This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

A Multi-stage Automatic Evaluation System for Sight-singing

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

***Abstract*—Sight-singing exercises are a fundamental part of music education. In this paper, we present an objective and complete automatic evaluation system for sight-singing, which has two critical stages: note transcription and note alignment. In the first stage, we use an onset detector based on the convolutional recurrent neural network (CRNN) for note segmentation and the pitch extractor described in [1] for note labeling. In the second stage, an alignment algorithm based on relative pitch modeling is proposed. Due to the lack of datasets for sight-singing note alignment and the overall system evaluation, we construct the sight-singing vocal dataset (SSVD). Each module of the system and the entire system are tested on this dataset. The onset detector achieves an F-measure of 90.61%, and the stages of note transcription and note alignment achieve an F-measure of 88.42% and 94.79%, respectively. In addition, we propose an objective criterion for the sight-singing evaluation system. Based on this criterion, our automatic sight-singing system achieves an F-measure of 77.95% on the SSVD dataset.**

***Index Terms*—Automatic sight-singing system, sight-singing transcription, note alignment, systematic evaluation measure.**

I. Introduction

THE practice of sight-singing refers to the process by which music students learn and improve their ability of musical reading through the repeated singing of musical notes from a musical score (usually given a reference note). In fact, beginning music students are often required to practice sight-singing to build their musical perceptions of each note. Sight-singing is considered a prerequisite for musical performances and effectively learning musical knowledge [2]. During sight-singing exercises, it is important to have constant feedback from a music expert who can detect every mistake made by the singer and pinpoint the best way to correct these mistakes. Traditionally, sight-singing evaluation is conducted one-to-one between an expert teacher and a student. However, music teachers' discernment may be affected by subjective factors and fatigue. When students practice sight-singing outside of the class, it is not easy to receive specific guidance and advice from teachers. Therefore, the construction of an objective and reliable automatic sight-singing evaluation system can be used to effectively solve this problem.

Usually, the sung notes need to be compared with the imitated notes from sheet music in sight-singing evaluation.

Manuscript received July 15, 2021; revised September 26, 2021, and February 17, 2022, and March 23, 2022; accepted April 9, 2022. This work is supported by the National Natural Science Foundation of China (No. 61877060). (Corresponding author: Wei Xu.)

The authors are with the Hubei Key Laboratory of Smart Internet Technology, and also with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China.(e-mail: yyweiming@hust.edu.cn; M202072113@hust.edu.cn; M202072111@hust.edu.cn; xuwei@hust.edu.cn; chengwq@mail.hust.edu.cn).

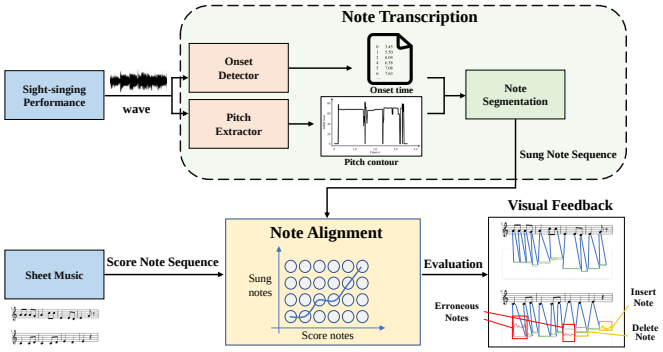


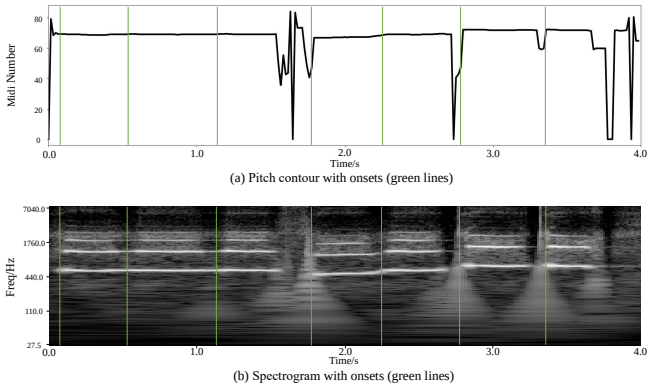
Fig. 1: The scheme of the proposed method for automatic sight-singing evaluation system.

Each of the sung notes should be evaluated correctly. Currently, the studies related to sight-singing evaluation mainly focus on the field of singing evaluation. Most of the existing singing evaluation systems [3][4][5][6] were adopted to score or classify the singers' performance in an overall manner. As a result, low-level features (e.g., frame-wise pitch contour) are first extracted from singing audio and reference audio/score. Then, the similarity of the features is focused on to make a global evaluation. In contrast, in sight-singing evaluation, the high-level note features need to be detected from musical audio signals to provide feedback at the note level. Due to the different requirements for singing evaluation and sight-singing evaluation, the proposed system can be better applied to assist sight-singing practice than existing singing evaluation systems.

To establish an automatic sight-singing evaluation system, two main problems must be solved: 1) obtaining the information of each note (usually including onset, offset, and pitch) from audio; 2) aligning the sequence of sight-singing notes with the sequence of score notes for evaluation. Therefore, the automatic sight-singing evaluation system proposed in this paper (as shown in Fig. 1) consists of two corresponding modules: *note transcription* and *note alignment*.

In the literature, the critical issue of note transcription is note segmentation, which separates the notes from the audio signals in the temporal dimension. Currently, there are two main approaches for note segmentation: the first approach is based on time-domain pitch information and the other approach is based on spectrograms. As shown in Fig. 2 (a), most of the former methods [7][8][9] used the pitch extraction algorithm to obtain the pitch contour and then onset detection was achieved through time-domain smoothing or a hidden Markov model (HMM) to achieve note segmentation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia



(a) Pitch contour with onsets (green lines)

(b) Spectrogram with onsets (green lines)

Fig. 2: Onset detection based on pitch contour and onset detection based on spectrogram.

This type of onset extraction method utilizes post-processing to the one-dimensional pitch contour, which is limited by the performance of the pitch extraction algorithm and specific strategies. In the latter methods [10][11], researchers concentrated on the spectrogram and modeled the transition stage of musical notes through the traditional autocorrelation algorithm or convolutional neural network (CNN) based on deep learning to obtain onsets, as shown in Fig. 2 (b). These approaches focus on the energy changes of the note's onset from both the time and frequency dimensions and achieve better performance.

Note alignment, which is much more complicated than the alignment for monophonic instruments, is another core problem in automatic sight-singing evaluation systems. First, due to many inevitable portamentos and slurrings, there is no apparent boundary between the former note and the latter one, resulting in high note transcription error rates. Second, the vocal pitch makes it difficult to reach the standard intonation comparable to that of musical instruments, and unstable pitches during singing can lead to consecutive errors in the sung note sequence. These problems introduce differences between the sequence of sung notes and the sequence of score notes, which is a great challenge for the alignment algorithm. Currently, the commonly used methods for music alignment are based on dynamic programming. For example, singing evaluation systems [5][6] extract the pitches of each frame of the test singing clip and the reference singing clip. Then, these clips are matched to measure the similarity between them by using dynamic time warping (DTW). Tsai *et al.* [4] converted the singing signals into pitch sequences, energy sequences, and rhythm sequences and then used DTW to find the time mapping relationship.

Although none of the datasets or evaluation criteria have been proposed for sight-singing evaluation systems, there are some datasets or evaluation criteria for note transcription and note alignment. The ISMIR2014 dataset [12], which contains 38 singing recordings, is often used for note transcription. The evaluation criteria of note transcription are to match the transcribed notes with the ground truth [12] to obtain precision, recall, and F-measure. The evaluation strategy for note alignment is to compare the alignment results of the algorithm with the standard alignment results to obtain the accuracy [13][14][15]. However, there is no dataset for note alignment in sight-singing. Currently, there are only subjective evaluation methods for sight-singing. For instance, Schramm *et al.* [16] trained a Bayesian classifier to simulate human sight-singing standards to provide a judgment of correct or incorrect for each note. Moreover, previous singing evaluation systems [17][18][19][20] were used to extract singing characteristics to classify or rank singers. Such methods are often subjective and general, and are not suitable for effective sight-singing feedback.

For our proposed automatic sight-singing evaluation system (Fig. 1), we carefully consider the above problems and learn from the score-informed assessment methods [21]. To achieve accurate transcription of sight-singing notes, we propose a combination of a CRNN-based onset detector and a CNN-based pitch extractor [1]. The onset detector and the pitch extractor are used to extract the onset time and frame-level pitch contour from the audio, respectively, and then note segmentation is used to combine the two results to obtain the sung notes. To achieve the alignment of the sung note sequence and score note sequence, we use the Needleman-Wunsch (NW) algorithm [22] based on relative pitch. Finally, according to the alignment results, the sight-singing performance is evaluated to provide objective and accurate feedback at the note level. Furthermore, to evaluate the proposed system, we sample and construct the sight-singing vocal dataset (SSVD)[1], which contains 127 sight-singing samples.

This paper is organized as follows: the related work is introduced in Section II. In Section III, we describe our proposal for the automatic sight-singing evaluation system in detail. In Section IV, the construction of the SSVD dataset is described. In Section V, we explore the contribution of the different submodules and analyze the overall performance of the proposed system. In Section VI, we summarize our work and draw conclusions.

II. Related Work

A. Note Segmentation

Note segmentation refers to extracting the onsets and offsets of the notes from given audio, which can be roughly divided into the methods based on pitch information and the methods based on spectrogram features.

A simple and commonly referenced note segmentation approach based on pitch information was proposed by McNab *et al.* [23]. Inspired by McNab, Molina *et al.* [7] observed the hysteresis process caused by the note change in the pitch-time curve to further improved the performance of note segmentation. Kroher *et al.* [24][25] developed a series of onset detection functions based on the volume and pitch characteristics to successfully segment the flamenco pitch contour into discrete note events. Mauch *et al.* [8] and Yang *et al.* [9] used HMM and hierarchical HMM to model the different states of a note, and then segmented the notes following Viterbi decoding. However, due to the limitation of the pitch

[1]https://github.com/itec-hust/Sight-Singing-Vocal-Data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

estimation algorithm, incorrect pitches will affect the results of note segmentation. Therefore, some studies considered the spectrogram feature to detect note onsets for note segmentation. For example, Chang *et al.* [10] extracted onsets and offsets based on the spectrogram characteristics and achieved good results on Korean singing data [26]. Moreover, Schlüter *et al.* [11] first proposed an onset detection function based on CNN by modeling the note transition of various musical instruments from a spectrogram and there were considerably improvements over the previous methods. Recently, many methods [27][28] based on spectrograms and deep learning have been proposed. Therefore, in this paper, we consider taking the spectrogram and deep learning methods to execute note segmentation.

B. Music Alignment

Music alignment has been applied in a variety of music information retrieval tasks, such as query-by-humming [29][30][31][32] and score following [14][33][34][35]. It is necessary to match the features in the aligned signal of music performance with the reference signal through an alignment algorithm, which requires a proper combination of alignment features and alignment algorithms.

The frame-level alignments usually use pitch contour [36], chroma [37][38][39] or spectral information [40][41] as alignment features. The note-level music alignment methods use the pitch value of note [13][14][15] or the triplets [30][42] that are modeled by the relative pitch between adjacent notes for alignment. Dynamic programming (DP) approaches such as DTW and NW [22] algorithms are often used to find the globally optimal alignment of sequences. Molina *et al.* [5] proposed that DP-based similarity is not only simple but also efficient. However, Grachten *et al.* [43] pointed out that DTW cannot handle structural differences (unexpected insertions or deletions made by performers) adequately without manual assistance. Therefore, they proposed an alignment method based on NW. This method achieves almost the same alignment accuracy as DTW when there is no structural difference between the sequences. When structural differences exist, the method proposed by Grachten *et al.* favors deleting the unexpected occurrence over a sequence of poor matches rather than forcing a match between elements as DTW does. Moreover, the choice of constraints in the alignment algorithms will cause different time warping, and result in different alignment effects.

C. Pitch Extraction

Pitch extraction, which refers to the estimation of the F0 trajectories of the audio signals, has been conducted on speech [44][45], singing voices [46] and musical instruments [47]. The most common traditional methods [48][49][50] are based on the analysis of local maxima of the autocorrelation function. These approaches are known to be prone to octave errors because the peaks of the ACF repeat at different lags. Therefore, several methods were introduced to be more robust to such errors, including the PRAAT [49] and the YIN algorithm [51]. Then, Mauch *et al.* [52] further proposed the pYIN algorithm, a joint probability model based on the YIN algorithm, which made the prediction results more reliable and became the best scheme among traditional algorithms. Recently, Kim *et al.* [1] proposed deep neural networks to demonstrate the best performance in pitch estimation tasks. Kim *et al.* used an end-to-end convolutional neural network to directly process the time-domain audio signal. Even at a very strict evaluation threshold of 10 cents, more than 90% of the pitch accuracy was maintained both on the RWC-synth dataset [53] and the MDB-stem-synth dataset [1].

D. Singing Evaluation System

Singing evaluation has recently been an area of interest. The existing studies can be primarily divided into two categories: reference-based methods [4][6][54] and non-reference-based methods [17][20]. In this paper, we only focus on reference-based methods. In these methods, it is common to first extract various acoustic features of singing clips, including pitch, volume, rhythm, and timbre. Then, these features can be compared with the reference features that usually come from the original music albums, such as CDs or VCDs [4][54]. For example, to improve singing evaluation capabilities, Tsai *et al.* [4] attempted to exploit various acoustic features to assess singing performances. Gupta *et al.* [6] explored different features of the audio signals that represent perceptual parameters to develop the singing assessment. In these works, researchers always compared the test samples for overall feedback or classifying good/bad singers. Although various characteristics from audio signals were extracted, these works aimed at approaching human-based judgments, resulting in subjectivity of their evaluations. However, we consider providing an objective sight-singing feedback note-by-note. Therefore, the current evaluation systems are not sufficient to reach our goal.

III. The Automatic Sight-singing Evaluation System

In the proposed system, the onsets and pitch contour of the input audio are first obtained by the onset detector and pitch extractor, respectively. Then, the pitch contour is segmented with the onsets to complete note transcription. The process of note alignment aligns the transcribed note sequence with the score note sequence. Finally, the feedback of the performance is provided to the singer in a visual manner. In the following sections, each module is described in detail.

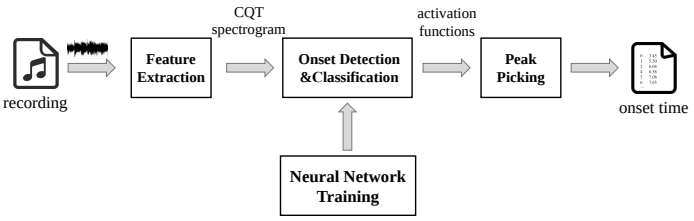


Fig. 3: The workflow of the onset detector.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

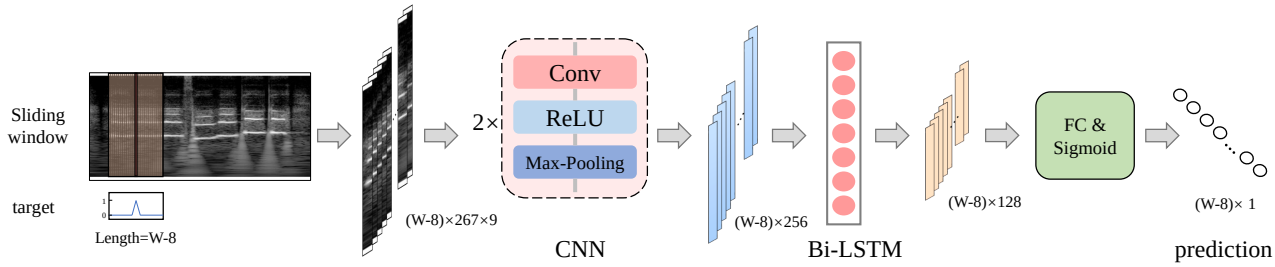


Fig. 4: The neural network architecture of the CRNN model.

A. Onset Detector Using CRNN

The onset detector utilizes the workflow shown in Fig. 3 and includes audio signal feature extraction, note onset detection, and peak picking.

1) Feature Extraction: In the field of audio signal processing, the time-frequency transform is often used to convert the original signal to a two-dimensional spectrogram. Common time-frequency transforms include short-term Fourier transform (STFT) and constant-Q transform (CQT). Compared with STFT, the frequency bins of CQT are distributed according to the exponential law similar to the twelve-tone equal temperament, which is more suitable for music signal processing. In this paper, the CQT function in *LibROSA* [55] is used for the time-frequency transform. We set F_{min} equal to the note $A0$ (i.e., 27.5Hz), and compute up to 267 CQT bins. We use a Hanning window with a hop length set equal to 512 samples, and one octave comprises 36 bins.

2) CRNN-based Onset Model: The structure of the CRNN model is shown in Fig. 4. First, the spectrogram is first passed to CNN. Then, the bidirectional long short-term memory (Bi-LSTM) network is used for time-dependent modeling, and finally, the binary classification results are obtained through the fully connected layers. A longer input spectrogram will provide more background information to improve the accuracy of onset detection. Therefore, we use the spectrogram of W frames as an input and use a 9-frame sliding window for segmentation. The sliding window moves one frame at a time, finally, we obtain a total of (W-8) groups of spectrogram segments, which are sent into the model for feature extraction and classification. After the experiments, we use the model with W=43 as our onset detector.

3) Details of Model Training: The parameters of the model are shown in Table I. The convolution layer parameters, H×W@C, refer to the height of the convolution kernel as H, width as W, and the number of channels as C. The max-pooling layer parameters, PH×PW/PSH×PSW, indicate that the height of the pooling area is PH, the width is PW, the step size along the height direction is PSH, and the step size along the width direction is PSW.

The loss function of the onset detection model is as follows:

$$loss = -\frac{1}{n}\sum_{x} \alpha y \ln \hat{y} + (1-y)\ln(1-\hat{y}) \qquad (1)$$

where y is the annotation and $\hat{y}$ is the predicted value. Due to the unbalanced distribution of positive and negative samples, we use the positive sample weight α to increase the loss weight of positive samples. Here we choose α as 5.

TABLE I: The network parameters of the CRNN.

Input	Layers & Parameters	Output
1×267×W	Group	(W-8) × 1 × 267 × 9
(W-8)×1×267×9	Convolution:25×3@21	(W-8)×21×243×7
(W-8)×21×243×7	Max-Pooling:3×2/3×2	(W-8)×21×81×3
(W-8)×21×81×3	Convolution:7×3@42	(W-8)×42×75×1
(W-8)×42×75×1	Max-Pooling:3×1/3×1	(W-8)×42×25×1
(W-8)×42×25×1	Reshape	(W-8)×1050
(W-8)×1050	Dropout+Fc:512+Relu	(W-8)×512
(W-8)×512	Dropout+Fc:256+Sigmoid	(W-8)×256
(W-8)×256	Bi-LSTM:512	(W-8)×1024
(W-8)×1024	Fc:128+Reshape	1×((W-8)×128)
1×((W-8)×128)	Fc:1	(W-8)×1

The onset detection model is trained on a dataset consisting of 111 sight-singing recordings with 5,443 onsets that total 3,920 s and contain only the onset annotations. To accelerate the convergence of the training process, we add batch normalization layers after each convolution layer and use the pooling layers, dropout layers, and regularization to prevent overfitting. We use the Adam optimizer [56], with a learning rate of 1e-5 and a batch size of 64. After training for 60 epochs, the onset detection model reached convergence. Our experimental platform graphics card is an NVIDIA GTX 1080.

4) Peak Picking: The peak detection method is used for post-processing the output probabilities of the onset detector to obtain the optimal onsets. Peak detection consists of four steps: (1) Smoothing and normalizing the onset probabilities: Smoothing is used to filter out noisy probabilities, while normalization allows the data to be compared on the same scale. (2) Threshold processing: Threshold processing is performed on input probabilities, with the onset probabilities greater than the threshold retained and those less than the threshold set to zero. As a result, some peaks of non-onset points less than the threshold can be eliminated. Here, we set the threshold as 0.5. (3) Peak selection: After threshold processing, the maximum selection is carried out for several consecutive probabilities greater than the threshold value. (4) Continuous note elimination: As it is impossible for sight-singing to have two onset events within a short period of time, we set 100 ms as the time threshold to eliminate the extra onsets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

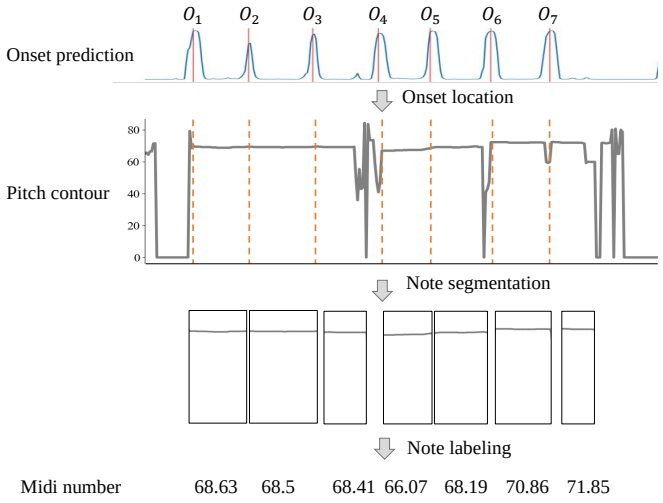


Fig. 5: The process of note segmentation and note labeling.

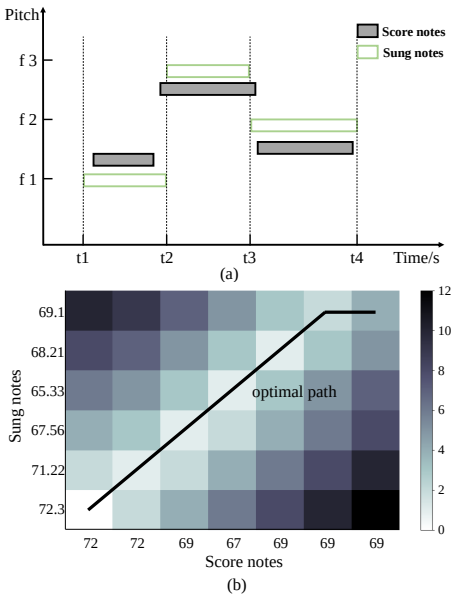


Fig. 6: (a) Illustration of common difference between sung notes an score notes; (b) Distance matrix and optimal path obtained by NW algorithm.

B. Pitch Extractor and Note Segmentation

1) Pitch Extractor: In this paper, the approach by the Kim *et al.* [1] is adopted to extract pitch contours. This model is based on a deep convolutional neural network, which is used to directly process temporal audio samples to obtain the fundamental frequency. Compared with pYIN [52], the model is based on deep learning and can achieve a better performance.

2) Note Segmentation: We combine the outputs of the onset detector and pitch extractor to complete note segmentation. Assuming the onset detector's prediction is $O = \{O_1, O_2, \ldots, O_n\}$, the output of pitch extractor is $F = \{F_1, F_2, \ldots, F_n\}$, where O_i is the onset probability of each frame, F_i is the fundamental frequency of each frame and n is the total number of CQT spectrogram frames. The segmentation algorithm is as follows:

- The onset peak values are selected from O. The frame index of these peak points is remembered as $k_1, k_2, ..., k_i, ...$
- Two adjacent peak onsets (O_{k_i} and $O_{k_{i+1}}$) segment F to obtain the pitch contour $F = \{F_{k_i}, F_{k_i+1}, \ldots, F_{k_{i+1}}\}$, which contains the current note and a silent area from the end of the current note until the next note onset.
- The pitch value for each separated sung note is labeled as described in the method by [7]: First, the extreme values of the pitch contour boundary $F = \{F_{k_i}, F_{k_i+1}, \ldots, F_{k_{i+1}}\}$ are removed, and then the pitch contour is traversed with the dynamic average to estimate the note pitch center. Estimates of the average pitch become more accurate as the length of the note increases. When the instantaneous pitch deviates greatly from the average value, it means that a note is finished. Finally, a median filter is utilized to determine the pitch of the current note. If the duration of the note is too short (e.g., 10 ms), it is directly discarded.
- The above steps are repeated to obtain all of the transcribed notes.

C. Note Alignment Based on Relative Pitch Modeling

1) Relative Pitch Modeling: It is challenging for the pitches of sung notes to be the same as the pitches of score notes. As shown in Fig. 6 (a), there are large differences between these notes. The traditional NW and DTW algorithms have larger difficulties with sight-singing note alignment. If the absolute pitches of the notes are directly taken as the alignment feature, it is difficult for the NW algorithm or DTW algorithm to handle with these differences, which results in many mismatches.

Based on the analysis of sight-singing samples, we find two primary characteristics. First, consecutive errors exist. If the initial pitch of a sung note deviates from the pitch of the score note, the subsequent pitches generally deviate to a certain extent, resulting in consecutive erroneous notes. Second, the pitch deviations of sung notes are generally in the same direction, that is,

$$\left(f^i_{\text{score}} - f^{i-1}_{\text{score}}\right) \times \left(f^i_{\text{sung}} - f^{i-1}_{\text{sung}}\right) \geq 0 \tag{2}$$

where f^i_{score} represents the pitch of the i-th note in the score and f^i_{sung} refers to the pitch of the i-th note in the sung notes. Therefore, the relationship of the adjacent sung notes is similar to that of the score notes. We use this relative relationship to encode the two sequences of sung notes and score notes, so that the matching effect can be achieved by only using the relative pitches of notes.

The steps of relative pitch modeling are as follows:

- Set the sequence of detected sung notes as $D = \{d_1, d_2, \ldots, d_m\}$ and the sequence of score notes as $S = \{s_1, s_2, \ldots s_n\}$. Then, use Eq. (3) and Eq. (4) to obtain the sequences X and Y, respectively. The outputs of the functions are $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$, where X and Y are the sequences

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

of sung notes and score notes based on relative pitch, respectively.

$$x_i = \begin{cases} \text{sgn}\left(d_{i+1} - d_i\right), & 1 \le i \le m-1 \\ \text{sgn}(0), & i = m \end{cases} \tag{3}$$

$$y_j = \begin{cases} \text{sgn}\left(s_{j+1} - s_j\right), & 1 \le j \le n-1 \\ \text{sgn}(0), & j = n \end{cases} \tag{4}$$

- After the relative pitch modeling, only three elements remain: 1) the pitch of the current note is higher than that of the previous note (Greater, G); 2) the pitch of the current note is the same as the previous note (Equal, E); and 3) the pitch of the current note is lower (L) than the previous note. As shown in Fig. 6 (b), suppose the two sequences are $D = \{72.3, 71.22, 67.56, 65.33, 68.21, 69.1\}$ and $S = \{72, 72, 69, 67, 69, 69, 69\}$. The results after relative pitch modeling are X=$\{L, L, L, G, G, E\}$ and Y = $\{E, L, L, G, E, E, E\}$.

2) NW Alignment Algorithm: We reconfigure some parameters of the NW algorithm. First, the gap penalty is set to be positive and greater than the mismatch penalty, which ensures that the cost of delete notes is greater than the cost of mismatches and minimizes the cost of the algorithm to fill the gap during backtracking. To reduce the probability of horizontal backtracking during the matching process, we fill the alignment matrix with a modified gap penalty that can narrow the search space to incorporate length constraints. We first set the position (1,1) to zero and then fill the first column and first row with the gap penalty. Then, the minimum value of the distance matrix is obtained in the process of backtracking. We set the gap penalty γ as 2, the matching cost σ as 0, and the insertion cost ω as 1.

$$nw(i,j) = \begin{cases} 0, & \text{if } i = 0, j = 0 \\ \gamma + nw(i, j-1), & \text{if } i = 0, j \neq 0 \\ \gamma + nw(i-1, j), & \text{if } j = 0, i \neq 0 \\ min \begin{cases} nw(i-1, j-1) + S(x_i, y_i), \\ \gamma + nw(i-1, j), \quad otherwise \\ \gamma + nw(i, j-1), \end{cases} \end{cases} \tag{5}$$

$$S(x_i, y_j) = \begin{cases} \sigma, & x_i = y_j \\ \omega, & x_i \neq y_j \end{cases} \tag{6}$$

The alignment algorithm is as follows:

- Distance matrix calculation: The distance matrix is completed according to the NW algorithm (Eq. (5) and Eq. (6)), as shown in Fig. 6 (b), where a lower score in the matrix means a smaller gap between two elements and a higher matching possibility.
- We will start from the upper-right corner of Fig. 6 (b) and backtrack to the lower-left corner. If the two elements are the same (i.e., $X_i = Y_j$), then the two elements are considered to be matched, and the next step is $(i-1, j-1)$. If the two elements are different (i.e., $X_i \neq Y_j$), the following three options exist for our next step: Move one space to the left, move one space to the bottom, or move one space to the bottom left. We select the position with the lowest similarity gap among the three positions.

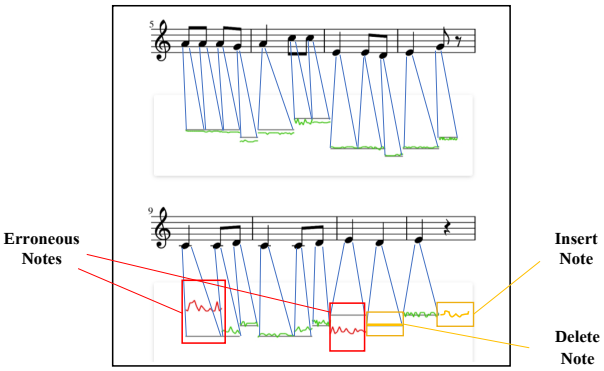


Fig. 7: Visual feedback of the system.

- The above steps are repeated until the left bottom corner is reached.

D. Evaluation and Visual Feedback

The system described in this paper is based on the evaluation of pitch accuracy to judge the correctness of notes in sight-singing. After the processing of the three stages introduced above, the alignment results, which are similar to those in Fig. 8, are obtained. By comparing the alignment results, the system can judge each note as a correctly sung note or an incorrectly sung note. Incorrectly sung notes include insert, delete and erroneous notes. For matched notes, if the pitch difference between the score note and the sung note is less than 0.5 MIDI number, then the note is considered to be sung correctly; otherwise, it is an erroneous note. To visualize the performance feedback, the pitch curve of each sung note is displayed, as shown in Fig. 7. Correctly sung notes are shown in green. Erroneous notes are shown in red, and insert notes and delete notes are shown in yellow.

IV. Dataset

We construct a sight-singing test dataset to evaluate the performance of the entire system and each submodule. This dataset contains the onset, pitch, and alignment annotations between the sung note sequence and the score note sequence of each sight-singing sample. In the following, we describe the collection of the sight-singing dataset and the annotation strategies for the ground truth.

A. Music Collection

Currently, there is no relevant dataset for sight-singing note alignment or sight-singing evaluation system. Therefore, we propose the sight-singing vocal dataset (SSVD). We randomly select 127 real sight-singing samples from the WeChat mini-program *Sight-singing Talent*[2]. Each sample has a corresponding sheet music. The dataset includes 91 singers, 46 different music scores, and a total of 5,206 notes. The duration of each

[2]Sight-singing Talent is an online WeChat mini-program that obtains timely reviews and has more than 600 long-term users and more than 60,000 effective sight-singing samples.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

song ranges from 15 s to 69 s, and the total duration of the whole dataset is 75 minutes. The singers include untrained students, ordinary amateur musicians, and music teachers. All the songs are rich in various notes and rhythms. All recording samples are single-channel MP3 files with a sampling rate of 44,100 Hz and a resolution of 16 bits.

We annotate the onsets and pitches for each song and use tuples to represent the alignment between the sung note sequence and the note sequence of the corresponding score. Fig. 8 (a) shows the manually labeled onsets and pitches of a certain piece of music. Fig. 8 (b) shows the note sequence of the score and Fig. 8 (c) shows the matching relationship between Fig. 8 (a) and Fig. 8 (b). The onset annotations in the second column of Fig. 8 (a) can be used independently to evaluate the onset detector. The combination of onset and pitch annotations can be applied to test the note transcription submodule. The annotation contained in Fig. 8 (c) can be used to test the note alignment algorithms. The alignment annotation contains two columns. The first column is the note index of the sung notes, and the second column is the note index of the score notes. Each row [i, j] represents a note mapping, that is, the i-th note of the sung note sequence is matched to the j-th note of the score note sequence. We use [i, -] or [-, j] to represent a note that cannot be matched to another note.

Row index	Onset	Pitch
0	1.16	72.31
1	1.73	72.35
2	2.29	67.11
3	2.58	65.63
4	2.88	62.83
5	3.45	65.14
6	3.74	65.0
7	4.05	62.87
8	4.33	65.19
9	4.67	62.35
10	5.8	65.92
11	6.44	69.91

(a) Annotation of onset and pitch

Row index	Score note
0	72
1	72
2	69
3	67
4	65
5	62
6	62
7	65
8	62
9	65
10	69

(b) Score note

Alignment annotation
(0,0)
(1,1)
(-,2)
(2,3)
(3,4)
(4,5)
(5,-)
(6,-)
(7,6)
(8,7)
(9,8)
(10,9)
(11,10)

(c) Alignment annotation

Fig. 8: An instance of annotation.

B. Ground Truth: Annotation Strategy

The music collection is manually annotated to build the ground truth. In this section, we introduce the annotation strategy in detail.

- Onset annotation: An onset is usually defined as the exact time when a note or instrument is played. However, such a moment is difficult to determine, so the most commonly used method is to mark the onset as the earliest time point at which humans can hear the sound. Referring to the onset annotation method proposed in [57], we annotate the onsets by manually listening to audios during slowed-down playback and observing the spectrograms obtained by the short-time Fourier transform. All annotation work is performed under the guidance of three well-trained experts in music, and each annotated onset is cross-confirmed multiple times.
- Pitch annotation: First, we transcribe the recordings with an existing note transcription method [8]. Then, all the transcription errors are corrected by three experts in music. All the annotations are checked until all of the experts agreed on their correctness. The pitch of each note is annotated with 10 cents resolution in the form of the MIDI number.
- Note alignment annotation: To obtain the ground truth of the note-level mapping, we first extract the sung note sequence from pitch annotation, and obtain the score note sequence from the corresponding score. Then, we run the note alignment algorithm in Section III-C to align the two sequences. All of the alignment results are examined and corrected by three experts.

V. Evaluation

In this section, we explore the performance of the automatic sight-singing evaluation system and evaluate each module in the paper. In Section V-A, the evaluation measures for the different submodules and the system are introduced. As the critical information of note segmentation, onsets and pitches affect the performance of note segmentation. Therefore, we evaluate and analyze the capabilities of different onset detection methods on the ISMIR2014 dataset and the SSVD dataset in Section V-B. Since we adopt state-of-the-art monophonic pitch extractor [1], we no longer test the performance of pitch extraction. In Section V-C, various note transcription algorithms are compared on the ISMIR2014 dataset and the SSVD dataset. Note alignment is a critical step in this system, and we investigate the performance of note alignment methods in Section V-D. The evaluation of the overall system will be discussed in Section V-E.

A. Evaluation Measures

In this section, we describe the evaluation measures used to test the performance of onset detection, note transcription, note alignment and the overall system. Here, precision, recall, and F-measure are considered as the main evaluation criteria. The performances of the algorithms on a certain dataset are represented by the averages of the precision, recall, and F-measure of all songs.

1) Evaluation Measure for Onset Detection: As in [12][57], we choose a window length of 100 ms as the onset criteria measure. If a detected onset is within 100 ms (i.e., ±50 ms) of an unmatched ground truth, it is regarded as a true positive (TP). Each detected onset can only be matched once. If there is more than one detected onset in the same evaluation window marked with a ground truth, the first one is regarded as a true positive (TP), and all others are regarded as false positives (FP). The ground truths that are not detected are false negatives (FN). The precision, recall, and F-measure are defined as follows:

$$P_{\text{onset}} = \frac{TP}{TP + NP} \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

$$R_{\text{onset}} = \frac{TP}{TP + FN} \tag{8}$$

$$F_{\text{onset}} = \frac{2P_{\text{onset}}\, R_{\text{onset}}}{P_{\text{onset}} + R_{\text{onset}}} \tag{9}$$

2) Evaluation Measure for Note Transcription: This study only focuses on the onset and pitch of the transcribed notes. According to the evaluation measure in [12], when a output note has the correct onset (within ±50 ms of the unmatched ground truth) and the correct pitch (within ±0.5 semitone of the ground truth), the note is correctly transcribed. Each output note and ground truth can only be matched once. Aussming that $\mathcal{J}_{\text{gt}}$ is the total number of annotated sung notes, $\mathcal{J}_{\text{output}}$ is the total number of output notes, and $\mathcal{J}_{\text{correct}}$ is the total number of correctly transcribed notes. The precision, recall, and F-measure are defined as follows:

$$P_{\text{note}} = \frac{\mathcal{J}_{\text{correct}}}{\mathcal{J}_{\text{output}}} \tag{10}$$

$$R_{\text{note}} = \frac{\mathcal{J}_{\text{correct}}}{\mathcal{J}_{\text{gt}}} \tag{11}$$

$$F_{\text{note}} = \frac{2P_{\text{note}}\, R_{\text{note}}}{P_{\text{note}} + R_{\text{note}}} \tag{12}$$

3) Evaluation Measure for Note Alignment: According to the evaluation measures used in [14], we use precision, recall, and F-measure to evaluate the effectiveness of the alignment methods. The parameter TP_{align} is the number of true positives, which are the intersections of the alignment result and the ground truth, FP_{align} represents the total number of false positives, which are the note-matching set of the alignment result that are absent in the ground truth, and FN_{align} is the number of false negatives, which are the note-matching set that are not correctly indicated by the alignment algorithm.

$$P_{align} = \frac{TP_{\text{align}}}{TP_{\text{align}} + FP_{\text{align}}} \tag{13}$$

$$R_{\text{align}} = \frac{TP_{\text{align}}}{TP_{\text{align}} + FN_{\text{align}}} \tag{14}$$

$$F_{\text{align}} = \frac{2P_{\text{align}}\, R_{\text{align}}}{P_{\text{align}} + R_{\text{align}}} \tag{15}$$

4) Evaluation Measure for System: If a correctly transcribed note is also correctly matched to the corresponding musical score note, we think that the system can correctly evaluate the sung note. Assuming $n_{correct}$ represents the number of the all correctly transcribed and aligned notes, n_{output} and n_{score} represent the total number of all detected notes and the total number of all score notes in the dataset, respectively.

$$P_{\text{system}} = \frac{n_{\text{correct}}}{n_{\text{output}}} \tag{16}$$

$$R_{\text{system}} = \frac{n_{\text{correct}}}{n_{\text{score}}} \tag{17}$$

$$F_{\text{system}} = \frac{2P_{\text{system}}R_{\text{system}}}{P_{\text{system}} + R_{\text{system}}} \tag{18}$$

TABLE II: Performance of different onset detection methods.

	ISMIR2014			SSVD		
	P(%)	**R(%)**	**F(%)**	**P(%)**	**R(%)**	**F(%)**
Molina [7]	61.26	65.26	62.73	72.03	80.89	75.85
Mauch [8]	72.69	64.65	68.16	76.23	76.2	76.21
Yang [9]	69.12	61.1	64.34	68.12	74.54	70.97
Chang [10]	67.1	72.1	69.5	79.62	78.97	79.23
Proposed	**93.03**	**85.67**	**89.02**	**91.27**	**90.11**	**90.63**

B. Evaluation for Onset Detector

According to the discussion in Section II-A, there are methods [7][8][9] based on pitch contour and methods [10] based on spectrograms for onset detection. In these pitch contour-based methods, notes are often detected by specific features from the pitch curve. In these spectrogram-based methods, notes are segmented by directly seeking onsets and offsets from the spectrogram.

The most common dataset for onset detection is the ISMIR2014 dataset, which consists of 38 melodies sung by untrained child and adult singers and includes 2,154 notes with onset, offset and pitch annotations. In this section, the above four methods and the CRNN-based onset detector proposed in this paper are compared on the ISMIR2014 dataset and the SSVD dataset. We use the function *mir_eval.onset.evaluate* in the Python library *mir_eval* [58] to make these evaluations.

1) Result: As shown in Table II, among the three pitch-based onset detection methods, the best method is proposed by Mauch *et al.*; F-measures of 68.16% and 76.21% are achieved on the ISMIR2014 dataset and the SSVD dataset, respectively. Meanwhile, when using the spectrogram-based method proposed by Chang *et al.*, F-measures of 69.5% and 79.23% are achieved on the ISMIR2014 dataset and the SSVD dataset, respectively. It can be seen that the traditional spectrogram-based method has a better performance than the methods based on the pitch contour. For the method proposed in this paper, F-measures of 89.02% and 90.63% are achieved on the ISMIR2014 dataset and the SSVD dataset, respectively, which are 19.52% and 11.4% higher than the method proposed by Chang *et al.*. Obviously, our onset detector outperforms all of the other four methods and has the best and the most stable performance. Although our onset detector is trained on the sight-singing dataset without lyrics, the proposed model still handles English songs in the ISMIR2014 dataset well.

2) Error Analysis: To further analyze the performance of each model, we count three kinds of errors (shown in Fig. 9) introduced by the five onset detection methods on the two datasets. As shown in Fig. 10, the total numbers of detection errors from the method proposed by Chang *et al.* and our onset detector are smaller than those of the other three methods on the two datasets. It can be seen that the one-dimensional pitch contour is less adequate than the two-dimensional time-frequency features for onset detection.

In addition, we observe that noise extra detections (extra detections caused by noise) are widespread among the other four methods. This demonstrates that these onset detection functions cannot handle noise well, and the interference of environmental noise and background sounds will affect their

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

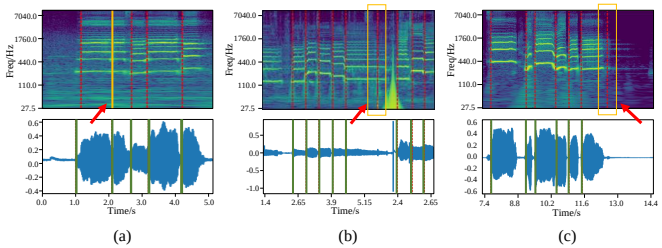


Fig. 9: Three kinds of errors in onset detection. Green lines indicate onset annotation, red dotted lines indicate predictions, and yellow boxes or yellow lines indicate errors. (a) missing detection, (b) successive detection in one note, (c) noise extra detection.

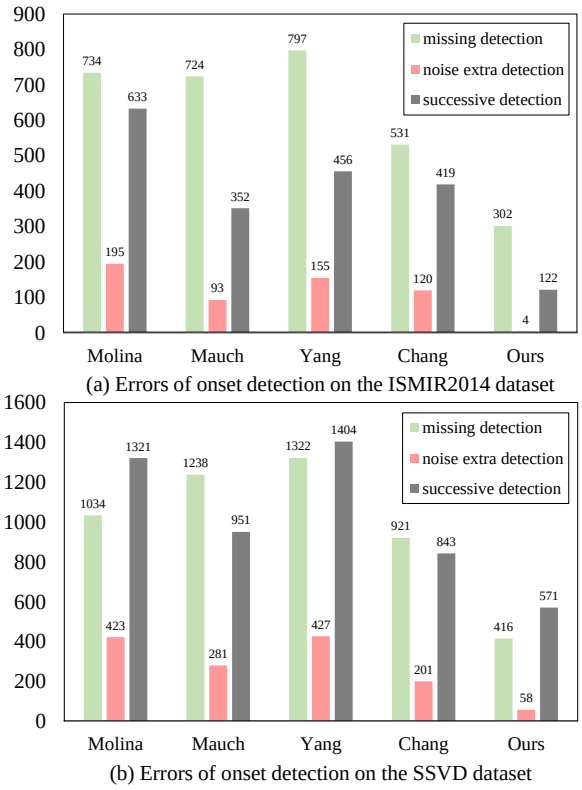


Fig. 10: Error analysis of onset detection methods.

onset judgment. In contrast, our onset detector produces fewer noise extra detections. This is because our model detects onsets by CNN from spectrograms which extracts more valid information from adjacent frames. Simultaneously considering the temporal and spectral information between adjacent frames can help the onset detector overcome the influences of noise. Neither the pitch contour-based methods nor the traditional spectrogram-based methods can solve the problem of successive extra and missing detections well. Therefore, we utilize Bi-LSTM to learn the feature between adjacent onsets by inputting the feature vectors in a period of time. This process can be used to solve some extra onsets on the same note that are obviously impossible. For example, it is generally impossible for a human to make two notes (onsets) within a small time interval. Some onsets hidden between consecutive same notes that are without apparent intervals can also be found using this method.

TABLE III: Performance of different note transcription methods.

	ISMIR2014			SSVD		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Molina [7]	59.45	63.11	60.79	71.02	79.8	74.8
Mauch [8]	71.27	63.41	66.85	73.17	73.29	73.15
Yang [9]	66.5	62.04	63.93	66.57	72.89	69.38
Proposed	**90.78**	**83.28**	**86.7**	**89.2**	**87.78**	**88.42**

C. Evaluation for Note Transcription

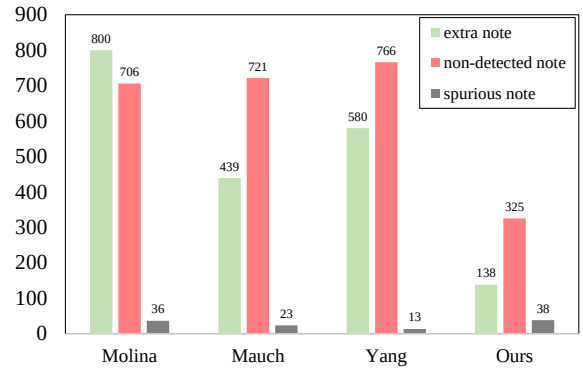
In this section, the results of note transcription are evaluated. We compare the traditional transcription methods from Molina *et al.* [7], Mauch *et al.* [8] and Yang *et al.* [9] with the method proposed in this paper on the ISMIR2014 dataset and the SSVD dataset. We use the *mir_eval.transcription.precision_recall_f1_overlap* [58] function to evaluate the performance.

1) Result: As shown in Table III, the method proposed by Mauch *et al.* performs better among the note transcription methods based on pitch contour, as F-measures of 66.85% and 73.15% are achieved on the ISMIR2014 dataset and the SSVD dataset, respectively. Our method achieves F-measures that are 19.85% and 15.27% higher than the values obtained by Mauch *et al.* on the ISMIR2014 dataset and the SSVD dataset, respectively. In addition, both the precision and recall of our method are the best on the two datasets. Combining the results in Table II, the performance of note transcription is positively correlated with the onset detection ability of each method. Moreover, we observe that the F-measures of note transcription of each method do not decrease significantly from the F-measures of onset detection. This shows that the note labeling of each scheme has minimal effect on the transcription results. Once the onsets of each note can be correctly detected, the pitch values extracted from pitch contour are sufficiently accurate; that is, onset detection results largely influence the performance of these note transcription methods. Since the results of our onset detector are already good enough, the performance of our note transcription is also stable.

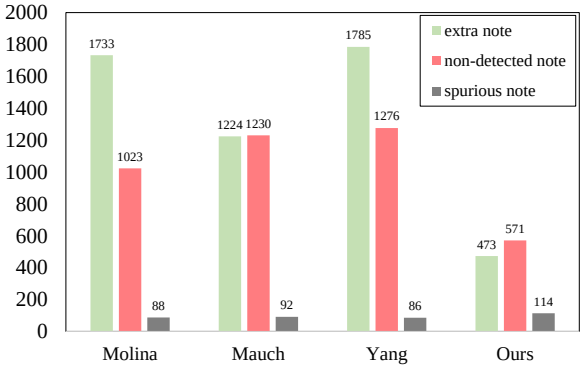
2) Error Analysis: To further analyze the performance of note transcription, we divide transcription errors into extra, non-detected and spurious notes. Extra notes represent the notes that the singer did not sing but the algorithm transcribed in the audio data, non-detected notes represent the notes that the singer sang but are not transcribed by the algorithm, and spurious notes represent transcribed notes that have the correct onset but the incorrect assigned pitch value.

The error cases of each note transcription method are shown in Fig. 11. It is obvious that the number of extra notes and non-detected notes when using our method is minimal on each of the datasets. Combined with the results in Fig. 10, since the missing and extra onsets of our onset detector are generally less than those of the other three methods, fewer extra notes and non-detected notes are generated in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia



(a) Errors of note transcription on the ISMIR2014 dataset



(b) Errors of note transcription on the SSVD dataset

Fig. 11: Error analysis of note transcription methods.

this submodule when using our method. However, we do not concentrate on precise offset estimation. Some segmented note generated by our method may not contain the correct pitch contour, which results in a few error pitch assignments, i.e., spurious notes. In general, what mainly affect the performance of note transcription are the extra notes and non-detected notes, and the method proposed in this paper minimizes these two errors very well.

D. Evaluation for Alignment Methods

In this section, we test the performance of the note alignment algorithms. We compare the alignment schemes that combine different note sequence modeling methods (relative pitch and absolute pitch) with different alignment algorithms (NW and DTW). In this section, RP-NW represents the alignment method of the combination of the NW algorithm and using the note sequence modeled by the relative pitch, and AP-NW represents the alignment method of the combination of the NW algorithm and using the note sequences of pitch values. The same is true for RP-DTW and AP-DTW.

1) Result: The results of RP-NW, AP-NW, RP-DTW, and AP-DTW are shown in Table IV. The F-measure of the NW algorithm is at least 8% higher than that of DTW algorithm, indicating that the NW algorithm handles sight-singing note alignment tasks better. After adding the relative pitch modeling, the F-measure of RP-NW is 5.2% higher than that of AP-NW, and the F-measure of RP-DTW is 1.84% higher than that of AP-DTW. These results suggest the effectiveness of the relative pitch modeling. Overall, RP-NW is a more suitable note alignment method for sight-singing.

TABLE IV: Performance of different alignment methods.

	P(%)	R(%)	F(%)
AP-DTW	81.75	81.33	81.54
AP-NW	89.63	89.57	89.59
RP-DTW	83.36	83.41	83.38
RP-NW	**94.76**	**94.83**	**94.79**

2) Further Analysis: To investigate the effectiveness of the note alignment methods, we observe the alignment annotations in the SSVD dataset and analyze the results of the alignment methods. We find that two factors affect the performance of the alignment methods: Erroneous notes and insert/delete notes (collectively called indel). Erroneous notes are the pitches of sung notes that deviate from those of target notes by more than 0.5 MIDI number. Indels are the notes in the score that are repeated or skipped by the singers. As shown in Fig. 12, the erroreous notes are divided into partly key transposition (KT, Fig. 12 (a)), consecutive random error (CRE, Fig. 12 (b)), and single random error (SRE, Fig. 12 (c)). The indels are divided into single random indel (RI, Fig. 12 (d)) and consecutive indel (CI, Fig. 12 (e)).

As shown in Fig. 12 (f), statistics are generated on the distribution of the five kinds of errors mentioned above on the SSVD dataset. It can be seen that the errors of key transposition account for the largest proportion of all cases. Insert notes and delete notes account for a small proportion of all cases and mostly appear as consecutive indels (CI, Fig. 12 (e)). Considering the principle of the RP-NW method, consecutive pitch shifts (pitches lower or higher than the target notes) between the sight-singing note sequence and the score note sequence can be better handled. The key transposition is one of these types of situations. Therefore, the RP-NW method can handle sight-singing note alignment well. In short, the NW algorithm with relative pitch can deal with the alignment problems in the situation with key transposition and a small amount of consecutive random errors, single random errors, and indels. When the errors of indels are prominent, various alignment schemes may fail.

E. Evaluation for the Overall System

In this section, we investigate the effectiveness of the overall system. To achieve this goal, we construct a comparative experiment. Next, we analyze the experimental results and provide a discussion.

Obviously, the result of the proposed system is the output of the cascade of note transcription and note alignment, so the output of note alignment is influenced by the results of note transcription. Here, we cascade the note transcription module proposed in this paper and the better-performing method proposed by Mauch *et al.* with AP-NW and RP-NW, respectively, to form four systems. The performances of these systems are shown in Table V. Comparing (1)(3) and (2)(4), we observe that the F-measure differences between these methods have a similar distribution as those between the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3168132, IEEE Transactions on Multimedia

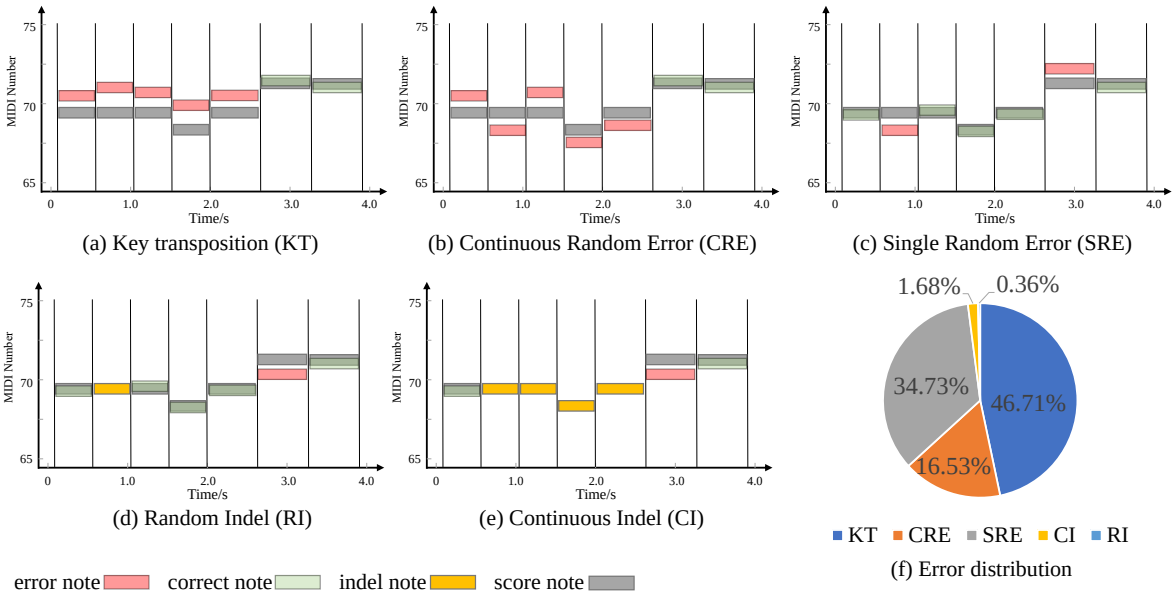


Fig. 12: Five kinds of errors in sight-singing (a $\sim$ e) and the distribution of the errors in the SSVD dataset (f).

TABLE V: The performance of the overall system.

	P(%)	R(%)	F(%)
Mauch[8] + AP-NW (1)	34.91	35.29	35.1
Mauch[8] + RP-NW (2)	63.3	63.99	63.64
Proposed + AP-NW (3)	48.8	48.45	48.62
Proposed + RP-NW (4)	**78.22**	**77.68**	**77.95**

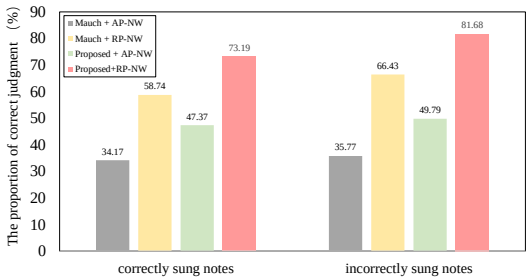


Fig. 13: The ability of system's discrimination for correctly sung notes and incorrectly sung notes.

two note transcription methods in Table III. This indicates that the effectiveness of the system is significantly correlated with the performance of note transcription. In addition, compared with Table IV, it can be seen that many extra notes and non-detected notes introduced by note transcription cause a rapid decline in the alignment performance. The decline in AP-NW is much greater than that in RP-NW. This result is consistent with the analysis in Section V-D 2).

Furthermore, we determine that the number of notes sung correctly by the singer is 2,111 and the number of incorrectly sung notes is 3,095 on the SSVD dataset. Therefore, we further explore the ability of the four systems to discriminate correctly and incorrectly sung notes (as defined in Section III-D). As shown in Fig. 13, for the proposed system, the proportion of correct discrimination of correctly sung notes in the dataset is 73.19%, and the proportion of correct discrimination of incorrectly sung notes is 81.68%, which surpasses the other three systems. Compared with AP-NW, the systems based on RP-NW are more sensitive to incorrectly sung notes, which can better detect the problems existing in sight-singing. At the same time, the lower discrimination of correctly sung notes also indicates that errors in sight-singing may affect the system's judgment of correctly sung notes. For example, we observe that when multiple insert notes surround a correctly sung note, the system has a high probability of misjudging the correctly sung note. This also implies that errors from note transcription will have negative effects on judging correctly sung notes.

In conclusion, the output of our system is mainly influenced by the extra notes and non-detected notes generated in the note transcription submodule. In Section V-C 2), we also find that the distribution of extra notes and non-detected notes produced by various note transcription methods is related to the results of onset detection. If extra and missing detections in the onset detection submodule can be reduced, the performance of the whole system will be improved.

VI. Conclusion

In this paper, a multi-stage automatic sight-singing evaluation system is proposed that focuses on note-level feedback and the SSVD dataset, which contains onset, pitch, and alignment annotations for evaluating each stage and the entire system, is established. This study mainly focuses on the onset detector and note alignment algorithm. The onset detector uses the CRNN model to extract the characteristics of onset and model the time dependence, as it has a better ability to resist noise interference and achieve a better effect than the other methods. By analyzing a large number of sight-singing problems, we propose a note sequence alignment algorithm that is more suitable for sight-singing. The NW algorithm based on

relative pitch modeling can more effectively solve the errors of key transposition than traditional NW and DTW algorithms. Finally, we propose an objective evaluation criterion for the sight-singing evaluation system, and with our system, an F-measure of 77.95% is obtained on the SSVD dataset.

Based on the experimental results in Section V, extra and missing detections introduced by onset detection and note transcription restrict the overall system performance, because the alignment algorithm is not always effective when there are a large number of insert notes and delete notes. One of the future research directions is to search for more accurate transcription and alignment methods that are sufficiently robust for insertions and deletions of notes. Moreover, this multi-stage system will cause the errors of the previous module to gradually propagate to the back end of the system and eventually introduce adverse effects to the entire automatic sight-singing evaluation system. An end-to-end sight-singing note transcription and alignment system may effectively alleviate this problem to a certain extent.

REFERENCES

[1] 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.

[2] D. Payne, "Essential skills for promoting a lifelong love of music and music making," *The American Music Teacher*, vol. 54, no. 4, p. 26, 2005.

[3] M. Oscar, B. Jordi, and L. Alex, "Performance analysis and scoring of the singing voice," in *35th International Conference: Audio for Games*, 2009, pp. 1–7.

[4] W. H. Tsai and H. C. Lee, "Automatic evaluation of karaoke singing based on pitch, volume, and rhythm features," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 4, pp. 1233–1243, 2011.

[5] E. Molina, I. Barbancho, E. Gómez, A. Barbancho, and L. Tardòn, "Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 744–748.

[6] C. Gupta, H. Li, and Y. Wang, "Perceptual evaluation of singing quality," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 577–586.

[7] 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 Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 252–263, 2014.

[8] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the tony software: Accuracy and efficiency," in *International Conference on Technologies for Music Notation & Representation*, 2015, pp. 23–30.

[9] L. Yang, A. Maezawa, J. B. Smith, and E. Chew, "Probabilistic transcription of sung melody using a pitch dynamic model," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 301–305.

[10] S. Chang and K. Lee, "A pairwise approach to simultaneous onset/offset detection for singing voice using correntropy," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 629–633.

[11] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 6979–6983.

[12] E. Molina, A. Barbancho, L. Tardón, and I. Barbancho, "Evaluation framework for automatic singing transcription," in *Proceeding of the 15th International Society for Music Information Retrieval Conference*. IEEE, 2014, pp. 567–572.

[13] B. Gingras and S. McAdams, "Improved score-performance matching using both structural and temporal information from midi recordings," *Journal of New Music Research*, vol. 40, no. 1, pp. 43–57, 2011.

[14] C. T. Chen, J. S. R. Jang, and W. Liou, "Improved score-performance alignment algorithms on polyphonic music," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 1365–1369.

[15] E. Nakamura, K. Yoshii, and H. Katayose, "Performance error detection and post-processing for fast and accurate symbolic music alignment," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 347–353.

[16] R. Schramm, H. Nunes, and C. Jung, "Automatic solfège assessment," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015, pp. 183–189.

[17] T. Nakano, M. Goto, and Y. Hiraga, "An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features," in *Ninth International Conference on Spoken Language Processing*, 2006, pp. 1706–1709.

[18] C. Gupta, H. Li, and Y. Wang, "Automatic evaluation of singing quality without a reference," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 990–997.

[19] N. Zhang, T. Jiang, F. Deng, and Y. Li, "Automatic singing evaluation without reference melody using bi-dense neural network," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 466–470.

[20] C. Gupta, H. Li, , and Y. Wang, "Automatic leaderboard: Evaluation of singing quality without a standard reference," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 13–26, 2019.

[21] J. Huang, Y.-N. Hung, A. Pati, S. K. Gururani, and A. Lerch, "Score-informed networks for music performance assessment," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.

[22] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[23] R. J. Mcnab, L. A. Smith, and I. H. Witten, "Signal processing for melody transcription," *proc.australasian computer science conf*, 1999.

[24] 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.

[25] N. Kroher, A. Pikrakis, and J. M. Diaz-Banez, "Discovery of repeated melodic phrases in folk singing recordings," *IEEE Transactions on Multimedia*, pp. 1–1, 2017.

[26] H. Heo, D. Sung, and K. Lee, "Note onset detection based on harmonic cepstrum regularity," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[27] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 161–165.

[28] J.-Y. Wang and J.-S. R. Jang, "On the preparation and validation of a large-scale dataset of singing transcription," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 276–280.

[29] J. S. R. Jang and H. R. Lee, "Hierarchical filtering method for content-based music retrieval via acoustic input," in *Proceedings of the ninth ACM international conference on Multimedia*, 2001, pp. 401–410.

[30] L. Lu, Y. Hong, and H. J. Zhang, "A new approach to query by humming in music retrieval," in *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.* IEEE, 2001, pp. 22–25.

[31] H. M. Yu, W. H. Tsai, and H. M. Wang, "A query-by-singing system for retrieving karaoke music," *IEEE Transactions on multimedia*, vol. 10, no. 8, pp. 1626–1637, 2008.

[32] N. H. Liu, "Effective results ranking for mobile query by singing/humming using a hybrid recommendation mechanism," *IEEE transactions on multimedia*, vol. 16, no. 5, pp. 1407–1420, 2014.

[33] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the 1984 International Computer Music Conference*, vol. 84, 1984, pp. 193–198.

[34] A. Arzt and S. Lattner, "Audio-to-score alignment using transposition-invariant features," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 592–599.

[35] S. Ewert, M. Muller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 1869–1872.

[36] A. Arzt, G. Widmer, and S. Dixon, "Adaptive distance normalization for real-time music tracking," in *2012 Proceedings of the 20th European*

Signal Processing Conference (EUSIPCO). IEEE, 2012, pp. 2689–2693.

[37] D. P. Ellis and G. E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4. IEEE, 2007, pp. 1429–1429.

[38] C. Joder, S. Essid, and G. Richard, "A comparative study of tonal acoustic features for a symbolic level music-to-score alignment," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 409–412.

[39] D. F. Silva, C.-C. M. Yeh, Y. Zhu, G. Batista, and E. Keogh, "Fast similarity matrix profile for music analysis and exploration," *IEEE transactions on multimedia*, vol. 21, no. 1, pp. 29–38, 2019.

[40] S. Dixon and G. Widmer, "Match: A music alignment tool chest," in *Proceedings of the 6th International Society for Music Information Retrieval Conference*, 2005, pp. 492–497.

[41] D. Bogdanov, J. Serrà, N. Wack, P. Herrera, and X. Serra, "Unifying low-level and high-level music similarity measures," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 687–701, 2011.

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

[43] M. Grachten, M. Gasser, A. Arzt, and G. Widmer, "Automatic alignment of music performances with structural differences," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013, pp. 607–612.

[44] D. Jouvet and Y. Laprie, "Performance analysis of several pitch detection algorithms on simulated and real noisy speech data," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 1614–1618.

[45] S. Strömbergsson, "Today's most frequently used f0 estimation methods, and their accuracy in estimating male and female pitch in clean speech." in *Conference of the International Speech Communication Association*. Dresden, 2016, pp. 525–529.

[46] 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 *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7815–7819.

[47] A. von dem Knesebeck and U. Zölzer, "Comparison of pitch trackers for real-time guitar effects," in *Proc. of the 13th Int. Conference on Digital Audio Effects*, 2010, pp. 525–529.

[48] J. Dubnowski, R. Schafer, and L. Rabiner, "Real-time digital hardware pitch detector," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 1, pp. 2–8, 1976.

[49] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proceedings of the institute of phonetic sciences*, vol. 17, no. 1193. Citeseer, 1993, pp. 97–110.

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

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

[52] M. Mauch and S. Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 IEEE International Conference on acoustics, speech and signal processing*. IEEE, 2014, pp. 659–663.

[53] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases." in *Proceedings of the 3rd International Society for Music Information Retrieval Conference*, vol. 2, 2002, pp. 287–288.

[54] W. H. Tsai, C. H. Ma, and Y. P. Hsu, "Automatic singing performance evaluation using accompanied vocals as reference bases," *Journal of information science and engineering: JISE*, vol. 31, no. 3, pp. 821–838, 2015.

[55] B. Mcfee, C. Raffel, D. Liang, D. Ellis, and O. Nieto, "LibROSA: Audio and music signal analysis in python," in *Python in Science Conference*, 2015.

[56] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[57] S. Bock, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012.

[58] 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 *Proceedings of the International Society for Music Information Retrieval Conference*, 2014.

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

Xianke Wang received the B.E. degree in electromagnetic field and wireless technology from Huazhong University of Science and Technology, Wuhan, China, in 2020. He is currently working toward to the M.S. degree at the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China. 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 from Huazhong University of Science and Technology, Wuhan, China, in 2020. He is currently working toward the M.S. degree at the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China. 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.

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.