SINGING VOICE MELODY TRANSCRIPTION USING DEEP NEURAL NETWORKS

François Rigaud and Mathieu Radenen
Audionamix R&D
171 quai de Valmy, 75010 Paris, France
<firstname>.<lastname>@audionamix.com

ABSTRACT

This paper presents a system for the transcription of singing voice melodies in polyphonic music signals based on Deep Neural Network (DNN) models. In particular, a new DNN system is introduced for performing the f_0 estimation of the melody, and another DNN, inspired from recent studies, is learned for segmenting vocal sequences. Preparation of the data and learning configurations related to the specificity of both tasks are described. The performance of the melody f_0 estimation system is compared with a state-of-the-art method and exhibits highest accuracy through a better generalization on two different music databases. Insights into the global functioning of this DNN are proposed. Finally, an evaluation of the global system combining the two DNNs for singing voice melody transcription is presented.

1. INTRODUCTION

The automatic transcription of the main melody from polyphonic music signals is a major task of Music Information Retrieval (MIR) research [19]. Indeed, besides applications to musicological analysis or music practice, the use of the main melody as prior information has been shown useful in various types of higher-level tasks such as music genre classification [20], music retrieval [21], music desoloing [4, 18] or lyrics alignment [15, 23]. From a signal processing perspective, the main melody can be represented by sequences of fundamental frequency (f_0) defined on voicing instants, *i.e.* on portions where the instrument producing the melody is active. Hence, main melody transcription algorithms usually follow two main processing steps. First, a representation emphasizing the most likely f_0s over time is computed, *e.g.* under the form of a salience matrix [19], a vocal source activation matrix [4] or an enhanced spectrogram [22]. Second, a binary classification of the selected f_0s between melodic and background content is performed using melodic contour detection/tracking and voicing detection.

In this paper we propose to tackle the melody transcription task as a supervised classification problem where each time frame of signal has to be assigned into a pitch class when a melody is present and an 'unvoiced' class when it is not. Such approach has been proposed in [5] where melody transcription is performed applying Support Vector Machine on input features composed of Short-Time Fourier Transforms (STFT). Similarly for noisy speech signals, f_0 estimation algorithms based on Deep Neural Networks (DNN) have been introduced in [9, 12].

Following such fully data driven approaches we introduce a singing voice melody transcription system composed of two DNN models respectively used to perform the f_0 estimation task and the Voice Activity Detection (VAD) task. The main contribution of this paper is to present a DNN architecture able to discriminate the different f_0s from low-level features, namely spectrogram data. Compared to a well-known state-of-the-art method [19], it shows significant improvements in terms of f_0 accuracy through an increase of robustness with regard to musical genre and a reduction of octave-related errors. By analyzing the weights of the network, the DNN is shown somehow equivalent to a simple harmonic-sum method for which the parameters usually set empirically are here automatically learned from the data and where the succession of non-linear layers likely increases the power of discrimination of harmonically-related f_0. For the task of VAD, another DNN model, inspired from [13] is learned. For both models, special care is taken to prevent over-fitting issues by using different databases and perturbing the data with audio degradations. Performance of the whole system is finally evaluated and shows promising results.

The rest of the paper is organized as follows. Section 2 presents an overview of the whole system. Sections 3 and 4 introduce the DNN models and detail the learning configurations respectively for the VAD and the f_0 estimation task. Then, Section 5 presents an evaluation of the system and Section 6 concludes the study.

2. SYSTEM OVERVIEW

2.1 Global architecture

The proposed system, displayed on Figure 1, is composed of two independent parallel DNN blocks that perform respectively the f_0 melody estimation and the VAD.

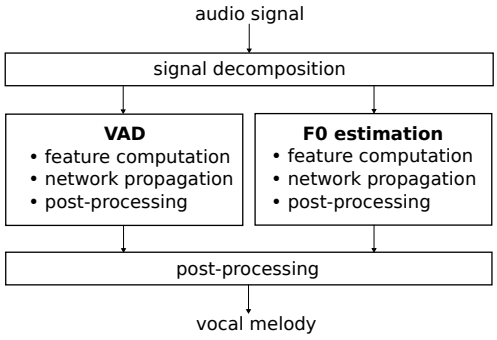


Figure 1: Architecture of the proposed system for singing voice melody transcription.

In contrast with [9,12] that propose a single DNN model to perform both tasks, we did not find such unified functional architecture able to discriminate successfully a time frame between quantified f_0s and 'unvoiced' classes. Indeed the models presented in these studies are designed for speech signals mixed with background noise for which the discrimination between a frame of noise and a frame of speech is very likely related to the presence or absence of a pitched structure, which is also probably the kind of information on which the system relies to estimate the f_0. Conversely, with music signals both the melody and the accompaniment exhibit harmonic structures and the voicing discrimination usually requires different levels of information, *e.g.* under the form of timbral features such as Mel-Frequency Cepstral Coefficients.

Another characteristic of the proposed system is the parallel architecture that allows considering different types of input data for the two DNNs and which arises from the application restricted to vocal melodies. Indeed, unlike generic systems dealing with main melody transcription of different instruments (often within a same piece of music) which usually process the f_0 estimation and the voicing detection sequentially, the focus on singing voice here hardly allows for a voicing detection relying only on the distribution and statistics of the candidate pitch contours and/or their energy [2, 19]. Thus, this constraint requires to build a specific VAD system that should learn to discriminate the timbre of a vocal melody from an instrumental melody, such as for example played by a saxophone.

2.2 Signal decomposition

As shown on Figure 1, both DNN models are preceded by a signal decomposition. At the input of the global system, audio signals are first converted to mono and re-sampled to 16 kHz. Then, following [13], it is proposed to provide the DNNs with a set of pre-decomposed signals obtained by applying a double-stage Harmonic/Percussive Source Separation (HPSS) [6,22] on the input mixture signal. The key idea behind double-stage HPSS is to consider that within a mix, melodic signals are usually less stable/stationary than the background 'harmonic' instruments (such as a bass or a piano), but more than the percussive instruments (such as the drums). Thus, according to the frequency resolution that is used to compute a STFT, applying a harmonic/percussive decomposition on a mixture spectrogram lead to a rough separation where the melody is mainly extracted either in the harmonic or in the percussive content.

Using such pre-processing, 4 different signals are obtained. First, the input signal s is decomposed into the sum of h_1 and p_1 using a high-frequency resolution STFT (typically with a window of about 300 ms) where p_1 should mainly contain the melody and the drums, and h_1 the remaining stable instrument signals. Second, p_1 is further decomposed into the sum of h_2 and p_2 using a low-frequency resolution STFT (typically with a window of about 30 ms), where h_2 mainly contains the melody, and p_2 the drums. As presented latter in Sections 3 and 4, different types of these 4 signals or combinations of them will be used to experimentally determine optimal DNN models.

2.3 Learning data

Several annotated databases composed of polyphonic music with transcribed melodies are used for building the train, validation and test datasets used for the learning (*cf.* Sections 3 and 4) and the evaluation (*cf.* Section 5) of the DNNs. In particular, a subset of RWC Popular Music and Royalty Free Music [7] and MIR-1k [10] databases are used for the train dataset, and the recent databases MedleyDB [1] and iKala [3] are split between train, validation and test datasets. Note that for iKala the vocal and instrumental tracks are mixed with a relative gain of 0 dB.

Also, in order to minimize over-fitting issues and to increase the robustness of the system with respect to audio equalization and encoding degradations, we use the Audio Degradation Toolbox [14]. Thus, several files composing the train and validation datasets (50% for the VAD task and 25% for the f_0 estimation task) are duplicated with one degraded version, the degradation type being randomly chosen amongst those available preserving the alignment between the audio and the annotation (*e.g.* not producing time/pitch warping or too long reverberation effects).

3. VOICE ACTIVITY DETECTION WITH DEEP NEURAL NETWORKS

This section briefly describes the process for learning the DNN used to perform the VAD. It is largely inspired from a previous study presented in more detail in [13]. A similar architecture of deep recurrent neural network composed of Bidirectional Long Short-Term Memory (BLSTM) [8] is used. In our case the architecture is arbitrarily fixed to 3 BLSTM layers of 50 units each and a final feed-forward logistic output layer with one unit. As in [13], different types of combination of the pre-decomposed signals (*cf.* Section 2.2) are considered to determine an optimal network: s, p_1, h_2, h_1p_1, h_2p_2 and $h_1h_2p_2$. For each of these pre-decomposed signals, timbral features are computed under the form of mel-frequency spectrograms obtained using a STFT with 32 ms long Hamming windows and 75 % of overlap, and 40 triangular filters distributed on a mel scale between 0 and 8000 Hz. Then, each feature of the input

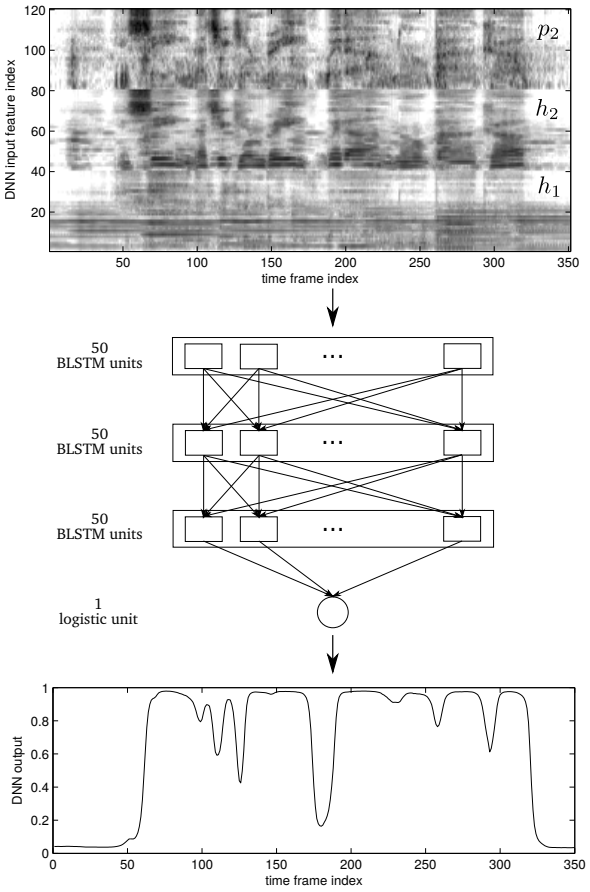


Figure 2: VAD network illustration.

data is normalized using the mean and variance computed over the train dataset. Contrary to [13] the learning is performed in a single step, *i.e.* without adopting a layer by layer training.

Finally, the best architecture is obtained for the combination of h_1, h_2 and p_2 signals, thus for an input of size 120, which corresponds to a use of the whole information present in the original signal ($s = h_1 + h_2 + p_2$). An illustration of this network is presented in Figure 2.

A simple post-processing of the DNN output consisting in a threshold of 0.5 is finally applied to take the binary decision of voicing frame activation.

4. F_0 ESTIMATION WITH DEEP NEURAL NETWORKS

This section presents in detail the learning configuration for the DNN used for performing the f_0 estimation task. An interpretation of the network functioning is finally presented.

4.1 Preparation of learning data

As proposed in [5] we decide to keep low level features to feed the DNN model. Compared to [12] and [9] which use as input pre-computed representations known for highlighting the periodicity of pitched sounds (respectively based on an auto-correlation and a harmonic filtering), we expect here the network to be able to learn an optimal transformation automatically from spectrogram data. Thus the set of selected features consists of log-spectrograms (logarithm of the modulus of the STFT) computed from a Hamming window of duration 64 ms (1024 samples for a sampling frequency of 16000 Hz) with an overlap of 0.75, and from which frequencies below 50 Hz and above 4000 Hz are discarded. For each music excerpt the corresponding log-spectrogram is rescaled between 0 and 1. Since, as described in Section 2.1, the VAD is performed by a second independent system, all time frames for which no vocal melody is present are removed from the dataset. These features are computed independently for 3 different types of input signal for which the melody should be more or less emphasized: s, p_1 and h_2 (*cf.* Section 2.2).

For the output, the f_0s are quantified between C#2 ($f_0 \simeq 69.29$ Hz) and C#6 ($f_0 \simeq 1108.73$ Hz) with a spacing of an eighth of tone, thus leading to a total of 193 classes.

The train and validation datasets including audio degraded versions are finally composed of, respectively, 22877 melodic sequences (resp. 3394) for a total duration of about 220 minutes (resp. 29 min).

4.2 Training

Several experiments have been run to determine a functional DNN architecture. In particular, two types of neuron units have been considered: the standard feed-forward sigmoid unit and the Bidirectional Long Short-Term Memory (BLSTM) recurrent unit [8].

For each test, the weights of the network are initialized randomly according to a Gaussian distribution with 0 mean and a standard deviation of 0.1, and optimized to minimize the cross-entropy error function. The learning is then performed by means of a stochastic gradient descent with shuffled mini-batches composed of 30 melodic sequences, a learning rate of 10^{-7} and a momentum of 0.9. The optimization is run for a maximum of 10000 epochs and an early stopping is applied if no decrease is observed on the validation set error during 100 consecutive epochs. In addition to the use of audio degradations during the preparation of the data for preventing over-fitting (*cf.* Section 2.3), the training examples are slightly corrupted during the learning by adding a Gaussian noise with variance 0.05 at each epoch.

Among the different architectures tested, the best classification performance is obtained for the input signal p_1 (slightly better than for s, *i.e.* without pre-separation) by a 2-hidden layer feed-forward network with 500 sigmoid units each, and a 193 output softmax layer. An illustration of this network is presented in Figure 3. Interestingly, for that configuration the learning did not suffered from overfitting so that it ended at the maximum number of epochs, thus without early stopping.

While the temporal continuity of the f_0 along timeframes should provide valuable information, the use of BLSTM recurrent layers (alone or in combination with

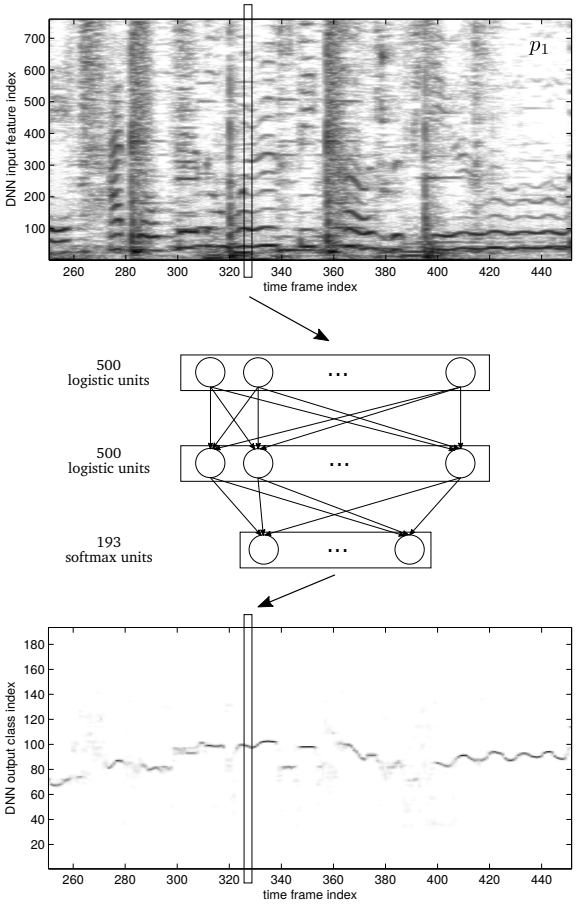


Figure 3: f_0 estimation network illustration.

feed-forward sigmoid layers) did not lead to efficient systems. Further experiments should be conducted to enforce the inclusion of such temporal context in a feed-forward DNN architecture, for instance by concatenating several consecutive time frames in the input.

4.3 Post-processing

The output layer of the DNN composed of softmax units returns a f_0 probability distribution for each time frame that can be seen for a full piece of music as a pitch activation matrix. In order to take a final decision that account for the continuity of the f_0 along melodic sequences, a Viterbi tracking is finally applied on the network output [5, 9, 12]. For that, the log-probability transition between two consecutive time frames and two f_0 classes is simply arbitrarily set inversely proportional to their absolute difference in semi-tones. For further improvement of the system, such transition matrix could be learned from the data [5], however this simple rule gives interesting performance gains (when compared to a simple 'maximum picking' post-processing without temporal context) while potentially reducing the risk of over-fitting to a particular music style.

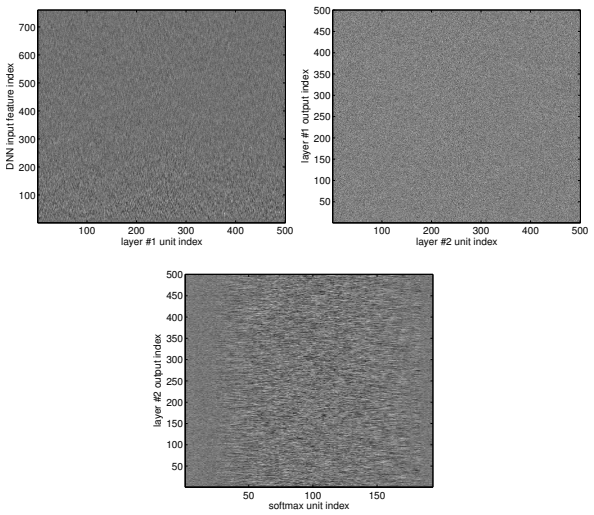


Figure 4: Display of the weights for the two sigmoid feed-forward layers (top) and the softmax layer (down) of the DNN learned for the f_0 estimation task.

4.4 Network weights interpretation

We propose here to have an insight into the network functioning for this specific task of f_0 estimation by analyzing the weights of the DNN. The input is a short-time spectrum and the output corresponds to an activation vector for which a single element (the actual f_0 of the melody at that time frame) should be predominant. In that case, it is reasonable to expect that the DNN somehow behaves like a harmonic-sum operator.

While the visualization of the distribution of the hidden-layer weights usually does not provide with straightforward cues to analyse a DNN functioning (*cf.* Figure 4) we consider a simplified network for which it is assumed that each feed-forward logistic unit is working in the linear regime. Thus, removing the non-linear operations, the output of a feed-forward layer with index l composed of N_l units writes

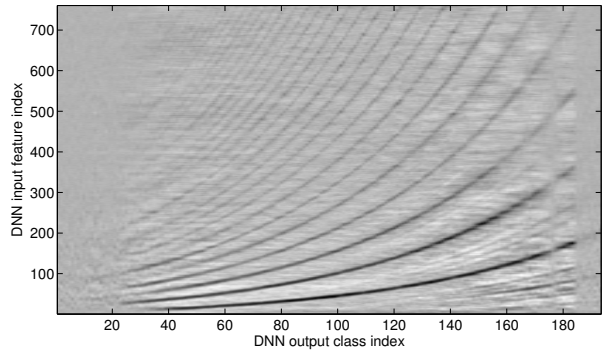
$$x_l = W_l \cdot x_{l-1} + b_l, \tag{1}$$

where $x_l \in \mathbb{R}^{N_l}$ (resp. $x_{l-1} \in \mathbb{R}^{N_{l-1}}$) corresponds to the ouput vector of layer l (resp. $l-1$), $W_l \in \mathbb{R}^{N_l \times N_{l-1}}$ is the weight matrix and $b_l \in \mathbb{R}^{N_l}$ the bias vector. Using this expression, the output of a layer with index L expressed as the propagation of the input x_0 through the linear network also writes

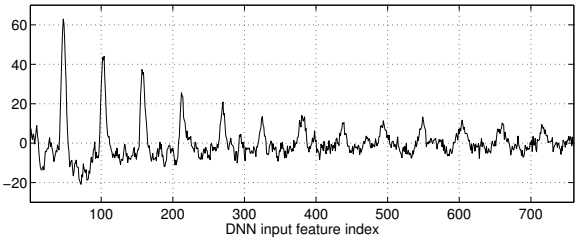
$$x_L = \mathbf{W} \cdot x_0 + \mathbf{b}, \tag{2}$$

where $\mathbf{W} = \prod_{l=1}^{L} W_l$ corresponds to a global weight matrix, and $\mathbf{b}$ to a global bias that depends on the set of parameters $\{W_l, b_l, \forall l \in [1..L]\}$.

As mentioned above, in our case x_0 is a short-time spectrum and x_L is a f_0 activation vector. The global weight matrix should thus present some characteristics of a pitch detector. Indeed as displayed on Figure 5a, the matrix $\mathbf{W}$ for the learned DNN (which is thus the product of the 3 weight matrices depicted on Figure 4) exhibits an harmonic structure for most output classes of f_0s; except for



(a)



(b)

Figure 5: Linearized DNN illustration. (a) Visualization of the (transposed) weight matrix $\mathbf{W}$. The x-axis corresponds to the output class indices (the f_0s) and the y-axis represents the input feature indices (frequency channel of the spectrum input). (b) Weights display for the f_0 output class with index 100.

some f_0s in the low and high frequency range for which no or too few examples are present in the learning data.

Most approaches dealing with main melody transcription usually relies on such types of transformations to compute a representation emphasizing f_0 candidates (or salience function) and are usually partly based on handcrafted designs [11, 17, 19]. Interestingly, using a fully data driven method as proposed, parameters of a comparable weighted harmonic summation algorithm (such as the number of harmonics to consider for each note and their respective weights) do not have to be defined. This can be observed in more details on Figure 5b which depicts the linearized network weights for the class index 100 ($f_0 \simeq$ 289.43 Hz). Moreover, while this interpretation assumes a linear network, one can expect that the non-linear operations actually present in the network help in enhancing the discrimination between the different f_0 classes.

5. EVALUATION

5.1 Experimental procedure

Two different test datasets composed of full music excerpts (*i.e.* vocal and non vocal portions) are used for the evaluation. One is composed of 17 tracks from MedleyDB (last songs comprising vocal melodies, from *MusicDelta_Reggae* to *Wolf_DieBekherte*, for a total of $\sim$ 25.5 min of vocal portions) and the other is composed of 63 tracks from iKala (from *54223_chorus* to *90587_verse* for a total of $\sim$ 21 min of vocal portions).

The evaluation is conducted in two steps. First the performance of the f_0 estimation DNN taken alone (thus without voicing detection) is compared with the state-of the art system *melodia* [19] using f_0 accuracy metrics. Second, the performance of our complete singing voice transcription system (VAD and f_0 estimation) is evaluated on the same datasets. Since our system is restricted to the transcription of vocal melodies and that, to our knowledge all available state-of-the-art systems are designed to target main melody, this final evaluation presents the results for our system without comparisons with a reference.

For all tasks and systems, the evaluation metrics are computed using the *mir_eval* library [16]. For Section 5.3, some additional metrics related to voicing detection, namely precision, f-measure and voicing accuracy, were not present in the original *mir_eval* code and thus were added for our experiments.

5.2 f_0 estimation task

The performance of the DNN performing the f_0 estimation task is first compared to *melodia* system [19] using the plug-in implementation with f_0 search range limits set equal to those of our system (69.29-1108.73 Hz, *cf.* Sec. 4.1) and with remaining parameters left to default values. For each system and each music track the performance is evaluated in terms of raw pitch accuracy (RPA) and raw chroma accuracy (RCA). These metrics are computed on vocal segments (*i.e.* without accounting for potential voicing detection errors) for a f_0 tolerance of 50 cents.

The results are presented on Figure 6 under the form of a box plot where, for each metric and dataset, the ends of the dashed vertical bars delimit the lowest and highest scores obtained, the 3 vertical bars composing each center box respectively correspond to the first quartile, the median and the third quartile of the distribution, and finally the star markers represent the mean. Both systems are characterized by more widespread distributions for MedleyDB than for iKala. This reflects the fact that MedleyDB is more heterogeneous in musical genres and recording conditions than iKala. On iKala, the DNN performs slightly better than *melodia* when comparing the means. On MedleyDB, the gap between the two systems increases significantly. The DNN system seems much less affected by the variability of the music examples and clearly improve the mean RPA by 20% (62.13% for *melodia* and 82.48% for the DNN). Additionally, while exhibiting more similar distributions of RPA and RCA, the DNN tends to produce less octave detection errors. It should be noted that this result does not take into account the recent post-processing improvement proposed for *melodia* [2], yet it shows the interest of using such DNN approach to compute an enhanced pitch salience matrix which, simply combined with a Viterbi post-processing, achieves good performance.

5.3 Singing voice transcription task

The evaluation of the global system is finally performed on the two same test datasets. The results are displayed as

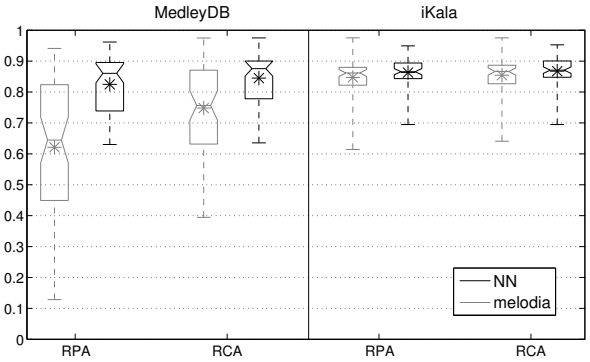


Figure 6: Comparative evaluation of the proposed DNN (in black) and *melodia* (in gray) on MedleyDB (left) and iKala (right) test sets for a f_0 vocal melody estimation task.

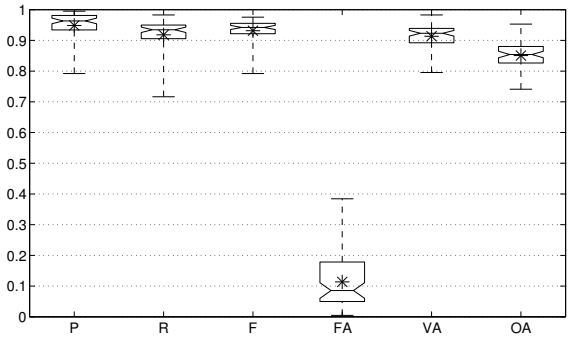
boxplots (*cf.* description Section 5.2) on Figures 7a and 7b respectively for the iKala and the MedleyDB datasets. Five metrics are computed to evaluate the voicing detection, namely the precision (P), the recall (R), the f-measure (F), the false alarm rate (FA) and the voicing accuracy (VA). A sixth metric of overall accuracy (OA) is also presented for assessing the global performance of the complete singing voice melody transcription system.

In accordance with the previous evaluation, the results on MedleyDB are characterized by much more variance than on iKala. In particular, the voicing precision of the system (*i.e.* it's ability to provide correct detections, no matter the number of forgotten voiced frames) is significantly degraded on MedleyDB. Conversely, the voicing recall which evaluate the ability of the system to detect all voiced portions actually present no matter the number of false alarm, remains relatively good on MedleyDB. Combining both metrics, a mean f-measure of 93.15 % and 79.19 % are respectively obtained on iKala and MedleyDB test datasets.

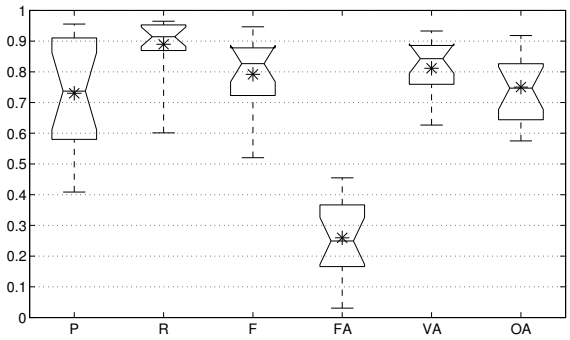
Finally, the mean scores of overall accuracy obtained for the global system are equal to 85.06 % and 75.03 % respectively for iKala and MedleyDB databases.

6. CONCLUSION

This paper introduced a system for the transcription of singing voice melodies composed of two DNN models. In particular a new system able to learn a representation emphasizing melodic lines from low level data composed of spectrograms has been proposed for the estimation of the f_0. For this DNN, the performance evaluation shows a relatively good generalization (when compared to a reference system) on two different test datasets and an increase of robustness to western music recordings that tend to be representative of the current music industry productions. While for these experiments the systems have been learned from a relatively low amount of data, the robustness, particularly for the task of VAD, could very likely be improved by increasing the number of training examples.



(a) iKala test dataset



(b) MedleyDB test dataset

Figure 7: Voicing detection and overall performance of the proposed system for iKala and MedleyDB test datasets.

7. REFERENCES

[1] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello. MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *Proc. of the 15th Int. Society for Music Information Retrieval (ISMIR) Conference*, October 2014.

[2] R. M. Bittner, J. Salamon, S. Essid, and J. P. Bello. Melody extraction by contour classification. In *Proc. of the 16th Int. Society for Music Information Retrieval (ISMIR) Conference*, October 2015.

[3] 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 *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 718–722, April 2015.

[4] J.-L. Durrieu, G. Richard, and B. David. An iterative approach to monaural musical mixture de-soloing. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 105–108, April 2009.

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

[6] D. FitzGerald and M. Gainza. Single channel vocal separation using median filtering and factorisation techniques. *ISAST Trans. on Electronic and Signal Processing*, 4(1):62–73, 2010.

[7] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proc. of the 3rd Int. Society for Music Information Retrieval (ISMIR) Conference*, pages 287–288, October 2002.

[8] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649, May 2013.

[9] K. Han and DL. Wang. Neural network based pitch tracking in very noisy speech. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 22(12):2158–2168, October 2014.

[10] C.-L. Hsu and J.-S. R. Jang. On the improvement of singing voice separation for monaural recordings using the mir-1k dataset. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(2):310–319, 2010.

[11] S. Jo, S. Joo, and C. D. Yoo. Melody pitch estimation based on range estimation and candidate extraction using harmonic structure model. In *Proc. of INTERSPEECH*, pages 2902–2905, 2010.

[12] B. S. Lee and D. P. W. Ellis. Noise robust pitch tracking by subband autocorrelation classification. In *Proc. of INTERSPEECH*, 2012.

[13] S. Leglaive, R. Hennequin, and R. Badeau. Singing voice detection with deep recurrent neural networks. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, April 2015.

[14] M. Mauch and S. Ewert. The audio degradation toolbox and its application to robustness evaluation. In *Proc. of the 14th Int. Society for Music Information Retrieval (ISMIR) Conference*, November 2013.

[15] A. Mesaros and T. Virtanen. Automatic alignment of music audio and lyrics. In *Proc. of 11th Int. Conf. on Digital Audio Effects (DAFx)*, September 2008.

[16] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. mir_eval: a transparent implementation of common MIR metrics. In *Proc. of the 15th Int. Society for Music Information Retrieval (ISMIR) Conference*, October 2014.

[17] M. Ryynänen and A. P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.

[18] M. Ryynänen, T. Virtanen, J. Paulus, and A. Klapuri. Accompaniment separation and karaoke application based on automatic melody transcription. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo*, pages 1417–1420, April 2008.

[19] J. Salamon, E. Gómez, 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, March 2014.

[20] J. Salamon, B. Rocha, and E. Gómez. Musical genre classification using melody features extracted from polyphonic music. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 81–84, March 2012.

[21] J. Salamon, J. Serrà, and E. Gómez. Tonal representations for music retrieval: From version identification to query-by-humming. *Int. Jour. of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, 2(1):45–58, 2013.

[22] H. Tachibana, T. Ono, N. Ono, and S. Sagayama. Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 425–428, March 2010.

[23] C. H. Wong, W. M. Szeto, and K. H. Wong. Automatic lyrics alignment for Cantonese popular music. *Multimedia Systems*, 4-5(12):307–323, 2007.