Melody-extraction-with-melodic-segnet
[9]https://github.com/s603122001/Vocal-Melody-Extraction

Table 2. Comparison of vocal melody extraction results.

Adc2004(vocal)					
Method	VR	VFA	RPA	RCA	OA
Salamon	81.5%	12.1%	77.8%	80.8%	73.9%
Bittner	83.0%	36.9%	81.3%	**85.8%**	72.1%
Hsieh	96.5%	45.9%	**83.9%**	85.3%	**78.1%**
Lu	73.9%	2.4%	67.0%	69.3%	70.9%
MultiDNN	77.8%	11.8%	71.4%	74.8%	73.4%
MultiDNN_RNN	83.7%	17.8%	74.9%	77.8%	75.9%
MIREX-05(vocal)					
Method	VR	VFA	RPA	RCA	OA
Salamon	87.0%	22.8%	80.4%	81.7%	76.8%
Bittner	79.1%	24.5%	80.1%	82.3%	73.6%
Hsieh	94.8%	42.2%	75.1%	76.4%	68.9%
Lu	87.4%	7.9%	**80.9%**	**82.5%**	**84.9%**
MultiDNN	75.6%	7.5%	70.4%	71.9%	78.9%
MultiDNN_RNN	79.0%	7.3%	73.2%	74.3%	80.9%

Figure 3 shows the min, max, mean of vocal melody extraction results based on songs(not frames), the results is slightly different from that of the original papers. For producing comparable evaluations, the code available online is used to calculate the results using the same evaluation program instead of citing original papers. As shown in Table 2, on average, the proposed model reaches high performance on par with the top-notch algorithms. Note that for the models evaluated using deep learning methods, Lu [26] and Hsieh [25] trained on MedleyDB and MIR-1K, while Bittner [12] trained on a subset of MedleyDB. Thus evaluation results on Adc2004 and MIREX-05 datasets are comparable. Among the five metrics, RPA and RCA measure the performance of the pitch estimation task while VA and VFA measure the performance of the voicing detection task. For another thing, OA combines the performance of both tasks to give an overall performance score for the system. As a consequence, comparison merely focus on the metric OA in this part. The proposed model is second only to Hsieh for Adc2004 in terms of OA, as for MIREX-05, the proposed model is second only to Lu. However, top methods on one test set was outperformed by the proposed model on the other test set, so the model has better generalization ability. It is worth mentioning that the architecture in this work is fairly simple and during training, only the MedleyDB training set is used without any additional datasets or data augmentation, which proves to be capable of improving performance in Kum [6].

5 Conclusions and Discussions

In this paper, we propose a novel DNN-LSTM based architecture for melody extraction. Notice that melody extraction can be decomposed into pitch estimation and voicing detection, we adopt the idea of multi-task learning and present a multi-task learning approach for this task, so as to perform pitch estimation and voicing detection simultaneously. Experiments show that by joint training the two tasks, the model can obtain higher accuracy and has better generalization ability. In addition, we show how the auxiliary weight takes effect in this work. Although the architecture is fairly simple and no more additional data or data augmentation is involved, the proposed model reach comparable performance to the state of the art. By expanding train dataset or increasing the complexity of the architecture, the proposed approach can be further improved.

6 Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant 61671156).

References

1. Justin Salamon, Joan Serrà, and Emilia Gómez. Tonal representations for music retrieval: From version identification to query-by-humming. *International Journal of Multimedia Information Retrieval*, 2(1):45–58, March 2013.
2. Joan Serrà, Emilia Gómez, and Perfecto Herrera. Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond. In Janusz Kacprzyk, Zbigniew W. Raś, and Alicja A. Wieczorkowska, editors, *Advances in Music Information Retrieval*, volume 274, pages 307–332. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
3. Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: Challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, December 2013.
4. Geoffroy Peeters. Sequence Representation of Music Structure Using Higher-Order Similarity Matrix and Maximum-Likelihood Approach. In *ISMIR*, pages 35–40, Vienna, Austria, 2007.
5. Graham E. Poliner, Daniel P. W. Ellis, Andreas F. Ehmann, Emilia Gomez, Sebastian Streich, and Beesuan Ong. Melody Transcription From Music Audio: Approaches and Evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1247–1256, May 2007.
6. Sangeun Kum, Changheun Oh, and Juhan Nam. Melody Extraction on Vocal Segments Using Multi-Column Deep Neural Networks. In *ISMIR*, pages 819–825, New York City, USA, 2016.
7. Benoit Fuentes, Antoine Liutkus, Roland Badeau, and Gael Richard. Probabilistic model for main melody extraction using Constant-Q transform. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5357–5360, Kyoto, Japan, March 2012. IEEE.

8. Karin Dressler. Towards computational auditory scene analysis: Melody extraction from polyphonic music. *Proc. CMMR*, pages 319–334, 2012.
9. Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.
10. Rachel M. Bittner, Justin Salamon, Slim Essid, and Juan Pablo Bello. Melody Extraction by Contour Classification. In *ISMIR*, pages 500–506, Malaga, Spain, 2015.
11. Hyunsin Park and Chang D. Yoo. Melody extraction and detection through LSTM-RNN with harmonic sum loss. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2766–2770, New Orleans, LA, March 2017. IEEE.
12. Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello. Deep Salience Representations for F0 Estimation in Polyphonic Music. In *ISMIR*, pages 63–70, Suzhou, China, 2017.
13. Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), June 5-9, 2008*, pages 160–167, Helsinki, Finland, 2008.
14. Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603, Vancouver, Canada, 2013. IEEE.
15. Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, Santiago, Chile, 2015.
16. Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv:1706.05098 [cs, stat]*, June 2017.
17. Daniel P. W. Ellis and Graham E. Poliner. Classification-based melody transcription. *Machine Learning*, 65(2):439–456, December 2006.
18. Sangeun Kum and Juhan Nam. Joint Detection and Classification of Singing Voice Melody Using Convolutional Recurrent Neural Networks. *Applied Sciences*, 9(7):1324, March 2019.
19. Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, Haifa, Israel, 2010.
20. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
21. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
22. Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, page 4, Long Beach, CA, USA, 2017.
23. 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, Taipei, Taiwan, 2014.
24. Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C. Colin Raffel. Mir_eval: A transparent implementation of common MIR metrics. In *In Proceedings of the 15th International Society*

for Music Information Retrieval Conference, ISMIR, Taipei, Taiwan, 2014. Citeseer.
25. Tsung-Han Hsieh, Li Su, and Yi-Hsuan Yang. A Streamlined Encoder/Decoder Architecture for Melody Extraction. *arXiv:1810.12947 [cs, eess]*, October 2018.
26. Wei Tsung Lu and Li Su. Vocal Melody Extraction with Semantic Segmentation and Audio-symbolic Domain Transfer Learning. In *ISMIR*, pages 521–528, Paris, France, 2018.