Singing Voice Separation and Pitch Extraction from Monaural Polyphonic Audio Music Via DNN and Adaptive Pitch Tracking

Zhe-Cheng Fan, Jyh-Shing Roger Jang
Dept. of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
{lambert.fan, jang}@mirlab.org

Chung-Li Lu
Chunghwa Telecom Laboratories
Taoyuan, Taiwan
chungli@cht.com.tw

***Abstract*—With the explosive growth of audio music everywhere over the Internet, it is becoming more important to be able to classify or retrieve audio music based on their key components, such as vocal pitch for common popular music. This paper proposes a novel and effective two-stage approach to singing pitch extraction, which involves singing voice separation and pitch tracking for monaural polyphonic audio music. The first stage extracts singing voice from the songs by using deep neural networks in a supervised setting. Then the second stage estimates the pitch based on the extracted singing voice in a robust manner. Experimental results based on MIR-1K showed that the proposed approach outperforms a previous state-of-the-art approach in raw-pitch accuracy. Moreover, the proposed approach has been submitted to the singing voice separation and audio melody extraction tasks of Music Information Retrieval Evaluation eXchange (MIREX) in 2015. The results of the competition show that the proposed approach is superior to other submitted algorithms, which demonstrates the feasibility of the method for further applications in music processing.**

Keywords- Audio melody extraction, singing pitch extraction, singing voice separation, multimedia, deep neural networks.

I. INTRODUCTION

In recent years, there are more and more music providers which offer digital distribution of music through online music stores and streaming services, such as Spotify, iTunes and Google Play. The rapid growth of audio music calls for an effective way to classify and retrieve audio contents via their key components, such as pitch, beat, chord progression, and so on. For common popular music with lead vocal, the most important component is the vocal pitch, which serves as the most memorable part of a song for most people. As a result, it is essential to perform singing pitch extraction (SPE) from monaural polyphonic audio music. SPE is critical to numerous real world applications of music analysis and classification, including singer identification, lyric recognition and synchronization, cover song detection, singing scoring, database construction for query by singing/humming, and so on. However, SPE is a very challenging task due to the severe interference from music accompaniments in a mixture music containing vocal. (Note that SPE denotes pitch tracking over the lead vocal in common popular music. It is a special case of audio melody extraction which aims to extract pitch from audio music with an instrument carrying the main melody. See J. Salamon et al. [29] for a more detailed definition for audio melody extraction.)

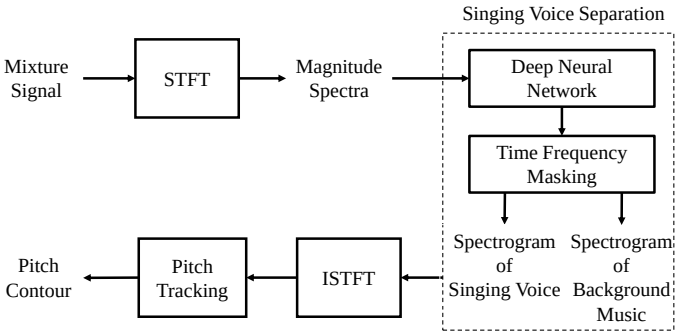

Figure 1. Block diagram of the proposed system.

Several approaches to audio melody extractions have been proposed in the literature after Goto's 2004 seminal paper [1] on using a parametric statistical model for audio melody extraction. Recently, J. Salamon et al. [6] come up with a comprehensive coverage of approaches to audio melody extraction. In general, there are two categories of SPE approaches. In the first category, pitch is selected directly from a set of pitch candidates which are derived from a periodicity detection function. For instance, Salamon et al. [2] propose a salience-based melody extraction method where a periodicity detection function (called salience function in their paper) is constructed by extracted spectral peaks, and the identified pitch is determined by a set of contour characteristics. In the second category, the pitch extraction is performed on the singing voice separated from the mixture music. The first SPE method in this category is proposed by Regnier et al. [3]. Other related work in this category can be found in [4, 5, 8, 10].

In this paper, we propose a novel and effective two-stage approach which explores the use of deep neural networks (DNN) for singing voice separation in a supervised setting. After extracting the singing voice from mixture music, the pitch is determined by using a robust pitch tracking method based on dynamic programming. The block diagram of the proposed system is shown in Figure 1.

The rest of the paper is organized as follows: Section 2 discusses the relation to previous work. Section 3 introduces the proposed method, including deep neural networks for singing voice separation and dynamic-programming-based robust approach to pitch tracking. Section 4 presents the experimental settings and the corresponding results using MIR-1K dataset, together with the results of two tasks (singing voice separation and audio melody extraction) in MIREX 2005. Concluding remarks and potential future directions are covered in Section 5.

II. RELATED WORK

Several approaches have been proposed to detect pitch after extracting singing voice. Hsu et al. [4] used a hidden Markov model (HMM) to detect singing voice with energy at semitones of interests and Mel-frequency cepstral coefficients as input features. In another paper, Hsu et al. [5] applied the method proposed in [3] for characterizing vibrato and tremolo in order to detect the presence of singing voice. Trend estimation of pitch was proposed in [7] to improve voice separation by detecting the pitch ranges of singing voice at each time frame and eliminating wrong pitch candidates by vibrato and tremolo features. Yeh et al. [8] proposed a hybrid approach consisting of [4] and [7] to achieve further improvement over SPE, which involves forward and backward trend estimation and training-based HMM to determine the pitch. Hsu et al. also proposed the Tandem algorithm [10] to better estimate the singing pitch and separate the singing voice iteratively. Their system can estimate rough pitches which were used to separate the singing voice by considering harmonicity and temporal continuity, and the separated singing voice can be used for better pitch tracking. The separated singing voice and estimated pitches were used to improve each other iteratively until convergence.

Along another direction, singing voice separation has been performed successfully by deep neural networks (DNNs). Deep learning methods have been applied to a variety of applications, including noise reduction [11, 12] which aims at creating a clean version of an utterance from a noisy one. Besides, DNN was also applied to speech recognition [13] via restricted Boltzmann machine and instrument extraction from music [28]. In the scenario of singing voice separation, given a mixture music regarded as a noisy signal, a DNN is trained to output the clean signal of vocal only. Similar work has been proposed in [14, 27] by using deep recurrent neural networks.

III. PROPOSED METHODS

A. Deep Neural Networks

For singing voice separation, we explore the use of deep neural networks to learn the optimum parameters under a given architecture to reconstruct the target spectra of singing voice.

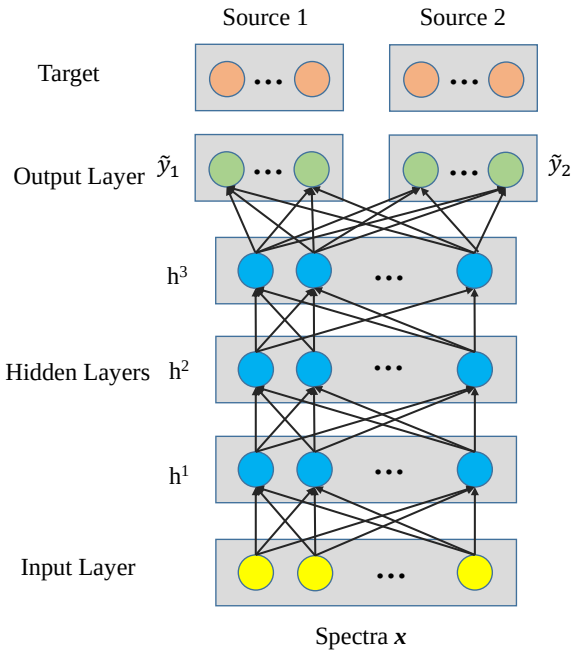

Figure 2. The architecture of deep neural network.

The architecture of DNNs are characterized by one or more hidden layers consisting of hidden nodes, with each hidden node representing a nonlinear activation function. Formally, we can define the scheme of DNN as follows. Suppose there is a DNN with L intermediate layers, the function performed by the l-th layer can be defined as follows:

$$\boldsymbol{h}^l = f(\mathbf{W}^l \boldsymbol{h}^{l-1} + \mathbf{b}^l), \quad (1)$$

and the overall output y of the DNN can be defined as:

$$y = f(\mathbf{W}^L \ldots f(\mathbf{W}^2 f(\mathbf{W}^1 \boldsymbol{h}^0 + \mathbf{b}^1) + \mathbf{b}^2) \ldots + \mathbf{b}^L), \quad (2)$$

where $\boldsymbol{h}^l$ is the hidden state of the l-th layer. $\mathbf{W}^l$ and $\mathbf{b}^l$ are the weight matrix and bias vector respectively for layer l, $1 \le l \le L$. For the first layer, $\boldsymbol{h}^0 = x$, where x is the input to the DNN, consisting of magnitude spectra of the mixture music which is performed by using short time Fourier transform (STFT). The function $f()$ is a nonlinear sigmoidal function which is applied to the output of matrix multiplication and element-wise addition. The weight matrix and bias vector were estimated by back-propagation [17] and stochastic gradient descent [18]. We have also tried several speedup techniques for gradient decent, including Momentum [19], adaptive subgradient [20], root mean squared gradient (RMSProp) [21], Adadelta [22], Nesterov's accelerated gradient [23] and so forth. We found that RMSProp performed the best in our experiments.

B. Model Architecure

As shown in Figure 2, given an input vector of mixture spectra $\boldsymbol{x}$, we can obtain the predicted spectra $\tilde{y}_1$ (spectra of the vocal) and $\tilde{y}_2$ (spectra of the background music) through the DNN. Given the original sources y_1 and y_2 (after normalization by dividing square of y_1 and y_2 respectively), the objective function J can be defined as follows:

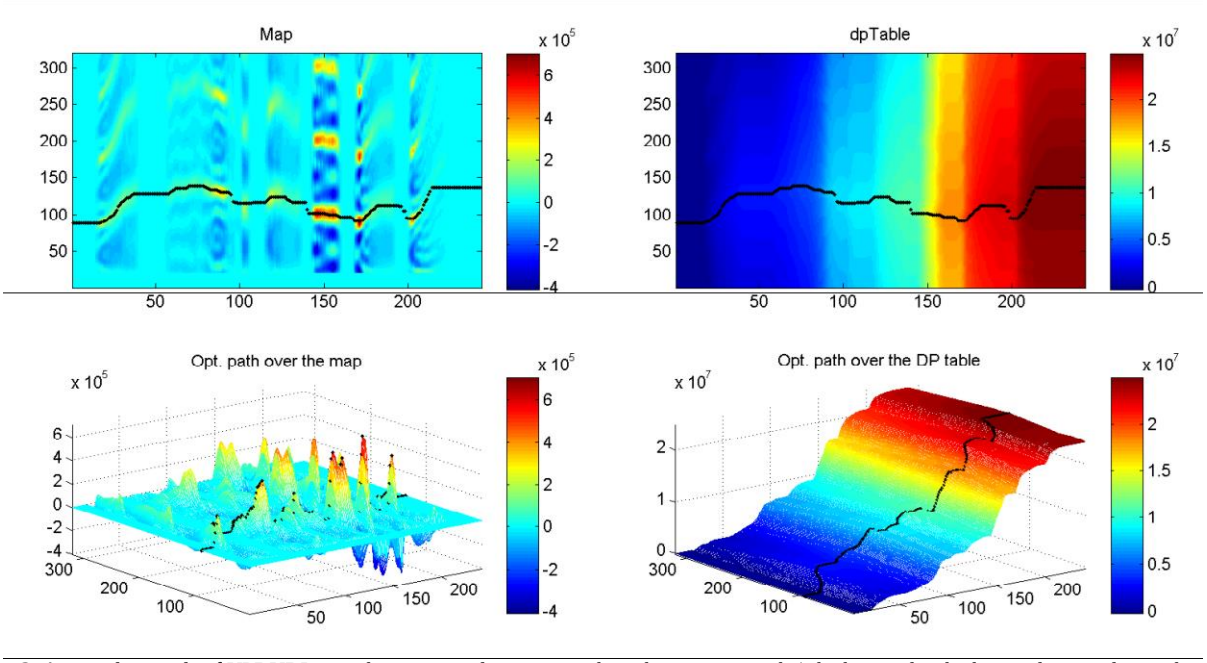

Figure 3. A typical example of UPDUDP over the auto-correlation map, where the optimum path (which considers both periodicity and smoothness of pitch) is obtained via dynamic programming. The black line is the optimum path over the auto-correlation map, which picks peaks most of the time. (For AMDF map, the optimum will picks valleys most of the time.)

$$J = \|\tilde{y}_1 - y_1\|_2^2 + \|\tilde{y}_2 - y_2\|_2^2. \quad (3)$$

Since the output is constrained to be within 0 and 1, we can define a soft time-frequency mask *m* [14] as follows:

$$m(f) = |\tilde{y}_1(f)| / (|\tilde{y}_1(f)| + |\tilde{y}_2(f)|), \quad (4)$$

where $f = 1, 2, \ldots F$, stands for different frequency bins. Then the estimated spectra $\tilde{s}_1$ and $\tilde{s}_2$, corresponding to vocal and music, respectively, can be defined accordingly:

$$\begin{aligned} \tilde{s}_1(f) &= m(f) z(f) \\ \tilde{s}_2(f) &= (1 - m(f)) z(f) \end{aligned} \quad (5)$$

where $z(f)$ is the magnitude spectra of the input frame.

The time-domain signals of estimated magnitude spectra are reconstructed by using inverse short time Fourier transform (ISTFT), which uses the phase information obtained from the original input signals.

C. Pitch Tracking

Once the vocal is extracted from the mixture, we need to perform pitch tracking to extract the vocal pitch. Here we propose a new adaptive method based on a previously proposed approach of unbroken pitch determination using dynamic programing (UPDUDP) [15] which is a robust pitch tracking method based on dynamic programming. Figure 3 shows a typical example of UPDUDP which considers both periodicity and smoothness to derive the final optimum path. To be more specific, given a frame of audio stream, we first compute the periodicity detection function (PDF) of each frame based on average magnitude difference function (AMDF) [16], with a frame size of 40 ms (640 samples) and a hop size of 10 ms (160 samples). The "shifted part" of our AMDF is actually only the first half of the frame, as shown in the following equation:

$$amdf(j) = \sum_{u=1}^{320} |frame[u] - frame[u + j]|, \; j = 0 \sim 320, \quad (6)$$

where $frame[u]$ is the u-th sample value of a given frame, and $amdf(j)$ is the j-th value of the AMDF vector. If we adopt a frame-based-only pitch tracking, we can simply pick the minimum AMDF of a frame within the index range of [16, 320] (corresponding to a frequency range of 50~1000Hz, or 31.35~83.21 semitones) to determine the frame's pitch. However, it is well-known that the undesirable effect of half or double frequencies is likely to happen, leading to an octave below or above the real pitch. As a result, we need to have a more robust mechanism to identify a smooth pitch contour. Note that we can put all the AMDF vectors of a given utterance into a $320 \times n$ matrix, where n is the number of frames. Our mission is to find a path through the AMDF matrix such that a balance between the value of AMDF (as small as possible) and the smoothness of the pitch contour is achieved. More specifically, for a given path $\mathbf{p} = [p_1, \cdots p_i, \cdots p_n]$ over an AMDF matrix where $16 \le p_i \le 320$, we can define a cost function as follows:

$$\text{cost}(\mathbf{p}, \theta, m) = \sum_{i=1}^{n} amdf_i(p_i) + \theta \times \sum_{i=1}^{n-1} |p_i - p_{i+1}|^m, \quad (7)$$

where $amdf_i$ is the AMDF vector of frame i, θ is the transition penalty term and m is the exponent for the difference in a path of two neighboring frames. As explained in [15], the above objective function can be minimized by a dynamical programming approach. More specifically, let the optimum-valued function $D(i, j)$ be defined as the minimum cost starting from frame 1 to i, with $p_i = j$. Then we can come up with the recurrent equation for $D(i, j)$, as follows:

$$D(i, j) = amdf_i(j) + \min_{k\in[16,320]}\left\{D(i-1,k) + \theta \times |k-j|^2\right\}, \quad (8)$$

where $i \in [1, n]$, $j \in [16,320]$, $i > 1$. The initial conditions are

$$D(1, j) = amdf_1(j), j \in [16,320]. \quad (9)$$

And the optimum cost is equal to $\min_{j\in[16,320]} D(n, j)$

In the above recurrent equation for dynamical programming (or in the original objective function), it is obvious that the value of θ controls the smoothness of the identified pitch curve. That is, a bigger θ will lead to a smoother pitch curve. However, if θ is too big, the resultant pitch curve will have low contrast and deviate from the true pitch. Our empirical studies indicate that under different recording conditions (different volume, different ambient noise, different microphone settings, etc.), it is hard to pinpoint a universally optimum value of θ that can achieve the best performance. As a result, this paper proposes an adaptive way to determine θ in UPDUDP (which is referred to as "adaptive UPDUDP") based on the continuity of pitch curve. The basic idea is based on the concept that the pitch curve of a person's voice should be continuous. In other words, we want to identify the (approximately) minimum value of θ that can make the pitch curve continuous. The continuity requirement of a given pitch curve $s = [s_1, \cdots s_i, \cdots s_n]$ (in terms of semitones) can be expressed as follows:

$$d(\theta) = \max_{i=1\sim n-1} |s_i - s_{i+1}| < \tau. \quad (10)$$

That is, the function $d(\theta)$, which stands for the max difference of pitch between neighboring frames, is require to be less than a given threshold τ. Empirically, we set the value of τ to 7 semitone for a hop size of 10 ms (or equivalently, a frame rate of 100 per second).

It is possible to increase the value of θ linearly until the pitch curve satisfies the continuity requirement shown in the previous equation. However, it is too time consuming. Here we propose a method that can identify the approximately minimum value of θ, denoted as $\hat{\theta}$, that can ensure the continuity requirement. The method can be described as follows.

1. Initial step: If $d(0) < \tau$, then $\hat{\theta} = 0$ and we are done.
2. Bracket: Our goal is to rapidly identify an interval $\hat{I} = [\theta_l, \theta_u]$ satisfying the bracket condition, that is, $d(\theta_l) \geq \tau$ and $d(\theta_u) < \tau$. This is achieved by the following steps:
   a. Set $I_0 = [\theta_0, \theta_1] = [0,1]$. If the bracket condition is fulfilled, then we are done with $\hat{I} = I_0$. Otherwise set $i = 1$ go to the next step.
   b. Set $I_i = [\theta_i, \theta_{i+1}]$ with $\theta_{i+1} = 2\theta_i$.
   c. If the bracket condition for I_i is fulfilled, then we are done with $\hat{I} = I_i$. Otherwise increment i and go back to step b.
3. Refine: Once we have the bracket interval $\hat{I}$, then we can employ binary-search-like algorithm to refine the interval efficiently. The iteration can be stopped when the range of the interval is less than, say 10. The final $\hat{\theta}$ is then selected as the upper bound of the refined interval.

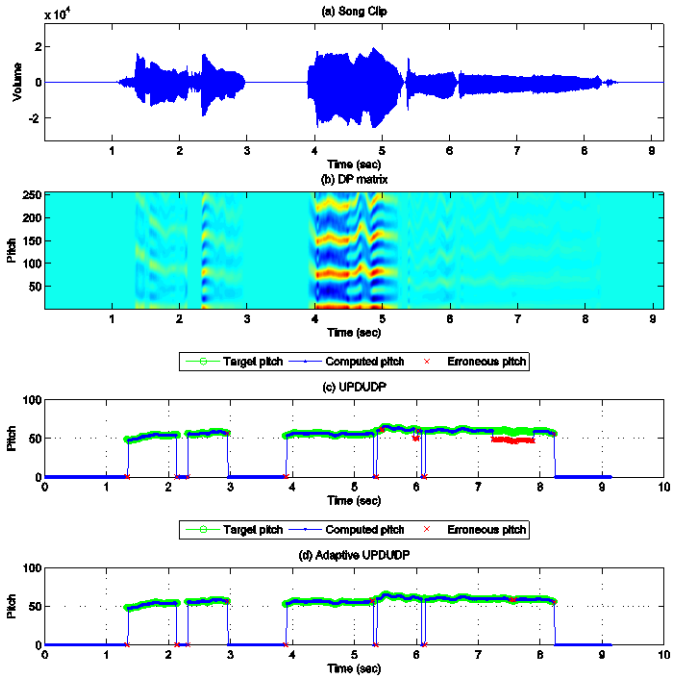

Figure 4. (a) A song clip without background music from MIR-1K. (b) Time-frequency energy plot of the utterance. The brighter area indicates strong energy. (c) The result of pitch tracking by using UPDUDP. The green labels represent the target pitch, and the blue labels represent the computed pitch. The red labels represent the erroneous pitch. (d) The result of pitch tracking by using adaptive UPDUDP.

The above procedure for selecting θ to ensure the continuity of the pitch curve is efficient in computation, and effective is enhancing the pitch accuracy, as described in the experiment section. Figure 4 shows a typical result of using UPDUDP and its adaptive version for pitch tracking. As shown in the figure, the proposed adaptive UPDUDP can effectively reduce the octave errors (double-pitch or half-pitch errors) due to its capability in forcing the pitch to be smooth.

IV. EXPERIMENTS

A. Experimental Settings

The proposed method is first evaluated by using the MIR-1K dataset [10] which consists of 1,000 song clips with a sample rate of 16 KHz and durations from 4 to 13 seconds. These clips are recorded from 110 Chinese popular karaoke songs performed by both male and female amateurs. Manual annotations of the pitch contours, lyrics, indices of voiced and unvoiced frames, and the indices of the vocal and non-vocal frames are provided. Each clip is a stereo recording, with one channel for singing voice and the other for background music.

In our three experiments, we use magnitude spectra of each frame (with a frame size of 1024 and a hop size of 512) as input features to DNN, which yields an input dimension of 513. Sigmoid function is employed as activation function in DNN and RMSProp is used to speed up gradient decent. A dropout [25] rate of 0.5 is employed for all hidden layers in the DNN. Moreover, the training data of these three experiments was divided into 186 batches, with each song in the training set divided into each batch as evenly as possible. To prevent the trained model from overfitting, the validation batch was chosen from training batch randomly and used along the training process. The training process was stopped when the linear cost was lower than a threshold (0.24) or when the maximum number of epochs (1000) is reached. To accelerate training, our implementation of DNN takes advantage of parallel computing via GPU.

In experiment 1, we tested different DNN architectures by changing numbers of hidden layers or numbers of nodes in each hidden layers, as describe in TABLE I. Here we used UPDUDP for pitch tracking on the extracted singing voice. By following the evaluation framework in [14], we used 175 song clips sung by one male and one female as training data, leading to approximately 141,000 frames for training, each with 513 dimensions as input features to DNN. It is a quite large number of training data for deep learning. The remaining 825 song clips of 17 singers are used for testing. The evaluation indices of singing voice separation are Source to Interference Ratio (SIR), Source to Artifacts Ratio (SAR) and Source to Distortion Ratio (SDR) by using BSS Eval toolbox [24]. We computed the normalized SDR by $\mathrm{SDR}(\hat{v},v)-\mathrm{SDR}(x,v)$, where $\hat{v}$ is reconstructed voice signal, v is original clean voice signal, and x is mixture signal. Moreover, we aggregate overall performance by taking a weighted average of NSDRs, SIRs and SARs to have GNSDR, GSIR and GSAR respectively.

In experiment 2, we compared three different pitch-tracking methods with the best performed DNN architecture (3 hidden layers of 1024 nodes each) obtained in experiment 1. The training set is the same as experiment 1 and the accuracy was calculated by testing remaining 825 song clips with 0.5 semitone tolerance to obtain 3 performance indices for pitch tracking, including overall, raw-pitch and raw-chroma accuracies.

In experiment 3, we compared the proposed approach to Yeh's method [8] which is submitted to MIREX contest and achieved the best performance in raw-pitch and raw-chroma accuracies on MIREX-09 dataset of audio melody extraction task from 2012 to 2014. The DNN architecture is the same as in experiment 2. The experimental settings are the same as in [8], which 5-fold singer-specific cross validation with 0.5 semitone tolerance to obtain average raw-pitch and raw-chroma accuracies.

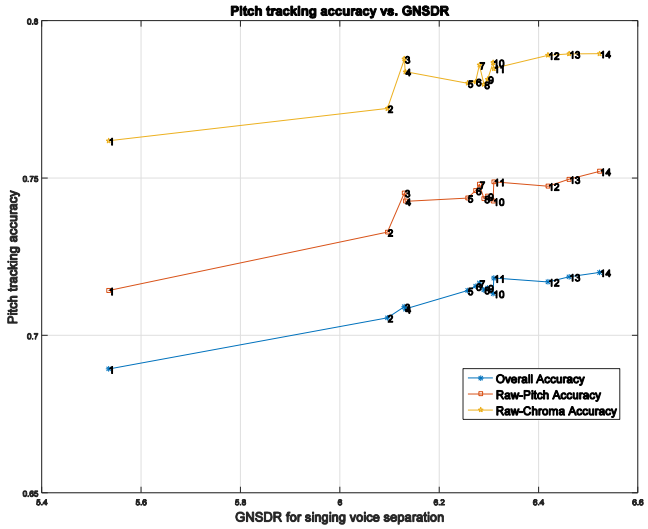

Figure 5. Pitch tracking accuracy (via UPDUDP) vs. vocal extraction performance. (The numbers around dots of lines represent different DNN architectures, as described in TABLE I.)

TABLE I. THE TYPES OF DNN ARCHITECTURE

Indices of DNN architecture	Numbers of nodes in each hidden layer	Numbers of hidden layers
1	64	3
2	128	1
3	128	2
4	256	2
5	1024	1
6	2048	1
7	512	2
8	768	1
9	512	1
10	256	3
11	768	2
12	512	3
13	768	3
14	1024	3

B. Experimental Results

Since the proposed method is composed of two stages of vocal extraction and pitch tracking, our first experiment is used to explore the effect of each stage's accuracy toward the overall accuracy. To this end, different DNN architectures are constructed to have vocal extraction of different GNSDR

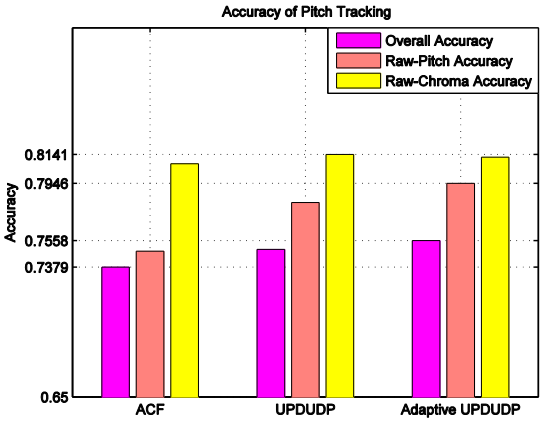

Figure 6. Accuracies of different pitch-tracking methods on the DNN-extracted singing voices.

TABLE II. COMPARISON BETWEEN PROPOSED METHODS AND YEH'S HYBRID APPROACH

	Raw-pitch accuracy	**Raw-chroma accuracy**
Yeh's Approach [8]	82.60%	82.78%
DNN+UPDUDP	82.73%	85.45%
DNN+Adaptive UPDUDP	85.35%	83.73%

indices. As shown in Figure 5, the higher GNSDR is, the better overall SPE accuracy we can obtain. Similar situations also apply to GSIR and GSAR. On the other hand, in our second experiment, with the same DNN architecture, we have tried different pitch-tracking methods, including simple autocorrelation function (ACF), UPDUDP and adaptive-UPDUDP respectively. As shown in Figure 6, the proposed adaptive-UPDUDP achieves the best result. This experiments indicates that any improvement in either vocal extraction or pitch tracking will enhance the overall SPE accuracy. And by dividing SPE into two stages, it is much easier for use to identify which part goes wrong if the SPE result is not desirable.

In the third experiment, we compared two versions of the proposed methods with Yeh's hybrid method [8]. As indicated in Table II, both the proposed versions outperform Yeh's approach in both raw-pitch and raw-chroma accuracies. (We do not use overall accuracy as a performance index in this experiment since Yeh's approach does not perform vocal detection.) In particular, DNN+UPDUDP has gained a large margin of 2.67% (or 15.51% in error rate reduction) in raw-chroma accuracy. When we switched to adaptive UPDUDP, the gain is 2.75% (or 15.8% in error reduction) in raw-pitch accuracy, which is a significant improvement considering the difficulty of SPE. It should be noted that raw-pitch accuracy is a more important performance index than raw-chroma accuracy. Therefore, we can clearly see the advantage offered by adaptive UPDUDP in SPE by increasing the raw-pitch accuracy significantly.

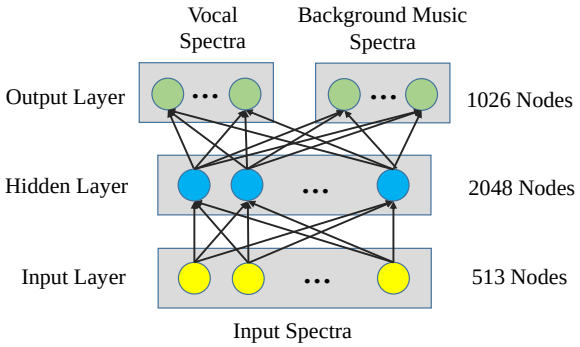

Figure 7. The architecture of FJ2 in the singing voice separation task.

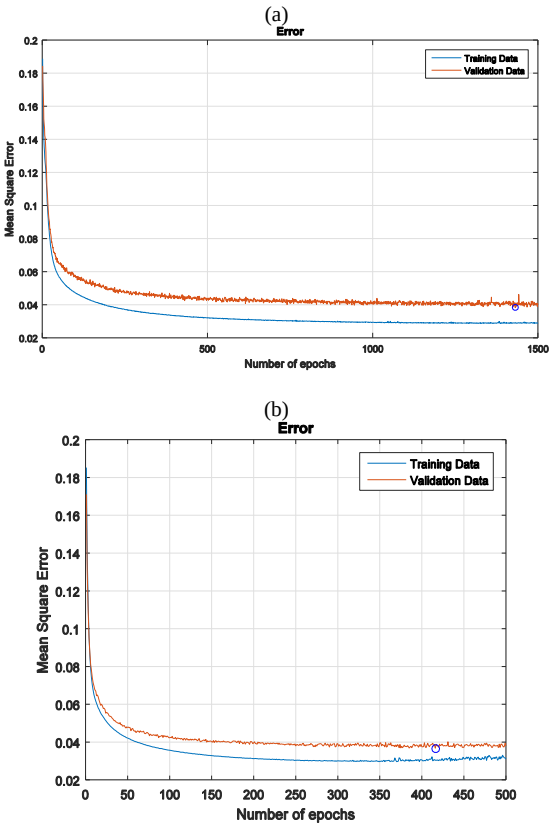

Figure 8. (a) The error profile of FJ1 during training, with the blue and red lines being the error profiles for training and validation data, respectively. The blue circle indicates where the lowest validation error occurs. (b) The error profile of FJ2 during training.

C. *MIREX Contest*

In order to further demonstrate the performance of the proposed methods, we also participated in the MIREX [30] contest in 2015. The proposed methods can be decomposed into two stages of source separation using deep learning and pitch tracking using adaptive-UPDUDP, so we took part in both the tasks of singing voice separation and audio melody extraction. Note that for unbiased evaluation, all submissions to these two tasks are tested on hidden datasets that are not available to the public.

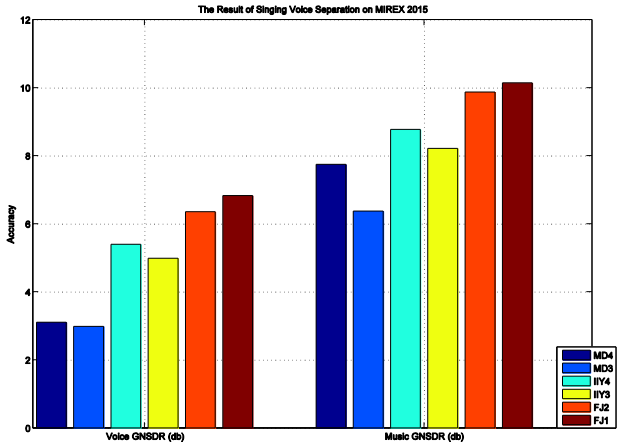

Figure 9. MIREX-2015 results of singing voice separation. (FJ1 and FJ2 are our submissions.)
Demo site of FJ1: http://mirlab.org/demo/singingVoiceSeparation/

In singing voice separation task, we have two submissions FJ1 and FJ2. FJ1 is based on a DNN of 3 hidden layers of 1024 nodes (see Figure 2), while FJ2 is based on a DNN of 1 hidden layer of 2048 nodes, as shown in Figure 7. Both DNNs adopt the sigmoid function, and the objective function is based on Equation 3. The training set for both DNNs is half of the public set of iKala dataset [26], while the other half is used as the validation set. The training set consists of approximately 318,000 frames, each with 513 features of magnitude spectra. The evaluation was performed by MIREX team to test the models on the hidden set of iKala dataset. To prevent overfitting during the process of training, the validation batch is randomly selected from the validation set, and the model was stored whenever the lowest validation error occurred so far during the training process. As shown in Figure 8, to achieve approximately the same error level, the number of epochs of FJ1 is much larger than that of FJ2. This is simply due to the fact that JF1 has a more complex architecture than FJ2. The result of singing voice separation task of MIREX-2015 is shown in Figure 9, where both our submissions FJ1 and FJ2 outperform all the other submissions in both performance indices of voice GNSDR and music GNSDR. In particular, FJ1 performs better than FJ2, indicating that a deeper model architecture with more hidden layers (and inevitably with longer training time) do have advantages over shallow ones, at least for the current task of singing voice separation. Since FJ1 has better performance than FJ2, its DNN architecture was adopted for the following task of audio melody extraction. Demo site of singing voice separation based on FJ1 can be found at http://mirlab.org/demo/singingVoiceSeparation.

For audio melody extraction task, we have two submissions FYJ1 and FYJ4 that are directly related to the proposed methods in this paper. These two submissions differ only in the pitch tracking methods for the extracted singing voices, while the DNN architecture for singing voice separation in stage 1 is the same (as the one of our submission FJ1 to singing voice separation of MIREX-2015, see Figure 2). The training set is the same as experiment 1, with 175 songs sung by one male and one female to have 141,000 frames, each with 513 dimensions of magnitude spectra. After singing voice

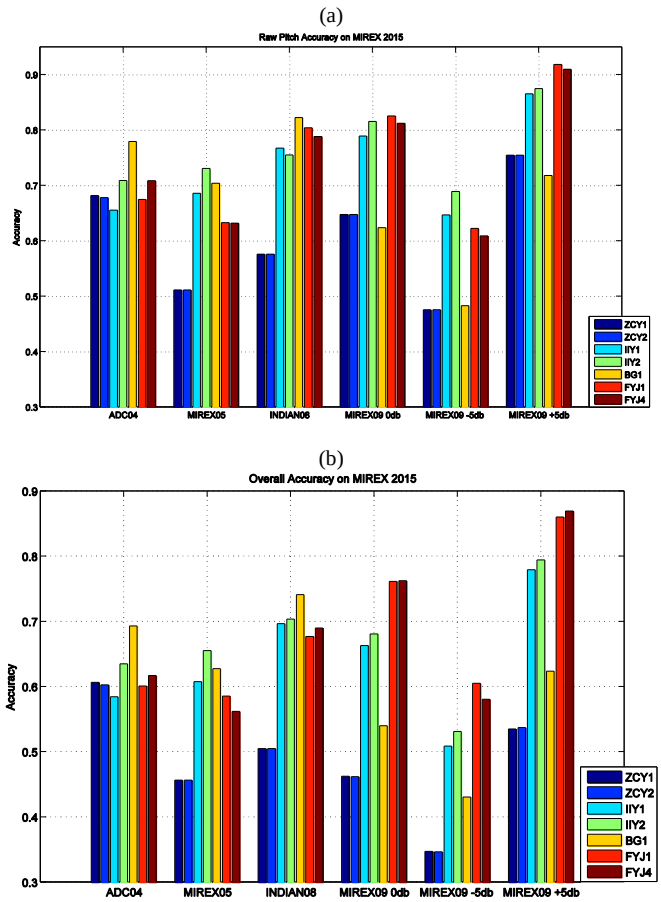

Figure 10. MIREX-2015 results (partial) of audio melody extraction: (a) Raw-pitch accuracy. (b) Overall accuracy. (FYJ1 and FYJ4 are our submissions)

separation, we extracted pitch by using UPDUDP and its adaptive version for FYJ1 and FYJ4, respectively. All the training settings are mostly the same as those used in training DNN for singing voice separation, as mentioned previously. Figure 10 demonstrates the accuracies of all submissions to audio melody extraction task in MIREX 2015, with (a) and (b) being the accuracies based on raw pitch and overall accuracy, respectively. As shown in Figure 10 (b) of the overall accuracy (the most important performance index of audio melody extraction), our submissions outperform all the others for three of the datasets, namely, MIREX 09 at 0db, MIREX 09 at -5db and MIREX 09 at +5db, which indicate the effectiveness of the proposed DNN for singing voice separation and adaptive UPDUDP for robust pitch tracking. For datasets of ADC04 and MIREX05, our submissions are not clear winners since, as explained by the MIREX webpage, the datasets contain music without lead vocals, which are not the target of the proposed methods. In other words, the proposed methods aim to deal with mixture music with lead vocal, such as those tracks in MIREX 09. For INDIANA08 dataset, it is not clear to us if the dataset contains music without lead vocal or not, but our submissions still have above-average performance. (In fact there is another dataset ORCH used in audio melody extraction task of MIREX 2015. But we do not list it here since the dataset consists of orchestra music without lead vocals at all.)

CONCLUSIONS AND FUTURE WORK

In this paper, we have combined DNN for singing voice separation and adaptive UPDUDP for pitch tracking to achieve the final goal of SPE for monaural polyphonic music. The experimental results demonstrate a significant overall error reduction rate of 15.8% in raw-pitch accuracy when compared with the previous state-of-the-art approach. More, the results of 2015 MIREX shows the proposed methods outperform other submissions in both the tasks of singing voice separation and audio melody extraction, indicating the effectiveness of the proposed methods.

Since the proposed approach is based on singing voice separation and pitch tracking, improvement in either aspect will enhance the accuracy of SPE. Our immediate future work will be focused on improving singing voice separation, which seems to be a better paid-off task than pitch tracking. To this end, we shall try other types of DNN for singing voice separation, such as recurrent neural networks or convolutional neural networks or combinations of different DNN architectures. Moreover, we shall try to use the proposed methods for several challenging tasks in music retrieval, including cover song identification and query by singing/humming based on audio database.

REFERENCES

[1] M. Goto, "A Real-Time Music Scene Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals," Speech Communication, vol. 43, no. 4, pp. 311-329, 2004.

[2] J. Salamon and E. G´omez, "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics," IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, pp.1759-1770, 2012.

[3] L. Regnier and G. Peeters, "Singing Voice Detection in Music Tracks Using Direct Voice Vibrato Detection," in IEEE Int. Conf. on International Conference on Acoustics, Speech, and Signal Processing, pp. 1685-1688, 2009.

[4] C. L Hsu, L. Y. Chen, J. S. Jang, and S. J. Li, "Singing Pitch Extraction from Monaural Polyphonic Songs by Contextual Audio Modeling and Singing Harmonic Enhancement," in Proc. 10th International Society for Music Information Retrieval (ISMIR), pp.201-206, 2009.

[5] C. L. Hsu and J. S. Jang, "Singing Pitch Extraction by Voice Vibrato/Tremolo Estimation and Instrument Partial Deletion," in Proc. 11th International Society for Music Information Retrieval (ISMIR), pp.525-530, 2010.

[6] J. Salamon and E. G´omez, D. P.W. Ellis and G. Richard, "Melody Extraction from Polyphonic Music Signals: Approaches, Applications and Challenges," IEEE Signal Processing Magazine, 31(2):118-134, 2014.

[7] C. L. Hsu, D. Wang, and J. S. Jang, "A Trend Estimation Algorithm for Singing Pitch Detection in Musical Recordings," in IEEE Int. Conf. on International Conference on Acoustics, Speech, and Signal Processing, 2011.

[8] T.-C. Yeh, M.-J. Wu, J.-S. Jang, W.-L. Chang, and I.-B. Liao, "A Hybrid Approach to Singing Pitch Extraction Based on Trend Estimation and Hidden Markov Models," in IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 457-460, 2012.

[9] C. L. Hsu and J. S. Jang, "On the Improvement of Singing Voice Separation for Monaural Recordings Using the MIR-1K Dataset," IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, pp.310-319, 2010.

[10] C. L. Hsu, D. Wang, J. S. Jang and K. Hu "A Tandem Algorithm for Singing Pitch Extraction and Voice Separation From Music Accompaniment," IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, pp.1482-1490, 2012.

[11] A. L. Mass, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng. "Recurrent Neural Networks for Noise Reduction in Robust ASR," in INTERSPEECH, 2012.

[12] S. Tamura and A. Waibel, "Noise Reduction Using Connectionist Models," in IEEE Int. Conf. on International Conference on Acoustics, Speech, and Signal Processing, 1988, pp. 553–556.

[13] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," in IEEE Signal Processing Magazine, 29:82–97, Nov. 2012

[14] P. S. Huang, M. Kim, M. Hasegawa-Johnson and P. Smaragdis, "Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation," in IEEE Transactions on Audio, Speech, and Language Processing, 2015.

[15] J. C. Chen and J.S. Jang, "TRUES: Tone Recognition Using Extended Segments," in ACM Transactions on Asian Language Information Processing, No. 10, Vol. 7, Aug 2008.

[16] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg and H. J. Manley, "Average Magnitude Difference Function Pitch Extractor," in IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-22, NO. 5, pp353-362, Oct, 1974.

[17] D.E. Rumelhart, G.E. Hinton, and R.J.Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533–536, 1986.

[18] H. Robinds and S. Monro, "A stochastic approximation method," Annals of Mathematical Statistics, vol. 22, pp.400–407, 1951.

[19] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. "On the Importance of Initialization and Momentum in Deep Learning," In Proceedings of the 30th International Conference on Machine Learning, ICML 2013.

[20] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online leaning and stochastic optimization," in COLT, 2010.

[21] T. Tieleman and G. Hinton, "Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning," 2012.

[22] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," arXiv:1212.5701, 2012.

[23] Y. Nesterov, "A Method of Solving a Convex Programming Problem with Convergence Rate O(1/sqr(k))," Soviet Mathematics Doklady, 27:372–376, 1983.

[24] E. Vincent, R. Gribonval, and C. F` evotte, "Performance measurement in blind audio source separation," in IEEE Trans. Audio, Speech & Language Processing, vol. 16, no. 4, pp. 766–778, 2008.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, pages 1929–1958, 2014.

[26] T. S. Chan, T. C. Yeh, Z. C. Fan, H. W. Chen, L. Su, Y. H. Yang and R. Jang, "Vocal Activity Informed Singing Voice Separation with the IKala Dataset," in IEEE Int. Conf. on International Conference on Acoustics,Speech, and Signal Processing, pp. 718-722, 2015.

[27] P. S. Huang, M. Kim, M. Hasegawa-Johnson and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in Proc. 15th International Society for Music Information Retrieval (ISMIR), 2014.

[28] U. Stefan, F. Giron and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2015.

[29] 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. 15th International Society for Music Information Retrieval Conference (ISMIR 2014), Taipei, Taiwan, Oct. 2014.

[30] Music Information Retrieval Evaluation eXchange (MIREX) : http://www.music-ir.org/mirex/wiki/2015:Main_Page.