Real-Time Monophonic Dual-Pitch Extraction Model

Ngoc-Son Tran*, Pei-Chin Hsieh*, Yih-Liang Shen†, Yen-Hsun Chu‡ and Tai-Shi Chi*
* Institute of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Taiwan
† Institute of Communications Engineering, National Yang Ming Chiao Tung University, Taiwan
‡ Realtek Semiconductor Corp., Hsinchu, Taiwan
E-mail: ngocsontran19398@gmail.com; wren.ee12@nycu.edu.tw; dennis831209@gmail.com;
yenhsun.chu@realtek.com; tschi@nycu.edu.tw

***Abstract*—Pitch estimation is a fundamental aspect of audio signal processing with applications in music information retrieval (MIR), speech analysis and more. This paper proposes a new model for real-time dual-pitch estimation to tackle the problem of estimating two pitches at the same time in duet songs. The model is designed for real-time processing, frame by frame and with streamlined components for efficiency. We also propose data augmentation methods for simulating duet singing scenarios. Our model is robust in single and dual-pitch scenarios through experiments with various datasets and metrics. Our results contribute to the pitch estimation techniques and provide a practical solution for real-time audio processing in MIR, speech analysis and beyond.**

I. Introduction

Pitch estimation has long been a pivotal topic in the domain of music information retrieval (MIR), finding applications in diverse areas such as melody extraction, and speech/music transcription. The existing landscape of pitch estimation methods can be broadly categorized into two types: those rooted in digital signal processing (DSP) and those driven by data. DSP-based methods encompass widely recognized algorithms like RAPT [1], YIN [2], and pYIN [3], which leverage time-domain periodicity features for pitch estimation. On the other hand, data-driven approaches, exemplified by neural network (NN)-based methods such as CREPE [4] and the hybrid model [5], rely on both time-domain and frequency-domain features, yielding superior performance compared with DSP-based methods.

Despite the effectiveness of these algorithms in pitch estimation tasks, their suitability for real-time applications remains limited — an imperative in scenarios like speaker identification [6], voice analysis [7], and music analysis [8]. Notably, the karaoke grading system should heavily rely not only on accurate single-pitch estimation for solo songs, but also on dual-pitch estimation for duet songs, to assess users' performance [9]. While the YIN algorithm stands out for its low computational overhead, due to its reliance on auto-correlation, it is faced with the challenge of handling two pitches simultaneously [2]. Recently, there has been a surge in the adoption of NN approaches for multi-pitch estimation. LATE/DEEP [10] uses a stack of 2D convolution layers to analyze the harmonic structure. RNN-BLSTM [11] uses the recurrent networks to capture the temporal information. However, a notable limitation of these methods lies in their inability to fulfill the real-time execution criterion. Motivated by this gap, our research endeavors to develop a real-time dual-pitch extraction algorithm for the karaoke grading system.

Our methodology builds upon the spectral branch in a previous study [12] while refining its efficiency. With the focus on efficiency, we adapt specific blocks to meet our objectives, prioritizing real-time performance on embedded systems with limited computational power. Concurrently, we leverage the log-scale spectrogram—an efficient representation to decipher the harmonic structure [5]—as the key feature input for our proposed NN model. Central to the architecture of the proposed model is the harmonic aware block, crucial for capturing underlying harmonic patterns essential for accurate pitch extraction [12]. Throughout the design process, careful attention is given to parameter optimization to minimize computational overhead while maintaining performance.

The rest of the paper is organized as follows. Section II presents the utilized features and details the proposed model. Section III outlines the datasets, and offers insights into the experimental configurations. Section IV delves into the experimental results and ensuing discussions. Lastly, Section V encapsulates the essence of our study through the concluding remarks.

II. Proposed Model

A. Input Features

Temporal and spectral features often serve as fundamental representations to NN models for audio applications [13]. To optimize computational efficiency, we adopted the log-scaled short-time Fourier transform (STFT)-based spectrogram, as the primary input representation for our model as in [5]. In addition to using the log-frequency representation, we applied the logarithmic scale to the magnitude of the log-spectrogram. As for the frequency resolution and the frequency coverage, we set 24 bins per octave, and took the first 176 frequency bins, covering 31 Hz to 4857 Hz, as the input representation to our model. The frequency coverage of 31 Hz to 4857 Hz was selected by following the setting in [14] to cover the pitch range of singing voices.

B. Proposed System Architecture

The original spectral branch proposed in [12] utilizes information from previous and subsequent frames to predict pitch of the current frame. However, for the real-time task, our system adopts a frame-based approach, necessitating adjustments to

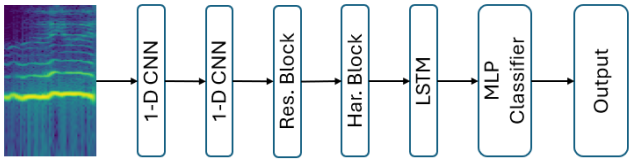

Fig. 1: The overall architecture of our proposed model.

the spectral branch architecture, as shown in Fig. 1. Specifically, we replaced all 2-D convolutional blocks in the original spectral branch with 1-D convolutional blocks and eliminated several blocks to streamline the model complexity. Details of the parameter settings in our model are outlined in Table I.

The model begins with two convolution blocks. Each block contains one layer of 1-D convolution with the kernel size of 8 and the stride of 1. The convolution layer is followed by batch normalization [15], ReLU activation, and the Maxpool layer with the pool size of 2 and the stride of 2. After the convolution blocks, there are the residual connection block (Res. Block) and the Harmonic-aware block (Har. Block). The residual connection block is shown in Fig. 2, and details of the Harmonic-aware block is illustrated in Fig. 3. At the end of the model, the layers of LSTM and MLP are deployed with the hidden size of 256 and 512, respectively. The zero-padding technique is employed at each convolution layer to maintain consistent input and output sizes.

The Harmonic-aware block is designed to capture specific harmonics based on Equation 1.

$$d_{i,j} = log_{2^{1/Q}}(jf_0) - log_{2^{1/Q}}(if_0) = Qlog_2(\frac{j}{i}) \quad (1)$$

where i, j represent the indexes of the harmonics, $d_{i,j}$ signifies the number of bins between i-th and j-th harmonics, Q denotes the number of bins per octave, and f_0 stands for the fundamental frequency. We set the parameter $Q = 24$. This setting yields interval computations between the 2nd, 3rd, and 4th to the 1st harmonic, resulting in approximately 24, 38, and 48 frequency bins, respectively. Two Max-pooling layers with the stride of 2 are used before the Harmonic-aware block, which implies the adoption of a $4\times$ downsampling. Therefore,

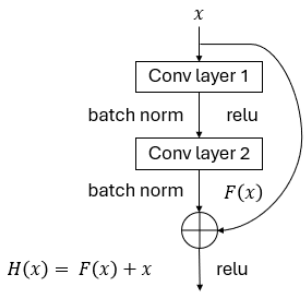

Fig. 2: The architecture of the residual block (Res. Block) in the proposed model.

TABLE I: Architecture configuration of the model. The Res. Block denotes for residual block, Har. block denotes for harmonic aware block.

Layer	Configuration	Input	Output
Conv1D	kernel 8, filters 3, stride 1	1 x 176	8 x 176
MaxPool1D	pool 2, stride 2	8 x 176	8 x 88
Conv1D	kernel 8, filters 3, stride 1	8 x 88	8 x 88
MaxPool1D	pool 2, stride 2	8 x 88	8 x 44
Res. Block	Conv1D: kernel 8, filters 1, stride 1	8 x 44	8 x 44
	Conv1D: kernel 8, filters 3, stride 1		
Har. Block	Conv1D: kernel 8, filters 1, stride 1	8 x 44	1 x 44
	Conv1D: kernel 1, filters 6/10/12, stride 1		
MaxPool1D	pool 2, stride 2	1 x 44	1 x 22
LSTM	hidden size 256	1 x 22	1 x 256
MLP layer	hidden size 512	1 x 256	127

three parallel 1-D convolution kernels with sizes 6, 10, and 12 are employed to identify harmonic structures and enhance classification accuracy. The idea of using 1-D convolution kernels with fixed sizes to decipher harmonics on the log-spectrogram was proposed and evaluated in the previous study [5].

C. Dual-pitch Extraction

The estimated pitch is taken from the model's final output layer. The 127-dimensional output shows the most likely pitch as the highest value, called $peak_1$. For duet songs, the second singer's pitch ($peak_2$) is identified as the index with the second-highest value in the output layer. We distinguish between silence, solo, and duet scenarios based on the peak's magnitude as follows:

$$\text{Number of Singer} = \begin{cases} 2 & \text{if } peak_2/peak_1 \geq 0.5 \\ 1 & \text{if } peak_2/peak_1 < 0.5 \\ 0 & \text{if } peak_1, peak_2 < \alpha \end{cases} \quad (2)$$

In other words, if the ratio between the magnitudes of $peak_2$ and $peak_1$ exceeds or equals to 0.5, the scenario is classified as a duet, and two predicted pitches are reported. Otherwise, if the ratio is below 0.5, it is seen as a solo scenario, and only $peak_1$ is given. If both $peak_1$ and $peak_2$ are less than α, the scenario is considered a silence, and no pitch is given.

III. Experiment Setting

A. Model Setting

Initially, raw audio signals undergo resampling to achieve a 48 kHz sampling rate. Subsequently, the signals are transformed into spectrograms using STFT with a window size of 4096 and a hop size of 1024. Then, the linear frequency axis is transformed into the log frequency axis using the setting mentioned in Section II.A, and the magnitude is also transformed into the logarithmic scale.

In this work, the pitch estimation problem is treated as a classification task. We partition frequency spanning from 31 Hz (B0) to 1175 Hz (D6) into 127 pitch classes, excluding an unvoiced class. To establish the mapping between frequencies and pitch classes, we employ the following equation:

$$f_n = 31 * 2^{(n-1)/2/12} \quad (3)$$

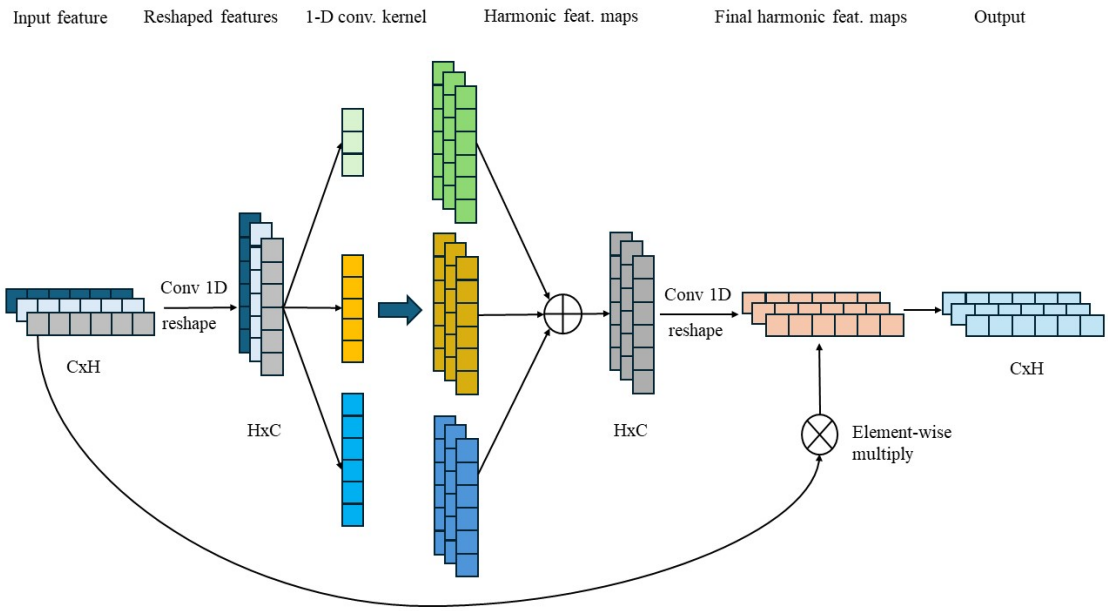

Fig. 3: Details of Harmonic-aware block (Har. Block) in our model. "C", and "H" respectively denote for the number of channels, and feature dimension.

where n is the index of the pitch class, and f_n is the corresponding frequency.

For the labeling process, we employ a multi-label strategy [16], depicted in Fig. 4. Frames featuring two singers are allocated two pitches, labeled as 1 each, whereas frames with a single singer are labeled with a single pitch marked as 1. Frames without vocalization are identified with all pitch classes set to 0. Additionally, to ensure consistency between the audio signal and labels, we resample the labels, using mir_eval library [17], in the dataset to match the frame length of the log-spectrogram.

In this work, we employ the binary cross-entropy loss function to compute the error between the ground truth pitch vector y_i and the prediction pitch vector $\hat{y}_i$ for each frame:

$$L(y, \hat{y}) = \sum_{i=1}^{127} -y_i log(p(\hat{y}_i)) - (1 - y_i) log(1 - p(\hat{y}_i)) \quad (4)$$

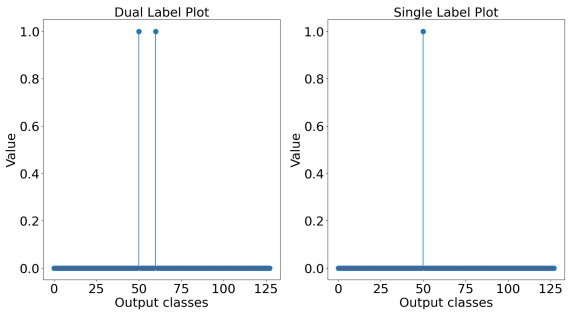

Fig. 4: Output samples of the labeling method (left panel: a duet frame; right panel: a solo frame).

We used the Adam optimizer [18] in this work, with the learning rate of 0.001. Adam facilitates faster convergence by dynamically adjusting the learning rate during training.

B. Training Datasets and Data Augmentation

The MIR-1k dataset [19], comprising 1000 audio clips featuring 8 females and 11 males, serves as our primary training dataset. Each audio clip spans a duration ranging from 4 to 13 seconds, with a cumulative duration of 133 minutes. In particular, this dataset encompasses both instrumental and vocal tracks, separated into the left and right channels, respectively. Given our focus on monophonic scenarios, training exclusively utilizes data from the vocal channel. In addition, the vocal data are augmented to enhance the diversity and robustness of our model. Specifically, we employ shift-up (10 semitones) and shift-down (8 semitones) methods on vocal data to broaden the frequency range seen by the model.

Besides the single-pitch training dataset extracted from MIR-1k, we derive a new dataset from MIR-1k to simulate duet singing scenarios. This involves a mixing process where the original vocal track is first pitch-shifted upwards, followed by a mixing operation utilizing two conditions:

$$Duet = \begin{cases} Original + (0.8 \sim 0.95) * (Shift\_up_{version}) \\ (0.8 \sim 0.95) * Original + (Shift\_up_{version}) \end{cases} \quad (5)$$

The above two mixing conditions simulate situations where one singer's volume is slightly lower than the other's.

As for shifting pitch upward, we consider two augmentation methods. In the first augmentation method (AUG1), we up-shift the pitch of the vocal track in intervals of 3, 4, 7, and

12 semitones, representing the minor third, major third, fifth, and octave, respectively. These intervals are commonly used in singing techniques [20]. In order to enhance our data-driven approach, we try the second augmentation method (AUG2), which considers all possible harmony singing scenarios by shifting up the vocal track with intervals from 1 to 12 semitones. For both augmentation methods, we use Eq. 5 to generate duet-singing datasets.

C. Evaluation Datasets and Metrics

For single pitch evaluation, we use the Vocadito dataset [21], which contains 40 short excerpts in 7 different languages of monophonic singing. Since we cannot find public duet singing datasets, we apply audio stems from the Dagstuhl ChoirSet dataset [22], which focuses on choral singing, to create our own test sets for duet singing. The ChoirSet dataset includes recordings from multiple singers performing three pieces in various harmonies: Soprano, Alto, Tenor, and Bass. Each singer was recorded using three types of microphones: a headset microphone (HSM), a larynx microphone (LRX), and a dynamic microphone (DYM). In practice, vocal pitch frequencies typically range from 100 Hz to 400 Hz [23]. Therefore, we extracted tenor and alto audios recorded by the LRX microphone from the dataset, pitch ranging from approximately 130 Hz to 698 Hz, and mixed them together to mimic duet singing for evaluation purposes.

To evaluate the models' efficacy on dual-pitch extraction, we employ metrics mentioned in [24]: Precision (P), Recall (R), F-score (F), FLOPs, and the number of parameters. Predicted pitches falling within the range of the true label, with the tolerant window of $\pm$ 0.5 semitones, are considered correct predictions.

IV. Results and Discussion

In this study, we conduct comprehensive experiments with different model configurations denoted as Proposed 0, Proposed 1, and Proposed 2, as detailed in Table II. Within the Proposed 0 configuration, the number of channels in Harmonic blocks is 64. The Proposed 1 configuration maintains consistency with the Proposed 0 configuration regarding components, albeit with a reduction in the count of Harmonic blocks' channels to 8. The Proposed 2 configuration represents a different combination of modules from the Proposed 1 configuration, lacking one Harmonic-aware block.

TABLE II: Configurations of compared models. It shows the included (O) and excluded (X) components for different models. Used channel numbers (C) are shown in parentheses.

	Conv. 1	**Conv. 2**	**Res. + Har. Block 1**	**Res. + Har. Block 2**
Proposed 0	O	O	O (C: 64)	O (C:64)
Proposed 1	O	O	O (C:8)	O (C:8)
Proposed 2	O	O	X	O (C:8)

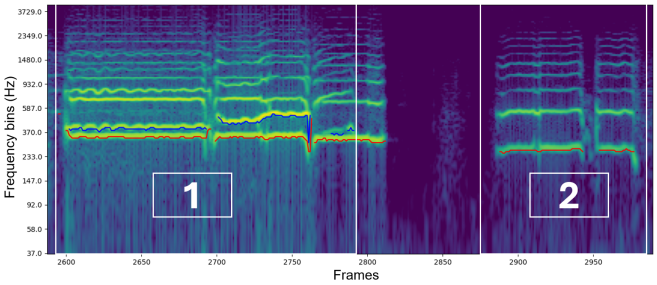

Fig. 5: The example output of the Proposed 2 configuration on a duet song. It shows the pitch tracks of two singers in red and blue colors respectively, overlaid on the log-spectrogram. The left region (1) indicates a duet singing situation, and the right region (2) indicates a solo singing situation.

A. Experiments on Different Thresholds

Firstly, we assess the performance of the proposed model using precision (P), recall (R), and F-score (F) metrics across varying thresholds, α, in Eq. 2. The threshold governs the trade-off between false positives and false negatives, crucial for refining the detection accuracy. Table III summarizes the results on dual-pitch extraction, using the Proposed 2 configuration, with thresholds ranging from 0.2 to 0.6.

Upon examination of the results, it is evident that the optimal F-score is achieved at a threshold of 0.2, with corresponding scores of P = 0.8669, R = 0.8592, and F = 0.8630. It indicates a good balanced precision and recall at this threshold. However, as the threshold is further increased, there is a significant decline in the performance of all metrics. Therefore, we set the threshold to 0.2 for all experiments.

B. Performance Comparison of Data Augmentation Methods on Duet Songs

Results in Tables IV and V demonstrate a huge improvement in the model when using the second data augmentation method (AUG2) for training. All configurations have significant increases in precision, recall, and F-score. Notable, the Proposed 2 configuration in AUG2 outperforms all configurations in AUG1 with P=0.8669, R=0.8592, and F=0.8630. These results show that the AUG2 method is more effective in generalizing the model without increasing the computation. Although these scores are slightly lower than scores of LATE/DEEP [10], the Proposed 2 configuration has a big advantage in computation efficiency with only 7.33M FLOPs compared to LATE/DEEP's

TABLE III: Performance scores of dual-pitch extraction thresholds α in Eq. 2.

α	**P**	**R**	**F**
0.2	0.8669	0.8592	0.8630
0.3	0.8494	0.8418	0.8455
0.4	0.8340	0.8266	0.8302
0.5	0.7974	0.7903	0.7938
0.6	0.7250	0.7186	0.7217

TABLE IV: Performance evaluation on duet songs using AUG1 duet dataset. FLOPS is computed based on an one-second-length signal.

	P	R	F	FLOPs	Params
Proposed 0	0.8227	0.8151	0.8188	39.51M	1.28M
Proposed 1	0.8109	0.8033	0.8070	11.04M	1.26M
Proposed 2	0.7825	0.7749	0.7786	7.33M	0.80M

TABLE V: Performance evaluation on duet songs using AUG2 duet dataset. FLOPS is computed based on an one-second-length signal.

	P	R	F	FLOPs	Params
Proposed 0	0.8829	0.8752	0.8790	39.51M	1.28M
Proposed 1	0.8834	0.8757	0.8795	11.04M	1.26M
Proposed 2	0.8669	0.8592	0.8630	7.33M	0.80M
LATE/DEEP [10]	0.8757	0.8684	0.8720	1.06G	0.59M

1.06G FLOPs. The huge reduction in computation makes Proposed 2 much more suitable for real-time applications. Fig. 5 shows an example output of Proposed 2 on a duet song. It showcases two estimated pitch tracks, represented by red and blue lines respectively, overlaid on the log-spectrogram.

C. Model Performance on Solo Songs

We also assess the performance of compared models on solo songs using a range of metrics, including overall accuracy (OA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), voicing recall (VR), and voicing false alarm (VFA) in Mir_eval tool [17], which are often used in literature of single-pitch detection. As illustrated in Table VI, Proposed 2 demonstrates commendable performance, achieving high accuracy while maintaining efficiency despite parameter reductions and reduced computational time. When compared against YIN [2], a trusted real-time single-pitch detection algorithm, Proposed 2 outperforms it across all metrics. In addition, results in the table also demonstrate Proposed 2 outperforms the spectral branch of the NN model [12] in almost all metrics. The spectral branch of [12] utilizes 11 frames of feature, which are from the first 352 frequency bin of the log-spectrogram, to predict the current pitch. Due to its high complexity, with FLOPs = 1.97G and parameters = 5.26M, it cannot be used for real-time inference.

V. Conclusions

In summary, we have presented a new neural network for real-time dual-pitch extraction to address the need for fast and accurate pitch estimation in audio processing. Our model works well for solo and duet songs. We get comparable results by using spectral features in one framework and reducing the computational cost. We also introduced a new data augmentation method for simulating duet singing. Our results show the potential of neural networks in audio processing and opens up new applications in MIR, speech processing and beyond. Next work will be to optimize and refine our model and explore more applications in real world scenarios.

TABLE VI: Performance evaluation on solo songs. SB is short for 'spectral branch'.

	OA	RPA	RCA	VR	VFA
Proposed 0	0.8386	0.8216	0.8230	0.9694	0.2507
Proposed 1	0.8407	0.8029	0.8066	0.9592	0.1957
Proposed 2	0.8289	0.7964	0.7978	0.9630	0.2215
YIN [2]	0.6106	0.6608	0.6710	0.9519	0.5011
SB of [12]	0.8449	0.7552	0.7604	0.9160	0.1179

References

[1] D. Talkin and W. B. Kleijn, "A robust algorithm for pitch tracking (rapt)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.

[2] A. De Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[3] M. Mauch and S. Dixon, "Pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, IEEE, 2014, pp. 659–663.

[4] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, *Crepe: A convolutional representation for pitch estimation*, 2018. arXiv: 1802.06182 [eess.AS]. [Online]. Available: https://arxiv.org/abs/1802.06182.

[5] H. Chou, M. T. Chen, and T. S. 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)*, 2018, pp. 381–385. DOI: 10.1109/ICASSP.2018.8461483.

[6] T. Kinnunen, E. Karpov, and P. Franti, "Real-time speaker identification and verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 277–288, 2006. DOI: 10.1109/TSA.2005.853206.

[7] X. Sun, "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2002, pp. I-333-I–336. DOI: 10.1109/ICASSP.2002.5743722.

[8] P. McLeod, "Fast, accurate pitch detection tools for music analysis," *Unpublished PhD thesis. Department of Computer Science, University of Otago*, 2008.

[9] O. Mayor, J. Bonada, and A. Loscos, "Performance analysis and scoring of the singing voice," in *Proc. 35th AES Intl. Conf., London, UK*, 2009, pp. 1–7.

[10] H. Cuesta, B. McFee, and E. Gómez, "Multiple f0 estimation in vocal ensembles using convolutional neural networks," *arXiv preprint arXiv:2009.04172*, 2020.

[11] J. Zhang, J. Tang, and L. R. Dai, "RNN-BLSTM based multi-pitch estimation.," in *Interspeech*, vol. 2016, 2016, pp. 1785–1789.

[12] S. Yu, Y. Yu, X. Sun, and W. Li, "A neural harmonic-aware network with gated attentive fusion for singing melody extraction," *Neurocomputing*, vol. 521, pp. 160–171, 2023.

[13] A. Natsiou and S. O'Leary, "Audio representations for deep learning in sound synthesis: A review," in *2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, 2021, pp. 1–8.

[14] W. Wei, P. Li, Y. Yu, and W. Li, "Harmof0: Logarithmic scale dilated convolution for pitch estimation," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2022, pp. 1–6.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pmlr, 2015, pp. 448–456.

[16] S. Stone and E. Spector, "Deep neural network for multi-pitch estimation using weighted cross entropy loss," in *2021 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, IEEE, 2021, pp. 1–3.

[17] C. Raffel, B. McFee, E. J. Humphrey, *et al.*, "Mir_eval: A transparent implementation of common mir metrics.," in *ISMIR*, vol. 10, 2014, p. 2014.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] C. L. Hsu and J. S. R. 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, no. 2, pp. 310–319, 2009.

[20] O. Zulfi, *How to sing harmony: The ultimate guide for beginners*, Accessed: 2024-03-08, 2024. [Online]. Available: https://deviantnoise.com/singing/how-to-sing-harmony/.

[21] R. M. Bittner, K. Pasalo, J. J. Bosch, G. Meseguer-Brocal, and D. Rubinstein, "Vocadito: A dataset of solo vocals with f_0, note, and lyric annotations," *CoRR*, vol. abs/2110.05580, 2021. arXiv: 2110.05580. [Online]. Available: https://arxiv.org/abs/2110.05580.

[22] S. Rosenzweig, H. Cuesta, C. Weiß, F. Scherbaum, E. Gómez, and M. Müller, "Dagstuhl ChoirSet: A multitrack dataset for MIR research on choral singing," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, pp. 98–110, 2020. DOI: 10.5334/tismir.48. [Online]. Available: http://doi.org/10.5334/tismir.48.

[23] iZotope. "How to eq vocals." (2023), [Online]. Available: https://www.izotope.com/en/learn/how-to-eq-vocals.html (visited on 07/17/2023).

[24] M. Bay, A. F. Ehmann, and J. S. Downie, "Evaluation of multiple-f0 estimation and tracking systems.," in *ISMIR*, 2009, pp. 315–320.