HIERARCHICAL CLASSIFICATION NETWORKS FOR SINGING VOICE SEGMENTATION AND TRANSCRIPTION

Zih-Sing Fu
Dept. EE, National Taiwan University, Taiwan
b04901015@ntu.edu.tw

Li Su
IIS, Academia Sinica, Taiwan
lisu@iis.sinica.edu.tw

ABSTRACT

Identifying the onset and offset time of a note is a challenging step in singing voice transcription, as the soft onset/offset, portamento, and vibrato phenomena are rich in singing voice signals. In this work, we utilize various types of signal representations with deep learning for onset and offset detection of monophonic singing voice. We consider onset and offset detection as a hierarchical classification problem, where every input segment is classified into one of all the possible states in monophonic singing, namely the silence, activation, and transition states,where the transition state is further classified into the onset and offset states. An objective function based on this hierarchical taxonomy nicely guides the model to capture complicated temporal dynamics of note sequences. Multiple input signal representations containing spectral differences and pitch saliency are employed to jointly enhance such temporal patterns. The proposed method implemented with residual networks provides improved performance over prior art in onset and offset detection. Moreover, by integrating with a pitch detection framework, the proposed method also outperforms previous singing voice transcription methods. This result emphasizes the importance of note segmentation in singing voice transcription.

1. INTRODUCTION

Note-level automatic music transcription (AMT) refers to converting a recorded music piece into its symbolic form containing the onset, offset, and pitch of every note [4, 22]. Note-level AMT is still a challenging problem, particularly in the case of singing voice transcription. The soft onset/offset and portamento patterns of singing voice hinder the positioning of onset and offset time in both the detection [8, 29] and the annotation process [10, 15, 19]. However, solving the onset and offset detection problem, or equivalently the *note segmentation* problem, [1] is mandatory in a note-level AMT system. How to improve a note segmentation model efficiently with limited scope of data, and how to incorporate the outcomes of detection into note-level AMT, are both important issues in developing a complete AMT system.

Previous note segmentation works on singing voice usually employ state-space machines such as the hidden Markov models (HMM), which consistently detect onset and offset by characterizing the temporal dynamics among the *states* (attack, sustain, and silence, etc.) of note events [16,20,24,29]. Recently, deep neural networks with objective functions optimized for onset and offset detection have demonstrated excellent performance in note-level AMT [1,12]. Some architectures such as the convolutional neural network (CNN) do achieve a great advance in modeling note transition by their compelling performance in pattern recognition on a local scale. One example is the CNN-based onset detection method in [25], where the local feature segments with CNN outperforms the temporal models based on the recurrent neural network (RNN) [9].

In this paper, we propose novel signal representations and objective functions in neural network-based singing voice segmentation. we regard onset and offset detection as a hierarchical classification problem that maps input segments/sequences onto our proposed state space, where a generalized hierarchical taxonomy of the states in a note sequence is specified to guide the learning process. Multiple data representations are also used to enhance signal-level expressivity of note transition events. Experiments using either the residual network (ResNet) [13] or the RNN with attention [2] demonstrate the effectiveness of hierarchical classification in note segmentation. Finally, a straightforward integration of the proposed note segmentation method and pitch detection provides improved note transcription performance over prior art.

2. RELATED WORK

The most challenging case of onset detection is arguably singing voice. According to the results from MIREX 2018 audio onset detection task, the best F1-score of singing voice onset detection among all submissions is 61.94%, lower than the best results of other instrument classes by at least 10%. [2] The state-of-the-art onset detection algorithms are based on either RNN [5, 11] or CNN [25]. In [25], the onset detection task is to classify whether the

[1] We refer to note segmentation as temporal segmentation of note objects, which is therefore equivalent to onset and offset detection [7].

[2] More details can be found in: https://nema.lis.illinois.edu/nema_out/mirex2018/results/aod/resultsperclass.html

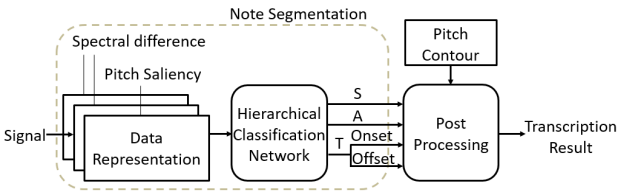

Figure 1: System overview of the proposed note segmentation and transcription framework.

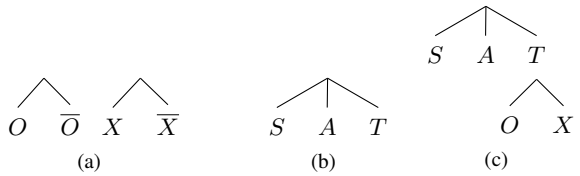

Figure 2: The taxonomy of the proposed models. Every tree represents an objective function, every siblings form a regularization term of the objective function, and every leaf of the tree represents a state label; S, A, O, $\overline{O}$, X, $\overline{X}$, and T represent silence, activation, onset, non-onset, offset, non-offset, and transition, respectively. Different trees therefore represent different optimization approaches: (a) On-Off model. (b) Tri-state model. (c) Hierarchical classification model. See Section 3.1 for more details.

middle of the input is at the onset time, where the inputs are short segments of spectrogram with various resolutions, each as one channel of the CNN. Besides spectrogram, other feature representations such as spectral difference, spectral flux and group-delay function are also widely-used in general-purpose onset detection [14].

Unlike onset detection, offset detection is seldom treated independently and is more often discussed in the context of note-level AMT [1, 3, 12]. The study carried out in [15] focuses on different playing styles of string instruments and summarizes several relevant features, including spectral difference, signal RMS energy, pitch confidence values, and pitch change, etc.

Previous methods in singing voice transcription widely adopt state-space machines to accomplish onset detection, pitch tracking, and offset detection in a single workflow. For example, the Tony software [16] uses an HMM containing three states, namely attack, stable, and silent, to characterize the temporal dynamics of a note sequence. The only allowed transition rules between these states are: 1) from attack to stable, 2) from stable to silent, and 3) from silent to attack of another note. However, these rules are oversimplified from real cases; for instance, an offset event is not always equivalent to a transition into the silent state. Rather, some offset events are followed immediately by the attack state of another consecutive note, which sometimes has the same pitch as the previous one. As a result, consecutive notes are merged and needs to be resolved by post-processing.

Recently developed note-level AMT methods utilizing deep learning has gained tremendous improvement, especially in offset detection. It is notable that in these methods, offset or onset detection sub-modules are optimized with more than one objective functions. Elowsson used two separate networks to learn 1) the offset curve, which outputs one at the instance of note offset, 2) the offset detection activation, which turns from zero to one when a note offset event turns into silence, and combined the results to describe offset events [1]. Hawthorne *et al.* used time-dependent object functions to infer the attack and decay of a musical note. These methods shed light on the note tracking of singing voice [12].

The above discussion inspires us two ways for improving singing voice segmentation. First, the objective functions can be designed to rely not merely on the onset and offset labels, but on an state space that describes all possible state transitions in a note sequence. Second, given the flexibility of neural network models, one may augment all the data representations related to onset/offset into the network to enhance the optimization process. The two ideas will be discussed in Section 3.1 and 3.2 respectively.

3. METHOD

Following previous discussion, we discuss the frame-wise onset and offset detection framework shown in Figure 1: for every time instance t, the hierarchical classifier predicts a set of labels y_t containing onset and offset information from a local feature representation $\mathbf{R}_t$. Note transcription is done by integrating pitch contour information.

3.1 Hierarchical classification for note segmentation

We consider the following states in a note sequence: silence (S), activation (A), and transition (T), where transition is further divided into two states, onset (O) and offset (X). When a transition (i.e. onset or offset) occurs, there are three possible *transition behaviors* of state evolution: $S \rightarrow T \rightarrow A$ where T represents an onset (O), $A \rightarrow T \rightarrow S$ where T represents an offset (X), and $A \rightarrow T \rightarrow A$ where T in this case contains an offset followed immediately by the onset of another note (XO). In other words, there is an important case that *an onset and an offset are presumably overlapped*. This fact motivates us to define such a state space that can encompass more general cases. As a result, there is a hierarchical taxonomy of these states, as shown in Figure 2 (c). See the caption of Figure 2 for more detailed information.

To investigate the behavior of this state space, we introduce several baselines and the proposed hierarchical classification model altogether to highlight the advantage of the proposed model in onset and offset classification.

1) First, we consider the note segmentation model consisting of two independent classifiers, one for onset detection and the other for offset detection. The 2-D onset label $y_{\text{on}} := [O, \overline{O}]$ is one-hot, where O represents the onset state while $\overline{O}$ represents the non-onset state. That means, $y_{\text{on}} = [1, 0]$ for onset and $y_{\text{on}} = [0, 1]$ for non-onset. Similarly, we have the offset label $y_{\text{off}} := [X, \overline{X}]$. Let the prediction of the two networks be $\hat{y}_{\text{on}}$ and $\hat{y}_{\text{off}}$, the model

is optimized by the following two objective functions:

$$L_{\text{on}}(y_{\text{on}}, \hat{y}_{\text{on}}) = \text{BCE}(y_{\text{on}}, \hat{y}_{\text{on}}) \,, \quad (1)$$

$$L_{\text{off}}(y_{\text{off}}, \hat{y}_{\text{off}}) = \text{BCE}(y_{\text{off}}, \hat{y}_{\text{off}}) \,. \quad (2)$$

where BCE is the binary crossentropy. This model is denoted as the on-off network (OON) model, and its taxonomy is illustrated in Figure 2 (a). Note that one tree represents one objective function, and every siblings form a regularization term in an objective function.

2) The onset and offset detection tasks share the same network, but with two task-specific layers, one for onset and the other for offset. The output label $y := [y_{\text{on}}, y_{\text{off}}]$ therefore has four dimensions. The total loss function is

$$L_{\text{M-OON}}(y, \hat{y}) := \text{BCE}(y_{\text{on}}, \hat{y}_{\text{on}}) + \text{BCE}(y_{\text{off}}, \hat{y}_{\text{off}}) \quad (3)$$

This model is denoted as the merged on-off network (M-OON) model hereafter.

3) The onset and offset are described implicitly by the three output states S, A, and T from a shared network. That means, the network outputs a multi-hot 3-D vector $y_{\text{tri}} := [S, A, T]$, where S, A and T are values between 0 and 1. The total loss function is

$$L_{\text{TSN}}(y, \hat{y}) := \text{BCE}(y_{\text{tri}}, \hat{y}_{\text{tri}}) \quad (4)$$

After obtaining the likelihood of S, T, A at every time instance t, we may follow the *transition behaviors* mentioned above to determine a T state to be an onset or an offset; the details can be found in Section 3.4. This model will be denoted as the tri-state notwork (TSN) model, and its taxonomy tree is constructed following Figure 2 (b).

Note that it is also possible to use categorical crossentropy rather than BCE in (4). However, using BCE allows possible overlapping of different states and therefore more flexibility for the model. Our pilot study also shows that using BCE achieves better performance.

4) We further consider the hierarchical structure that T can be onset, offset or an overlap of onset and offset. The output label is then a six-dimension space $y := [S, A, O, \overline{O}, X, \overline{X}]$, and the total objective function is:

$$\begin{aligned} L_{\text{HCN1}}(y, \hat{y}) := \text{BCE}(y_{\text{tri}}, \hat{y}_{\text{tri}}) \\ + \text{BCE}(y_{\text{on}}, \hat{y}_{\text{on}}) + \text{BCE}(y_{\text{off}}, \hat{y}_{\text{off}}) \end{aligned} \quad (5)$$

where we define the likelihood of the transition state as $T := \max(O, X)$. That means, if one of O or X is higher than a threshold (0.5 in the logistic regression case), then the state will be also predicted as T. The taxonomy tree of this case is illustrated in Figure 2 (c).

Finally, since T is in minority, optimizing the term $\text{BCE}(y_{\text{tri}}, \hat{y}_{\text{tri}})$ would suffer from data imbalance. To mitigate this issue, we enhance the *activity* classification between S and A by adding a new set of labels $y_{\text{act}} := [S, A]$, to enforce the output that only one of S and A would have high likelihood. The total objective function is then

$$\begin{aligned} L_{\text{HCN2}}(y, \hat{y}) := \text{BCE}(y_{\text{tri}}, \hat{y}_{\text{tri}}) + \text{BCE}(y_{\text{act}}, \hat{y}_{\text{act}}) \\ + \text{BCE}(y_{\text{on}}, \hat{y}_{\text{on}}) + \text{BCE}(y_{\text{off}}, \hat{y}_{\text{off}}) \end{aligned} \quad (6)$$

For clarity, (5) is denoted as the hierarchical classification network 1 (HCN1) model and (6) is denoted as the the hierarchical classification network 2 (HCN2) model.

3.2 Data representations

Based on the discussion in [15], we consider the spectral differences and the pitch salience representation in as the input of the proposed model. Given the input audio signal $\mathbf{x} := \mathbf{x}[n]$, where n is the time index. Let the amplitude part of the short-time Fourier transform (STFT) of $\mathbf{x}$ be $\mathbf{X}$. The forward spectral difference $\mathbf{S}^+$ and the backward spectral difference $\mathbf{S}^-$ are the time-forward and the time-backward differences of two neighbouring spectra in $\mathbf{X}$, as shown in the followings:

$$\mathbf{S}^+ = \text{ReLU}\left(\mathbf{X}[k, n+1] - \mathbf{X}[k, n-1]\right), \quad (7)$$

$$\mathbf{S}^- = \text{ReLU}\left(\mathbf{X}[k, n-1] - \mathbf{X}[k, n+1]\right), \quad (8)$$

where $\text{ReLU}(\cdot)$ represents the element-wise rectified linear unit: $\text{ReLU}(x) = x$ if $x > 0$, and 0 otherwise. That means, we split the first-order temporal difference of the spectrogram $\mathbf{X}$ into two channels, one is the part with positive temporal difference, and the other one is with negative temporal difference.

For the pitch saliency feature of $\mathbf{x}$, we adopt the one proposed in the combined frequency and periodicity (CFP) approach, which combines a frequency-domain feature indicating its fundamental frequency (f_0) and harmonics (nf_0), in a time-domain feature revealing its f_0 and sub-harmonics (f_0/n) to form a succinct, localized pitch feature with suppressed harmonic and sub-harmonic peaks [21, 28]. The feature is computed with the following process. Given a DFT matrix $\mathbf{F}$, high-pass filters $\mathbf{W}_f$ and $\mathbf{W}_t$, and activation functions σ_i, we consider three features, namely, spectrogram $\mathbf{Z}_0$, generalized cepstrum (GC) $\mathbf{Z}_1$, and generalized cepstrum of spectrum (GCoS) $\mathbf{Z}_2$:

$$\mathbf{Z}_0[k, n] := \sigma_0\left(\mathbf{W}_f \mathbf{X}\right) , \quad (9)$$

$$\mathbf{Z}_1[q, n] := \sigma_1\left(\mathbf{W}_t \mathbf{F}^{-1} \mathbf{Z}_0\right) , \quad (10)$$

$$\mathbf{Z}_2[k, n] := \sigma_2\left(\mathbf{W}_f \mathbf{F} \mathbf{Z}_1\right) . \quad (11)$$

The index k in $\mathbf{Z}_0$ and $\mathbf{Z}_2$ is frequency, while the index q in $\mathbf{Z}_1$ is called *quefrency*, which has the same unit as time. The nonlinear activation function is defined as a rectified and root-power function $\sigma_i(\mathbf{Z}) = |\text{ReLU}(\mathbf{Z})|^{\gamma_i}$, where $i = 0, 1, 2 \cdots$, $0 < \gamma_i \leq 1$, and $|\cdot|^{\gamma_0}$ is an element-wise root function. $\mathbf{W}_f$ and $\mathbf{W}_t$ are two high-pass filters designed as diagonal matrices used to remove slow-varying portions, where $\mathbf{W}_f$ applies cutoff frequency k_c and $\mathbf{W}_t$ applies cutoff quefrency q_c. In this paper we set $k_c = 80$ Hz and $q_c = 1/800$ sec. Based on the CFP approach, unwanted harmonics and sub-harmonics can be suppressed by merging $\mathbf{Z}_1$ and $\mathbf{Z}_2$ together. Note that $\mathbf{Z}_1$ should be mapped into the frequency domain because it is in the quefrency domain. Hence, we apply two sets of filter banks, both of which contain 174 triangular filters ranging from 80 Hz to 1000 Hz and with 48 bands per octave, respectively in the time and frequency domains. More specifically, the mth filter in frequency (or time) takes the weighted sum of the components whose frequency (or period) is between 0.25 semitones above and below the frequency at $f_m = 80 \times 2^{(m-1)/48}$ Hz (or the period at $1/f_m$

seconds). The filtered representations $\tilde{\mathbf{Z}}_1$ and $\tilde{\mathbf{Z}}_2$ are then both in the time-pitch scale. The CFP representation $\mathbf{Z}$ is

$$\mathbf{Z}[p,n] = \tilde{\mathbf{Z}}_1[p,n]\tilde{\mathbf{Z}}_2[p,n]\,, \tag{12}$$

where p is the pitch index. Details and source codes of computing the CFP representations can be found in [27].

In this work, the audio recordings are resampled to 16 kHz and are merged into mono-channel. Following [5], the input features are of multiple resolution.We compute $\mathbf{S}^+$, $\mathbf{S}^-$, and $\mathbf{Z}$ using the Hann window with 3 different sizes of 186, 372, and 743 samples (i.e. 11.61, 23.22, and 46.44 ms), resulting in nine data representation. The hop size is 320 samples (i.e. 20 ms). In CNN, $\mathbf{S}^+$, $\mathbf{S}^-$, and $\mathbf{Z}$ form the three input channels, and in each channel the data representations with three different window sizes are concatenated together. In RNN, all the nine data representations are concatenated as the input.

3.3 Model

We investigate two networks that stand for two strategies in modeling note sequences: ResNet for image classification [13] and RNN with attention for sequence classification [2]. Denote the frame-level feature at the time instance t as $\mathbf{r}_t$. For every t, we take the sequence $\mathbf{R}_t := [\mathbf{r}_{t-k}, \mathbf{r}_{t-k+1}, \cdots, \mathbf{r}_t \cdots, \mathbf{r}_{t+k}]$ as the input of the model to predict the presence of onset and offset at t. We set $k = 9$ according to the optimal loss on the validation set. That means, the dimension of every input $\mathbf{R}_t$ is $(c, 174, 19)$ (for ResNet) or $(c * 174, 19)$ (for RNN with attention mechanism), where c represents the number of channels: if $\mathbf{S}^+$, $\mathbf{S}^-$, and $\mathbf{Z}$ are stacked as the input, then $c = 3$.

Our implementation of the ResNet model basically follows the ResNet-18 architecture in [13]. The network is composed of eight sub-networks, each of which has two convolutional layers. The convolutional layers mostly have kernel of size $(3, 3)$. Batch normalization is used after each convolutional layer. The spatial pooling process is done by using convolutional layers with stride of two. Shortcut paths link the feature maps by skipping every two convolutional layers. After the convolution stages, the feature maps are pooled by averaging, and then are mapped to the output space through fully connected layers. See [13] for the implementation details. The output format and the objective functions follow the discussion in Section 3.1.

The RNN with attention is composed of a three bidirectional long-short-term memory (BLSTM) [26] layers, an attention layer, and two fully connected layers. For the three-layer BLSTM, the dimension of every hidden unit is 150. The outputs of the BLSTM are weighted and summed by the $2k + 1$ attention weights derived from the hidden units of the last BLSTM layer [2]. Layer normalization is used to stabilize training and inference processes. The results are then fed into the two-layer fully-connected network, each with a dimension of 150 and 6. The output format and the objective functions of the model also follows the discussion in Section 3.1.

Each data representation is normalized to zero mean before fed into the model. The manual labels in the dataset are not always exact since the exact time of an onset/offset event is hard to determine [5]. To solve this issue, we extend the labels to a *tolerance window* δ that can allow uncertainty in the onset/offset time labels: if a frame is within $\delta = \pm 50$ms to the true label, the label is also set to 1. This δ value is chosen according to the evaluation convention of onset detection in MIREX. This can mitigate the issue of data imbalance. In this work, all the models are obtained after 80 epochs of training on an Nvidia TITAN Xp GPU, using the Adam optimizer with the learning rate of 0.001. The source code, supplementary materials, and listening examples are available at: `https://github.com/Itachi6912110/Hierarchical-Note-Segmentation`.

3.4 Post-processing and note segmentation

We employ a linear filter with impulse response $h(n) = [0.25, 0.5, 1, 0.5, 0.25]$ to smooth the predicted onset and offset sequences. Then we apply a threshold at 0.5 and a peak picking process on the sequences to determine possible onset and offset positions. At this stage, minor mismatches between the predicted onset and offset positions still remain. To ensure that every onset is followed by exactly one offset, additional procedures are used.

For the OON and the M-OON models, the procedure includes: 1) if there are two onsets having no offset between them, we insert an offset at the time when the second onset occurs; 2) if there are two offsets without any onset between them, we directly discard the second one.

For the TSN model, consistent segmentation results can be derived directly from the relationship among S, A and T, so there is no issue on onset/offset mismatching. Onsets and offsets are determined by the following steps: 1) obtain the peak positions of the predicted sequence of T; 2) sum over the likelihood values of S and A in every interval separated by those peaks obtained in 1). If the sum of S is higher than the sum of A, then the interval is determined to be S. Otherwise, the interval is determined to be A; 3) for every selected T in 1), if its left-side interval is S and its right-side interval is A, a $S{\rightarrow}T{\rightarrow}A$ pattern is detected and the transition is determined as an onset. Conversely, if we detect a $A{\rightarrow}T{\rightarrow}S$ pattern, the transition is determined as an offset; 4) if we detect an $A{\rightarrow}T{\rightarrow}A$ pattern, the transition is determined as an offset and an onset; 5) if we detect a S$\rightarrow$T$\rightarrow$S pattern, the transition is directly discarded.

For HCN1 and HCN2, the procedure is a combination of the two strategies above: 1) if there are two onsets having no offset between them, we insert an offset specified to the time when S firstly surpasses A at that interval; 2) similarly, if there are two offsets having no onset between them, the inserted onset is specified to the time when A firstly surpasses S at that interval; 3) any detection violating the rules of 1) and 2) is deleted.

3.5 Note-level transcription

We combine the note segmentation method with a simple pitch estimation process for note-level singing voice transcription. This is implemented by: 1) obtain the onset and

offset times of each note with the note segmentation model, and 2) use the vocal melody extraction method in [27] to obtain the pitch contour of every note, and 3) the final pitch value is simply determined by the median of the pitch contour of that note.

4. EXPERIMENTS

4.1 Data and evaluation metrics

To test the robustness of our model, we set a cross-dataset scenario for the experiments on note segmentation. We use TONAS [10, 19], a dataset of 71 flamenco a cappella sung melody, as our training dataset. In addition, we evaluate our proposed method on the ISMIR2014 sung melody dataset [17]. It contains singing data from 11 female adults, 13 male adults and 14 children.

Section 4.2 first compares the results using different input features. Section 4.3 further compares the results of training with five different objective functions mentioned in Section 3.1. Section 4.4 then compares the ResNet-18 model, the RNN model with attention and the onset detector in the `MADMOM` library [6]. The latter is known as the state of the art for general-purpose onset detection.

For the evaluation metrics, we report the F1-scores of onset detection, offset detection and note transcription and the average overlap ratio (AOR) by using the utilities in the `mir_eval` library with default parameters [23]. To quantify the mismatch between the detected onsets and offsets in note segmentation results, we further compare their *conflict ratio* (CFR), which is defined as the ratio between the number of unpaired detection and the number of all predicted transitions (i.e. onsets plus offsets):

$$\text{CFR} := \frac{\text{\# of unpaired transitions}}{\text{\# of predicted transitions}} \tag{13}$$

The unpaired transition is defined as the onset/offset that cannot be derived from, or that violates the relationship of the states used in the model. For example, in the OON model, if there are two consecutive onsets having no offset in between, the second offset violates the relationship between onset and offset and is accounted as an unpaired detection. On the other hand, the TSN model produces zero unpaired transition and therefore has zero CFR, as discussed in Section 3.4. CFR can be seen as a criterion of systematic consistency for a note segmentation model.

4.2 Comparison of input features

The first five rows of Table 1 lists the results of both onset and offset detection with various inputs: $\mathbf{X}$, $\mathbf{S}^+$, $[\mathbf{S}^+, \mathbf{S}^-]$, $[\mathbf{S}^+, \mathbf{Z}]$, and $[\mathbf{S}^+, \mathbf{S}^-, \mathbf{Z}]$. In comparison to others, using only the spectrogram ($\mathbf{X}$) with less feature engineering gives competitive result, which indicates the power of ResNet in pattern recognition. However, it should be emphasized that using a detailed set of features relevant to onset and offset such as $[\mathbf{S}^+, \mathbf{S}^-, \mathbf{Z}]$ achieves the best note transcription F1-score at 59.5%, which is better than the case using only $\mathbf{X}$ by 3.9%. Such improvement can be seen from other interesting comparisons. For example, adding either $\mathbf{S}^-$ or $\mathbf{Z}$ to $\mathbf{S}^+$ greatly improves the F1-scores of both the onset and offset. Adding $\mathbf{S}^-$ to $\mathbf{S}^+$ also results in 14.5% improvement on onset F1-score, meaning that the backward spectral difference may also be relevant to an onset event. These observations can all be explained by the fact that an onset event can be highly overlapped by an offset event of another notes, and the feature set revealing different aspects of the signal characteristics helps resolve such ambiguity. For simplicity, we adopt $[\mathbf{S}^+, \mathbf{S}^-, \mathbf{Z}]$ in the following experiments.

4.3 Comparison of objective functions

The lower part of Table 1 compares the results of models trained by four baseline objective functions, including OON, M-OON, TSN, and HCN1. Comparing the F1-scores of OON and M-OON, we observe that M-OON slightly degrades onset detection but greatly improves offset detection by 29.2%. This indicates the importance of joint training: incorporating onset information in a shared network can help offset detection.

Although the F1-score of TSN is worse than the one of M-OON, TSN achieves zero CFR as all onsets/offsets can be completely inferred from the rule mentioned in section 3.1 and 3.4. This shows that training on S, A, T and the temporal constraints make highly consistent prediction. However, the poor performance on onset and offset detection implies that using a single T state is not sufficient to describe the behavior of both onset and offset.

HCN1 and HCN2 therefore combine the advantage of both the M-OON model and TSN model. Result shows that the HCN1 model enhances the segmentation quality (reducing CFR to half) compared to the M-OON model and improves the onset and offset detection F1-score compared to the TSN model, then achieves the F1-score of 56.7% on note transcription. In addition, HCN2 model outperforms the HCN1 model in almost all evaluation metrics, where a 2.7% improvement on note transcription F1-score is obtained. Such advancement indicates the importance of regularizing activation/silence detection in note segmentation and transcription tasks.

4.4 Comparison of models

Table 1 also compares two implementations of HCN2 using different modules for the hierarchical classifier: ResNet-18, and the RNN with attention (denoted as RNN-attn) as a sequence classification network for comparison.

Results show that ResNet-18 outperforms RNN-attn in every performance metrics, probably because that an image-based classification network can extract more detailed features considering local information where sequential dependency is not that significant. These findings are partly in line with that in [25], where a CNN outperforms sequence models such as RNN.

4.5 Singing Voice Note Transcription

Table 2 shows the results of singing voice transcription compared with five previous methods: Ryynänen *et al.*

Objective	Classifier	Feature	F1 (onset)	F1 (offset)	CFR	AOR	P (note)	R (note)	F1 (note)
HCN2	ResNet-18	$\mathbf{S}^+$	0.599	0.409	0.078	0.862	0.430	0.394	0.409
		$\mathbf{X}$	0.757	0.740	0.050	0.873	0.576	0.538	0.555
		$[\mathbf{S}^+, \mathbf{S}^-]$	0.744	0.715	0.057	0.870	0.532	0.506	0.517
		$[\mathbf{S}^+, \mathbf{Z}]$	0.745	0.713	0.050	0.870	0.553	0.506	0.527
		$[\mathbf{S}^+, \mathbf{S}^-, \mathbf{Z}]$	0.786	**0.759**	0.043	0.869	**0.625**	**0.569**	**0.594**
	RNN-attn	$[\mathbf{S}^+, \mathbf{S}^-, \mathbf{Z}]$	0.699	0.722	0.050	0.840	0.520	0.502	0.510
HCN1	ResNet-18	$[\mathbf{S}^+, \mathbf{S}^-, \mathbf{Z}]$	0.751	0.739	0.051	0.872	0.608	0.535	0.567
TSN			0.691	0.705	**0.000**	0.864	0.472	0.480	0.474
M-OON			0.778	0.707	0.129	**0.874**	0.574	0.526	0.547
OON			**0.790**	0.415	0.210	0.846	0.313	0.305	0.308

Table 1: Evaluation results for various input features objective functions, and classifier models.

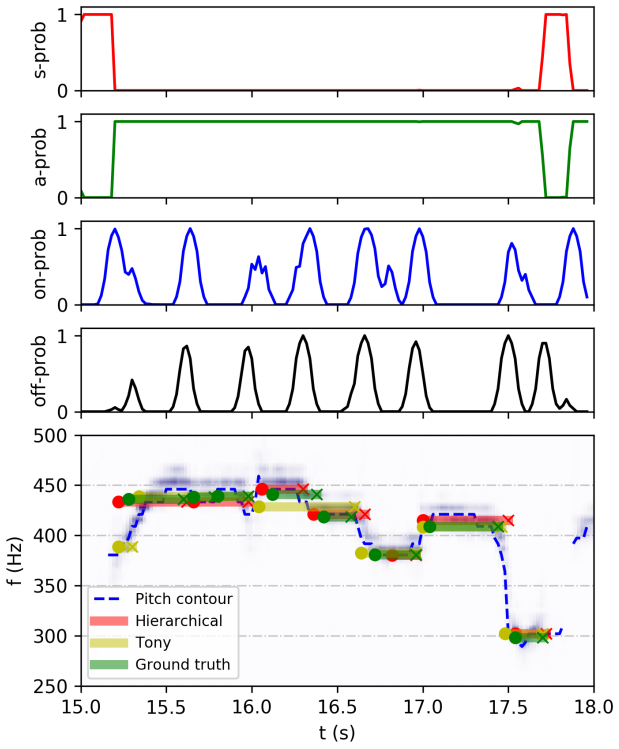

Figure 3: Transcription results from the 15th to the 18th second of 'child10.wav' in the ISMIR 2014 dataset. From top to bottom: predicted likelihood for S, A, O, X, and transcription results. Background of the bottom subfigure: the pitch saliency function $\mathbf{Z}$. Blue dashed lines: estimated pitch contour. Bullet: onset time. X mark: offset time.

Method	Precision	Recall	F
Ryynänen [24]	0.304	0.315	0.308
Gómez & Bonada [10]	0.430	0.373	0.398
SiPTH [18]	0.397	0.440	0.415
Yang [29]	0.409	0.436	0.421
Tony [16]	0.510	0.534	0.520
Proposed	**0.625**	**0.569**	**0.594**

Table 2: Comparison of singing transcription results.

[24] , Gómez & Bonada [10], SiPTH [18], Yang *et al.* [29], and Tony [16]. The results for these five methods are reported in [29]. Our proposed method outperforms all the previous methods by more than 7.4% in terms of the F1-measure. It is important to note that although our model is trained on a dataset with the singing style (flamenco singing) quite different from the testing data, the model still outperforms the Tony software, which performance is actually based on a parametric grid search on the testing dataset [16]. This fact indicates that our method is potentially generalizable over various data modalities. Besides, since we do not directly deal with issues such as vibrato, unstable pitches and tuning shift [29], our model actually benefits more from a stable note segmentation method. This highlights the importance of note segmentation in note transcription.

Fig. 3 illustrates an example of the predicted silence, activation, onset, offset likelihood curves and note transcription results of a clip in the testing dataset. The transcription result from the Tony software is also provided for comparison. It can be shown that Tony tends to miss onsets for consecutive notes, while the proposed model successfully captures almost all the note transitions except the onset at 16.71 sec, which is a challenging case due to the bent pitch contour around the onset event and a relatively short note duration.

5. CONCLUSION

We have presented the effectiveness of the proposed hierarchical classification networks in note segmentation and transcription in singing voice. By unfolding the structure of the state evolution patterns in note sequences and by applying multi-channel data representations to modeling note transitions, the general, robust, and consistent note segmentation procedure plays a vital role in achieving state-of-the-art performance. One important aspect omitted in our discussion is using temporal modeling (e.g., HMM) over the hierarchical state space rather than using post-processing rules to complete the note transcription process. Based on the positive result of this study, this direction is with high potential and will be left as future work.

6. ACKNOWLEDGEMENT

This work is partially supported by the MOST of Taiwan under Grant No. 106-2218-E-001-003-MY3.

7. REFERENCES

[1] E. Anders. *Modeling Music: Studies of Music Transcription, Music Perception and Music Production*. PhD thesis, KTH Royal Institute of Technology, 2018.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[3] E. Benetos and S. Dixon. Polyphonic music transcription using note onset and offset detection. In *Proc. IEEE ICASSP*, pages 37–40. IEEE, 2011.

[4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *J. Intelligent Information Systems*, 41(3):407–434, 2013.

[5] S. Böck, A. Arzt, F. Krebs, and M. Schedl. Online real-time onset detection with recurrent neural networks. In *Proc. DAFx*, 2012.

[6] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In *Proc. ACM MM*, pages 1174–1178, 2016.

[7] P. Brossier, J. P. Bello, and M. D Plumbley. Real-time temporal segmentation of note objects in music signals. In *Proceedings of ICMC 2004, the 30th Annual International Computer Music Conference*, 2004.

[8] S. Chang and K. Lee. A pairwise approach to simultaneous onset/offset detection for singing voice using correntropy. In *Proc. IEEE ICASSP*, pages 629–633, 2014.

[9] F. Eyben, S. Böck, B. Schuller, and A. Graves. Universal onset detection with bidirectional long-short term memory neural networks. In *ISMIR*, pages 589–594, 2010.

[10] E. Gómez and J. Bonada. Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, 37(2):73–90, 2013.

[11] R. Gong and X. Serra. Singing voice phoneme segmentation by hierarchically inferring syllable and phoneme onset positions. In *Interspeech*, pages 716–720, 2018.

[12] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck. Onsets and frames: Dual-objective piano transcription. In *ISMIR*, pages 50–57, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] A. Holzapfel, Y. Stylianou, A. C Gedik, and B. Bozkurt. Three dimensions of pitched instrument onset detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1517–1527, 2010.

[15] C.-Y. Liang, L. Su, Y.-H. Yang, and H.-M. Lin. Musical offset detection of pitched instruments: The case of violin. In *ISMIR*, pages 281–287, 2015.

[16] 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 *Proc. SMC*, 2015.

[17] E. Molina, A. M. Barbancho-Perez, L. J. Tardón, I. Barbancho-Perez, et al. Evaluation framework for automatic singing transcription. 2014.

[18] 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*, 23(2):252–263, 2015.

[19] J. Mora, F. Gómez, E. Gómez, F. Escobar-Borrego, and J. M. Díaz-Báñez. Characterization and melodic similarity of a cappella flamenco cantes. In *ISMIR*, pages 351–356, 2010.

[20] R. Nishikimi, E. Nakamura, K. Itoyama, and K. Yoshii. Musical note estimation for F0 trajectories of singing voices based on a bayesian semi-beat-synchronous hmm. In *ISMIR*, pages 461–467, 2016.

[21] G. Peeters. Music pitch representation by periodicity measures based on combined temporal and spectral representations. In *Proc. IEEE ICASSP*, pages 53–56, 2006.

[22] M. Pesek, A. Leonardis, and M. Marolt. Robust real-time music transcription with a compositional hierarchical model. *PloS one*, 12(1), 2017.

[23] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel. mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, pages 367–372, 2014.

[24] M. P. Ryynänen and A. P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.

[25] J. Schlüter and S. Böck. Improved musical onset detection with convolutional neural networks. In *Proc. ICASSP*, pages 6979–6983, 2014.

[26] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[27] L. Su. Vocal melody extraction using patch-based cnn. In *Proc. IEEE ICASSP*, pages 371–375, 2018.

[28] L. Su and Y.-H. Yang. Combining spectral and temporal representations for multipitch estimation of polyphonic music. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 23(10):1600–1612, 2015.

[29] L. Yang, A. Maezawa, J. B. Smith, and E. Chew. Probabilistic transcription of sung melody using a pitch dynamic model. In *Proc. IEEE ICASSP*, pages 301–305, 2017.