VOCAL MELODY EXTRACTION WITH SEMANTIC SEGMENTATION AND AUDIO-SYMBOLIC DOMAIN TRANSFER LEARNING

Wei-Tsung Lu and Li Su
Institute of Information Science, Academia Sinica
s603122001@gmail.com, lisu@iis.sinica.edu.tw

ABSTRACT

The melody extraction problem is analogue to semantic segmentation on a time-frequency image, in which every pixel on the image is classified as a part of a melody object or not. Such an approach can benefit from a signal processing method that helps to enhance the true pitch contours on an image, and, a music language model with structural information on large-scale symbolic music data to be transfer into an audio-based model. In this paper, we propose a novel melody extraction system, using a deep convolutional neural network (DCNN) with dilated convolution as the semantic segmentation tool. The candidate pitch contours on the time-frequency image are enhanced by combining the spectrogram and cepstral-based features. Moreover, an adaptive progressive neural network is employed to transfer the semantic segmentation model in the symbolic domain to the one in the audio domain. This paper makes an attempt to bridge the semantic gaps between signal-level features and perceived melodies, and between symbolic data and audio data. Experiments show competitive accuracy of the proposed method on various datasets.

1. INTRODUCTION

Melody extraction of polyphonic music has been accounted a key towards bridging the semantic gap in music processing, as melody is an intermediate object that correlates to both low-level signal attributes such as pitch and high-level semantics, i.e. the difference between melody and accompaniment, of music [3, 12, 29]. However, it is challenging because the notion of melody is complicated by two levels of information extraction and data modalities. For information extraction, both pitch detection and *semantic segmentation* levels are required to specify the position and shape of a melody out of other pitch contours in a time-frequency representation. As to data modalities, the problem arises from the difference of melody-related features between the *composed* data (e.g., symbolic data such as MIDI) and the *performed* data (e.g., audio data): the former provides structural information such as voiced/unvoiced segments and chord/non-chord notes, while the latter provides interpretational information such as sliding and vibrato. Both kinds of information are essential for accurately identifying the melody pitch contour.

We perform vocal melody extraction using semantic segmentation techniques. Semantic segmentation partitions an image into semantically meaningful objects with precise boundaries. Rendered as a pixel-wise classification problem and able to be implemented by an encoder-decoder network with 2-D convolutional feature mappings, it brings great success in computer vision [6,7,14,25]. Semantic segmentation also makes a breakthrough in solving the source separation problem in music processing [17], which analogously needs to resolve components coexisting in a time-frequency image. In this work, a deep convolutional neural network (DCNN) is adopted with dilated convolution for semantic segmentation as it achieves better performance in multi-resolution images.

To fully utilize the advance of semantic segmentation in vocal melody extraction, we further attend to the aforementioned issues, pitch detection and multiple data modalities, both of which are absent from typical image-based semantic segmentation. For pitch detection, we notice that when performing melody extraction with semantic segmentation, the spectrogram is usually suboptimal since it captures the harmonic peaks and information unrelated to the melody, which accounts for one of the major errors among all the melody extraction methods. This issue is addressed by modifying the spectrogram with cepstral-features, which results in a novel time-frequency representation that enhances the true pitch contour while also suppresses harmonic contours [26, 32].

The modality difference between symbolic and audio data is relatively less noticed in melody extraction. We address this issue with transfer learning: we first train a melody extraction model with symbolic data, and the model parameters are then reused in the vocal melody extraction model trained with audio data. In this way, the symbolic-based model assists in music language modeling that audio-based models may fall short of. Incorporating symbolic music data is of great potential to mitigate the data scarcity problem, since building a symbolic dataset with melody annotations is much easier than building an audio one, and it is also very straightforward to perform data augmentation on symbolic data. In this work, we adopt the *progressive neural network* (PNN) [1], a network structure providing cross-domain network parameter shar-

Attribution: Wei-Tsung Lu and Li Su. “Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning”, 19th International Society for Music Information Retrieval Conference, Paris, France, 2018.

ing to accomplish symbolic-audio transfer learning task.

To sum up, this paper attempt to apply image semantic segmentation to vocal melody extraction, forming a systematic method to perform singing voice activity detection, pitch detection and melody extraction all at the same time. This segmentation method gives competitive results on pitch accuracy, and even works unprecedentedly well on singing voice activity detection compared to other deep-learning-based methods. With the integration with the PNN, we leverage large-scale symbolic data to train the model, and attain similar performance to the segmentation method with less training time.

2. RELATED WORK

Melody extraction of polyphonic music has been widely investigated with various signal processing and machine approaches. Recent works using convolution neural networks (CNNs) or recurrent neural networks (RNNs) are mostly classification-based, where the output is the frame-level likelihood score of every pitch at a time instance [4,22,27,30,37]. [2] adopts a fully convolution neural network and output a salience representation at the song level. Advanced semantic segmentation networks such as the U-net [28] have been utilized in source separation [17] and shows high potential in melody extraction.

Most of the melody extraction studies focus on the signal processing level, possibly because signal-level characteristics such as slides and vibrato are still the principal factors in recognizing a melody contour. In contrast, melody extraction on symbolic data is rarely discussed in the literature. Although not the main topic of this work, we manage to pose the problem of *symbolic melody extraction* and emphasize its importance in music language modeling for cross-domain transfer learning.

Previous works on transfer learning for music information retrieval mostly aim under the same type of input data representation [8,13]. Contrarарily, transfer learning across the data from different *domains*, such as adapting a model learned from symbolic data to another learned from audio data, is relatively less discussed. Previous works dealing with cross-domain data mainly focus exploring audio-to-MIDI or audio-to-sheet correspondence [10,11].

3. METHOD

An overview of the proposed model is shown in Fig.1. The model contains a feature extractor which computes the audio data representation and a PNN which consists of two segmentation models, with one trained on the symbolic data and the other on the audio data. The filter is for dimension reduction of the audio representation to fit the symbolic segmentation model in the PNN. Details of the model are discussed below.

3.1 Audio data representation

In music processing, designing a data representation suitable for the machine learning models to better identify and capture the information of interest can help significantly improve the performance [18]. In the task of pitch detection in polyphonic music, related methods include the feature scaling [18], the harmonic constant-Q transform (HCQT) that combines the CQTs based on different octave numbers [2], the combined frequency and periodicity (CFP) representation that intergrates a temporal or spectral representation with its Fourier dual [26, 32, 34], and others. All of these methods are designed to emphasize the saliency of pitch contours in the music signal.

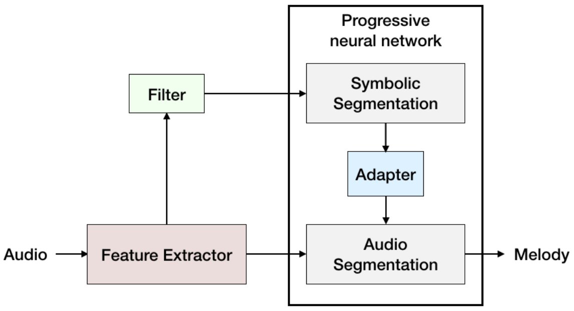

Figure 1: The system diagram of the proposed method.

We adopt the data representation used in [33], which has been shown effective in enhancing the true pitch components of polyphonic signals. The adopted data representation is essentially the product of a *generalized cepstrum* (GC), a classical time-based pitch detection function [16,20,21,35,36], and a *generalized cepstrum of spectrum* (GCoS), a modified spectrum lying in the frequency domain [32]. The GC and GCoS are complementary: a GCoS reveals the presence of a pitch object by its fundamental frequency (f_0) and harmonics (nf_0), while a GC reveal it by its f_0 and sub-harmonics (f_0/n) [26, 32, 34]. By simply multiplying GC by GCoS, we effectively suppress the harmonic and sub-harmonic peaks, and at the same time localize a pitch object.

The GC and GCoS are both computed by the discrete Fourier transform (DFT) and nonlinear activation functions. Consider an input signal $\mathbf{x} := \mathbf{x}[n]$ where n is the index of time. Let the magnitude of the short-time Fourier transform (STFT) of $\mathbf{x}$ be $\mathbf{X}$. Given an N-point DFT matrix $\mathbf{F}$, high-pass filters $\mathbf{W}_f$ and $\mathbf{W}_t$ for eliminating the DC terms, and activation functions σ_i, the power-scaled spectrogram, GC and GCoS are represented as:

$$\mathbf{Z}_{\text{S}}[k,n] := \sigma_0\left(\mathbf{W}_f\mathbf{X}\right), \tag{1}$$

$$\mathbf{Z}_{\text{GC}}[q,n] := \sigma_1\left(\mathbf{W}_t\mathbf{F}^{-1}\mathbf{Z}_{\text{S}}\right), \tag{2}$$

$$\mathbf{Z}_{\text{GCoS}}[k,n] := \sigma_2\left(\mathbf{W}_f\mathbf{F}\mathbf{Z}_{\text{GC}}\right), \tag{3}$$

$$\sigma_i\left(\mathbf{Z}\right) = |\text{relu}(\mathbf{Z})|^{\gamma_i}, \quad i = 0, 1, 2 \tag{4}$$

where relu$(\cdot)$ represents a rectified linear unit, $|\cdot|^{\gamma_0}$ is an element-wise root function, and we choose $(\gamma_0, \gamma_1, \gamma_2) = (0.24, 0.6, 1)$ for a feature scaling in the power scale [32].

Besides, to fit the perceptive scale of musical pitches, $\mathbf{Z}_{\text{GC}}$ and $\mathbf{Z}_{\text{GCoS}}$ are mapped onto the log-frequency scale, by $88 * 4 = 352$ triangular filters ranging from 27.5 Hz

(A0) to 4487 Hz , with 48 bands per octave. The GC and GCoS after the filterbank are then both on the pitch scale, as denoted by $\tilde{\mathbf{Z}}_{\text{GC}}$ and $\tilde{\mathbf{Z}}_{\text{GCoS}}$. The final 2-D data representation for semantic segmentation is

$$\mathbf{C}[p,n] = \tilde{\mathbf{Z}}_{\text{GC}}[p,n]\tilde{\mathbf{Z}}_{\text{GCoS}}[p,n]\,, \tag{5}$$

where p is the index on the log-frequency scale. The audio files are resampled at 16 kHz and merged into one mono channel. Data representations are computed with a Hann window of 2048 samples. The hop size is 320 samples, and therefore the time step is 20ms. The upper two subplots of Figure 4 illustrate a comparison between the spectrogram and $\mathbf{C}$. We can observe that with the aid of cepstral feature, the unwanted harmonic peaks are highly suppressed in $\mathbf{C}$.

3.2 Semantic segmentation

The proposed segmentation model for vocal melody extraction is mainly based on the DeepLabV3 and its improved version, DeepLabV3+ [6,7], which are the state-of-the-art models for semantic segmentation tasks. The model is a fully convolution neural network with an encoder-decoder architecture. The encoder is implemented by a ResNet [15], followed by an atrous spatial pyramid pooling process, and a decoder implemented by stacks of decoder blocks, as shown in Figure 2.

One major utility in DeepLabV3 is the use of dilated convolution, which can be represented as a generalized version of the standard convolution as follows:

$$\mathbf{y}[i] = \sum_k \mathbf{x}[i + r \cdot k]\mathbf{w}[k] \tag{6}$$

where $\mathbf{x}$ and $\mathbf{y}$ denotes the input and output 2-D feature maps, respectively, $\mathbf{w}$ is the convolution filter and i indicates the locations on the feature maps. The number r is the dilated rate which determines the stride with which the input are sampled and standard convolution is a special case when $r = 1$. To capture the context in different ranges, one can apply dilated convolution with different values of r on the same input feature map parallely, called Atrous Spatial Pyramid Pooling (ASPP) in [6]. The outputs of these parallel convolution operations are then concatenated to provide information collected from various scales, as shown in Figure 2c.

Different from normal image segmentation task that target objects usually holds certain area compared to the whole image, the melody part of music occupies only a small portion and appears as thin lines when visualized in a 2-D image. To overcome this difficulty, We proposed two modifications to improve the performance of the model.

First, the decoder module in DeepLabV3, which is originally an up-sampling operation, is replaced by stacks of convolution and transpose convolution layers for fine-grained outputs. It is shown in [7] that by doing this, the small and detailed objects in an image can be better recognized. Also, better performance is achieved by introducing the U-net [28] structure, which lets the output from each layer of the encoder be concatenated to the corresponding block of the decoder. This idea is also mentioned in [7].

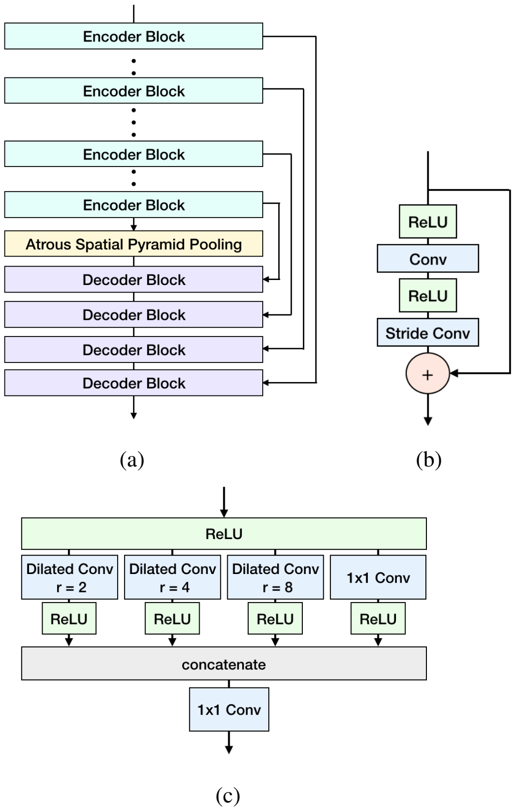

Figure 2: Model descriptions. (a) The overall structure of the segmentation model. (b) The encoder block. The stride rate in *Stride Conv* is (2,2). *Stride Conv* can be replaced with standard convolution so it allows more layers in the encoder. It can also be changed to transpose convolution with stride (2,2), so the block can serve as a decoder block. (c) The Atrous Spatial Pyramid Pooling unit.

Second, we adopt the *focal loss* [23] as the loss function for the proposed model, in order to solve the class imbalance problem, where the negative labels, i.e., the time-frequency pixels corresponding to accompaniment and silence parts, could dominate in the input feature and thus affect the performance. The focal loss is represented as:

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)\,, \tag{7}$$

where p_t denotes the model's estimated probability for an input to be classified to class t, $\alpha_t \in [0,1]$ is a weighting factor for balancing the importance of positive and negative examples and the term $(1 - p_t)^\gamma$ acts as a modulating factor with γ controlling the rate at which dominant examples are down-weighted. Following [23], we set $\alpha_t = 0.25$, $\gamma = 2$ in this work.

3.3 Domain adaptation

Most of the existing deep learning models require a large amount of training data to reach good performance. However, annotating melody pitch contours on audio data precisely is quite challenging; it is labor-intensive and also needs strong expertise in music. Recent attempts to address the issue of data scarcity mostly focus on weakly supervised learning [17, 24, 31].

In this work, we consider the potential of domain-adaptive transfer learning, which incorporates the information in MIDI data to assist in training the audio melody extraction model. The primary motivation for using MIDI files is the capability of data augmentation: one can use MIDI files to easily create large-scale symbolic dataset with detailed and precise notations. Besides, the symbolic data also present some musical characteristics clearer than the audio data do. This therefore gives more insights to the music language modeling, such as musical structures and phrases. Moreover, the space efficiency of symbolic data also allows more training examples than audio data given the same memory resource.

To discuss transfer learning between audio and symbolic data, we first discuss the difference in their data formats. One difference is the pitch resolution, which is 0.25 semitones in the audio data (i.e., 48 bins per octave), and 1 semitone in the symbolic data; this results in the difference of dimension between the audio and the symbolic data. As for the time resolution, there are some more flexible ways to define it. Therefore, we consider two types of time resolution for the symbolic data: the first is *time-based* resolution with its unit length in time (e.g., 20 ms), and the second is *note-based* resolution with its unit length in note name (e.g., a 32nd note). Both the symbolic and audio data can be represented in *time-based* resolution. Symbolic data can also be represented in a more musically informative *note-based* resolution since obtaining beat and tempo information in symbolic data is more straightforward.

To achieve domain-adaptive transfer learning for two different domains, we adopt the progressive neural network (PNN) [1], in which an *adapter* network (see Figure 3) is designed to make one network connected to another in different domains, regardless of the difference in data dimension. In the general scenario of PNN, multiple networks trained on various tasks are connected layer-to-layer in parallel through the adapters, so the trained networks can transfer the previously learned knowledge into a new task and to accelerate the training speed or to improve the performance of the new task.

In our melody extraction method, we first trained a segmentation model using the symbolic dataset. We connect the symbolic segmentation model to another segmentation model, and the latter model is then trained on the audio dataset, with the parameters in the symbolic segmentation model frozen. In the testing phase, the input audio representation is fed into both of the segmentation models. To make the dimension of audio representation match the symbolic segmentation model, a triangular filterbank is used to map the pitch resolution from 0.25 to 1 semitone, as illustrated in Figure 1.

The adapter between two models in the PNN is illustrated in Figure 3. It modifies the dimension of the inter-layer outputs and make such information be propagated ahead. In the proposed method, transpose convolution layers are adopted for the adapter networks, since transpose convolution can up-sample the output of the symbolic representation (with lower pitch resolution) in order to fit the audio representation (with higher pitch resolution).

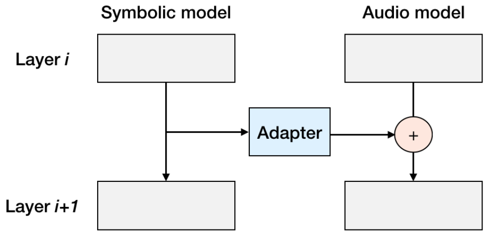

Figure 3: The connection between the two networks in the proposed model. The parameter of the i-th layer in the symbolic model is first fed into the adapter, and then connected to the $(i + 1)$-th layer of the audio model with an addition operation.

3.4 Inference

Since the segmentation model only allows a limited range of input at one time, to perform melody extraction on a given score, we slide a window along the score and then superpose all the resulting matrices. The analysis window with a fixed dimension is shifted from one time-step to another. As to the beginning and ending time, we pad the score with zeros for it captures the process in which information feeds only the last column then gradually filling up all the columns in the beginning, and gradually leaving the window column by column at the end. After the process above, the segmentation output is a superposed image representing the salience of vocal melody in the time-frequency plane. We then find the max value for each column of the image and set all the other elements to zero, i.e., unvoiced. Finally, the elements smaller than the average of each column's maximum are also set to zero, and the remaining non-zero elements is considered as voiced.

3.5 Implementation details

The models are implemented using the Keras [9] library with tensorflow as the back end. The width of the input window equals 128 timesteps, and for computational convenience, we pad the dimension of pitch from 88 to 128, and 352 to 384 for the symbolic and audio data, so the input dimension will be $(128, 128, 1)$ and $(128, 384, 1)$ for the symbolic and audio model, respectively. As shown in Fig.1, the input feature will first be passed into a 29-layer encoder based on Resnet. Then, the output from the encoder which is 16 times smaller than the original input will be fed into the ASPP unit. Finally, a decoder which contains 4 decoder blocks will up-sample the dense features to the original shape by transpose convolutional layers with strides equal $(2, 2)$. The output dimension will be $(128, 128, 2)$ and $(128, 384, 2)$ for the symbolic and audio model, respectively, with the first channel indicating the presence melody and the other is for non-melody.

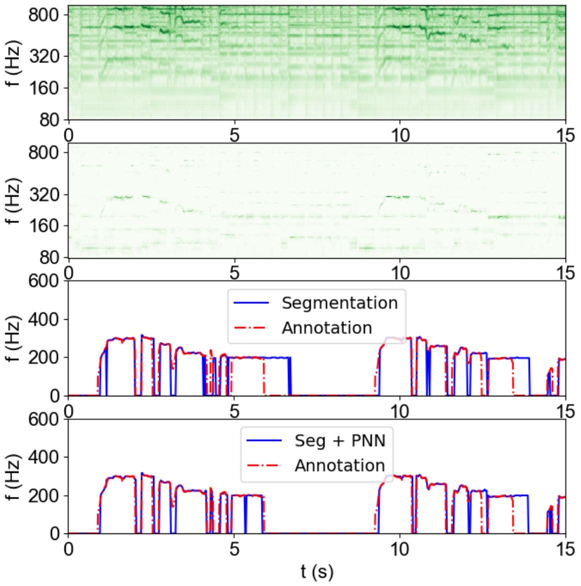

Figure 4: Data representation and melody extraction results of the first 15s of 'train06.wav' in MIREX2005 as input. From top to bottom: power-scale spectrogram, data representation $\mathbf{C}$, the result using segmentation, and the result using segmentation and note-based PNN.

The superposition in the inference process is performed on the first channel. To implement the PNN, two segmentation networks with same structure are connected using the adapters which is composed of transpose convolution layer. These connections happen in layers with dimension changing. Batch normalizations are applied after each activations, and a dropout rate of 30% is added after the batch normalizations. ADAM [19] is used for optimization. Source codes can be found at `https://github.com/s603122001/Vocal-Melody-Extraction`.

4. EXPERIMENT

4.1 Data

The training data for the audio comes from two datasets, one is the MIR1K [1] , which contains 1000 Chinese karaoke clips, another is MedleyDB [5], where 48 songs with vocal tracks are included. The total dataset contains about 3 hours of audio and without data augmentation.

A MIDI corpus contains 600 folk songs with a melody track is used as the training data for the symbolic model. [2] In the training process, we perform data augmentation, by pitch-shifting each song up and down by at most 6 semitones in order to cover all possible keys. In addition, half of the pieces in the dataset are modified by shifting the melody by one octave down. As a result, we produce 7673 pieces of symbolic training data. The pieces in the dataset are represented in two different formats. One is the *time-based* with 20 ms length in each time step and the other is the *note-based* that each time step equals a thirty-second note. Due to limited computational resources, we only use 2048 pieces when training the time-based model since time-based data is space consuming.

The testing data are from three datasets: ADC2004, MIREX05, [3] and MedleyDB. As the proposed model is trained solely for singing voice melody, we follow [22] and select only samples having melody sung by human voice from ADC2004 and MIREX05. As a result, 12 clips in ADC2004 and 9 clips in MIREX05 are selected. To obtain the annotation of singing voice in medleyDB, 12 songs having singing voice included in their 'MELODY2' annotations are selected. The vocal melody labels are obtained from the MELODY2 annotations occurring in the intervals labeled by 'female singer' or 'male singer'. These 12 songs are not included in the training data.

4.2 Experiment setting

To assess the performance of semantic segmentation and the effects of transfer learning on vocal melody extraction, we experiment on the following three different settings:

1) *Segmentation*: using simply the audio-level semantic segmentation model. This audio-only semantic segmentation model is trained on the MIR1K dataset.

2) *Segmentation with note-based progressive neural network (Seg + note PNN)*: using both the audio-level and symbolic-level segmentation models. The symbolic segmentation model is first trained using the note-based symbolic dataset, then this model is incorporated into the training stage of the audio segmentation model with the PNN.

3) *Segmentation with time-based progressive neural network (Seg + time PNN)*: similar to 2), while the symbolic model in trained with the time-based symbolic dataset.

We compare the above-mentioned models with three baseline methods in deep learning approaches: the multi-column DNN (MCDNN) [22], the patch-based CNN (pathc-CNN) [33], and the deep salience map (DSM), for which on-line source code with the vocal option is available [2]. Since the detection results of DSM are sensitive to the thresholding parameter, the parameter is tuned from 0 to 0.9 for all datasets to find the optimal value for better comparison. The resulting optimal threshold th=0.1 is used in the experiment.

The performance metrics include overall accuracy (OA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), voice recall (VR) and voice false alarm (VFA); [4] all these metrics are computed from the mir_eval standard with the tolerance of pitch detection being 50 cents.

4.3 Result

Table 1 lists the performance metrics of all the proposed methods together with the baselines on the three testing datasets. Among the three proposed models, *Segmentation* outperforms the other two PNN-based models in terms of OA for all datasets except MedleyDB, where *Segmentation* performs on par with *Seg + note PNN*. Through the

[1] https://sites.google.com/site/unvoicedsoundseparation/mir-1k
[2] https://goo.gl/aPgzrW
[3] https://labrosa.ee.columbia.edu/projects/melody/
[4] http://www.music-ir.org/mirex/wiki/2016:Audio_Melody_Extraction

Method	OA	RPA	RCA	VR	VFA
Segmentation	**74.9**	71.7	74.8	73.8	**3.0**
Seg + note PNN	73.5	70.2	73.2	72.2	3.1
Seg + time PNN	73.2	70.4	72.9	73.2	5.4
MCDNN [22]	73.1	75.8	78.3	88.9	41.2
Patch-CNN [33]	72.4	74.7	75.7	90.1	41.3
DSM [2]	70.8	**77.1**	**78.8**	**92.9**	50.5

(a) ADC2004 (vocal)

Method	OA	RPA	RCA	VR	VFA
Segmentation	**85.8**	**82.2**	**82.9**	87.3	7.9
Seg + note PNN	84.5	79.6	80.3	84.7	**6.9**
Seg + time PNN	84.8	82.3	83.0	87.3	9.9
MCDNN	68.4	76.3	77.4	87.0	49.0
Patch-CNN	74.4	83.1	83.5	95.1	41.1
DSM	69.6	76.3	77.3	**93.6**	42.8

(b) MIREX2005 (vocal)

Method	OA	RPA	RCA	VR	VFA
Segmentation	**70.0**	68.3	70.0	77.9	22.4
Seg + note PNN	**70.0**	67.1	68.7	77.0	**21.5**
Seg + time PNN	69.1	67.4	69.0	78.7	23.6
Patch-CNN	55.2	59.7	63.8	78.4	55.1
DSM	66.2	**72.0**	**74.8**	**88.4**	48.7

(c) MedleyDB (vocal)

Table 1: Vocal melody extraction results of the proposed methods and other methods on various datasets. The proposed methods are: segmentation, segmentation with note-based progressive neural network (Seg + note PNN), and segmentation with time-based progressive neural network (Seg + time PNN).

melody extraction accuracies of the segmentation model are not improved by introducing the PNN structure, there is still a notable improvement when comparing training efficiency. In fact, it takes 6 epochs for *Segmentation* to converge, but *Seg + note PNN* reach similar performance with only 2 epochs of training. Therefore, introducing the PNN improves the training speed.

One reason why PNN does not improve the accuracy is related to the symbolic dataset we are using: the symbolic data contains only one style of music and turns out to be of low diversity. Another reason is the lack of *intensity* labels in symbolic data. Our pilot study indicated that a segmentation model trained on symbolic data may result in high RCA and RPA but also relatively high VFA. However, a segmentation model trained on the audio data gives inverse results, with low VFA, as shown here. This might have something to do with the sound intensity in the audio signal, which is an important sign for to determine the present of melody. However, our symbolic data do not have such labels on intensity. Model training with a larger symbolic music dataset with higher diversity and with MIDI velocity labels are for future investigation.

The two PNN-based methods, *Seg + note PNN* and *Seg + time PNN*, achieve similar OA, while the former model has lower VFA. This implies that the performance of the symbolic model trained with note-based symbolic data is better than training with time-based data. One reason may be that compiling symbolic data in time-based resolution may result in the ambiguity of musical information; in time-based data, the same type of note may have different lengths in time due to different tempi among the music pieces. This could affect the model capability in learning the musical structure.

Comparing the proposed *Segmentation* model to the baseline methods, we observe that *Segmentation* outperforms all of them in terms of OA. Particularly, in MIREX2005, *Segmentation* achieves an OA at 85.8%, a high accuracy outperforming DSM by 16.2%, patch-CNN by 11.4% and MCDNN by 17.4%. In other two datasets, *Segmentation* also outperforms other methods by around $1 \sim 4\%$ in terms of OA. These experiment results reveal the competitiveness of the proposed semantic segmentation method in audio melody extraction. On the other hand, when focusing on the pitch accuracy (i.e., RPA and RCA), DSM is still competitive among all.

The high OA of *Segmentation* is mainly resulted from the excellent performance of VFA with the semantic segmentation approach. Among all methods and datasets, the proposed methods significantly outperform the baseline methods by a 20-40% reduction in VFA. In ADC2004, *Segmentation* further achieves a low VFA of 3.0%. This implies that the proposed melody extraction method itself is highly robust to non-vocal interference, and is without the need of a voice activity detector [27]. In other words, the semantic segmentation model with fully convolutional layers itself behaves as a melody pitch classifier and a voice activity detector at the same time.

Finally, the lower two subplots in Figure 4 illustrate two melody extraction results using *Segmentation* without and with a note-based PNN. Both methods perform well in segmenting the main melody part from the representation $\mathbf{C}$ shown in second subplot in Figure 4. This example also demonstrates one part that using the note-based PNN does well: in the lowest subplot, the *Seg + note PNN* method well detects the unvoiced part between the 6th and the 9th second, in which the *Segmentation* method regards the extended instrument part as melody.

5. CONCLUSION

We proposed a melody extraction method utilizing the semantic segmentation model, the input combining spectral and cepstral representations, and domain-adaptive transfer learning. Experiments using a low-diversity training data indicate the competitiveness of the segmentation model with the data representation, especially in reducing voice false alarm. Incorporating large-scale symbolic data provides better efficiency and exhibits potential in enhancing contextual information. Future work will focus on the improvement of domain adaption. Note-level segmentation can be considered as a future work as it is also feasible applying symbolic-audio transfer learning and would also benefit the melody extraction task.

6. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work is partially supported by MOST Taiwan, under the contract MOST 106-2218-E-001-003-MY3.

7. REFERENCES

[1] R. Andrei A., R. Neil C., D.Guillaume, S. Hubert, K. James, K. Koray, P. Razvan, and H. Raia. Progressive neural networks. *eprint arXiv:1606.04671*, 2016.

[2] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello. Deep salience representations for f_0 estimation in polyphonic music. In *18th Int. Soc. for Music Info. Retrieval Conf.*, Suzhou, China, Oct. 2017.

[3] R. M. Bittner, J. Salamon, J. J. Bosch, and J. P. Bello. Pitch contours as a mid-level representation for music informatics. In *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. Audio Engineering Society, 2017.

[4] R. M. Bittner, J. Salamon, S. Essid, and J. P. Bello. Melody extraction by contour classification. In *Proc. ISMIR*, pages 500–506, 2015.

[5] 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. ISMIR*, volume 14, pages 155–160, 2014.

[6] L.-C. Chen, P. George, S. Florian, and A. Hartwig. Rethinking atrous convolution for semantic image segmentation. *eprint arXiv:1706.05587*, 2017.

[7] L.-C. Chen, Y. Zhu, P. George, S. Florian, and A. Hartwig. Encoder-decoder with atrous separable convolution for semantic image segmentation. *eprint arXiv:1802.02611*, 2018.

[8] K. Choi, G. Fazekas, M. Sandler, and K. Cho. Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*, 2017.

[9] F. Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[10] R. B. Dannenberg and C. Raphael. Music score alignment and computer accompaniment. *Communications of the ACM*, 49(8):38–43, 2006.

[11] M. Dorfer, A. Arzt, and G. Widmer. Learning audio-sheet music correspondences for score identification and offline alignment. In *18th Int. Soc. for Music Info. Retrieval Conf.*, Oct.

[12] M. Goto. A predominant-F0 estimation method for polyphonic musical audio signals. In *Proc. Int. Cong. Acoustics*, pages 1085–1088, 2004.

[13] P. Hamel, M. Davies, K. Yoshii, and M. Goto. Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity. 2013.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

[15] K. He, X. Zhang, S Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[16] H. Indefrey, W. Hess, and G. Seeser. Design and evaluation of double-transform pitch determination algorithms with nonlinear distortion in the frequency domain-preliminary results. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, pages 415–418, 1985.

[17] A. Jansson, E. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde. Singing voice separation with deep u-net convolutional networks. In *18th Int. Soc. for Music Info. Retrieval Conf.*, Suzhou, China, Oct. 2017.

[18] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer. On the potential of simple framewise approaches to piano transcription. *arXiv preprint arXiv:1612.05153*, 2016.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2014.

[20] A. Klapuri. Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Trans. Audio, Speech, Lang. Proc.*, 16(2):255–266, 2008.

[21] T. Kobayashi and S. Imai. Spectral analysis using generalized cepstrum. *IEEE Trans. Acoust., Speech, Signal Proc.*, 32(5):1087–1089, 1984.

[22] S. Kum, C. Oh, and J. Nam. Melody extraction on vocal segments using multi-column deep neural networks. In *Proc. ISMIR*, pages 819–825, 2016.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dolla r. Focal loss for dense object detection. *eprint arXiv:1708.02002*, 2017.

[24] J.-Y. Liu and Y.-H. Yang. Event localization in music auto-tagging. In *Proc. ACM Multimedia*, pages 1048–1057. ACM, 2016.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[26] G. Peeters. Music pitch representation by periodicity measures based on combined temporal and spectral representations. In *Proc. IEEE ICASSP*, 2006.

[27] F. Rigaud and M. Radenen. Singing voice melody transcription using deep neural networks. In *ISMIR*, pages 737–743, 2016.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] J. Salamon and E. Gómez. Melody extraction from polyphonic music audio. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2010.

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

[31] J. Schlüter. Learning to pinpoint singing voice from weakly labeled examples. In *ISMIR*, pages 44–50, 2016.

[32] L. Su. Between homomorphic signal processing and deep neural networks: Constructing deep algorithms for polyphonic music transcription. In *Asia Pacific Signal and Infor. Proc. Asso. Annual Summit and Conf. (APSIPA ASC)*, 2017.

[33] L. Su. Vocal melody extraction using patch-based cnn. In *Proc. ICASSP*, 2018.

[34] L. Su and Y.-H. Yang. Combining spectral and temporal representations for multipitch estimation of polyphonic music. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 23(10):1600–1612, 2015.

[35] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai. Mel-generalized cepstral analysis: a unified approach to speech spectral estimation. In *Proc. Int. Conf. Spoken Language Processing*, 1994.

[36] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. *IEEE Speech Audio Processing*, 8(6):708–716, 2000.

[37] P. Verma and R. W. Schafer. Frequency estimation from waveforms using multi-layered neural networks. In *INTERSPEECH*, pages 2165–2169, 2016.