Deep Learning Models for Melody Perception: An Investigation on Symbolic Music Data

Wei-Tsung Lu and Li Su
Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: lisu@iis.sinica.edu.tw Tel/Fax: +886-2-27883799/+886-2-27824814

***Abstract*—We investigate the deep learning approaches on the melody extraction problem on symbolic music data. Specifically, we compare two different approaches: the first one employs recurrent neural networks (RNN) by considering melody extraction as a sequence prediction problem, while the second employs fully convolutional networks (FCN) by considering it as a image semantic segmentation problem. Both methods are tested against a MIDI dataset with melody tracks acting as ground truth. A more challenging case that the melodies are shifted by one octave is also considered. Evaluation results show the advantage of the semantic segmentation approach in terms of the accuracy.**

I. Introduction

Melody is the essence of music.[1] A music piece can be efficiently summarized, and thereby uniquely identified through the elements (e.g., theme, motif) in its melody, while its accompaniment parts such as the chord progression are secondary to the melody.[2] Because of this, melody has long been considered as an important subject in the research of music information retrieval (MIR). For example, melody has been used as a convenient query for searching in large-scale music database, such as query-by-humming [1] with audio input and snippet search [2], [3] with symbolic input. Moreover, pitch contours in music are a general facility for active music discovery [4].

To this end, a wide class of problems related to *melody extraction* in polyphonic music data is of great importance. Related MIR tasks include audio melody extraction [5]–[11], audio note tracking [12]–[15], symbolic voice separation [16]–[23], to name but a few. Although with development for decades, there is still plenty of room for improvement in melody extraction, probably because our perception of melody is such an intricate cognition process that encompasses several types of musical information (e.g., time, space, pitch, dynamics, timbre, etc.) embedded in the data with different modalities including audio and symbolic (e.g., MIDI), and governed by various descriptive perceptual rules (e.g., temporal continuity, pitch proximity, etc.) [24]. Illusion effects, such as Deutsch's scale illusion [25], and high-level cognition effects, such as expectancy/attention [26], and dynamic pattern structures [27] further complicate the process of melody perception. As a result, when discussing the difference between melody and accompaniment (or harmony) in music, we usually retreat to use metaphoric rather than exact descriptions, such as foreground and background, surface and structure, and others.

Most of the studies on melody extraction in these years have been focusing on audio melody extraction.In this problem, solutions typically takes the pitch range, loudness, timbre and interpretation factors (e.g., portamento, vibrato) of the estimated pitch contours the system can not only rely on pitch but also on timbre and interpretation factors (e.g., vibrato), which can be more related to the melody. On the other hand, less efforts can be paid on symbolic-level melody extraction, the problem that aims at extracting melodies from the symbolic data, given only the pitch structure of music but without any audio-level cues. Such a task is important as doing this can simulate the high-level cognitive processes of melody in our brain and will mark an important step toward music language understanding. Most of the previous studies on symbolic-level melody extraction were based on hand-crafted rules, which might be limited in capturing musically-interesting ideas. Learning-based approaches to symbolic melody extraction are still rarely investigated, except some of the recent works using probabilistic modeling [16], [17] or neural networks [18].

With the great success of deep learning in processing high-level semantics for pattern recognition in recent years, we revisit the problem of symbolic melody extraction in this paper. In particular, we consider advanced deep learning models from two different perspectives to tackle this problem. The first model is a revision from the DeepBach model [28], a state-of-the-art music generation model built with long-short-term-memory recurrent neural networks (LSTM-RNN). The second model, constructed with fully convolution networks (FCN), is based on DeepLabV3 and its improved version, DeepLabV3+ [29], [30], which are the state-of-the-art models for semantic segmentation of images. That means, the first model solves melody extraction from the perspective of sequence prediction, while the second model solves it from the perspective of semantic segmentation. We will give a systematic investigation on both perspectives and compare them according to the experiment results. Source code is available on-line[3].

[1]Quoted from the famous words of Wolfgang Amadeus Mozart: "Melody is the essence of music. I compare a good melodist to a fine racer, and counterpointists to hack post-horses; therefore be advised, let well alone and remember the old Italian proverb: Who knows most, knows least."

[2]Unless otherwise specified, the musical texture we discuss in this paper is the homophonic music, i.e., the music composed of *one* predominant melody and accompaniment.

[3]https://github.com/s603122001/Vocal-Melody-Extraction

II. Related work

In symbolic musical data processing, the melody extraction task can be regarded as a subtask of the general *voice separation* task, which deals with not the separation of homophonic melody but also the separation of multiple concurrent melodies in the *polyphonic* music. Based on the concept of *voice leading* in the literature [24]. Most of the proposed methods in this direction apply the perceptual principles established in psychological studies, such as the principles of temporal continuity and pitch proximity [24], to specify a *perceptually independent* musical lines . Karydis *et al.* proposed the Voice Integration/Segregation Algorithm (VISA), which considers voice assignment as a order-aware bipartite matching problem [19] and is the revised in the later works [20], [21]. Chew and Wu proposed the concept of *contig*, the short segment where the number of played notes inside a contig is constant. The voice separation process includes the extraction of the contigs, and then reconnect the contigs according to similar perceptual noises [31]. The contig approach has gained attention in recent years; several improvements have been proposed [22], [23]. The above-mentioned methods, however, relies on a number of hand-crafted criteria that might rule out many musically-interesting things. The order of how to process these criteria also affect the result and therefore make it hard to be repeated. There were also less source codes released.

Some recent studies have come to the use data-driven methods that are also compatible with the perceptual principles while being more flexible. Temperley proposed a Bayesian model that based on the principles that 1) melodies tend to remain within a narrow pitch range; 2) note-to-note intervals within a melody tend to be small; and 3) notes tend to conform to a distribution (or key profile) that depends on the key [16]. Similar approach can also be seen in HMM-based voice separation [17]. Gray and Bunescu [18] proposed a neural network based method for voice separation of both polyphonic and homophonic music.

Most of the above-mentioned studies considered the separation multiple voices/streams in symbolic data. In this paper, we consider the problem of predominant melody extraction in homophonic music. This is a special case in voice separation. In other words, we assume only one predominant melody in this special case, and exclude the case of polyphony (e.g., Fugue). We consider this case because this is more useful in analyzing the music people usually listen to nowadays.

In audio music processing, the problem of melody extraction from polyphonic music has been discussed extensively [5]–[11]. Most of these methods have focused on acoustic modeling that incorporate acoustic features, while the language model is less investigated.

III. Methods

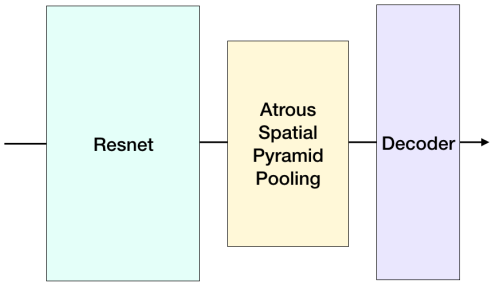
A. Semantic segmentation

We adopt DeepLabV3 and its successor, DeepLabV3+ [29], [30], one of the state-of-the-art performance in image semantic segmentation tasks, as the core of the proposed semantic segmentation model for melody recognition. Since this model has been applied in an related work on audio melody extraction [32], we mainly follow the settings in [32] for the semantic segmentation model used in this work. The model is a FCNN with an encoder-decoder architecture, where the encoder is implemented by a ResNet [33] followed by an dilated spatial pyramid pooling process, and the decoder is implemented by the reverse of the encoder, as shown in Fig. 1a. The major characteristic of DeepLabV3 is the dilated convolution blocks as a generalized version of the standard convolution:

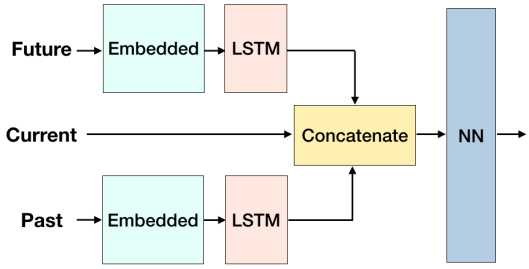
$$\mathbf{y}[i] = \sum_k \mathbf{x}[i + r \cdot k]\mathbf{w}[k] \tag{1}$$

where $\mathbf{y}$ and $\mathbf{x}$ denotes the input and the output feature maps, respectively, $\mathbf{w}$ is the convolution filter, and i indicates the location of the feature maps. The number r, named the dilated rate, determines the stride with which the input are sampled. For example, when $r = 1$, (1) stands for the standard convolution. To capture the context in different ranges, one can apply dilated convolution with different values of r on the same input feature map in parallel; this process is called the Spatial Pyramid Pooling (ASPP) [29], [30]. The outputs of these parallel convolution operations are then concatenated to provide information collected from various scales.

There is a problem of data imbalance as a melody object (i.e., positive data) typically occupies only a small portion of



(a) The FCN-based model for semantic segmentation.



(b) The LSTM-based model for sequence prediction.

Fig. 1: The models adopted for symbolic melody extraction.

the input piano roll in comparison to the accompaniment and silence part (i.e., negative data). To address this problem, we adopt the focal loss [34] as the loss function for the model:

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t)\,, \tag{2}$$

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 the imbalanced classes which balances 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 easy examples are down-weighted. Following [34], we set $\alpha_t = 0.25$, $\gamma = 2$ in this work. Besides, to recover the melody in fine resolution in the decoding process, we replace the simple decoder module in DeepLabV3 with an inverted version of the encoder module for fine-grained outputs, and the up-sampling process is replaced with stacks of convolution and transpose convolution layers. Also, better performance is achieved by introducing the U-net [35] structure that the output from each block of the encoder is concatenated to the corresponding block of the decoder. Implementation details can be found in [32].

B. Recurrent neural networks

We utilize the Deepbach model [28] to develop another model for melody extraction. In this model, the two-sided LSTM structure, which can simultaneously access the past and future information of interest, is well suited for polyphony generation. Here we utilize the model for melody extraction, by treating it as a sequence prediction problem.

The Deepbach model uses four identical networks cooperating together to model Bach's four part chorales with each network modeling one of the four voices:

$$\max_{\theta_i} \sum_{j} \log p_i\left(V_{ij}|V_{\setminus ij}, M, \theta_i\right), \text{ for } i \in [1,4]\,, \tag{3}$$

where V_{ij} is the pitch number for voice i at time j, M is the meta information such as beats and the fermata symbol, and θ_i is the latent parameters of the model for voice i. Each network in Deepbach outputs the activated pitch number for the corresponding voice at a time.

To extract melody, we take only one sub-network from Deepbach and use only piano roll as the data representation. This allows us to model multiple concurrent notes in homophonic music. Also, the current part of the input is modified to the timestep which we want to predict the melody on piano roll. The diagram of the model is showed in Fig. 1b.

C. Training and Inference

Instead of using all the data in the dataset during training, we randomly select a portion of training samples from the dataset for each training epoch, and set the number of samples to be a constant. This is an alternative way for efficient training, since training all data for every epoch is time-consuming. Because of the different topologies of the two models, their allowed batch sizes during training are also different. Therefore, in our experiments, for the training of the segmentation model, 60k samples are used for every epoch, while for the LSTM model, 768k samples are used.

To extract melody from a given score using the segmentation model, a window based analyzing method is used. The size of the window equals the input dimension of the segmentation model, and the hop size is one. The score will first be padded with 128 zeros, which equals the width of window at the beginning and the end, so each time step can be processed once in every location in the analysis window. As the window slides over the piano roll, all the results are superposed so that at the end we will have a time-frequency representation for the salience of melody. After the sliding process, we pick the maximum value in each column and zero out all other values. The remaining nonzero entities with values smaller than the average of each column's maximum are then set to zero.

To perform melody extraction on a given score using the LSTM model, we use the same method as the inference of segmentation model that a fixed dimension analysis window will slide through the padded score. The difference is that the predictions are concatenated to get the result for the whole score because the LSTM model predict one time step at a time. At last, the same zero out process will be applied in order to get the final result.

D. Implementation

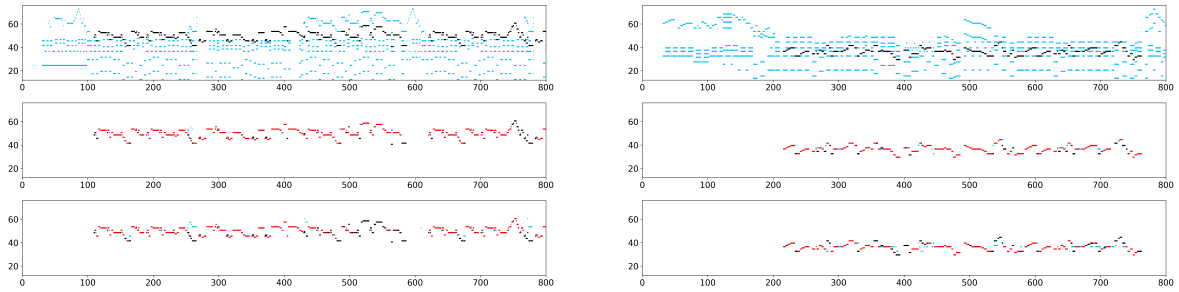
Both models are implemented using the Keras library with tensorflow as the back end, and optimized using ADAM. Before being processed, the pitch dimension of the input for segmentation model is padded with zeros from 88 to 128 for computational convenience and for the LSTM model, it is augmented from 88 to 90 which presents the start and end symbol respectively.

For the segmentation model, the input dimension is 128 timesteps in width and 128 in length which indicates the pitch. As shown Fig.1a, the input feature will first be processed by a 29-layer Resnet encoder. Then, the dense features output which is 16 times smaller than the original will be passed to the ASPP unit. Finally, a decoder which is composed of transpose convolutional layers with strides equal (2, 2) will up-sample the dense features to its original shape. The final output dimension will be (128, 128, 2), with the first channel indicating the presence melody and the other is for non-melody. The superposition in the inference process is performed on the first channel. Batch normalizations are applied after each activations, and a dropout rate of 30% is added after the batch normalizations.

For the LSTM model, the past and future part of the input score each contains 128 timesteps. Both parts are first separated, such that each unit of the input data contains a fragment of four time steps which is actually a 90-by-4 matrix, and there are no overlaps between these fragments. Every fragment is first flattened, and then a shared fully connected layer will reduct its dimension into a 90-D vector. By doing so, larger context information can be considered with a smaller model capacity. Both LSTM networks take a

TABLE I: Experiment results (in %) on the American Folk test set containing songs from various sub-genres.

Training data	Method	w/o melody shift in testing					w/i melody shift in testing				
		OA	RPA	RCA	VR	VFA	OA	RPA	RCA	VR	VFA
w/o melody shift in training	Semantic segmentation	79.26	79.14	81.2	86.57	**17.21**	61.89	54.71	68.56	82.28	19.23
	LSTM RNN	**80.36**	**79.27**	**81.56**	**86.94**	17.26	58.54	46.73	62.05	79.18	15.89
w/i melody shift in training	Semantic segmentation	78.67	76.75	79.30	84.18	14.03	**76.60**	**73.79**	**76.60**	**84.25**	**12.92**
	LSTM RNN	75.80	72.56	76.18	81.18	16.01	73.03	68.47	72.01	80.14	13.67
	Baseline (max pitch)	70.00	82.52	91.40	100	53.81	48.36	52.95	75.56	100	55.55



(a) (b)

Fig. 2: Melody extraction result. Top: the ground truth piano rolls, where the melodies are in black and the accompaniments are in cyan. Middle: extracted melodies using the semantic segmentation model. Bottom: extracted melodies using the RNN model. In the middle and bottom sub-figures, true positive detections are in red, false negative detections are in black, and false positive detections are in cyan.

series of embedded features with 32 time steps, i.e., a 90-by-32 matrix, as the input. Both networks contain 3 LSTM layers, each having 90 hidden units. The outputs of the two networks are concatenated with the current part of the input score and then transformed to an 88-D vector with another fully-connected layer.

IV. Experiment

A. Data

A MIDI corpus contains 600 American folks with a melody track is used as the training data for the symbolic model (see https://goo.gl/aPgzrW; last retrieved: 2018/09/08). All the pieces are first parsed to piano rolls. A timestep in the piano roll is a 32nd note. In the training process, we perform data augmentation, by pitch-shifting each song in the dataset up and down by at most 6 semitones 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, the training dataset comprise 7,800 pieces.

Besides using the ground truth dataset, we further consider a more challenging case: shifting down the melody by one octave to make the melody and accompaniment are interleaved. In this case, most of the melody parts lie within the pitch region of the accompaniment, making the it more difficult to model the true melody contour. We assume that if the training data contains data which melodies are shifted down, the model should be able to better predict such a challenging case, and make the model able to simulate human perception that the melody is robust to octave shifting. Therefore, we consider the cases that the training or testing data contain shifted melody or not. As a result, we experiment on four different experimental settings by consider whether the melody is shifted in either training or testing data, as shown in Table I.

We also compare the two models with a baseline that naively regards the highest note at every time stamp as the melody. This is a reasonable baseline because accompaniments tend to be arranged in the low-pitch region for this dataset that contains mainly folk music. In the case that the melody is shifted, this method becomes no more applicable.

B. Results

Table I lists the overall accuracy (OA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), voice recall (VR) and voice false alarm (VFA) of the proposed methods on the testing dataset. All data is computed from mir_eval [36].

The left-hand part of Table I shows that when neither the training nor the testing data contains melody shifting, LSTM RNN achieves slightly better performance than semantic segmentation; it achieves an OA of 80.36%, 1.1% higher than the OA of semantic segmentation. Interestingly, when the training data contain shifted melodies, the resulting performance degrades for both methods, but only the VFA values are improved. This implies that the modeling of melody objects is still highly related to its interleaved notes for both methods. However, perhaps such interleaved contents provide better cue for the model to classify the time intervals without melodies, i.e., no interleaved melody and accompaniment.

The right-hand part of the results in Table I shows that when the testing data contain shifted melody, the performance of both models degrade a lot. In particular, the RPA is much lower

than the RCA, indicating that the number of octave errors increases a lot. This implies that both models are still limited in modeling the dynamic pattern of melody independent from the structure. When the training data contain shifted melodies, the performance values of both models are improved a lot. Comparing the two models, we observe that semantic segmentation performs consistently better than LSTM RNN in this case. More specifically, the OA of the semantic segmentation model is better than the one of the LSTM RNN model by around 3%. Therefor when the melody and the accompaniment are interleaved, semantic segmentation appears to be a better method. Such higher accuracies might be caused by the large model capacity and high flexibility of the FCN structure in capturing contextual information.

Finally, Fig. 2 shows two examples with melody extraction results using the two models. Note that the second example is a challenging that the melody is shifted down by one octave. Results show that both models are capable in resolving the case that the melody is shifted. In general, the semantic segmentation model performs better than the LSTM RNN model in capturing detailed behaviors of the melody contour.

V. Conclusions

We have investigated two deep learning frameworks, FCN-based semantic segmentation and RNN-based sequence prediction, on the machine-based symbolic melody recognition problem. Results positively show the high performance of both frameworks in modeling melody when the accompaniment part is interleaved with. When the pitch difference between melody and accompaniment is large, the RNN-based method performs better; when the pitch difference is small, the FCN-based method performs better. The results also suggest future work on improving model robustness to interleaved accompaniment, and on the separation of multiple voices or tracks.

References

[1] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: musical information retrieval in an audio database," in *Proceedings of the third ACM international conference on Multimedia*. ACM, 1995, pp. 231–236.

[2] M. Melucci and N. Orio, "Musical information retrieval using melodic surface," in *Proceedings of the fourth ACM conference on Digital libraries*. ACM, 1999, pp. 152–160.

[3] C. S. Sapp, Y.-W. Liu, and E. Selfridge-Field, "Search effectiveness measures for symbolic music queries in very large databases." in *ISMIR*. Citeseer, 2004.

[4] J. Salamon, "Pitch analysis for active music discovery," in *Machine Learning for Music Discovery workshop, International Conference on Machine Learning (ICML)(cit. on p. 137)*, 2016.

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

[6] F. Rigaud and M. Radenen, "Singing voice melody transcription using deep neural networks." in *ISMIR*, 2016, pp. 737–743.

[7] 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.

[8] P. Verma and R. W. Schafer, "Frequency estimation from waveforms using multi-layered neural networks," in *INTERSPEECH*, 2016, pp. 2165–2169.

[9] R. M. Bittner, J. Salamon, S. Essid, and J. P. Bello, "Melody extraction by contour classification." in *Proc. ISMIR*, 2015, pp. 500–506.

[10] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[11] 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.

[12] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the tony software: Accuracy and efficiency."

[13] L. Yang, A. Maezawa, J. B. Smith, and E. Chew, "Probabilistic transcription of sung melody using a pitch dynamic model," in *Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 301–305.

[14] E. Molina, A. M. Barbancho-Perez, L. J. Tardón, and I. Barbancho-Perez, "Evaluation framework for automatic singing transcription," 2014.

[15] R. Nishikimi, E. Nakamura, K. Itoyama, and K. Yoshii, "Musical note estimation for f0 trajectories of singing voices based on a bayesian semi-beat-synchronous hmm." in *ISMIR*, 2016, pp. 461–467.

[16] D. Temperley, "A probabilistic model of melody perception," *Cognitive Science*, vol. 32, no. 2, pp. 418–444, 2008.

[17] A. McLeod and M. Steedman, "Hmm-based voice separation of midi performance," *Journal of New Music Research*, vol. 45, no. 1, pp. 17–26, 2016.

[18] P. Gray and R. C. Bunescu, "A neural greedy model for voice separation in symbolic music." in *ISMIR*, 2016, pp. 782–788.

[19] I. Karydis, A. Nanopoulos, A. Papadopoulos, E. Cambouropoulos, and Y. Manolopoulos, "Horizontal and vertical integration/segregation in auditory streaming: a voice separation algorithm for symbolic musical data," in *Proceedings 4th Sound and Music Computing Conference (SMC)*, 2007.

[20] D. Rafailidis, E. Cambouropoulos, and Y. Manolopoulos, "Musical voice integration/segregation: Visa revisited," in *Proceedings of the 6th Sound and Music Computing Conference*, 2009, pp. 42–47.

[21] D. Makris, I. Karydis, and E. Cambouropoulos, "Visa3: Refining the voice integration/segregation algorithm," 2016.

[22] N. Guiomard-Kagan, M. Giraud, R. Groult, and F. Levé, "Comparing voice and stream segmentation algorithms," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 493–499.

[23] ——, "Improving voice separation by better connecting contigs," in *International Society for Music Information Retrieval Conference (ISMIR 2016)*, 2016.

[24] D. Huron, "Tone and voice: A derivation of the rules of voice-leading from perceptual principles," *Music Perception: An Interdisciplinary Journal*, vol. 19, no. 1, pp. 1–64, 2001.

[25] D. Deutsch, "An illusion with musical scales," *The Journal of the Acoustical Society of America*, vol. 56, no. S1, pp. S25–S25, 1974.

[26] W. J. Dowling, "Expectancy and attention in melody perception." *Psychomusicology: A Journal of Research in Music Cognition*, vol. 9, no. 2, p. 148, 1990.

[27] M. R. Jones, "Dynamic pattern structure in music: Recent theory and research," *Perception & psychophysics*, vol. 41, no. 6, pp. 621–634, 1987.

[28] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *International Conference on Machine Learning*, 2017, pp. 1362–1371.

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

[30] 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.

[31] E. Chew and X. Wu, "Separating voices in polyphonic music: A contig mapping approach," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2004, pp. 1–20.

[32] W.-T. Lu and L. Su, "Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning," in *ISMIR*, 2018.

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

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

[35] 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*. Springer, 2015, pp. 234–241.

[36] 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. ISMIR*, 2014.