




MusicYOLO: A Vision-Based Framework for Automatic Singing Transcription

Xianke Wang , Bowen Tian , Weiming Yang, Wei Xu , *Member, IEEE*, and Wenqing Cheng

Abstract—Automatic singing transcription (AST), which refers to the process of inferring the onset, offset, and pitch from the singing audio, is of great significance in music information retrieval. Most AST models use the convolutional neural network to extract spectral features and predict the onset and offset moments separately. The frame-level probabilities are inferred first, and then the note-level transcription results are obtained through post-processing. In this paper, a new AST framework called MusicYOLO is proposed, which obtains the note-level transcription results directly. The onset/offset detection is based on the object detection model YOLOX, and the pitch labeling is completed by a spectrogram peak search. Compared with previous methods, the MusicYOLO detects note objects rather than isolated onset/offset moments, thus greatly enhancing the transcription performance. On the sight-singing vocal dataset (SSVD) established in this paper, the MusicYOLO achieves an 84.60% transcription F1-score, which is the state-of-the-art method.

Index Terms—AST, note object detection, spectrogram peak search.

I. INTRODUCTION

SIGHT singing is a process in which music beginners read scores and sing to improve their perception of rhythm and pitch, which is of great significance in primary music education. In traditional sight-singing practice, students often need one-on-one, face-to-face guidance from teachers, and the learning cost is high. An efficient and accurate automatic singing transcription (AST) model can reduce the burden on teachers. However, it can also enable students to receive timely feedback and facilitate self-practice, which is useful for music beginners. Therefore, this paper focuses on the issue of singing transcription in sight-singing and also investigates general AST.

AST infers the onset, offset, and pitch from the singing audio. At present, there are three main AST methods. In the first class of methods, end-to-end approaches are used to obtain frame-level onset, offset, and pitch probabilities, and then note-level

Manuscript received 10 March 2022; revised 5 August 2022 and 20 October 2022; accepted 20 October 2022. Date of publication 10 November 2022; date of current version 2 December 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFC3340803 and in part by the National Natural Science Foundation of China under Grant 61877060. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Chuang Gan. (Corresponding author: Wei Xu.)

The authors are with the Hubei Key Laboratory of Smart Internet Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: m202072113@hust.edu.cn; m202072111@hust.edu.cn; yyweiming@hust.edu.cn; xuwei@hust.edu.cn; chengwq@mail.hust.edu.cn).

Digital Object Identifier 10.1109/TASLP.2022.3221005

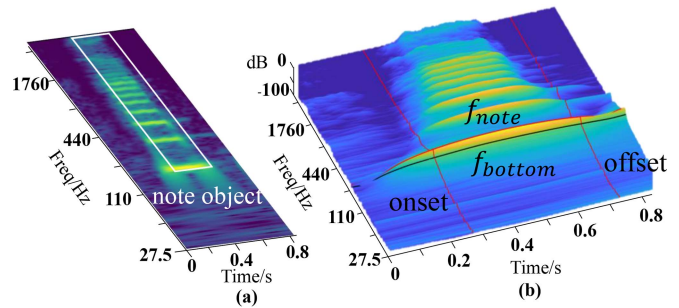


Fig. 1. Design of the transcription scheme.

results are obtained through post-processing. For example, a convolutional neural network (CNN) was first used [1] to extract high-dimensional features of the spectrogram, and then used fully connected layers to obtain frame-level transcription results. In the second class of methods, the onset and offset are obtained first, and the pitch is obtained through pitch extraction. For example, correntropy [2] was used to detect onset/offset first, and then the pitch was obtained using standard pitch tracking algorithms. In the third class of methods, F0 tracking is first conducted to obtain a frame-level pitch curve. The note-level onset, offset, and pitch are obtained through post-processing of the pitch curve. For example, Tony software [3] first tracked the pitch through the pYIN algorithm [4]. Then, the Viterbi decoding algorithm was used based on the hidden Markov model (HMM) to obtain the note-level transcription results.

This paper proposes a note-level AST framework called MusicYOLO. MusicYOLO is inspired by the perspective of sound event detection and based on object detection. It includes pre-processing, note detection, and pitch labeling. The pre-processing module transforms one-dimensional (1D) audio sequences into a two-dimensional (2D) spectrogram. The note detection module obtains the note-level onset/offset. The pitch labeling module extracts each note's pitch. The innovation of this paper is that we use the objection detection method to accomplish the note detection task from a macro perspective, and we obtain the pitch using a search method instead of signal calculation. The following sections will introduce note detection and pitch labeling briefly.

Note detection obtains the note object using the object detection method YOLOX. Fig. 1(a) shows that MusicYOLO derives the bounding boxes from a macro perspective through the YOLOX object detection model. After that, a post-processing method obtains the onset, offset, and f_{bottom} from the left,

right, and bottom boundaries, respectively. Compared with previous frame-based models, MusicYOLO detects a complete note rather than a single onset or offset time, so its onset/offset detection performance is better. Moreover, MusicYOLO base on object detection has a good positive sample matching strategy, which can better handle the imbalance of positive and negative samples. This can greatly reduce the risk that frame-based models tend to miss notes.

Pitch labeling uses a spectrogram peak search method to obtain the pitch of each note. Fig. 1(b) shows that MusicYOLO obtains the onset, offset, and f_{bottom} through the note detection module. The peak value in the rectangular spectrogram region with the frequency range of $[f_{bottom} - \Delta f, f_{bottom} + \Delta f]$ is searched, where Δf represents the frequency range of our search algorithm. The frequency value of this point is the final pitch value, which is called f_{pitch} . The previous pitch labeling algorithm calculates the frequency value of each frame through signal processing methods. This paper proposes a novel and effective peak search approach to obtaining the pitch directly using f_{bottom} . Compared with the previous pitch labeling methods, the method in this paper uses a search method rather than calculation to gain the pitch.

In general, the innovations of this paper are as follows. First, this paper uses an object detection model to detect notes in the singing transcription field for the first time. Our object-detection-based approach's onset/offset detection performance is better than the previous frame-based approach. Second, this paper uses the peak search method to extract the precise pitch, called f_{pitch} . The technique is straightforward. The experiment results show that the onset detection, offset detection, and transcription of MusicYOLO achieve F1-scores of 96.17%, 98.07%, and 85.95% on our sight-singing vocal dataset (SSVD),¹ respectively. At the same time, our MusicYOLO performs better than other methods on other datasets. These experiment results prove that our MusicYOLO is a state-of-the-art framework in the field of AST.

The second chapter introduces the related work of AST. The third chapter gives a detailed implementation of the MusicYOLO framework. The fourth chapter introduces the dataset and evaluation metrics used in this paper. The fifth chapter describes the experiment results of MusicYOLO compared with other approaches. The final chapter provides the conclusion. Our code will be released when this paper is published.

II. RELATED WORK

A. Note Segmentation

Note segmentation refers to extracting the onsets and offsets from the given audio. Note segmentation methods can be roughly divided into methods based on pitch information and methods based on spectrogram features. A high-performance note segmentation algorithm is of great significance for subsequent pitch extraction.

Part of the research first extracts the pitch curve and then completes the note segmentation through post-processing of the

pitch curve. Pitch and audio amplitude curves were used in [5] for note segmentation. Inspired by McNab, an improved method called Spith was proposed in [6], whereby the YIN algorithm [7] was used to obtain a novel hysteresis process on the pitch-time curve, which further improved the performance of note segmentation. Much research has also been [8] done on singing using a novel contour filtering process so as to eliminate pitch contour segments originating from the guitar accompaniment and formulate a set of onset detection functions based on volume and pitch characteristics.

Some studies have considered the spectrogram information to obtain better note segmentation performance. In [2], a new detection function and a peak-picking method were proposed for onset/offset detection in the singing voice based on frequency domain information, which achieved good results in note segmentation on the Korean Cappella dataset [9]. At the same time, deep learning networks have demonstrated strong capabilities in detecting onsets and offsets in automatic music transcription [10]. In [11], an onset detection method was first proposed based on the CNN network, which significantly outperformed previous methods. However, the onset detector still produces many errors owing to a large amount of portamento and rich vibrato in vocal music. According to the MIREX 2018 audio onset detection competition, the best F1-score of singing onset detection is only 61.94%, which is at least 10% lower than the onset F1-scores of other music instruments.

B. Pitch Extraction

Pitch extraction refers to the estimation of the F0 trajectories of the audio signals, which has been conducted on speech [12], [13], singing voices [14], and music instruments [15]. Some audio transcription tasks first perform onset/offset detection and then use the pitch extraction module to extract pitches separately. At present, pitch extraction is mainly divided into traditional signal processing methods and deep learning methods.

In traditional methods, most pitch estimation methods can be divided into those based on time-domain information [7], [16], [17], [18], frequency domain information [19], [20], [21], and both [22], [23]. In [24], a frequency-domain method was proposed for pitch extraction from speech signals, which is rugged and can tolerate a considerable amount of additional noise. Most time-domain methods are based on analyzing the local maxima of the autorelation function [16], which generates interval errors. More robust algorithms like PRAAT [17], PART [18], and YIN [7] have been proposed to solve this problem. For instance, pYin was proposed in [4], which greatly improved the recall and precision and became the best solution among traditional methods.

However, deep learning also promotes the development of pitch extraction. In [25], machine learning methods were pioneered to address pitch estimation in speech. The authors first extracted hand-picked spectral features and then introduced neural networks for further computation. In [26], a novel pitch extraction method was proposed based on a jointly trained

¹[Online]. Available: <https://github.com/xk-wang/SSVD-v2.0>

neural network, investigated BLSTM-RNN for pitch estimation, and considered the bottleneck features in a pitch estimation task, which was superior to other state-of-the-art pitch estimation algorithms at that time [25], [27]. In 2018, CREPE was proposed [28], which achieved the best performance of the deep neural networks (DNNs) in pitch estimation tasks, maintaining over 90 cents raw pitch accuracy even for a strict evaluation threshold of just 10 cents. CREPE is a data-driven pitch tracking algorithm based on a deep CNN that operates directly on the time-domain waveform. Considering the difficulty of obtaining large accurate pitch annotations, it was proposed [29] that a neural network be trained with a self-supervised learning mechanism, which could generate absolute pitches without access to manually labeled datasets and achieved accuracy close to that of supervised learning.

C. Singing Transcription

The sight-singing transcription presented in this paper is also a form of singing transcription. Singing transcription is challenging even in the case of monophonic signals without accompaniment as a result of singers' peculiarities and singing skills [30].

Singing transcription has a long research history. Traditional methods are mainly based on signal processing and analysis. Initially, the transcription accuracy was improved in [31] by changing the acoustic front-end of the QBM. Moreover, in [8], the author explored the limitations of state-of-the-art transcription systems to improve flamenco singing transcription and achieved 85% accuracy. In 2021, a contour filtering model was proposed [32] based on audio signal analysis. The author introduced the edge detection method in image processing to search for all contours, which realized the study of melody contour features.

There has been much research on transcription based on deep learning frameworks. A transcription system was proposed in [33] for a singing melody in polyphonic music signals based on the DNN model. Compared with the well-known state-of-the-art method [34] at the time, the system's way of distinguishing different f_0 by the DNN showed a significant improvement, with an average raw pitch accuracy gain of 20%. Using HMM, a probabilistic transcription method was proposed in [35] for monophonic singing melodies. Compared with the existing Tony [3] (which uses a similar HMM note model) and SiPTH [6] methods, the model achieved a better effect. In [36], the author proposed a Bayesian hierarchical hidden semi-Markov model (HHSMM), which generates a note sequence and consists of three sub-models describing local keys, pitches, and onset score times. Later, a CRNN-HSMM hybrid model was proposed in [37], estimating the most likely notes from the music signal using the Viterbi algorithm. This method improved the performance of AST and was superior to HSMM-based method, the most advanced method at that time [36].

D. Audio Event Detection

Audio event detection (AED) has been an active research area in recent years. Its research methods also can be applied to human voice transcription or monophonic music instrument transcription. In addition to focusing on improving accuracy [38], [39], [40], [41], other aspects of research have been applied to AED, including noise robustness [42], [43], [44], overlapping event processing [45], [46], [47], [48], early event detection [49], multi-channel fusion [50], and universal representation [51].

Among traditional approaches, a common trend in AEDs is through machine learning (e.g., by support vector machines (SVMs) [52]), which is performed on long sliding windows. In [53], a high-quality event classifier was proposed to increase the verification step to improve the detection system. When a single-channel AED integrates multiple audio channels, a naive fusion strategy will deteriorate the system instead [54], [55], [56]. In [57], an efficient and straightforward multi-channel fusion framework was proposed based on the probabilities for joint acoustic event detection and classification to solve this problem.

At the same time, the development of deep learning has also promoted the development of audio event detection. In [58], a novel approach based on R-FCN was proposed for audio event detection and classification. The input can be a 2D representation of any length, and the location of the audio event can be directly output. Inspired by Faster RCNN, a region-based convolutional recurrent neural network (R-CRNN) was proposed in [59] for audio event detection, and its event-based error rate (ER) was reduced by half on the DCASE 2017 Challenge dataset [60]. With the advancement of deep learning methods, the amount of data required has also increased substantially, but in practice, richly annotated data are often difficult to obtain and generate. In terms of weakly supervised learning, both [61] and [62] proposed a framework to learn acoustic event detectors using only weakly labeled data, which can be obtained directly from online platforms. In recent years, there are also some studies [63] that consider both audio and video modalities.

III. THE PROPOSED MODEL

As shown in Fig. 2, the MusicYOLO framework is divided into three modules, including pre-processing, note detection and pitch labeling. The pre-processing module uses the constant Q transform (CQT) to convert the 1D audio sequence into a 2D spectrogram. The note detection module uses the YOLOX object detection model to locate the notes and then obtains the onset, offset, and f_{bottom} through the post-processing of the bounding boxes. The pitch labeling module searches the peak value in the spectrogram matrix to obtain the fundamental frequency, called f_{pitch} .

A. Preprocessing

The pre-processing module includes noise reduction, time-frequency transformation, linear intensity mapping, and spectrogram cutting. The audio noise reduction is based on the

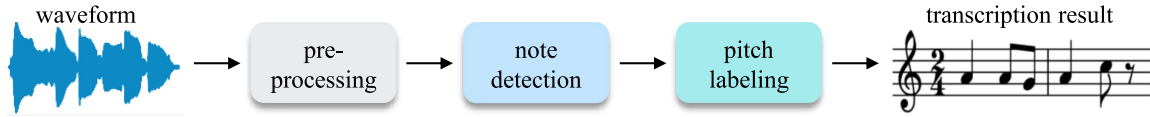


Fig. 2. The MusicYOLO framework.

Spleeter [64] to separate the interference of background music and metronome to make the human voice clearer. The time-frequency transformation module uses the CQT to convert the singing audio into a 2D spectrogram. Linear intensity mapping maps the spectrogram matrix into RGB images according to the value of matrix elements. Owing to the long time dimension of the spectrogram, a spectrogram cutting module is used to cut the spectrogram into square blocks with similar aspect ratio along the time axis.

1) *Audio Noise Reduction*: In sight-singing, practitioners often use metronomes or background music to assist in practice, making the task of singing transcription more difficult. Traditional filtering algorithms are powerless against these non-stationary noises. This paper uses the Spleeter [64], based on deep learning to eliminate background and metronome sounds.

The Spleeter is a method of source separation based on frequency domain. Each source track corresponds to a U-Net network [65] structure in the Spleeter. The input of each U-Net network is the spectrogram amplitude of the mix audio, and the output is the spectrogram amplitude of each source. The loss function calculates the L1 distance between the predicted amplitude spectrogram and the standard amplitude spectrogram. During prediction, the Spleeter predicts the mask of each source. Then, each source's output spectrogram is obtained by multiplying the input spectrogram by the corresponding mask.

We use the Spleeter model of two sound sources. Human voice is separated from one sound source. The sound of an instrument accompaniment or a metronome is separated from another source. This way, we realize the separation of human voice and noise. The experimental results show that in the case of background noise, the F1-score of note onset detection using the Spleeter is 1% higher than that without noise reduction.

2) *Time Frequency Transformation*: The 1D audio sequence is often transformed into a 2D spectrogram. The standard transformation methods include short-time Fourier transform (STFT), Mel transform, and CQT [66]. The time and frequency resolution of STFT does not change with frequency. The music signal often meets a specific frequency harmonic relationship, so STFT is not suitable for music signal analysis. The frequency resolution of the Mel transform decreases, and the time resolution increases with the increase of frequency. The frequency distribution of CQT is similar to that of the music signal, which is suitable for music signal processing. However, owing to the time blurring problem [67] in the low frequency, CQT will cause some interference in the onset and offset detection.

This study observes the effects of the Mel transform and CQT on the MusicYOLO singing transcription system. The results of the experiment show that the onset/offset detection performance of the CQT is slightly better than that of the Mel transform.

Algorithm 1: Obtaining Segmentation Points.

```

1: initialize splits;
2: set the iteration number  $r = 1$  and the maximum
   iteration number  $M$ ;
3: repeat
4:    $last\_time = splits(-1)$ ;
5:    $ratio = R(end_i - last\_time)$ ;
6:   if  $ratio \geq max\_ratio$  then
7:      $splits = [splits, end_i]$ ;
8:   else if  $best\_ratio < ratio \leq max\_ratio$  then
9:      $ratio\_silence = R(start_{i+1} - last\_time)$ ;
10:     $ratio\_next = R(end_{i+1} - start_{i+1})$ ;
11:    if  $ratio\_silence > max\_ratio$  then
12:       $splits = C(splits, S_{end_i}, start_{i+1})$ ;
13:    else if  $ratio\_next > max\_ratio$  then
14:       $splits = [splits, start_{i+1}]$ ;
15:    else
16:       $splits = [splits, end_i]$ ;
17:    else
18:      A process similar to the above;
19:     $r \leftarrow r + 1$ ;
20: until  $r > M$ .

```

3) *Linear Intensity Mapping*: A linear intensity mapping converts a single-channel spectrogram into a three-channel RGB image. A linear intensity mapping effectively quantifies the original spectrogram according to the spectral intensity. The strong spectrogram values are more prominent in one channel, and the weak ones are more prominent in other channels. It is shown that the features with linear intensity mapping are better than those with single-channel input [68].

4) *Spectrogram Cutting*: The singing audio is converted into a 2D matrix after CQT. Then, the spectrogram matrix is converted into a spectrogram image using a linear intensity mapping. As the aspect ratio of the spectrogram image is too large to conveniently process, we use an automatic slice generation algorithm to obtain approximately square slices along the spectrogram time axis. We use $ratio$ to describe the slice shape, and its calculation formula is as follows:

$$ratio = w/h - 1 \quad (1)$$

where w and h represent the width and height of slice, respectively, and their calculation formulas are as follows:

$$\begin{cases} w = frame_len \cdot scale \\ h = n_bins \cdot scale \end{cases} \quad (2)$$

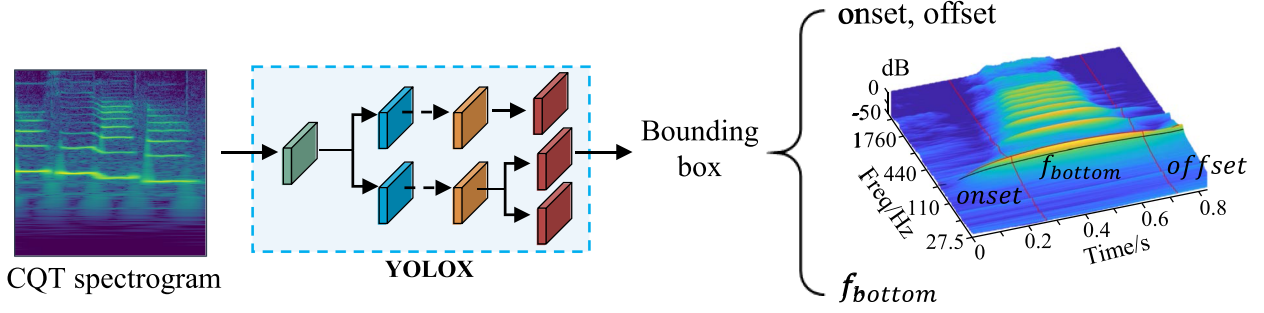


Fig. 3. Note object detection.

where $frame_len$ is the frame length of the CQT matrix, n_bins is the frequency bins of CQT, and $scale$ is the scaling factor from the matrix size to image size. $frame_len$ is calculated as follows:

$$frame_len = t \cdot sr / hop_len \quad (3)$$

where sr and hop_len are the CQT parameters, and t is the time length corresponding to the slice. From the above, we derive the $ratio$ calculation function written as R :

$$R(t) = t \cdot \frac{sr}{hop_len \cdot n_bins} \quad (4)$$

Our automatic slice generation algorithm is as follows:

- Use the `librosa.effects.split` function [69] to split audio into unmute segments S , where $S_i = (start_i, end_i)$ represents the start time and end time of the i th unmute segment, $i=1, 2, 3, \dots, M$.
- Obtain segmentation points $splits$ of the spectrogram image. We divide the types of S_i into three categories according to the $ratio$, which represents the appropriate segment time, longer or shorter time, or extremely long or extremely short time, respectively:

$$\begin{cases} |ratio| \leq best_ratio \\ best_ratio < |ratio| \leq max_ratio \\ |ratio| > max_ratio \end{cases} \quad (5)$$

For long silent segments, we specially design the silent segment cutting function C , which cuts the silent segment into squares. The details for obtaining segmentation points for different segment types is as shown in Algorithm 1.

- The spectrogram image is segmented into slices using the previously obtained segmentation points $splits$.

The parameters of the constant Q transform in the above algorithm are as follows. The lowest frequency f_{min} is set to 27.5 Hz. The sampling rate sr is set to 44100 Hz. The frame shift hop_len is set to 512. The octave parameters of $bins_per_octave$ and n_bins are set to 24 and 178, separately. `librosa.effects.split` function parameters are as follows. The top_db is set to 20. The $frame_len$ is set to 1024. The hop_len is set to 512. The ratio values of $best_ratio$ and max_ratio are set to 0.2 and 0.65, separately.

B. Note Detection

Note detection obtains the onset and offset from audio. This module includes the note object detection, bounding box post-processing, and time shift. The previous note detection models usually use CNNs to detect the onset and offset separately. However, the onset, offset, and f_{bottom} are obtained together from the note object bounding box in this paper.

1) *Note Object Detection*: The essential idea of note object detection is to transform the onset/offset detection problem into an object detection problem. As shown in Fig. 3, we regard the note spectrogram feature as a note object. Then, we use the object detection model to obtain the note bounding box. From the left, right, and bottom boundaries, we can easily derive the onset, offset, and f_{bottom} , respectively. The f_{bottom} will assist in the determination of the subsequent fundamental frequency, f_{pitch} .

We use the small version of YOLOX [70] as our note object detection model. Its parameter is 9 M and computational complexity per second of audio is 8.9Gflops. YOLOX uses Darknet53 [71] as the backbone of feature extraction and then uses two detection heads for classification and regression tasks. Compared with the previous YOLO series models, YOLOX has three advantages.

- The decoupling detection heads make the model converge faster and perform better.
- The use of anchor free for training eliminates the process of clustering to obtain prior bounding boxes from the dataset, which reduces the overfitting potential.
- SimOTA enables the model to automatically analyze the number of positive samples corresponding to each ground truth.

Mainstream object detection models are the two-stage models represented by Faster RCNN [72] and the one-stage model represented by YOLO [73]. Therefore, we compare the note object detection performance of YOLOX with that of the Faster RCNN. The experiment results in Table I from the later section show that the YOLOX model performs better.

2) *Post-Processing the Bounding Box*: Owing to differences between object detection and sight-singing onset/offset detection, we perform post-processing of the bounding box of YOLOX according to sight-singing characteristics.

- We remove the box within another box. In object detection, the box-in-box phenomenon means that small objects are

TABLE I
NOTE DETECTION RESULTS OF DIFFERENT MODEL CONFIGURATIONS

		CQT	MEL	YOLOX	Faster RCNN	SSVD			ISMIR2014			Mixed		
						P	R	F	P	R	F	P	R	F
Onset	M1	✓		✓		95.03	95.65	95.32	92.40	89.39	90.71	93.23	92.00	92.50
	M2		✓	✓		95.25	95.28	95.24	89.85	86.46	89.85	94.37	90.45	92.21
	M3	✓			✓	92.33	94.79	93.45	89.53	85.20	87.00	91.44	89.86	90.43
	M4		✓		✓	92.06	94.57	93.19	89.87	84.60	86.85	91.15	89.39	90.03
Offset	M1	✓		✓		97.49	98.13	97.78	87.10	84.53	85.65	92.34	91.32	91.73
	M2		✓	✓		97.72	97.75	97.71	88.87	81.88	85.04	93.66	89.96	91.62
	M3	✓			✓	94.85	97.41	96.02	85.53	81.65	83.26	90.97	89.58	90.07
	M4		✓		✓	94.64	97.25	95.81	85.72	80.93	82.97	90.81	89.25	89.79

included in large objects. There is only one note object in each note range in the sight-singing spectrogram because the human voice is monophonic. Therefore, if box A, whose upper left coordinate is (x_{a1}, y_{a1}) and lower right coordinate is (x_{a2}, y_{a2}) , is contained by box B, whose upper left coordinate is (x_{b1}, y_{b1}) and lower right coordinate is (x_{b2}, y_{b2}) , then we remove box A, and box B and put in a larger box C, whose upper left coordinate is $\min(x_{a1}, x_{b1}), \min(y_{a1}, y_{b1})$ and lower right coordinate is $\max(x_{a2}, x_{b2}), \max(y_{a2}, y_{b2})$, as the detection result.

- As a characteristic of human voice is that only after one sound ends can the next sound begin, two adjacent bounding boxes should not intersect. If two adjacent boxes (A, B) intersect, where box A is in front of box B, then we use the right boundary of box A as the left boundary of box B.

Then, the onset, offset, and f_{bottom} are obtained according to the corresponding relationship between the image size and time duration. The onset corresponds to the left boundary of the detection box, offset corresponds to the right boundary, and f_{bottom} corresponds to the bottom boundary.

3) *Time Shift*: We shift the onset and offset to obtain the complete audio's onset and offset values. As we have previously cut the spectrogram, we splice all image slices along the time axis and add the corresponding time shift to the previously detected onset and offset.

C. Pitch Labeling

In audio signal processing, pitch refers to the perception of human psychology on the fundamental frequency of different notes. The fundamental frequency is often used to represent pitch. For counting convenience, the MIDI number (p) is often used to represent the fundamental frequency (f). The conversion relationship between p and f is as follows:

$$p = 69 + 12 \times \log_2 \left(\frac{f}{440} \right) \quad (6)$$

In object detection, f_{bottom} has some errors from the precise pitch, f_{pitch} . There are two main causes of this error. First, the frequency resolution is sacrificed for a higher time resolution for more accurate onset and offset in note object detection. Hence, the frequency resolution is reduced, making the f_{bottom} not close enough to f_{pitch} . Second, the vocal signal is non-stationary,

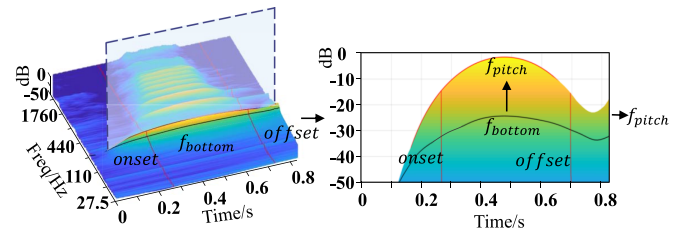


Fig. 4. Pitch labeling process.

and the vibrato and portamento in the singing process will lead to pitch fluctuations in the spectrogram. The error between the f_{bottom} and the f_{pitch} will be greater in these cases. Therefore, we propose an additional peak search algorithm to solve this problem.

A novel spectrogram peak search method is proposed to realize pitch labeling. The process of the peak search is shown in Fig. 4. Based on the onset, offset, and f_{bottom} of the note object detection, the peak point of the spectrogram matrix is searched to obtain the accurate pitch, f_{pitch} . The specific steps are as described in Algorithm 2.

IV. EXPERIMENT ENVIRONMENT

A. Dataset

As there is no dataset for sight-singing transcription, we have established the SSVD. Additionally, current singing transcription datasets are also used to evaluate our MusicYOLO framework. At present, the methods of monophonic instrument transcription are not so different from singing transcription. Therefore, a monophonic instrument dataset is also used to evaluate the generalization ability of different models. The following is a detailed description of each class of datasets.

1) *Sight-Singing Dataset*: The SSVD is recorded from the Sight-singing Talent WeChat applet,² providing practitioners with free sight-singing practice services. The SSVD dataset contains audio of children and adults performing sight-singing exercises. Among the audio samples, 67 audio samples are used as training and validation sets, and 127 audio samples are used

²Sight-singing Talent is an online WeChat applet that has more than 600 long-term users and more than 60,000 effective Sight-singing samples.

Algorithm 2: Pitch Labeling.

Require: Cqt : spectrogram matrix; $onset_i, offset_i$: onset and offset of the i th detected note; f_{bottom_i} : bottom frequency of the i th detected note; bpo : the number of frequency bins per octave; Δb : the search range of frequency bins.

Ensure: $Pitch$: a pitch list of the detected notes.

```

1: set  $N$  as the number of the notes
2: foreach  $i \in \{1, 2, \dots, N\}$  do
3:    $f_{init} = f_{bottom_i}$ 
4:    $\Delta f1 = f_{init} \times (1 - 2^{-\Delta b/bpo})$ 
5:    $\Delta f2 = f_{init} \times (2^{\Delta b/bpo} - 1)$ 
6:    $f_{low} = f_{init} - \Delta f1$ 
7:    $f_{high} = f_{init} + \Delta f2$ 
8:    $f_0 = 0$ 
9:   foreach  $t \in \{onset_i, \dots, offset_i\}$  do
10:     $f_0 = f_0 + f_{low} + \text{argmax}(Cqt[f_{low} : f_{high}, t])$ 
11:     $f_{pitch} = f_0 / (offset_i - onset_i)$ 
12:     $Pitch_i = 69 + 12 \times \log(f_{pitch}/440)$ 

```

as test sets. The audio sampling rate is 44100 Hz, and the sampling resolution is 16 bits. It contains 8549 notes in total, with a cumulative time of 114 minutes. Each audio file includes a note level annotation of the onset, offset, pitch. The annotation method is as follows:

- Four researchers who had received professional sight-singing training gave the approximate onset time of each note through slowing down and listening to the singing audio under the guidance of music score;
- The researchers used the audition software to observe the spectrogram with a high time resolution set. Then, they watched the spectrogram characteristics near the approximate onset time given in the last step and took the occurrence time of the second harmonic signal as the accurate onset time;
- After obtaining the onset annotation of each note, the offset annotation was given by listening to the audio. The specific rules were as follows: the time when the sound could not be heard was regarded as the approximate offset. Combined with the spectrogram, the accurate offset time was defined when most harmonic signals disappeared;
- After making the audition to be more logarithmic and higher frequency resolution, the four researchers took the center frequency of each note as the fundamental frequency and converted the frequency into the MIDI number as the pitch annotation;
- Finally, the four researchers checked each other’s annotation files until no annotation errors were found.

2) *Singing Dataset*: The singing transcription datasets include the ISMIR2014 dataset [74] and MIR-ST500 dataset [1]. The ISMIR2014 dataset contains 38 melodies sung by untrained adults and children, with a 16 KHz sampling rate. The length of each audio is from 15 s to 86 s, and the total length is 1154 s. The MIR-ST500 contains 500 popular songs, which are from

YouTube. These songs contain background music, with a total of 30 hours and 162438 notes.

3) *Monophonic Instrument Dataset*: Bach10 dataset [75] is a polyphonic dataset, including ensembles of bassoon, clarinet, saxophone, and violin. There are ten tracks in total, and each track is about 30 s. The Bach10 dataset also contains 40 monophonic audio files and note-level annotations played independently by each instrument. We use the monophonic audio subset to test different models’ generalization ability.

B. Experiment Configuration

At present, the method proposed in [1] in singing transcription works well on the MIR-ST500 dataset. Therefore, we use the MIR-ST500 training set and SSVD training set to train Wang’s model and our note detection model, respectively. We use the ISMR2014 dataset, MIR-ST500 test set, SSVD test set, and the Bach10 dataset to test different models.

C. Evaluation Metrics

We use the evaluation metrics proposed in [1], including *Onset* (Correct Onset), *Offset* (Correct Offset), *Note* (Correct Onset and Pitch), and *Note w/ offset* (Correct Onset, Pitch and Offset), to evaluate different AST models. If the onset/offset/pitch difference between two notes is within a certain threshold, the onset/offset/pitch is regarded as “correct.” By comparing ground truth and predicted notes, we can compute the F1-score of each metric. In our experiments, the onset threshold is set to 50 ms, the pitch threshold is set to 50 cents, and the offset threshold is set to the maximum of (50 ms, $0.2 \times \text{note_duration}$). *Onset*, *Offset*, *Note*, and *Note w/ offset* are calculated using precision, recall, and F1 values by using the *mir_eval* library [76]. The calculation formulas are as follows:

$$P = \frac{TP}{TP + FP} \quad (7)$$

$$R = \frac{TP}{TP + FN} \quad (8)$$

$$F1 = \frac{2 \times P \times R}{P + R} \quad (9)$$

where TP represents the number of correctly predicted notes, FP represents the number of extra detected notes, and FN represents the number of missed notes.

V. EXPERIMENT

In this chapter, we conduct three experiments. In the first experiment, the effects of different model configurations on note detection are studied. In the second experiment, the performance of our note detection method is compared with that of previous models [1], [3], [77], and the performance of different models is analyzed. In the third experiment, the performance of our MusicYOLO framework is compared with that of other models, and the advantages and disadvantages of our MusicYOLO framework are also discussed.

TABLE II
NOTE DETECTION RESULTS OF DIFFERENT MODELS

		ISMIR2014			MST500			SSVD			Bach10		
		P	R	F	P	R	F	P	R	F	P	R	F
Onset	Tony	72.13	64.19	67.63	50.98	56.88	53.45	79.00	78.96	78.90	91.94	88.55	90.15
	Fu&Su	83.00	75.43	78.76	50.78	53.97	52.10	79.70	81.04	80.11	87.17	79.41	83.00
	Wang I	81.09	82.49	81.60	73.50	78.59	75.67	76.88	82.44	79.31	87.38	88.65	87.61
	MusicYOLO I	89.55	82.76	85.88	81.89	75.02	78.17	88.02	86.88	87.41	92.56	84.68	88.31
	Wang II	88.38	40.15	53.90	65.07	29.70	40.34	91.36	65.79	75.70	84.74	24.48	37.21
	MusicYOLO II	91.01	89.24	90.01	62.53	65.20	63.59	95.99	96.38	96.17	93.97	94.19	94.07
Offset	Tony	79.88	70.37	74.47	55.82	62.44	58.59	93.59	93.50	93.43	95.71	92.16	93.84
	Fu&Su	79.60	72.94	75.87	53.26	56.82	54.74	84.48	86.09	84.99	92.39	84.07	87.91
	Wang I	76.39	77.86	76.95	65.25	69.89	67.25	71.86	77.78	74.43	87.00	88.70	87.43
	MusicYOLO I	83.43	77.07	79.99	78.78	72.01	75.10	85.14	84.21	84.64	97.24	88.80	92.68
	Wang II	21.39	9.12	12.46	26.50	12.05	16.38	13.84	10.00	11.45	42.47	12.33	18.80
	MusicYOLO II	85.79	84.34	84.96	62.89	65.57	63.94	97.89	98.28	98.07	96.80	97.06	96.91

A. Model Configurations

To test the effect of different configurations (i.e., CQT and Mel transform, YOLOX and Faster RCNN object detection models), the SSVD training set is used to train our note detection model. The SSVD test set, ISMIR2014 dataset, and the Mix dataset (all ISMIR2014 data and 38 tracks of the SSVD test set) are used to test different configurations. The experimental results of onset detection and offset detection are shown in Table I.

As shown in Table I, the onset F1-score of the M1 configuration is 0.29% higher than that of the M2 configuration owing to the use of YOLOX model. The offset F1-score of the M1 configuration is only 0.11% higher than that of the M2 configuration. These indicate that the Mel transform and CQT transform have little difference in the impact on onset/offset detection. Considering the applicability of CQT to music signals, we finally choose CQT as the time-frequency transform configuration. On the SSVD, ISMIR2014, and Mixed datasets, the M1 configuration with the YOLOX model has better performance than that of the M3 configuration with the Faster RCNN model. The onset F1-scores of the M1 configuration are 1.87%, 3.71%, and 2.07% higher than that of the M3 configuration. The offset F1-scores in the M1 configuration are 1.76%, 2.39%, and 1.66% higher than that of the M3 configuration. Therefore, we use YOLOX as our object detection model.

B. Note Detection Results

We evaluate the onset/offset detection performance of different models. Wang proposed a new singing transcription dataset, MIR-ST500. We regard the MusicYOLO trained on the MIR-ST500 training subset and SSVD training subset as MusicYOLO I and MusicYOLO II, respectively. Wang's models trained on MIR-ST500 and SSVD are called Wang I and Wang II, respectively. The note detection results of different models are shown in Table II.

1) *Experiment Results*: The Table II shows that the MusicYOLO II achieves the best onset and offset detection performance on the ISMIR2014 dataset, SSVD test set, and Bach10 dataset. However, its performance on the MIR-ST500 test set is not

as good as that on Wang I. This is because the training set of MusicYOLO I, the SSVD training set, is a sight-singing dataset, whose spectral characteristics are not as complex as those of the MIR-ST500 dataset, so it cannot be well adapted. We retrain MusicYOLO on the MIR-ST500 training set to obtain the MusicYOLO II model. The experiment results show that MusicYOLO II achieves a 78.17% onset F1-score and a 75.10% offset F1-score on the MIR-ST500 test set, which is better than Wang I. This is because MusicYOLO detects note objects from a macro point of view, so it can be rarely disturbed by the spectral characteristics of singing details. These results show that MusicYOLO's approach for note object detection is more effective than the traditional frame-based note detection.

The frame-based deep learning methods, like those proposed in [77] and [1], are not stable on different datasets, while MusicYOLO adapts well to all datasets. Both of those works, based on the CNN, have obtained satisfactory detection performance on ISMIR2014, MST500, and SSVD datasets. However, these two models cannot reach the performance of the Tony software [3] based on signal processing on the Bach10 dataset, which shows that the two models overfit the human vocal dataset and cannot generalize well to the common monophonic note detection problem. In contrast, MusicYOLO achieves a 94.07% onset F1-score and a 96.91% offset F1-score on the Bach10 dataset, which greatly outperforms other methods.

2) *Error Analysis*: We find that portamento, legato, volume changes, and pitch jitter in singing produce many errors in different note detection algorithms. We classify the detection errors into four categories. Fig. 5(a) shows that the second note is missed in two consecutive notes, called successive missing (SM). The ones directly missed by the note detection model are called isolated missing (IM), as shown in Fig. 5(b). Fig. 5(c) shows the extra detection errors in the middle of long notes, called successive extra detection (SE). Extra detection errors caused by pitch fluctuations are called isolated extra detection (IE), as shown in Fig. 5(d).

We make the error distribution statistics of different methods on ISMIR2014, MIR-ST500, SSVD, and Bach10 datasets. The results are shown in Fig. 6. As shown in Fig. 6, other methods

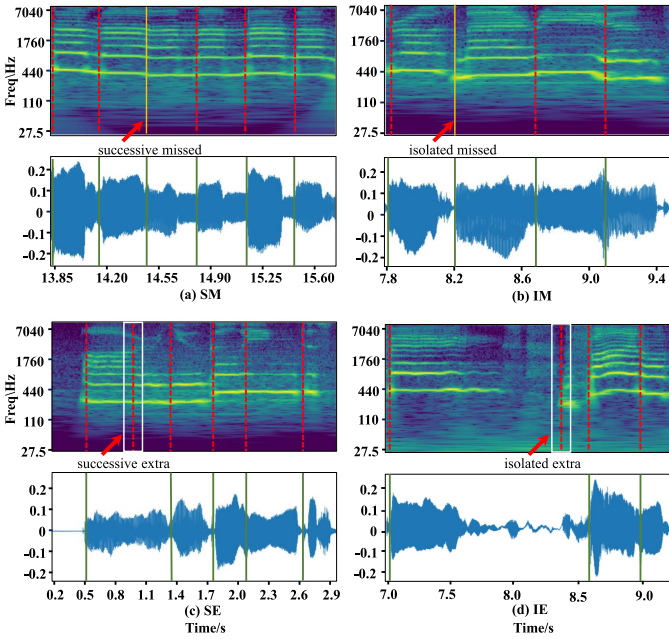


Fig. 5. Problems of frame-based transcription methods. The red dotted lines represent the predictions. The green solid lines represent the annotations. (a) Continuous missed detection (SM) (b) Isolated missed detection (IM) (c) Continuous extra detection (SE) (d) Isolated extra detection (IE).

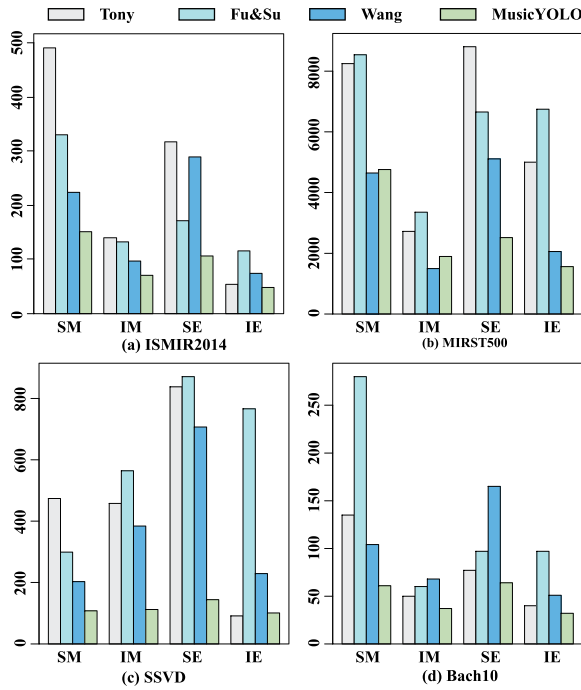


Fig. 6. Distribution of note detection errors on the four datasets.

generally perform well on one dataset, but not so well on the other three datasets. MusicYOLO, proposed in this paper, performs well on all four datasets, indicating that its generalization ability is also good.

As shown in Fig. 6, SM and SE are the main problems. These two errors decrease the precision and recall rate, resulting

in a poor note detection F1-score. Compared with the other three methods, MusicYOLO has few detection errors, with relatively balanced error distribution and no significant difference on different datasets, indicating MusicYOLO's excellent note detection performance and generalization ability.

As can be seen from Fig. 6, Tony performs poorly on the ISMIR2014 dataset and MIR-ST500 dataset, but it performs well on the SSVd dataset and Bach10 dataset. Tony is based on signal processing methods without a training process for different data. Therefore, it can perform well on simple sight-singing data and monophonic instrument data, while it can not complete the note segmentation task well for complex patterns in pop singing data. Fu&Su is optimized for flamenco data, so it performs poorly on all the four datasets. Furthermore, the performance of Fu&Su is unstable on different datasets, which is a common problem after overfitting a specific dataset.

Wang uses a lot of pop music singing as a training set and is able to adapt to a wide range of singing styles, so Wang's performance is better than the first two methods. However, the average note duration in the SSVd dataset is longer than that in pop singing, and Wang's method is based on frame-level spectral features, so it is easily disturbed by local spectrogram features. Therefore, Wang has many successive extra detections.

MusicYOLO has more SM and IM on the MIR-ST500 dataset. The MST dataset is composed of popular Chinese songs, so there are many continuous singing notes and the separation of adjacent notes is not obvious on this dataset. This causes MusicYOLO to detect some successive notes as one note, resulting in SM. Moreover, the singing pronunciation is affected by the lyrics, and it is difficult to detect the unvoiced sound, which leads to IM. In general, the MusicYOLO II model trained on the MIR-ST500 training set outperforms the other four methods.

The visualization note detection results of the four models are shown in Fig. 7. From top to bottom are the detection results of Tony, Fu&Su, Wang, MusicYOLO. The bottom is the label. In Fig. 7(a), Tony, Fu&Su have two continuous missed note detection errors. Wang has one continuous missed detection error. This will reduce the recall of their models. In Fig. 7(b), Tony, Fu&Su and Wang all have one continuous extra error if offset is not considered. If offset is considered, they will have two continuous extra errors. This will reduce the precision of their models. In Fig. 7(c), Tony, Fu&Su and Wang all have one predicted onset shift. This results in one missed detection error and one extra detection error, resulting in both precision and recall degradation. In Fig. 7(d), Tony, Fu&Su and Wang all have one predicted offset shift. If offset is considered, the offset shift note will be regarded as wrong detection, resulting in one missed detection error and one extra detection error. The precision and recall of their models will decrease.

C. Transcription Results

We compare the transcription results of different models. Table III shows the results under the *Note* and *Note w/ offset* metrics. Wang II note detection performance is not good and negatively influences transcription results. Its final transcription results are not shown in Table III.

TABLE III
TRANSCRIPTION RESULTS OF DIFFERENT MODELS

	ISMIR2014			MST500			SSVD			Bach10			
	P	R	F	P	R	F	P	R	F	P	R	F	
Note	Tony	65.69	58.60	61.70	38.92	43.29	40.78	71.54	71.44	71.41	91.07	87.74	89.31
	Fu&Su	77.81	70.62	73.80	36.46	38.55	37.33	63.08	64.07	63.37	83.93	76.48	79.92
	Wang I	66.79	68.15	67.33	66.09	70.88	68.18	58.24	62.25	60.00	82.12	83.65	82.55
	MusicYOLO I	85.24	78.84	81.78	74.69	68.61	71.42	79.19	78.15	78.64	89.27	81.72	85.20
	MusicYOLO II	84.00	82.44	83.12	45.16	47.08	45.94	85.77	86.10	85.92	91.77	91.99	91.86
Note w/ offset	Tony	49.99	44.87	47.10	24.98	27.98	26.27	67.12	67.16	67.07	84.65	81.77	83.13
	Fu&Su	62.52	56.87	59.40	22.69	24.03	23.25	56.96	57.60	57.17	77.79	70.96	74.12
	Wang I	53.09	54.09	53.49	46.61	50.04	48.12	44.36	47.24	45.64	73.77	75.04	74.19
	MusicYOLO I	72.21	66.90	69.34	61.22	56.27	58.56	68.62	67.77	68.17	87.28	79.91	83.30
	MusicYOLO II	72.18	70.96	71.49	31.11	32.54	31.70	84.45	84.78	84.60	89.28	89.49	89.37

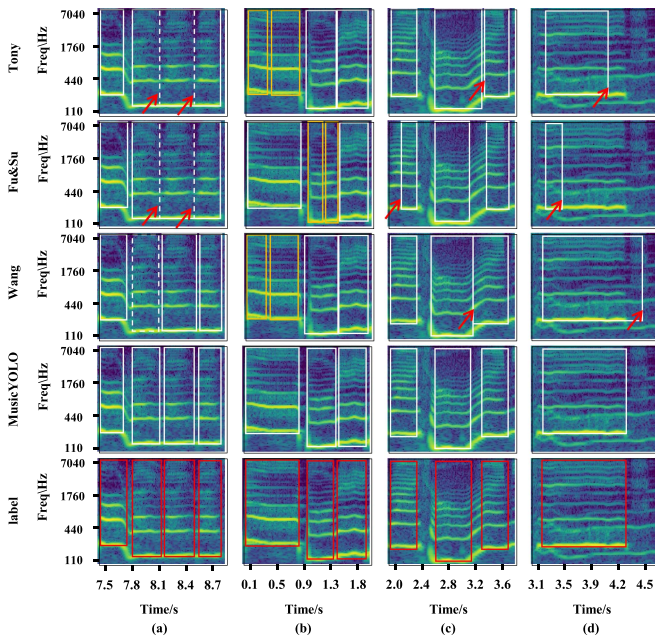


Fig. 7. The detection results of different methods. (a) Continuous missed detection. The dashline indicates specific missed notes. (b) Continuous extra detection. The yellow color indicates specific extra notes. (c) The onset shift. The arrow indicates the deviation of the predicted onsets. (d) The offset shift. The arrow indicates the deviation of the predicted offsets.

1) *Experiment Results*: As can be seen from Table III, the MusciYOLO framework still achieves the best results across all datasets. MusicYOLO obtains an 85.92% *Note* F1-score and an 84.60% *Note w/ offset* F1-score on the sight-singing dataset, SSVD, making it a state-of-the-art transcription method at present. Furthermore, MusicYOLO outperforms other methods on the above four datasets, indicating that MusicYOLO can grasp the essential characteristics of notes from a macro perspective and is effective for the general monophonic audio transcription.

By comparing *Note* and *Note w/ offset*, we find that the *Note w/ offset* of different methods on the ISMIR2014 and MIR-ST500 dataset decreases by about 12% compared with *Note*, which shows that pitch fluctuations and pitch

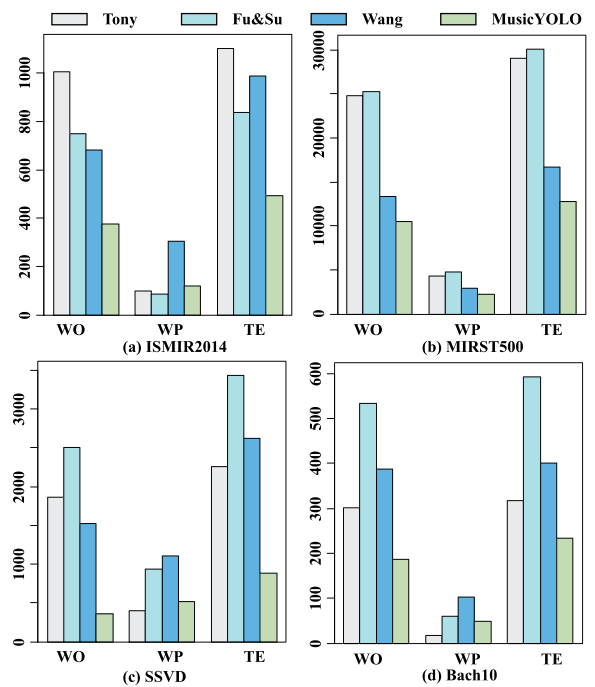


Fig. 8. Error distributions of transcription error on four datasets.

changes are relatively complex. In comparison, offset detection in sight-singing is relatively easy. For example, Tony's *Note w/ offset* drops only 4% compared with *Note*, and Fu&Su's *Note w/ offset* drops only 6% compared with *Note*. Sight-singing is more staccato, so the singing skills are not as complicated as for pop songs. Thus, the end features of the sight-singing spectrogram are more evident, making detection of offsets less complicated than those in pop music.

2) *Error Analysis*: We divide transcription errors into two categories only considering the onset and pitch. One class of transcription errors is caused by onset errors, called the wrong onset (WO), which is the sum of SM, IM, SE, and IE, mentioned above. However, when the onset is correct, but the difference between the predicted pitch and annotated pitch exceeds the pitch threshold, this class of errors is called the wrong pitch (WP). We call the sum of these two transcription errors the total

error (TE). The distributions of errors across different datasets are shown in Fig. 8.

As shown in Fig. 8, onset detection errors are the main contributor to transcription errors on different datasets, while pitch errors are rarely found. This indicates that extracting pitch in singing transcription is easy, and onset/offset detection is the key to transcription performance. It is worth making an effort to minimize the influence of portamento, legato, volume change, and pitch fluctuation on onset/offset detection. The MusicYOLO proposed in this paper affords an interesting attempt to do so.

VI. CONCLUSION

This paper has proposed an AST framework based on object detection, called MusicYOLO. MusicYOLO first uses the YOLOX object detection model to obtain the onset, offset, and approximate fundamental frequency, f_{bottom} . Then, a simple peak search method is used to gain the specific pitch of each note. Compared with previous methods, the MusicYOLO detects note objects rather than isolated onset/offset moments, which greatly enhances the performance of onset/offset detection. As for AST, MusicYOLO has reached 84.60% of the F1-score under the *Note w/ offset* metric on the SSVD dataset, making it a state-of-the-art algorithm. Of course, the current MusicYOLO is not perfect. For example, it does not work well with accompaniment and multiple singers. These require continuous improvement.

Establishing a set of objective sight-singing evaluation metrics can make sight-singing practitioners more aware of the shortcomings. AST is the first step in establishing an automatic sight-singing evaluation system. The onset can evaluate the singing rhythm, the offset can evaluate the singing duration, and pitch facilitates intonation assessment. In the follow-up research, we will think about how to establish an objective evaluation system to assess the condition of the vocal area to give guidance to practitioners. That will enrich the content of our automatic sight-singing evaluation system and guide practitioners more effectively.

REFERENCES

- [1] J. Y. Wang and J. S. R. Jang, "On the preparation and validation of a large-scale dataset of singing transcription," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 276–280.
- [2] S. Chang and K. Lee, "A pairwise approach to simultaneous onset/offset detection for singing voice using correntropy," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 629–633.
- [3] M. Mauch et al., "Computer-aided melody note transcription using the Tony software: Accuracy and efficiency," in *Proc. Int. Conf. Technol. Music Notation Representation*, 2015, pp. 23–30.
- [4] M. Mauch and S. Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 659–663.
- [5] R. J. McNab, L. A. Smith, and I. H. Witten, "Signal processing for melody transcription," in *Proc. 19th Aust. Comput. Sci. Conf.*, 1997, pp. 301–307.
- [6] E. Molina, L. J. Tardón, A. M. Barbancho, and I. Barbancho, "SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 2, pp. 252–263, Feb. 2015.
- [7] A. D. Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Amer.*, vol. 111, no. 4, pp. 1917–1930, 2002.
- [8] N. Kroher and E. Gómez, "Automatic transcription of flamenco singing from polyphonic music recordings," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 5, pp. 901–913, May 2016.
- [9] H. Heo, D. Sung, and K. Lee, "Note onset detection based on harmonic cepstrum regularity," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2013, pp. 1–6.
- [10] A. Elowsson, "Modeling music: Studies of music transcription, music perception and music production," Ph.D. dissertation, KTH Royal Inst. Technol., Stockholm, Sweden, 2018.
- [11] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6979–6983.
- [12] D. Jouviet and Y. Laprie, "Performance analysis of several pitch detection algorithms on simulated and real noisy speech data," in *Proc. IEEE Eur. Signal Process. Conf.*, 2017, pp. 1614–1618.
- [13] S. Strömbergsson, "Today's most frequently used F0 estimation methods, and their accuracy in estimating male and female pitch in clean speech," in *Proc. Interspeech*, 2016, pp. 525–529.
- [14] O. Babacan, T. Drugman, N. d' Alessandro, N. Henrich, and T. Dutoit, "A comparative study of pitch extraction algorithms on a large variety of singing sounds," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7815–7819.
- [15] A. V. D. Knesebeck and U. Zölzer, "Comparison of pitch trackers for real-time guitar effects," in *Proc. 13th Int. Conf. Digit. Audio Effects*, 2010, pp. 525–529.
- [16] J. Dubnowski, R. Schafer, and L. Rabiner, "Real-time digital hardware pitch detector," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 1, pp. 2–8, Feb. 1976.
- [17] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," *Proc. Inst. Phonetic Sci.*, vol. 17, pp. 97–110, 1993.
- [18] D. Talkin and W. B. Kleijn, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.
- [19] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *J. Acoustical Soc. Amer.*, vol. 124, no. 3, pp. 1638–1652, 2008.
- [20] D. J. Hermes, "Measurement of pitch by subharmonic summation," *J. Acoustical Soc. Amer.*, vol. 83, no. 1, pp. 257–264, 1988.
- [21] Y. Ikemiyama, K. Yoshii, and K. Itoyama, "Singing voice analysis and editing based on mutually dependent F0 estimation and source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 574–578.
- [22] T. Ramabadran, A. Sorin, and M. McLaughlin, "The ETSI extended distributed speech recognition (DSR) standards: Server-side speech reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, pp. I-53.
- [23] H. Kawahara, A. Cheveigné, and H. Banno, "Nearly defect-free F0 trajectory extraction for expressive speech modifications based on straight," in *Proc. Interspeech*, 2005, p. 4.
- [24] M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio Electroacoustics*, vol. 16, no. 2, pp. 262–266, Jun. 1968.
- [25] K. Han and D. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 2158–2168, Dec. 2014.
- [26] B. Liu, J. Tao, and D. Zhang, "A novel pitch extraction based on jointly trained deep BLSTM recurrent neural networks with bottleneck features," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 336–340.
- [27] S. Gonzalez and M. Brookes, "PEFAC-A pitch estimation algorithm robust to high levels of noise," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 2, pp. 518–530, Feb. 2014.
- [28] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 161–165.
- [29] B. Gfeller, C. Frank, and D. Roblek, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1118–1128, 2020.
- [30] E. Gómez and J. Bonada, "Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing," *Comput. Music J.*, vol. 37, no. 2, pp. 73–90, Jun. 2013.
- [31] T. D. Mulder, J. P. Martens, and M. Lesaffre, "Recent improvements of an auditory model based front-end for the transcription of vocal queries," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, vol. 4, pp. 257–260.

- [32] Z. He and Y. Feng, "Singing transcription from multimedia audio signals using contour filtering," in *Proc. IEEE Int. Conf. Commun., Inf. Syst. Comput. Eng.*, 2021, pp. 114–117.
- [33] F. Rigaud and M. Radenen, "Singing voice melody transcription using deep neural networks," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 737–743.
- [34] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2013.
- [35] L. Yang, A. Maezawa, J. B. L. Smith, and E. Chew, "Probabilistic transcription of sung melody using a pitch dynamic model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 301–305.
- [36] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian singing transcription based on a hierarchical generative model of keys, musical notes, and f0 trajectories," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1678–1691, 2020.
- [37] R. Nishikimi, E. Nakamura, and M. Goto, "Audio-to-score singing transcription based on a CRNN-HSMM hybrid mode," *APSIPA Trans. Signal Inf. Process.*, vol. 10, 2021, Art. no. e7.
- [38] A. Temko, R. Malkin, and C. Zieger, "Clear evaluation of acoustic event detection and classification systems," in *Proc. Int. Eval. Workshop Classification Events, Activities Relationships*, 2006, pp. 311–322.
- [39] A. Mesaros et al., "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 2, pp. 379–393, Feb. 2018.
- [40] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 1, pp. 20–31, Jan. 2015.
- [41] J. Schröder, S. Goetze, and J. Anemüller, "Spectro-temporal Gabor filterbank features for acoustic event detection," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 23, no. 12, pp. 2198–2208, Dec. 2015.
- [42] J. Dennis, H. D. Tran, and E. S. Chng, "Image feature representation of the subband power distribution for robust sound event classification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 367–377, Feb. 2013.
- [43] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 3, pp. 540–552, Mar. 2015.
- [44] H. Phan, L. Hertel, and M. Maass, "Robust audio event recognition with 1-max pooling convolutional neural networks," in *Proc. Interspeech*, 2016, pp. 3653–3657.
- [45] J. Dennis, H. D. Tran, and E. S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 1085–1093, 2013.
- [46] E. Cakir, T. Heittola, and H. Huttunen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2015, pp. 1–7.
- [47] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 6440–6444.
- [48] H. D. Tran and H. Li, "Jump function kolmogorov for overlapping audio event classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 3696–3699.
- [49] H. Phan, M. Maass, R. Mazur, and A. Mertins, "Early event detection in audio streams," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2015, pp. 1–6.
- [50] P. Giannoulis, G. Potamianos, A. Katsamanis, and P. Maragos, "Multi-microphone fusion for detection of speech and acoustic events in smart spaces," in *Proc. Eur. Signal Process. Conf.*, 2014, pp. 2375–2379.
- [51] H. Phan, L. Hertel, M. Maass, R. Mazur, and A. Mertins, "Learning representations for non-speech audio events through their similarities to speech patterns," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 4, pp. 807–822, Apr. 2016.
- [52] L. Lu, F. Ge, and Q. Zhao, "A SVM-based audio event detection system," in *Proc. Int. Conf. Elect. Control Eng.*, 2010, pp. 292–295.
- [53] H. Phan et al., "What makes audio event detection harder than classification?," in *Proc. IEEE Eur. Signal Process. Conf.*, 2017, pp. 2739–2743.
- [54] R. Stiefelhagen, K. Bernardin, and R. Bowers, "The clear 2007 evaluation," in *Multimodal Technologies for Perception of Humans*. Berlin, Germany: Springer, 2007, pp. 3–34.
- [55] A. Temko, C. Nadeu, and J. I. Biel, "Acoustic event detection: SVM-based system and evaluation setup in clear'07," in *Multimodal Technologies for Perception of Humans*. Berlin, Germany: Springer, 2007, pp. 354–363.
- [56] A. Waibel, R. Stiefelhagen, and R. Carlson, "Computers in the human interaction loop," in *Handbook of Ambient Intelligence and Smart Environments*. Berlin, Germany: Springer, 2010, pp. 1071–1116.
- [57] H. Phan, M. Maass, and L. Hertel, "A multi-channel fusion framework for audio event detection," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2015, pp. 1–5.
- [58] K. Wang, L. Yang, and B. Yang, "Audio event detection and classification using extended R-FCN approach," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2017, pp. 128–132.
- [59] C. C. Kao, W. Wang, and M. Sun, "R-CRNN: Region-based convolutional recurrent neural network for audio event detection," in *Proc. Interspeech*, 2018, pp. 1358–1362.
- [60] A. Mesaros, T. Heittola, and A. Diment, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2017, pp. 85–92.
- [61] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 1038–1047.
- [62] T. W. Su, J. Y. Liu, and Y. H. Yang, "Weakly-supervised audio event detection using event-specific Gaussian filters and fully convolutional networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 791–795.
- [63] C. Gan, D. Huang, H. Zhao, J. B. Tenenbaum, and A. Torralba, "Music gesture for visual sound separation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10475–10484.
- [64] R. Hennequin, A. Khelif, F. Voituret, and M. Moussallam, "Spleeter: A fast and efficient music source separation tool with pre-trained models," *J. Open Source Softw.*, vol. 5, no. 50, 2020, Art. no. 2154.
- [65] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
- [66] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. Proc. Sound Music Comput. Conf.*, 2010, pp. 3–64.
- [67] J. Nam, G. J. Mysore, and J. Ganseman, "A super-resolution spectrogram using coupled PLCA," in *Proc. Interspeech*, 2010, pp. 1696–1699.
- [68] J. Dennis, H. D. Tran, and H. Li, "Spectrogram image feature for sound event classification in mismatched conditions," *IEEE Signal Process. Lett.*, vol. 18, no. 2, pp. 130–133, Feb. 2011.
- [69] B. McFee, C. Raffel, and D. Liang, "librosa: Audio and music signal analysis in python," in *Proc. 14th Python Sci. Conf.*, 2015, pp. 18–25.
- [70] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," 2021, *arXiv:2107.08430*.
- [71] J. Redmon, "Darknet: Open source neural networks in c," 2016. [Online]. Available: <http://pjreddie.com/darknet/>
- [72] S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, vol. 28, pp. 91–99.
- [73] J. Redmon, S. Divvala, and R. Girshick, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
- [74] E. Molina, A. M. Barbancho, and L. J. Tardon, "Evaluation framework for automatic singing transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 567–572.
- [75] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2121–2133, Nov. 2010.
- [76] C. Raffel, B. McFee, and E. J. Humphrey, "mir_eval: A transparent implementation of common MIR metrics," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 367–372.
- [77] Z. S. Fu and L. Su, "Hierarchical classification networks for singing voice segmentation and transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2019, pp. 900–907.



Xianke Wang received the B.E. degree in electromagnetic field and wireless technology, in 2020 from the Huazhong University of Science and Technology, Wuhan, China, where he is currently working toward the M.S. degree with the School of Electronic Information and Communications. His research interests include music information retrieval, speech recognition, signal processing, and machine learning.



Bowen Tian received the B.E. degree in electronic and information engineering, in 2020 from the Huazhong University of Science and Technology, Wuhan, China, where he is currently working toward the M.S. degree with the School of Electronic Information and Communications. His research interests include computer vision, machine learning, and human-computer interactions.



Wei Xu (Member, IEEE) received the Ph.D. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008. He is currently an Associate Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include machine learning, automatic singing/piano transcription and evaluation.



Weiming Yang received the B.E. degree in electronic and information engineering, in 2020 from the Huazhong University of Science and Technology, Wuhan, China, where she is currently working toward the M.S. degree with the School of Electronic Information and Communications. Her research interests include machine learning, signal processing, and automatic music transcription.



Wenqing Cheng received the B.E. degree in telecommunication engineering and the Ph.D. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1985 and 2005, respectively. She is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. Her research interests include mobile communications and wireless sensor networks, information systems, and e-learning applications.