



A Joint Network for Singing Melody Extraction from Polyphonic Music with Attention Aggregation and Self-Consistency Training

Jiabo Jing^{1,3}, Ying Hu^{1,3}, Hao Huang¹, Liang He^{1,2}, Zhijian Ou²

¹College of Computer Science and Technology, Xinjiang University, Urumqi, China

²BNRist, Department of Electronic Engineering, Tsinghua University, Beijing, China

³Key Laboratory of Signal Detection and Processing, Xinjiang, Urumqi, China

jingjb@stu.xju.edu.cn, huying@xju.edu.cn

Abstract

Singing melody extraction (SME) is an important task in music information retrieval (MIR). In this paper, we propose a separate spectrum-based SME model and a joint network that combines the pre-trained and spectrum-based models. In the joint network, we design an attention aggregation module (AAM) consisting of cross-attention (CA) and adaptive decision fusion (ADF) to effectively fuse the intermediate features from two models. Furthermore, we introduce a self-consistency training strategy, which utilizes hard and soft labels to supervise two separate models to better obtain the SME task-relevant information. Experimental results show that our proposed method, Joint Network, outperforms six compared state-of-the-art methods, achieving overall accuracy (OA) scores of 91.6%, 92.5%, and 78.9% on the ADC 2004, MIREX 05, and MEDLEY DB datasets, respectively. Visualized results show that the Joint Network can reduce the octave and melody detection errors.

Index Terms: Singing melody extraction, attention aggregation, self-consistency training, music information retrieval

1. Introduction

Singing melody extraction (SME) from polyphonic music is an important task in the field of music information retrieval (MIR), which focuses on producing a sequence of frequency values corresponding to the pitch of the dominant melody from polyphonic music records [1]. SME plays a key role in downstream applications such as music transcription [2], cover song identification [3], and query-by-humming [4].

In SME-related works [5–11], two types of music data representations are usually used as the input of the network: the spectrogram transformed via the short-time Fourier transform (STFT) [5–9] and raw waveform [10, 11]. Wei et al. proposed a robust model (RMVPE) that uses a deep U-Net to extract hidden features and utilizes GRU for pitch prediction [7]. Gao et al. proposed a multi-band time-frequency attention network (MTANet), which uses the band partition scheme to learn the position distribution relationship between the fundamental frequency (F0) and non-F0 components [8]. Yu et al. proposed a review network (REVNet), which pairs low- and high-level features into several groups, and uses dilated convolution with different dilation factors to achieve feature complementary and increase the receptive field [9]. The aforementioned methods all take the STFT-based spectrum features as model’s input, termed as spectrum-based model. On the other hand, early works such as CREPE [10] and DeepF0 [11] utilized raw waveform as input and employed models based on one-dimensional convolutional neural networks for supervised training to estimate the pitch.

Since spectrogram and raw waveform reflect the time-frequency structure and time-domain characteristics of music

respectively, some researchers have attempted to design networks that jointly model features extracted from both data types to further improve SME task performance. Chou et al. proposed a hybrid neural network to simulate the spectral model and the temporal model, then merged the features of the two paths, and used a fully connected layer for melody prediction [12]. Yu et al. proposed a neural harmonic-aware network (NHAN-GAF), which extracts feature representations in parallel from both the raw waveform and spectrogram, employing gated attention mechanisms to fuse the features [13]. However, limited by effective modeling of raw waveform and simple fusion strategy, those attempts do not achieve ideal results in SME task.

Pretrained models learn powerful representations by training large models on large-scale general audio datasets. For example, Dinkel et al. proposed the Dasheng pre-trained encoder model, which was trained on speech corpus of 272,356 hours (containing 3,768 hours of music data) and showed excellent classification performance on 18 different downstream tasks [14]. Some researchers have used pre-trained models (e.g., WavLM [15], Wav2Vec2 [16], etc.) as feature extractor [17], transferring the learned knowledge to downstream tasks, and have achieved promising results in areas such as speech emotion recognition [18] and sound event detection [19].

Motivated by the above observations, we propose a joint network for SME, which consists of spectrum-based and pre-trained models, and design an attention aggregation module to effectively fuse the intermediate features of the two single-branch models. The contributions of our work are as follows:

i) We propose a spectrum-based model for SME, which consists of a stacked convolutional (Conv) block, three time-frequency attention Res2blocks (TFA-Res2Blocks), and a selective feature fusion (SFF) module. Compared to state-of-the-art methods, the model achieves competitive performance.

ii) We also propose a joint network integrating the above-mentioned spectrum-based and pre-trained models, and design an attention aggregation module (AAM) including the cross-attention (CA) and adaptive decision fusion (ADF) to effectively fuse the intermediate features of both.

iii) We introduce a self-consistency training strategy that transfers the knowledge of hard and soft labels from the joint network to two single-branch models. This strategy encourages each single-branch model to extract more task-relevant information without additional parameters.

2. Proposed Method

Fig.1 illustrates the proposed SME framework. Our proposed joint SME network as shown in Fig.1(A), combines the pre-trained and spectrum-based SME models, and the intermediate features F_s and F_a of the two single-branch models are fused

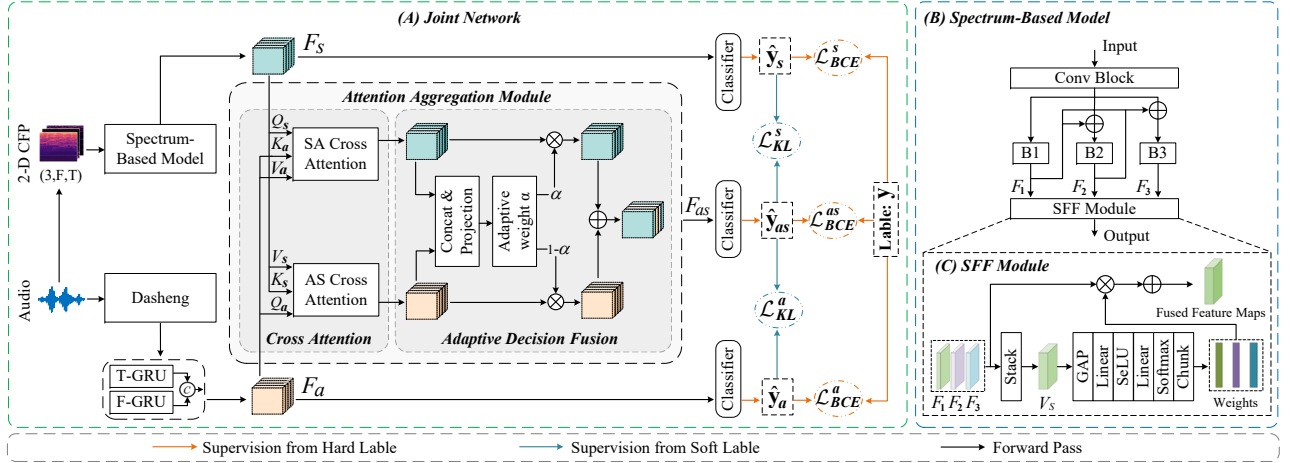


Figure 1: The illustration of our proposed SME framework. Dasheng is the pre-trained model. (A) The overall architecture of the Joint Network. (B) The Spectrum-Based Model. (C) Selective Feature Fusion (SFF) Module. B1, B2, and B3 denote TFA-Res2Blocks. The symbol \odot denotes the concatenating operation along the channel dimension, while \oplus and \otimes indicate element-wise addition and the Hadamard product, respectively.

by an attention aggregation module (AAM). T-GRU and F-GRU are used to model the hidden representations extracted from the pre-trained model, ensuring that F_a has the same output dimensions as F_s . During training, the intermediate features from the pre-trained and spectrum-based models, as well as the fused features output from AAM, are fed into a classifier respectively to obtain melody predictions. Furthermore, we calculate two KL divergences, \mathcal{L}_{KL}^s and \mathcal{L}_{KL}^a , between the outputs of the top and bottom branch classifiers and that of the joint network, respectively. Note that in the testing phase, only the classifier fed by fused features of AAM outputs the melody predictions (i.e., the top and bottom branch classifiers in Fig.1(A) exist only during the training phase). The Combined Frequency and Periodicity (CFP) representation [20] $X \in \mathbb{R}^{3 \times F \times T}$ in Fig.1(A) is obtained by pre-processing the audio signal. It contains three components: a power-scaled spectrogram, a generalized cepstrum (GC) [21] and a generalized cepstrum of spectrum (GCoS) [22]. F and T denote the numbers of frequency bins and time frames.

2.1. Spectrum-Based Model

Inspired by the powerful multi-scale ability of Res2Net [23], we designed a spectrum-based SME model, as shown in Fig.1(B). Table 1 describes the components of each block. The model consists of three components: i) Conv block is used to extract shallow features. ii) TFA-Res2Blocks (B1, B2, and B3) with various dilations, where the time-frequency attention block (TFA-Block) [24] is applied to assign different weights along the time and frequency axes, mimicking human hearing. iii) Selective feature fusion (SFF) module for feature aggregation. Specifically, the input is first processed by conv blocks and then fed into three cascading TFA-Res2Blocks to extract three groups of deep features F_i ($i \in [1, 2, 3]$), each with distinct semantics. The process of operation can be written as:

$$F_1 = B1(X_{con}) \quad (1)$$

$$F_2 = B2(F_1 + X_{con}) \quad (2)$$

$$F_3 = B3(F_1 + F_2 + X_{con}) \quad (3)$$

where X_{con} denotes the features processed by conv block.

Moreover, we design the SFF module to dynamically select three groups of features F_i with different semantics and fuse them. As shown in Fig.1(C), three groups of features

F_i are firstly stacked together to obtain V_s , and then a global average pooling (GAP) operation is performed to obtain the initial weights of each channel in the feature map. Two linear layers are used to recalibrate the importance of each channel, and *SeLU* is added between these two linear layers. The *softmax* layer is applied to obtain the final weights. Finally, the Hadamard product is performed on the three groups of features F_i and their corresponding weights to obtain the weighted feature maps, which are fused by element-wise addition. The fused features contain rich information selected from the three groups of features F_i .

2.2. Pre-trained Model

We use the Deep Audio-Signal Holistic Embeddings (Dasheng) [14] to extract music-related acoustic features from the waveforms of audio. Dasheng is a pre-trained encoder model based on the masked autoencoders [25] architecture, which is composed of a transformer-based asymmetric encoder-decoder. The pre-trained Dasheng model applied in this work was trained on 272,356 hours of diverse audio datasets, including 268,588 hours of general audio and 3,768 hours of music data, with a total of 1.2 billion parameters.

The pre-trained Dasheng model excels at capturing richer speech and music information, and the extracted Dasheng features can be directly applied to a variety of downstream classification tasks without parameterization [14]. Inspired by this, we froze the pre-trained Dasheng model and directly utilized it to extract features from audio signals. T-GRU and F-GRU are then used to model the acoustic representations extracted from the Dasheng model, which contain rich musical information.

2.3. Attention Aggregation Module

We design an attention aggregation module (AAM) to effectively fuse the features extracted from the spectrum-based model branch and the pre-trained model branch. As shown in Fig.1(A), AAM consists of two parts: (SA/AS) cross-attention (CA) and adaptive decision fusion (ADF).

Cross Attention. The cross attention we used is based on multi-head self-attention [27]. This way enables the network to combine features F_a and F_s from both branches of the pre-trained model and the spectrum-based model, providing complementary information to each other while retaining key cues.

Table 1: The spectrum-based model architecture and configuration. For the Conv2D($k \times k$, $d \times d$) operation, $k \times k$ denotes the kernel size, and $d \times d$ denotes the dilation spacing. Each Conv layer is followed by the Mish [26] activation function and a batch normalization layer. Dimensions refer to (channels, frequency, time). For each TFA-Res2Block, feature maps were split into s (we set $s=8$ in this work) feature map subsets along channel dimension after the 1×1 convolution. TFA-Block denotes time-frequency attention block [24].

Layer	Block Structure	Output Size
Conv Block $\times 2$	Conv2D(3×3 , 1×1)	$64 \times F \times T$
	Conv2D(5×3 , 1×1)	
TFA-Res2Block (B1)	Conv2D(1×1 , 1×1), $s = 8$	$128 \times F \times T$
	Conv2D(3×3 , 2×2)	$16 \times F \times T$
	Conv2D(1×1 , 1×1)	$64 \times F \times T$
	TFA-Block	$64 \times F \times T$
TFA-Res2Block (B2)	Conv2D(1×1 , 1×1), $s = 8$	$128 \times F \times T$
	Conv2D(3×3 , 3×3)	$16 \times F \times T$
	Conv2D(1×1 , 1×1)	$64 \times F \times T$
	TFA-Block	$64 \times F \times T$
TFA-Res2Block (B3)	Conv2D(1×1 , 1×1), $s = 8$	$128 \times F \times T$
	Conv2D(3×3 , 4×4)	$16 \times F \times T$
	Conv2D(1×1 , 1×1)	$64 \times F \times T$
	TFA-Block	$64 \times F \times T$
SA Module	—	$64 \times F \times T$

$F=360$, $T=128$. The downsampling operation is not applied to avoid potential information loss.

Here, we use the intermediate feature F_s , extracted spectrum-based model, as the query to illustrate the computation method of the SA cross-attention mechanism. Define:

$$\hat{F}_s = LN(MHA(Q_s, K_a, V_a) + F_s) \quad (4)$$

$$MHA(Q_s, K_a, V_a) = Concat(head_1, \dots, head_h) \quad (5)$$

$$\begin{aligned} head_h &= Attention(Q_h^s, K_h^a, V_h^a) \\ &= Softmax\left(\frac{Q_h^s (K_h^a)^T}{\sqrt{d_k}}\right) V_h^a \end{aligned} \quad (6)$$

$$Q_h^s = F_s W_h^Q + B_h^Q \quad (7)$$

$$K_h^a = F_a W_h^K + B_h^K \quad (8)$$

$$V_h^a = F_a W_h^V + B_h^V \quad (9)$$

$$F_{out}^s = LN(FFN(\hat{F}_s) + \hat{F}_s) \quad (10)$$

where W_h^u and B_h^u are learnable weight matrices and biases, here $u=(Q, K, V)$. The h (set to 8 in this work) denotes the h -th cross-attention. LN , MHA , and FFN denote the layer normalization, multi-head attention, and feed-forward layer, respectively. When calculating F_{out}^a , the method remains consistent with F_{out}^s , except that the query is based on the features F_a extracted from the branch of the pre-trained model.

Adaptive Decision Fusion. We introduce adaptive decision fusion to dynamically adjust the joint network’s dependence on each output feature (F_{out}^s/F_{out}^a) from the (SA/AS) CA. This adaptivity optimizes the network by selectively emphasizing task-relevant features and discarding inessential information, thereby allowing for more adaptive and robust decision-making. Specifically, the process re-fuses features F_{out}^s and F_{out}^a , a linear layer and *sigmoid* activation operation are used to generate a gated parameter α for F_{out}^s and thus $1-\alpha$ for F_{out}^a . The above process can be formulated as:

$$\alpha = Sigmoid(w(Concat(F_{out}^s, F_{out}^a)) + bias) \quad (11)$$

$$F = \alpha \odot F_{out}^s + (1 - \alpha) \odot F_{out}^a \quad (12)$$

where w and $bias$ are trainable parameters, and \odot denotes the Hadamard product.

2.4. Self-consistency Training Strategy

Inspired by the self-distillation proposed by Li et al. [28], we design a self-consistency training strategy to transfer the latent knowledge existing in hard and soft labels from the joint network to the two single-branch models. This strategy aims to improve the quality of the single-branch features before their fusion, which in return benefits the joint network to learn more discriminative representations. To achieve this, it combines binary cross entropy (BCE) loss and Kullback-Leibler (KL) divergence loss.

The feature F_m is fed into a classifier consisting of a linear layer followed by a *softmax* layer to output the probabilities of frame-level melodies. Define:

$$E_m = w_m \cdot ReLU(F_m) + bias_m \quad (13)$$

$$\hat{y}_m = Softmax(E_m) \quad (14)$$

$$\hat{y}_m^\tau = Softmax(E_m/\tau) \quad (15)$$

where $m \in (s, a, as)$, w_m and $bias_m$ are trainable parameters. τ is the temperature to soften \hat{y}_m (written as \hat{y}_m^τ after softened) and a higher τ produces a softer probability distribution. Here, we empirically set τ to 3.

Binary Cross Entropy Loss. The BCE loss is used to minimize the difference between the predicted probability of the feature F_m and the ground truth (i.e., hard label).

$$\mathcal{L}_{BCE}^m = BCE(\hat{y}_m, y) \quad (16)$$

where \hat{y}_m denotes the prediction of the features F_m by the classifier, and y the one-hot ground truth label.

KL Divergence Loss. Soft label containing informative dark knowledge can be used as training supervision [28]. To make the output probabilities of the two single-branch models approximate the output probability (i.e., soft label) of the joint network, the KL divergence loss between them is minimized.

$$\mathcal{L}_{KL} = KL(\hat{y}_m^\tau, \hat{y}_{as}^\tau) \quad (17)$$

where \hat{y}_m^τ and \hat{y}_{as}^τ are soften probability distributions of the feature F_m by the classifier.

In summary, the above loss function is used to achieve joint optimization of the network.

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{BCE} + \lambda_2 \mathcal{L}_{KL} \quad (18)$$

$$\mathcal{L}_{BCE} = \sum_{m \in \{s, a, as\}} \mathcal{L}_{BCE}^m \quad (19)$$

$$\mathcal{L}_{KL} = \sum_{m \in \{s, a\}} \mathcal{L}_{KL}^m \quad (20)$$

where λ_1 and λ_2 are hyperparameters empirically selected.

3. Experiment

3.1. Experimental Setup

Following the literature [6], we also select 1,000 Chinese pop song clips from the MIR-1K dataset [29] and 35 vocal tracks from the MedleyDB [30] for training. For the test set, we choose 12 clips from ADC2004, 9 clips from MIREX05, and 12 clips from MedleyDB. All samples contain the singing melodies. Note that the training and testing datasets do not overlap.

For the signal processing part, we adopt the same parameter settings as TONet [6], with a sampling rate of 8 KHz, a window size of 768 samples, and a hop size of 80 samples. For CFP representation, we set the time dimension to $T=128$ frames (1.28 secs) and the number of frequency bins to $F=360$, covering 6

Table 2: The comprehensive results of the Joint Network and compared methods on three datasets.

Dataset	-	-	ADC 2004						MIREX 05						MEDLEY DB					
Metrics	Reference	Par. (M)	VR↑	VFA↓	RPA↑	RCA↑	ROA↑	OA↑	VR↑	VFA↓	RPA↑	RCA↑	ROA↑	OA↑	VR↑	VFA↓	RPA↑	RCA↑	ROA↑	OA↑
FTANet [†] [5]	ICASSP'21	3.4	85.6	8.6	78.3	78.4	83.6	81.7	85.7	5.8	80.4	81.3	84.1	84.5	63.2	11.6	57.8	58.8	59.8	71.2
TONet [†] [6]	ICASSP'22	147.0	86.1	8.9	80.2	80.3	84.9	82.2	88.4	7.8	83.4	83.5	86.7	85.4	64.6	12.3	56.5	58.7	60.4	71.6
MTANet [†] [8]	INTERSPEECH'23	0.3	89.3	11.4	84.4	84.5	87.5	85.4	88.7	5.4	83.6	83.6	87.6	88.3	70.6	15.4	62.4	64.1	65.9	73.9
RMVPE [†] [7]	INTERSPEECH'23	470.4	83.4	14.6	81.1	81.4	82.2	82.7	88.5	6.8	85.4	85.5	86.9	88.5	70.7	17.5	55.5	59.6	62.6	67.7
NHAN-GAF [‡] [13]	NEUROCOMPUTING'23	9.8	85.8	29.3	79.0	80.2	-	81.7	84.3	21.7	79.6	79.9	-	83.8	78.2	19.4	54.6	55.3	-	71.6
REVNet [†] [9]	ICME'24	-	86.1	5.5	84.1	85.0	-	85.6	86.6	8.9	81.7	82.2	-	85.5	72.4	10.0	67.2	68.3	-	74.8
Joint Network	-	8.8	93.9	5.1	91.1	91.5	91.8	91.6	94.8	4.5	90.4	90.4	93.7	92.5	83.5	13.7	72.6	73.8	75.4	78.9

[†]: The results of our re-implemented according to their official codes.

[‡]: The results of their reported.

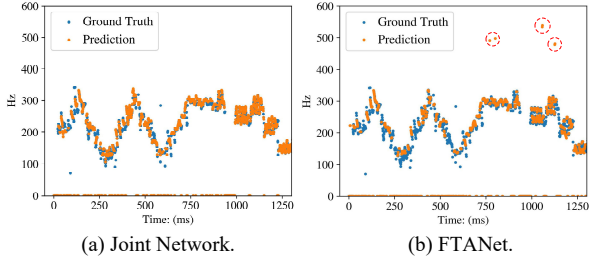


Figure 2: Visualized comparison between our proposed Joint Network and FTANet on the song of “opera_male3”. The circle marks in the figure are octave error (e.g., mistake A3 as A2 or A4, etc).

Table 3: Ablation study of network components on three datasets. OA_{ADC} , OA_{MIR} and OA_{MED} represent the OA of the joint network on the ADC2004, MIREX05, and MEDLEY DB datasets after removing different components, respectively.

Systems	Par.(M)	$OA_{ADC}↑$	$OA_{MIR}↑$	$OA_{MED}↑$
Spectrum_Based	0.7	86.2	88.3	75.1
Pre-trained_Based	5.7	84.9	87.1	74.5
Joint Network	8.8	91.6	92.5	78.9
w/o CA	6.5(↓2,3)	89.2	89.9	76.8
w/o ADF	8.7(↓0,1)	90.4	91.0	77.8
w/o AAM	6.4(↓2,4)	88.5	89.1	76.3

octaves, each containing 60 bins. The frequency range spans from 32.5 Hz to 2050 Hz, covering C1 to B6. All models are trained and tested in NVIDIA A40 GPUs and implemented in PyTorch. For trainable parameters update, we use a batch size of 8, the Adam optimizer [31] with a learning rate of 0.0001.

We use six metrics for performance evaluation: overall accuracy (OA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), voicing recall (VR), and voicing false alarm (VFA) from `mir_eval` library [32]. In the literature [1], OA is often considered the most crucial evaluation metric. The raw octave accuracy (ROA) [6] serves as an additional metric for evaluating octave prediction accuracy.

3.2. Ablation Study

We conducted multiple ablations to verify the effectiveness of key components in Joint Network, as shown in Tables 3 and 4. Combining the results in Tables 3 and 4, we summarize the following observations: i) Our proposed spectrum-based model achieves better performance with fewer parameters. ii) Modeling the representations extracted from the pre-trained Dasheng model using T-GRU and F-GRU achieved scores comparable to existing state-of-the-art methods. iii) The joint network containing all components achieves the best OA on the three datasets.

Table 4: Ablation study of the loss function on three datasets.

\mathcal{L}_{BCE}^{as}	\mathcal{L}_{BCE}^s	\mathcal{L}_{BCE}^a	\mathcal{L}_{KL}^s	\mathcal{L}_{KL}^a	$OA_{ADC}↑$	$OA_{MIR}↑$	$OA_{MED}↑$
✓	✗	✗	✗	✗	90.1	91.4	77.4
✓	✓	✓	✗	✗	91.1	92.0	78.2
✓	✗	✗	✓	✓	90.6	91.8	77.7
✓	✓	✓	✓	✓	91.6	92.5	78.9

OA_{ADC} , OA_{MIR} and OA_{MED} have the same description as Table 3 above.

iv) When the designed CA, ADF and AAM modules were removed, the OA on all three datasets decreased to varying degrees. v) When removing the losses \mathcal{L}_{BCE}^s and \mathcal{L}_{BCE}^a between the intermediate features F_s and F_a extracted by two single branches and the label, the OA decreases by about 1% on average across the three datasets. After further removal of losses \mathcal{L}_{KL}^s and \mathcal{L}_{KL}^a , OA continues to decrease.

3.3. Comprehensive Performance Comparison

We compared our proposed Joint Network with six state-of-the-art methods, including FTANet [5], TONet [6], MTANet [8], RMVPE [13], NHAN-GAF [13], and REVNet [9] on three public datasets as shown in Table 2. By comparison, we can observe that the Joint Network achieves the best performances on all three datasets, significantly exceeding the existing methods. Compared with the similar structure of NHAN-GAF, our proposed Joint Network achieves the improvement on OA scores of 12.1%, 10.4%, and 10.2% on the ADC 2004, MIREX 05 and MEDLEY DB datasets, respectively.

To explore what types of errors are solved by the proposed Joint Network, a case study is performed on an opera song: “opera_male3.wav” in the ADC2004 dataset. We choose FTANet [5] to compare with due to its effectiveness and popularity. As depicted in Fig.2, diagram (a) has no octave errors (i.e., vertical jumps in the contours) compared to (b). Furthermore, from 0-1200 ms in Fig.2 (b), we can find a large number of melody detection errors (i.e., predicting a melody frame as a non-melody one), which are significantly reduced in Fig.2 (a). The above results fully verify the effectiveness and advantages of our proposed Joint Network in the SME task.

4. Conclusion

In this paper, we propose a separate spectrum-based SME model and a joint network combining pre-trained and spectrum-based models, and design an attention aggregation module including the cross-attention and adaptive decision fusion for the joint network to fuse the intermediate features from two models. We introduce a self-consistency training strategy to optimize the joint network. Ablation study results indicate that our proposed separate spectrum-based model achieves better performance with fewer parameters. Experimental results show that the Joint Network outperforms the existing state-of-the-art methods on three datasets.

5. Acknowledgements

This work is supported by the Open Research Fund Program of Beijing National Research Center for Information Science and Technology (04410307724), Key research and development plan projects of the au-tonomous region (2023B01030).

6. References

- [1] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard, “Melody extraction from polyphonic music signals: Approaches, applications, and challenges,” *IEEE Signal Processing Magazine*, vol. 31, no. 2, pp. 118–134, 2014.
- [2] N. Kroher and E. Gómez, “Automatic transcription of flamenco singing from polyphonic music recordings,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 901–913, 2016.
- [3] X. Du, K. Chen, Z. Wang, B. Zhu, and Z. Ma, “Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 616–620.
- [4] C.-C. Wang and J.-S. R. Jang, “Improving query-by-singing/humming by combining melody and lyric information,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 798–806, 2015.
- [5] S. Yu, X. Sun, Y. Yu, and W. Li, “Frequency-temporal attention network for singing melody extraction,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 251–255.
- [6] K. Chen, S. Yu, C.-i. Wang, W. Li, T. Berg-Kirkpatrick, and S. Dubnov, “Tonet: Tone-octave network for singing melody extraction from polyphonic music,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 621–625.
- [7] H. Wei, X. Cao, T. Dan, and Y. Chen, “Rmvpe: A robust model for vocal pitch estimation in polyphonic music,” in *INTERSPEECH 2023*, 2023, pp. 5421–5425.
- [8] Y. Gao, Y. Hu, L. Wang, H. Huang, and L. He, “Mtanet: Multi-band time-frequency attention network for singing melody extraction from polyphonic music,” in *Proc. INTERSPEECH 2023*, 2023, pp. 5396–5400.
- [9] S. Yu, X. He, and Y. Zhang, “Revnet: A review network with group aggregation fusion for singing melody extraction,” in *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2024, pp. 1–6.
- [10] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, “Crepe: A convolutional representation for pitch estimation,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.
- [11] S. Singh, R. Wang, and Y. Qiu, “Deepf0: End-to-end fundamental frequency estimation for music and speech signals,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 61–65.
- [12] 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)*. IEEE, 2018, pp. 381–385.
- [13] 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.
- [14] H. Dinkel, Z. Yan, Y. Wang, J. Zhang, Y. Wang, and B. Wang, “Scaling up masked audio encoder learning for general audio classification,” in *Interspeech 2024*, 2024.
- [15] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [16] A. Baeovski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
- [17] H. Zou, Y. Si, C. Chen, D. Rajan, and E. S. Chng, “Speech emotion recognition with co-attention based multi-level acoustic information,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7367–7371.
- [18] Y. Hu, H. Yang, H. Huang, and L. He, “Cross-modal features interaction-and-aggregation network with self-consistency training for speech emotion recognition,” in *Proc. Interspeech 2024*, 2024, pp. 2335–2339.
- [19] Y. Li, X. Wang, H. Liu, R. Tao, L. Yan, and K. Ouchi, “Semi-supervised sound event detection with local and global consistency regularization,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 271–275.
- [20] L. Su and Y.-H. Yang, “Combining spectral and temporal representations for multipitch estimation of polyphonic music,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 10, pp. 1600–1612, 2015.
- [21] T. Kobayashi and S. Imai, “Spectral analysis using generalised cepstrum,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1235–1238, 1984.
- [22] L. Su, “Between homomorphic signal processing and deep neural networks: Constructing deep algorithms for polyphonic music transcription,” in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 884–891.
- [23] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, “Res2net: A new multi-scale backbone architecture,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 652–662, 2019.
- [24] Q. Zhang, X. Qian, Z. Ni, A. Nicolson, E. Ambikairajah, and H. Li, “A time-frequency attention module for neural speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 462–475, 2022.
- [25] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.
- [26] D. Misra, “Mish: A self regularized non-monotonic activation function,” *Accepted by BMVC*, 2020.
- [27] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.
- [28] M. Li, D. Yang, Y. Lei, S. Wang, S. Wang, L. Su, K. Yang, Y. Wang, M. Sun, and L. Zhang, “A unified self-distillation framework for multimodal sentiment analysis with uncertain missing modalities,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 9, 2024, pp. 10 074–10 082.
- [29] 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.
- [30] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, “Medleydb: A multitrack dataset for annotation-intensive mir research,” in *ISMIR*, vol. 14, 2014, pp. 155–160.
- [31] D. Kingma, “Adam: a method for stochastic optimization,” in *Int Conf Learn Represent*, 2014.
- [32] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “Mir.eval: A transparent implementation of common mir metrics,” in *ISMIR*, vol. 10, 2014, p. 2014.