A Multi-Scale Fully Convolutional Network for Singing Melody Extraction

Ping Gao, Cheng-You You and Tai-Shih Chi
Department of Electrical and Computer Engineering
National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C.
E-mail: {gaoping.eed06g, chengyou.eed07g}@nctu.edu.tw, tschi@mail.nctu.edu.tw

***Abstract*—The melody extraction can be considered as a sequence-to-sequence task or a classification task. Many recent models based on semantic segmentation have been proven very effective in melody extraction. In this paper, we built up a fully convolutional network (FCN) for melody extraction from polyphonic music. Inspired by the state-of-the-art architecture of the semantic segmentation, we constructed the encoder in a dense way and designed the decoder accordingly for audio processing. The combined frequency and periodicity (CFP) representation, which contains spectral and cepstral information, was adopted as the input feature of the proposed model. We conducted performance comparison between the proposed model and several methods on various datasets. Experimental results show the proposed model achieves state-of-the-art performance with less computation and fewer parameters.**

I. INTRODUCTION

Melody extraction, which extracts the melody pitch contour from the polyphonic music audio, is an active topic in the research field of music analysis and music information retrieval (MIR). With the advancement of deep learning, researchers develop various neural networks for melody extraction. Music is like language, composition and arrangement are subject to certain rules. Therefore, for human, melody perception might not only stem from the lower level of pitch perception but also from the higher level of semantic analysis. Consequently, melody extraction can be thought of as an audio translation to a melody contour such that one might apply the techniques used in natural language processing on this subject to produce good results. For instance, some studies consider melody extraction as a sequence to sequence task [1] [2], where the input audio sequence can be one-dimensional raw data, or a two-dimensional representation through the discrete Fourier transform or the constant-Q transform (CQT). The audio sequence is then put through the neural network to produce a sequence of the pitch contour. These architectures can determine the current pitch through information from distant past or future to boost performance of melody extraction in some situations.

From another point of view, melody extraction can be considered as a classification problem of semantic segmentation [3] [4]. This type of approach first enhances the pitch contour as the input feature, and then put the feature through a semantic segmentation network. Later on, a threshold is used to binarize the result in a similar way to thresholding operations in many other music related tasks [3] [5]. The study in [6] compared these two different approaches in details and showed the semantic segmentation method produces 3% higher overall accuracy (OA) scores than the LSTM-based sequence-to-sequence method. It also showed the semantic segmentation method produces better results when melody and accompaniment interlace with each other.

Context information is very important to melody extraction. In music, it is quite common to have tone change across time. The changing translations of the pitch contour are encoded as the up and down sweeping patterns on the spectrogram, which can be captured by the convolutional neural network (CNN). Although the semantic segmentation approach shows good results, it possesses a tradeoff between performance and the computational load. If the sizes of the CNN kernels are not large enough, the system cannot capture a wide range of context information. If the kernel sizes are enlarged, a lot of parameters and calculations are brought into the system. To address this tradeoff, we used the dilated convolution in the fully convolutional networks (FCN). Features from different scales were collected using dilated convolutions with different sizes. In addition, the recently proposed dense connection with atrous spatial pyramid pooling in the DenseASPP image segmentation architecture [7] was also adopted in our model to significantly reduce the number of parameters while retaining state-of-the-art performance.

There are two main contributions in this paper. The first one is we proposed a model outperforming several recently developed methods on singing melody extraction. The second one is we successfully identified that the DenseASPP image segmentation architecture can be applied to the relatively distant domain of music pitch estimation. The rest of the paper is organized as follows. In Section II, we briefly review related work in literature. In Section III, we demonstrate the architecture of the proposed model. We then evaluate the model and compare it with several recently developed methods on melody extraction in Section IV. Finally, the conclusion and potential future work are given in Section V.

II. RELATED WORK

Melody extraction is a long-standing topic in music information retrieval. With the popularity of deep learning, more and more researchers have begun to use the deep learning architecture for melody extraction and demonstrated much better performance than traditional methods [4]. Some studies are inspired by high-order semantic segmentation, while others construct neural networks by mimicking human perception [8] [9].

Inspired by DeepLabV3+ [10] [11], a novel melody extraction system using a semantic segmentation tool for deep convolutional expansion convolution neural networks was proposed in [4]. The encoder was implemented by a ResNet [12], and followed by an atrous spatial pyramid pooling layer [13]. Experiment results showed the segmentation model is competitive, especially in reducing the rate of voice false positives. Another study compared the two different melody extraction approaches for symbolic music [6]. The first approach considered melody extraction as a sequence prediction problem and used recurrent neural networks (RNN) as the system architecture. The second approach considered melody extraction as a semantic segmentation problem and used fully convolutional networks (FCN) as the system architecture. Experiment results showed the semantic segmentation approach produced more accurate results when using the same data set.

For image processing, the DenseNet with the architecture of connecting each layer to every other layer in a feed-forward fashion was proposed in [14]. The study showed the DenseNet requires fewer parameters and fewer calculations to achieve the most advanced performance due to the enhanced feature propagation and feature reuse. The study also showed that the DenseNet with convolution features exhibits a compact internal representation, which reduces functional redundancy, such that it is well suited for various tasks. A DenseNet-based full convolutional network (FCN) model was proposed for sematic segmentation in [15]. The study showed the DenseNet-based FCN can achieve higher accuracy than other methods without pre-training and its architecture is 10 times smaller than other methods that achieve the same performance. Later on, a densely connected atrous spatial pyramid pooling, which constructs with a set of atrous convolutional layers in the dense way, was proposed in [7] to generate multi-scale features from a larger scale range.

Inspired by these studies in image processing, we propose a dilated-convolution based multi-scale FCN with a dense connection and atrous spatial pyramid pooling for melody extraction in this paper. Experiment results show the proposed model achieves state-of-the-art performance in a very efficient way.

III. PROPOSED MODEL

A. Data Pre-processing

We first downsampled the audio to 16 kHz sampling rate. It was shown the combined frequency and periodicity (CFP) feature [16] is better than features from CQT and other transformation methods for melody extraction [3] [16] such that we used the CFP feature as the input feature. The CFP feature consists of the power-scaled spectrogram(S), generalized cepstrum (GC) and generalized cepstrum of spectrum (GCoS) [17] [18] [19].

The frequency range of the power-scaled spectrogram was set from 31 Hz to 1250 Hz with the resolution of 48 bins per octave. The x represents the input audio signal in the time domain and in the short-time Fourier transform (STFT) domain. $\mathbf{W}^{(2)}$ and $\mathbf{W}^{(3)}$ are high-pass filters for removing DC component. $\mathbf{F}$ is the N-point discrete Fourier transform (DFT) matrix and $\sigma^{(i)}$ are activation functions. The formula for calculating CFP are given as follows:

$$\mathrm{z}^{(1)} = \sigma^{(1)}\left(|\mathbf{F}\mathrm{x}| + \mathrm{b}^{(1)}\right) \quad (1)$$

$$\mathrm{z}^{(2)} = \sigma^{(2)}\left(\mathbf{W}^{(2)}\mathbf{F}^{-1}\mathrm{z}^{(1)} + \mathrm{b}^{(2)}\right) \quad (2)$$

$$\mathrm{z}^{(3)} = \sigma^{(3)}\left(\mathbf{W}^{(3)}\mathbf{F}\mathrm{z}^{(2)} + \mathrm{b}^{(3)}\right) \quad (3)$$

where $\sigma^{(i)}(x) = \begin{cases} x^{\gamma_i} & if\ x > 0 \\ 0 & if\ x \le 0 \end{cases}$, the considerations for setting γ_i can be viewed in [16][17][18][19].

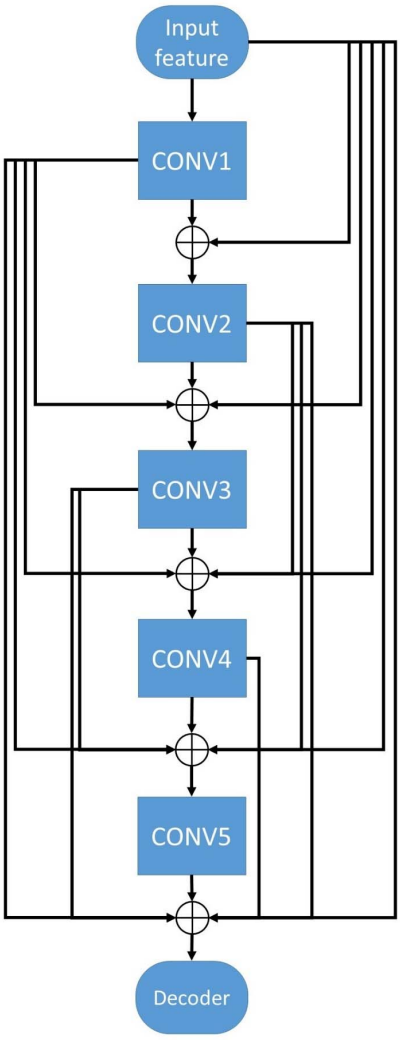

Fig. 1 Architecture of the encoder of the proposed model

B. Model Architecture

The proposed model consists of two major modules, the encoder and the decoder. The architectures of the encoder and the decoder are respectively shown in Fig. 1 and Fig. 2. The encoder contains five dilated convolution layers (named CONV1~CONV5 in Table I) which are connected in a dense way with atrous spatial pyramid pooling.

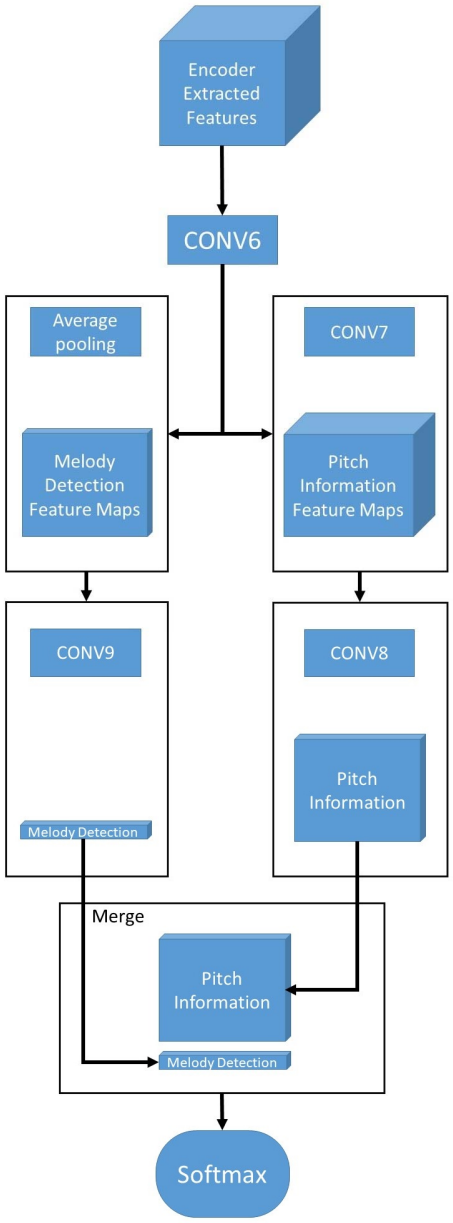

Fig. 2 Architecture of the decoder of the proposed model

First, the atrous spatial pyramid pooling is used in the encoder to increase the size of the receptive field [7], which can be calculated by:

$$R_l = (d_l - 1) \times (K_l - 1) + K_l \quad (4)$$

where d_l is the dilation rate of layer l, and K is the kernel size of layer l.

If one dilated convolutional layer is connected to another, the size of the receptive field after stacking is:

$$R_l = R_{l-1} + R_{l-2} - 1 \quad (5)$$

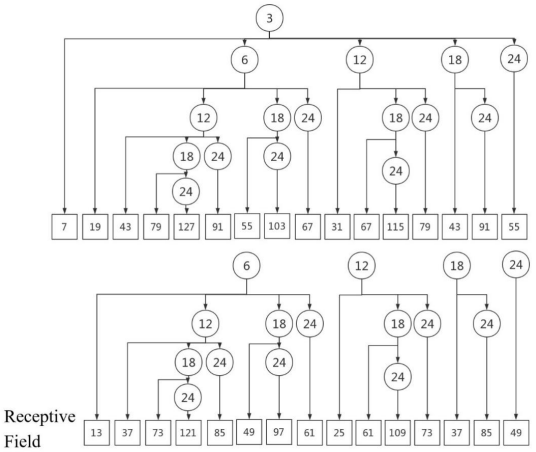

Fig. 3 Illustration of different dilation rate layers stacking. The number in the circle represents the dilation rate. The number in the square represents the size of receptive field.

For example, a convolutional layer 1 with the dilation rate of 3 and the kernel size of 3 would have a receptive field of size 7. The other connected convolutional layer 2 with the dilation rate of 6 and the kernel size of 3 would have a receptive field of size 13. Stacking these two layers would produce a new layer 3 with receptive field of size 19. In this study, we stacked dilated layers of different sizes to obtain multi-scale features to improve the performance of the model. Fig. 3 shows the size of the receptive field by combining different layers with different dilation rates when the kernel size is 3.

Inspired by [14], the input of each layer in the encoder is directly sent to all successive layers to have feature reuse. Meanwhile, each layer of the encoder is particularly designed to be "thin", that is, only very few feature maps are learned to reduce redundancy. In this way, the encoder would have fewer parameters, hence less computation, and lower change of overfitting. The features extracted by each layer are equivalent to a nonlinear transformed representation of the input data. As the network depth increases, the complexity of the transformation also increases (i.e., composite of more nonlinear functions). Compared with a general NN classifier, whose performance directly depends on the features of the last layer (with the most accumulated complexity) of the network, the DenseNet utilizes low complexity features such that it is easier to get a smooth decision function for better generalization performance. As for the activation function, we used the scaled exponential linear unit (SELU) in each layer. The formula for the SELU activation function is as follows:

$$\text{SELU}(x) = \lambda \begin{cases} x & if\ x > 0 \\ \alpha e^x - \alpha & if\ x \le 0 \end{cases} \quad (6)$$

The details of the encoder settings are described below. The size of the kernel of the proposed FCN model is set to 3. Since the atrous spatial pyramid pooling is capable of combining context information, a large kernel size is not necessary. Batch normalization is done in each layer and the SELU is used as the activation function [20]. To be able to combine the output channels, the output feature maps at each stage are ensured to be

with the same size using zero padding. The number of the output channels is set to 10. In simulations, we have also tried 20 output channels. However, the benefit of using 20 channels is negligible (2% increase in the overall accuracy score) but with a very high computational cost of doubling parameters of the encoder. To have the connection of feature reuse, the input of the n-th layer is combined from inputs of all previous layers (1st, 2nd, …, (n-1)-th) and the output of the (n-1)-th layer. The settings of the dilation rates and the number of input/output channels of each layer of the encoder are shown in Table I.

Table I: Architecture of the encoder. The kernel size of any convolutional layer was set to 3×3. Batch normalization was done in each layer and the activation function was SELU. Zero padding was used on each layer to make the feature map with the same size.

Layers	Input channels	Output channels	Dilation
CONV1	3	10	3
CONV2	13	10	6
CONV3	23	10	12
CONV4	33	10	18
CONV5	43	10	24

The decoder consists of two parts. The first part contains three two-dimensional convolutions (named CONV6, CONV7 and CONV8 in Table II). The goal is to refine the features while reducing the number of channels. The size of the input of the decoder is F×T with the channel number of 77, where F and T are the total numbers of the frequency bin and time frame of the CFP features. This input contains outputs of all dilation layers in the encoder and the original CFP features. After the three convolutional layers, the number of channels will be reduced to 1. The size of the output of this first part is 1×F×T. The second part is for melody detection. It contains an average pooling layer, which integrates all frequency information at a certain time frame, and a two-dimensional convolution (named CONV9 in Table II), which integrates information across channels and reduces the channel number to 1. The size of the output of this second part is 1×1×T. The output of these two parts are combined to form the final feature with the size of 1×(F+1)×T. The Softmax is used to get the final detection result. Details of the input/output dimensions of each layer of the decoder are given in Table II. If no melody is detected in a frame, the output pitch will be 0 Hz. If melody is detected, the output pitch will be the frequency value of the frequency bin corresponding to the classification result.

Table II. Architecture of the decoder. The kernel size of any convolutional layer was set to 3×3. Batch normalization was used in each layer and the activation function was SELU.

Layers	Input size	Output size
CONV6	(77,F,T)	(64,F,T)
CONV7	(64,F,T)	(32,F,T)
CONV8	(32,F,T)	(1,F,T)
Average pooling	(64,F,T)	(64,1,T)
CONV9	(64,1,T)	(1,1,T)
Softmax	(1,F+1,T)	-

[1] https://github.com/eed0650745/singing_melody_extraction

For model update, we chose the binary cross entropy as the loss function. The Adam optimizer was used with the learning rate of 0.001. The source code is given here[1] for reproducing our results.

IV. Experiment

A. Datasets

We followed previous studies [8][9] in setting up our datasets. For training, we used 740 clips from the MIR1K dataset which contains 1000 song clips extracted from 110 Chinese karaoke pop songs sung by 8 female and 11 male nonprofessional singers. The remaining 260 clips were used for test. We also used 200 clips from the iKala dataset for training, and the remaining 52 clips for test. The ADC2004 and MIREX05 datasets were entirely used for test. Note that music clips containing singing melody were used for training and test and those clips with the main melody from musical instruments were not used. Since the frame duration of the input feature is 0.016 second, we used the mir_eval tool [21] to resample the ground truth pitch label by linear interpolation. When dealing with the ground truth, we mapped the frequency of the label to the frequency of the closest CFP frequency bin. In calculating the accuracy score, the frequency of the predicted pitch needed to be in the upper and lower quarter tone of the true pitch so that the mapping operation did not produce artificial errors.

B. Results

For evaluation, we used the regular metrics in literature, the voicing recall rate (VR), the voicing false alarm rate (VFA), raw pitch accuracy (RPA), raw chroma accuracy (RCA) and overall accuracy (OA). Except the VFA measure, the higher the score, the better the performance. All scores were calculated by Python library of the mir_eval tool [17]. For comparison, we have implemented the latest methods of using deep learning for melody extraction. Comparison results for each dataset are showed in Table III.

DenseASPP_10 in Table III represents the model whose encoder has 10 output channels in each layer. Similarly, the encoder of the DenseASPP_20 has 20 output channels in each layer. We also set all the dilations in our proposed model to 1 to have the architecture of the DenseNet (named DenseNet in Table III). The purpose of this is to investigate the performance gain by combining atrous spatial pyramid pooling with dense connections. From the results, the proposed model outperforms the DenseNet in all metrics on all test datasets. On ADC2004, the OA score increases around 8%. On MIR1K and MIREX2005, the OA scores raise between 3% and 4%. The results indicate that multi-scale information plays a very important role in melody extraction. When comparing with SegNet [8], the proposed model has higher RPA and RCA scores, which means the proposed model is more capable of capturing pitch. On ADC2004, the proposed model produces significantly higher RPA and RCA scores than all other compared methods. It achieves at least 7% improvement, reaching 77%

and 79% respectively. It also produces the best VR score but not the best VFA score. On MIR1K and MIREX2005, the proposed model produces at least 2% higher RPA and RCA scores than SegNet. On iKala, the proposed model produces slightly higher RPA and RCA scores than SegNet.

Table III: Evaluation results (in %) in terms of VR, VFA, RPA, RCA and OA on various datasets. The bold numbers represent the best results of all compared methods.

MIR1K	VR	VFA	RPA	RCA	OA
Duplex[8]	80.97	14.74	70.30	73.88	74.67
Seg-Net[3]	85.93	6.09	81.03	82.47	84.69
Two-stage[9]	88.27	16.65	79.27	81.67	80.46
DenseASPP_10	87.43	8.80	83.25	84.87	85.41
DenseASPP_20	**88.68**	7.66	**85.00**	**86.14**	**87.04**
DenseNet	84.66	8.00	79.07	81.16	82.64
DA_Reversed	86.37	**7.18**	82.85	84.30	85.60

(a)MIR1K

ADC2004	VR	VFA	RPA	RCA	OA
Duplex[8]	56.65	9.88	50.20	55.03	56.54
Seg-Net[3]	75.90	**6.81**	70.38	72.56	72.77
Two-stage[9]	62.19	15.78	53.49	58.00	58.37
DenseASPP_10	82.59	10.94	76.84	78.89	78.06
DenseASPP_20	**84.91**	10.91	**80.14**	**81.86**	**80.67**
DenseNet	75.71	8.96	68.46	71.62	70.68
DA_Reversed	77.82	6.95	72.15	75.01	74.36

(b)ADC2004

MIREX2005	VR	VFA	RPA	RCA	OA
Duplex[8]	81.91	7.37	74.36	76.22	80.67
Seg-Net[3]	88.83	**5.60**	83.96	84.69	**87.71**
Two-stage[9]	86.63	12.57	78.84	80.07	81.81
DenseASPP_10	87.31	6.20	82.58	83.44	86.55
DenseASPP_20	**90.43**	9.03	**85.38**	**85.91**	87.18
DenseNet	89.67	10.68	82.75	83.83	85.05
DA_Reversed	86.59	6.29	81.90	82.83	86.12

(c)MIREX2005

iKala	VR	VFA	RPA	RCA	OA
Duplex[8]	83.65	17.30	74.50	76.97	77.21
Seg-Net[3]	87.16	6.27	83.16	84.27	86.56
Two-stage[9]	89.53	15.21	80.74	82.13	82.07
DenseASPP_10	86.54	6.49	83.18	84.38	86.32
DenseASPP_20	**90.43**	9.03	**85.38**	**85.91**	**87.18**
DenseNet	86.25	7.53	81.91	83.12	85.00
DA_Reversed	85.93	**5.01**	83.56	84.74	87.02

(d)iKala

In addition, we also investigated another setting which was referred to as the DA_Reversed system. It had the order of the dilated rates reversed from 3-6-12-18-24 to 24-18-12-6-3 while keeping the same number of channels as DenseASPP_10. This setting was tested to see performance of the neural network with decreasing sizes of receptive fields. From the experimental results shown in Table III, we can observe reversing the order of dilated rates has no significant impact on the network efficiency. It is because the overall size of the stacked receptive fields remains the same based on equations (4) and (5). In other words, what matters is not the order of the dilated rates but the overall size of the stacked receptive fields.

Finally, we calculated the number of parameters used by different FCN-based models. We found that using DenseNet did reduce a large amount of parameters. Comparing with SegNet, the DenseNet uses less than 20% parameters and 30% computation as listed in Table IV in terms of #Params and GFLOPs. Dilation does not increase the number of parameters, but it does improve performance.

Table IV: Number of parameters and GFLOPs used by different FCN-based models.

Method	#Params	GFLOPs
SegNet	540.2K	18.76
DenseASPP_10	74.5K	5.48
DenseASPP_20	120.2K	8.79
DenseNet	74.5K	5.48
DA_Reversed	74.5K	5.48

V. CONCLUSIONS

In this paper, we proposed a method based on semantic segmentation for melody extraction. It combined DenseNet and atrous spatial pyramid pooling to achieve state-of-the-art performance. It outperforms other popular methods in almost all evaluation metrics on all test datasets. In the category of FCN-based model, the proposed model has fewer parameters, requires less computation, and achieves faster training speed. Significant improvement in voicing recall rate, raw pitch accuracy, raw chroma accuracy and overall accuracy was observed in experiments, which means the proposed model has great capability in capturing pitch. In the future, we will explore the potential usage of this model in other MIR tasks such as audio chord estimation.

REFERENCES

[1] Dogac Basaran, Slim Essid, and Geoffroy Peeters, "Main melody extraction with source-filter nmf and crnn," in *19th International Society for Music Information Retreival*, 2018.

[2] Hyunsin Park and Chang D Yoo, "Melody extraction and detection through lstm-rnn with harmonic sum loss," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2766-2770, 2017.

[3] Hsieh, Tsung-Han, Li Su, and Yi-Hsuan Yang, "A Streamlined Encoder/Decoder Architecture for Melody Extraction," in *IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 156-160, 2019.

[4] Wei-Tsung Lu and Li Su, "Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning," in *ISMIR*, pp. 521-528, 2018.

[5] Hao-Wen Dong and Yi-Hsuan Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," *arXiv preprint* arXiv:1804.09399, 2018.

[6] Wei-Tsung Lu and Li Su, "Deep learning models for melody perception: An investigation on symbolic music data," in *Proceedings, APSIPA Annual Summit and Conference*, pp. 12-15, 2018.

[7] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3684-3692, 2018.

[8] Hsin Chou, Ming-Tso Chen, and Tai-Shih Chi, "A hybrid neural network based on the duplex model of pitch perception for singing melody extraction," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 381-385. IEEE, 2018.

[9] Chen, Ming-Tso, Bo-Jun Li, and Tai-Shih Chi, "CNN Based Two-stage Multi-resolution End-to-end Model for Singing Melody Extraction," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1005-1009, 2019.

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint* arXiv:1706.05587, 2017.

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801-818, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, pp. 630- 645. Springer, 2016.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834-848, 2018.

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian QWeinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708, 2017.

[15] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 11-19, 2017.

[16] Li Su and Yi-Hsuan Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 10, pp. 1600-1612, 2015.

[17] Takao Kobayashi and Satoshi Imai, "Spectral analysis using generalised cepstrum," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1235-1238, 1984.

[18] Li Su, "Between homomorphic signal processing and deep neural networks: Constructing deep algorithms for polyphonic music transcription," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 884-891, 2017.

[19] Keiichi Tokuda, Takao Kobayashi, Takashi Masuko, and Satoshi Imai, "Mel-generalized cepstral analysisa unified approach to speech spectral estimation," in *Third International Conference on Spoken Language Processing*, 1994.

[20] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter, "Self-normalizing neural networks," in *Advances in neural information processing systems*, pp. 971-980, 2017.

[21] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel, "mir_eval: A transparent implementation of common mir metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.