VOCANO: A NOTE TRANSCRIPTION FRAMEWORK FOR SINGING VOICE IN POLYPHONIC MUSIC

[1]**Jui-Yang Hsu** [2]**Li Su**
[1]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan
B05901022@ntu.edu.tw, lisu@iis.sinica.edu.tw

ABSTRACT

High variability of singing voice and insufficiency of note event annotation present a huge bottleneck in singing voice transcription (SVT). In this paper, we present VOCANO, an open-source VOCAl NOte transcription framework built upon robust neural networks with multi-task and semi-supervised learning. Based on a state-of-the-art SVT method, we further consider virtual adversarial training (VAT), a semi-supervised learning (SSL) method for SVT on both clean and accompanied singing voice data, the latter being pre-processed using the singing voice separation (SVS) technique. The proposed framework outperforms the state of the arts on public benchmarks over a wide variety of evaluation metrics. The effects of the types of training models and the sizes of the unlabeled datasets on the performance of SVT are also discussed.

1. INTRODUCTION

Singing voice transcription (SVT), the task to map singing voice to common music notation of note events, is a critical step to drive novel applications in music retrieval, content creation, musicology, education, and human-computer interaction [1]. Similar to many of the automatic music transcription (AMT) tasks, the SVT task typically encompasses several sub-tasks of AMT, which are pitch detection, onset detection, offset detection, as well as sequence-level modeling [2, 3]. In the literature of music information retrieval (MIR), one of the most extensively investigated sub-tasks of SVT might be *vocal melody extraction*, the task to transcribe the *frame-level* instantaneous pitch (i.e., fundamental frequency (F0)) contours of singing voice in either monophonic (no accompaniment) or polyphonic (mixed with accompaniment) audio signals. Specifically, recent endeavours mostly focus on leveraging deep learning techniques to transcribe the singing voice which serves as a predominant melody in polyphonic music [4–7]. Though achieving breakthrough performance, vocal melody extraction is however not yet a complete solution of SVT, as it does not specify *note-level* events distinct from its outputs rendered in time frames.

The challenges of note-level SVT are multi-fold. Vocal signals are highly variable in singing timbre, articulation, intonation, discernible patterns such as *vibrato*, *glissando*, note transitions and ornaments, and even lyrics. These variables blurs the boundaries between notes and notes, and make the note-level data annotation of singing voice an extremely challenging job. It seems to be impossible to compile large-scale, accurate and consistent human-annotated datasets, especially on note transition (e.g., onset and offset) time. Such variability also challenges a model to discriminate local time-frequency patterns. For example, an offset event can be overlapped with another onset event to a flexible overlapping ratio, making the transition a non-Markovian process [8]. This issue is even worsen in polyphonic music in which the note transitions in accompaniments are much denser than in the vocal melody.

In this paper, we propose a novel SVT framework to address these issues. We notice that, as the annotated datasets are limited, using advanced regularized neural networks against overfitting, and semi-supervised learning (SSL) to leverage massive amounts of unlabeled data emerge as an efficient solution. Based on the hierarchical classification approach of transcription [8], we utilize the PyramidNet with ShakeDrop regularization to reduce overfitting [9], and also incorporate it with virtual adversarial training (VAT) [10] for SSL. These techniques have been found useful in the fields of computer vision, while their potential on MIR tasks has not been thoroughly discussed.

The major technical novelty and contribution of this paper are as follows. First, to the best of our knowledge, this paper represents one of the first implementations of note-level SVT considering mixture audio inputs. Second, the proposed SVT method outperforms state-of-the-art methods. Also, the effect of SSL mechanism together with the model choice on SVT performance are discussed. Section 2 will give an overview and paper survey on the SVT problem scenario that will be discussed in this paper. Method and experiment results will be given in Section 3 and 4, respectively. Conclusion will be made in Section 5.

2. PROBLEM SCENARIOS AND BACKGROUND

The problem scenario of SVT is not consistently defined in the literature. First, the transcription results can be in either frame-level (e.g., vocal melody extraction) or note-level. Second, the input data can be either monophonic singing or with accompaniment. Third, the target of transcription can be either solo voice or multiple concurrent voices (e.g., choir). In this work, we consider SVT of a single voice without or with instrument accompaniment, which will be referred to as the monophonic or polyphonic SVT later on. There are two approaches to deal with the case when the accompaniment is present: 1) train a general SVT model using the singing voice data mixed with accompaniment, and 2) train a specialized SVT model using clean singing voice data, and use singing voice separation (SVS) tools to remove accompaniments of the input before inference. In the second approach, monophonic and polyphonic SVT can be regarded as the same task, based on the fact that SVS is a relatively well developed technology. For simplicity, we will focus on the second approach in this paper. A pilot study comparing the two approaches will also be reported in Section 4.3.

Previous note segmentation works on SVT usually employ state-space machines such as Bayesian models or 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 [11–14]. Tony [13], a widely-used note transcription software, is also based on this approach. In the deep learning approach, the connectionnist temporal classification (CTC) loss [15], self-attention mechanism [3] also play similar roles in temporal decoding of note-level SVT. However, it has been pointed out that the state space in note transition can be ambiguous in several cases when onset and offset events are overlapped or when note pitch are repeated, and this issues can be solved by extending the output dimension of the network to describe the different classes of transition states [8]. Similar ideas such as multi-state note models have also been discussed recently in piano AMT tasks [16].

To our knowledge, a note-level SVT tool specifically for the singing voice signals mixed with accompaniment has rarely been implemented. It is not until 2020 that the task of "Singing Transcription from Polyphonic Music" was proposed in Music Information Retrieval Evaluation eXchange (MIREX), while there was only one submission to this campaign. [1] Related tasks include singing voice separation (SVS) [6,17] and automatic transcription of multiple concurrent singing voices such as *a cappella* [18,19], most of which are restricted to frame-level transcription.

3. METHOD

The proposed SVT framework is shown in Figure 1. In the training stage, data representations are extracted for each frame, and are then fed into two neural network models, one for pitch contour extraction and the other for note segmentation. SSL is performed on the note segmentation network. Note-level transcription results are obtained through a temporal decoding process over the frame-level outputs.

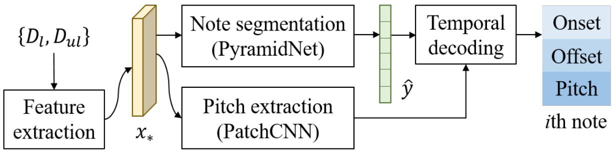


Figure 1. The proposed SVT framework. D_l, D_{ul}, x, and $\hat{y}$ represent labeled dataset, unlabeled dataset, input sample and predicted label, respectively.

3.1 Data representation

All the input audio signals are sampled at 16 kHz. For all the polyphonic data (e.g., singing plus accompaniment), an SVS algorithm, Demucs [17], is employed to separate out singing voice for use before the feature extraction stage.

Following the previous state-of-the-art method [8], the input of the SVT network is a multi-channel feature consisting in spectrum, generalized cepstrum and the generalized cepstrum of spectrum (GCoS) [20]. Such a combination has been shown effective in enhancing F0 components while suppressing unwanted harmonic components [20]. All channels of feature are mapped into the log-frequency scale with a filterbank containing 174 overlapped triangular filters allocated from 80 Hz to 1 kHz with 48 bins per octave. To adapt to signal-level attributes in different resolution, three windows with different sizes are employed to compute these data representations. As a result, the input feature has 9 channels. For every time step at t, the input $x(t)$ contains the data representations at frame t and also at its previous and future 9 frames, totaling 19 frames. In other word, the shape of $x(t)$ is: (number of channel, height, width) := (9, 174, 19).

3.2 Note segmentation networks

We decompose the SVT process into two parts: frame-level pitch extraction and note segmentation. For pitch extraction, we directly use a vocal melody extraction network, Patch-CNN [21], to obtain frame-level pitch contours. Since the frame-level pitch extraction has been a widely investigated technique (see discussion on vocal melody extraction in Section 1), the proposed network therefore focuses on note segmentation.

The note segment network can be regarded as an re-implementation and extension of [8]. First, while the model in [8] concatenates the 9 data representations as a single-channel inputs, in this work we reshape them into 9 individual channels, as shown in Section 3.1. Second, while [8] was based on the ResNet-18 network [22], we instead consider the PyramidNet with ShakeDrop regularization [23] to reduce overfitting. The PyramidNet improves the performance by gradually increasing the numbers of feature maps through the layers such as to effectively increase the diversity of high-level attributes [9]. ShakeDrop regularization further diversifies the feature maps by assigning different random weights in forward and backward

[1] `https://www.music-ir.org/mirex/wiki/2020:Singing_Transcription_from_Polyphonic_Music_Results`

stages at each residual layer [23]. In this work, we adopt the PyramidNet-110 architecture, which has 28.49M parameters, a size larger than ResNet-18 by 2.5 times.

Following [8], the output of the note segmentation network is optimized with multiple sub-tasks to capture the complex dynamics of music note transition. For each time step, the network outputs a 6-dimensional vector $\hat{y} := \left[s, a, o, \bar{o}, f, \bar{f}\right]$, where s represents the silence state, a represents the activation (i.e. a note is on) state, o represents the onset state, and f represents the offset state. $\bar{o}$ and $\bar{f}$ are the non-onset and non-offset states, respectively. The output at time t is denoted as $\hat{y}(t)$, in which the states are denoted as $s(t)$, $a(t)$, and so on. Each state in $\hat{y}$ represents a probability value between zero and one, and we simply set $\bar{o} := 1 - o$, $\bar{f} := 1 - f$, and $s = 1 - a$. We also define the transition state $t := \max(o, f)$ to describe the state that either an onset of an offset occur. Defining the subspaces $\hat{y}_{\text{tri}} := [s, a, t]$, $\hat{y}_{\text{act}} := [s, a]$, $\hat{y}_{\text{on}} := [o, \bar{o}]$, $\hat{y}_{\text{off}} := [f, \bar{f}]$, then the total objective function for note segmentation is

$$\begin{aligned}\mathcal{L}_{\text{SEG}}(y, \hat{y}) := \text{BCE}(y_{\text{tri}}, \hat{y}_{\text{tri}}) + \text{BCE}(y_{\text{act}}, \hat{y}_{\text{act}}) \\ + \text{BCE}(y_{\text{on}}, \hat{y}_{\text{on}}) + \text{BCE}(y_{\text{off}}, \hat{y}_{\text{off}}),\end{aligned} \quad (1)$$

where y, y_{tri}, y_{act}, y_{on}, and y_{off} are the ground truth, and BCE is binary cross-entropy. In brief, the note segmentation mechanism here is not merely to classify onset and offset individually, but is a combination of four classification sub-tasks over the subspaces in the output space y: one sub-task of multi-class classification over transition, activation, and silence, and three sub-tasks of binary classification (i.e. activation/silence, onset/non-onset, and offset/non-offset). Such design facilitates the discrimination between possibly overlapped events, such as onset and offset (when the offset of a note followed by the onset of its next note) and smooth note transition.

3.3 Semi-supervised learning

The Virtual Adversarial Training (VAT) technique is used for semi-supervised learning on both the labeled and unlabeled training data. VAT can be regarded as an effective data/ label augmentation technique without the needs of prior domain knowledge. Let x_l and x_{ul} be labeled and unlabeled samples sampled from a labeled dataset $\mathcal{D}_l$ and and unlabeled dataset $\mathcal{D}_{ul}$, respectively. Given a sample x_* which is either x_l or x_{ul}, the output distribution can be represented as $p(y|x, \theta)$, in which θ represents the parameters of the note segmentation model. In our case, VAT aims at minimizing the below local distributional smoothness (LDS) function for every x_*:

$$\begin{aligned}\text{LDS}(x_*, \theta) =& \text{BCE}\left(p(y|x_*, \theta), p(y|x_* + r_{\text{adv}}, \theta)\right), \quad (2) \\ r_{\text{adv}} :=& \underset{r; \|r\|_2 < \epsilon}{\arg\max}\, \text{BCE}\left(p(y|x_*, \theta), p(y|x_* + r)\right).\end{aligned}$$

Let N_l and N_{ul} are the number of samples in $\mathcal{D}_l$ and $\mathcal{D}_{ul}$, respectively, we have the total VAT loss being $\mathcal{L}_{\text{VAT}} := 1/(N_l + N_{ul}) \sum_{x \in \mathcal{D}_l \cup \mathcal{D}_{ul}} \text{LDS}(x_*, \theta)$. Combined with the supervised loss function (Equation (1)), the total loss function is represented as $\mathcal{L} := \mathcal{L}_{\text{SEG}} + \lambda \mathcal{L}_{\text{VAT}}$, and we set $\lambda = 1$ throughout this work. The note segmentation network is implemented with PyTorch v1.5, and is obtained after 20 epochs of training on an Nvidia TITAN RTX GPU, using the AdamW optimizer with a learning rate of 10^{-4}. Typically, it takes around 8 hours to accomplish training a model.

3.4 Temporal decoding

Post-processing is needed to derive temporally consistent onset/ offset/ activation timestamps from the 6-D distribution (i.e. $\hat{y}(t)$) outputted from the network. We call this process *temporal decoding*. First, we employ a linear filter with impulse response as a 5-tap triangular window to smooth each dimension in $\hat{y}(t)$ in the time axis. Then, we perform peak picking on $\hat{o}(t)$ and $\hat{x}(t)$ with a threshold at 0.5 to determine possible onset and offset positions, respectively. At this stage, there are inevitable mismatches between the predicted onset and offset positions. To ensure that every onset is followed by exactly one offset, additional procedures are used: 1) if there are two onsets having no offset between them, we insert an offset specified to the time when s firstly surpasses a with 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 in that interval; and 3) any predicted result violating rules 1) and 2) is removed and is not recognized as an onset or an offset.

After having the onset-offset interval of every predicted note, the pitch of every note is determined by the median value of the pitch contour within that onset-offset interval.

4. EXPERIMENT

4.1 Data

To test the robustness of our model, a cross-dataset scenario (i.e. the training and testing datasets are compiled independently) is employed for the experiments. The dataset used for supervised learning (denoted as $\mathcal{D}_l$) is the TONAS dataset, which contains 71 flamenco a cappella sung melody, each of which has high-quality note-level annotation [24]. We consider three datasets for semi-supervised learning (denoted as $\mathcal{D}_{ul}$), which are MIR-1K,[2] MedleyDB [25] and DALI [26]. MIR1K contains 1,000 excerpts of Chinese karaoke songs sung by amateur singers. MedleyDB is a multi-track dataset, and we select the tracks labeled as ‘female singer’ or ‘male singer’ (76 tracks in total) as our unlabeled training data. The DALI dataset contains a large-scale polyphonic music (mostly Western pop music). In this dataset We select 65 songs from this dataset as unlabeled training data, and this subset is denoted as DALI-train hereafter.

For evaluation, we consider three testing datasets (denoted as $\mathcal{D}_{test}$), which are ISMIR2014, DALI-test, and Cmedia. ISMIR2014 [27] is a monophonic vocal singing dataset containing singing data from 11 female adults, 13

[2] https://sites.google.com/site/unvoicedsoundseparation/mir-1k

male adults and 14 children. The DALI-test set, also selected from DALI, contains 20 songs with automated annotation of notes. Finally, the Cmedia dataset [28] is used in the MIREX campaign on polyphonic SVT (see footnote 1), and on the list we retrieve 99 pop songs (mostly Chinese songs) with vocal annotation publicly available. The list of the songs selected from DALI and Cmedia are provided on the project website (see Section 5).

4.2 Evaluation metrics

We use the metrics of note transcription in the `mir_eval` library for evaluation [29]. In the evaluation rules, a predicted note is considered as correct (i.e., true positive) for a ground truth note if it fulfills the three rules: 1) the difference in pitch number between the predicted note and the ground truth note is less than a pitch tolerance value δp (in cents), 2) the difference in onset time is less than an onset tolerance value δo (in seconds), and 3) the difference in offset time is less than $\max(\delta o, \delta f \times g)$, where δx is an offset tolerance ratio and g is the duration of the ground truth note (in seconds). The F1-score is the harmonic mean of the precision and recall values obtained from these criteria.

Therefore, the F1-score is parametrized by $(\delta p, \delta o, \delta f)$, and is denoted as $\mathcal{F}_{(\delta p, \delta o, \delta f)}$ in this paper. This incorporates several conventional metrics of note-level transcription. Setting $\delta p = 50$ cents, $\delta o = 50$ms and $\delta f = 0.2$, we consider the following F1-scores (a tolerance value of ∞ means that it is not consider in the evaluation):

- Onset-only F1-score: $\mathcal{F}_{(\infty, 0.05, \infty)}$
- Offset-only F1-score: $\mathcal{F}_{(\infty, \infty, 0.2)}$
- Onset-offset F1-score: $\mathcal{F}_{(\infty, 0.05, 0.2)}$
- Onset-pitch F1-score: $\mathcal{F}_{(50, 0.05, \infty)}$
- Onset-offset-pitch F1-score: $\mathcal{F}_{(50, 0.05, 0.2)}$

For example, $\mathcal{F}_{(50,0.05,0.2)}$ means that a note is considered as a true positive if its pitch deviates from the ground truth pitch by less than 50 cents, its onset deviates from the ground truth onset by less than 0.05s, and its offset deviation is less than 0.2 times the duration of the ground truth note. The F1-scores of only onset (or only offset) events are the cases of $\delta p = \delta f = \infty$ (or $\delta p = \delta o = \infty$).

Besides the note-level F1-scores, we also propose a high-level metric called the sequence-level Note Accuracy (NAcc), which is based on matching the MIDI pitches of predicted and ground truth note sequences rather than the timestamps of onset/ offset. More specifically, NAcc is the Levenshtein distance between the ground truth and the predicted MIDI sequences: $\mathrm{NAcc} := 1 - (D + I + S)/N$, where D denotes the number of deletions, I is the number of insertions, S is the number of substitutions, and N is the length of the ground truth sequence. Unlike the F1-score, NAcc can better reveal the performance on the entire pitch sequence, rather than the performance on the time stamps of note events. This evaluation is useful when accurate time stamps of the output are not of primary importance while the global information of pitch sequence is required.

4.3 Results

4.3.1 *Effect of singing voice separation*

First, as a pilot study, we compare the two SVT approaches for polyphonic audio mentioned in Section 2: 1) a model directly trained with polyphonic $\mathcal{D}_{ul}$, and 2) a model trained with SVS-processed (i.e. monophonic) $\mathcal{D}_{ul}$, and requiring SVS in the inference stage. Using MIR1K as $\mathcal{D}_{ul}$ and ISMIR2014 as $\mathcal{D}_{test}$, results show that the onset-offset-pitch F1-score is 30.04% for the first model, while the second model achieves 68.38%, a much better performance. This is mainly due to the domain difference between $\mathcal{D}_{l}$ and $\mathcal{D}_{ul}$ (the former is purely monophonic while the latter is polyphonic). We therefore focus on the second approach in evaluating the proposed SVT framework.

4.3.2 *Comparison of models*

Table 1 compares the performance metrics of two models (ResNet-18 and PyramidNet) trained under a supervised scheme (w/o VAT), and a semi-supervised schemes (w/i VAT) with three different unlabeled datasets ($\mathcal{D}_{ul}$) having different scales: MIR1K, MIR1K + MedleyDB, and also MIR1K + MedleyDB + DALI-train.

A comparison of the two models is first made from the left three columns of Table 1 (without VAT). ResNet-18 outperforms PyramidNet for onset F1-score, onset-pitch F1-score and NAcc, while PyramidNet prevails on offset detection and gives better onset-offset-pitch F1-scores on the three datasets. In short, PyramidNet, with a larger size of training parameters, performs better on strict note transcription metric such as onset-offset-pitch F1-score.

4.3.3 *Effects of semi-supervised learning*

By comparing the results without VAT and the ones with VAT, we observe two different trends for the two models. For PyramidNet, using VAT improves the performance for almost all $\mathcal{D}_{test}$ and all the metrics. For example, with MIR1K as $\mathcal{D}_{ul}$ improves the onset-offset-pitch F1-score of the three test datasets by 5.73, 0.08 and 3.01 percentage points, respectively. Since all the methods adopt the same pitch extraction results, such improvement is fully contributed by the improvement of note segmentation network with semi-supervised learning.

For ResNet-18, however, using VAT only improves the performance of offset-related metrics rather than all metrics. This is possibly because VAT performs more effectively on larger models with regularization mechanism. Among the improvement of offset detection metrics, it is worth mentioning that the offset F1-score of the ISMIR2014 dataset is improved by 3.83 percentage points (from 74.68% to 78.51%) with the $\mathcal{D}_{ul}$ being MIR1K+Med+DALI. In summary, although VAT does not improve the performance consistently over all types of models on all performance metrics, it still exhibits a trend to improve more challenging metrics such as offset.

4.3.4 *Effects of the unlabeled dataset $\mathcal{D}_{ul}$*

Table 1 also demonstrates that the size, quality, and diversity of the unlabeled dataset ($\mathcal{D}_{ul}$) affect the performance

	w/o VAT			w/i VAT								
$\mathcal{D}_{ul}$	–			MIR-1K			MIR-1K + MedleyDB			MIR1K + Med + DALI		
$\mathcal{D}_{test}$	I	D	C	I	D	C	I	D	C	I	D	C
	PyramidNet + ShakeDrop											
Onset-only	78.02	27.79	58.15	**84.04**	**32.87**	**64.56**	81.10	28.98	60.65	82.25	30.40	61.81
Offset-only	75.50	**36.32**	45.24	**80.06**	33.95	**51.86**	78.52	33.95	48.29	78.42	34.87	47.43
Onset-offset	62.92	11.16	25.95	**68.60**	**11.95**	**33.65**	67.06	10.85	29.97	67.12	11.13	29.65
On-off-pitch	62.65	2.69	22.36	**68.38**	**2.76**	**28.28**	66.73	2.65	25.37	66.92	2.68	25.05
Onset-pitch	75.32	5.26	45.57	**80.58**	**5.99**	**48.33**	78.26	5.35	45.95	78.72	5.50	47.28
NAcc	69.76	12.30	50.54	**78.68**	12.06	48.52	75.51	12.78	47.88	77.41	**15.43**	**52.36**
	ResNet											
Onset-only	**82.01**	**30.05**	**60.12**	79.39	26.70	55.87	78.85	26.71	55.80	78.38	26.68	55.68
Offset-only	74.68	**35.38**	45.31	76.76	33.23	44.18	76.11	33.32	46.94	**78.51**	33.29	**46.47**
Onset-offset	61.80	**10.46**	**26.60**	62.93	9.91	25.10	62.93	9.65	26.53	**63.32**	9.49	26.32
On-off-pitch	61.71	**2.51**	**22.95**	62.76	2.42	21.66	62.77	2.21	22.60	**63.04**	2.28	22.44
Onset-pitch	**77.97**	**5.89**	**47.87**	75.90	5.19	43.76	74.87	4.93	43.12	74.78	5.11	43.51
NAcc	**80.08**	**16.06**	**56.24**	73.29	4.11	43.08	74.20	11.17	48.28	78.16	10.79	48.38

Table 1. Evaluation results on three test datasets (I: ISMIR2014; D: DALI-test; C: Cmedia). The evaluation metrics are (from top to bottom): onset F1, offset F1, onset-offset F1, onset-offset-pitch (on-off-pitch) F1, and pitch-onset F1. See Section 4.2 for more details on the evaluation metrics. The best performances of each dataset are marked in bold. Upper: PyramidNet with ShakeDrop. Lower: ResNet-18.

with VAT in a quite complicated way. First, it should be noted that a larger-scale of unlabeled dataset ($\mathcal{D}_{ul}$) does not always imply better performance, and this phenomenon is also model-dependent. First, for PyramidNet, optimal performances mostly occur when only MIR1K is taken as $\mathcal{D}_{ul}$, and adding MedleyDB and DALI-train does not guarantee better results. For ResNet, its can be observed that a larger unlabeled dataset (MIR1K+Med+DALI) does give better results, but this trend is more obvious only in offset-related metrics. A possible reason is that the genres of the three $\mathcal{D}_{ul}$ are quite different. Both MedleyDB and DALI-train contain a much wider ranges of singing styles, usually with chorus singing, while MIR1K is less diverse and can be better optimized when training in batch. Nevertheless, using MIR1K+MedleyDB+DALI-train still outperforms the case using MIR-1K+MedleyDB, and this indicates that there is still room for improvement if incorporating more unlabeled data for semi-supervised learning.

Among the three testing datasets, DALI-test is obviously the most challenging and is hard to be improved by VAT. This is because that the note event annotation in DALI is obtained automatically from global alignment, and is reported to be error prone [30]. Besides, the chorus singing part, which is commonly seen in the DALI dataset, may confound the result of monophonic pitch extraction. This can be seen from the fact that the performance greatly drops when considering pitch for DALI-test set: its onset-offset-pitch F1-scores are always much lower than its onset-offset F1-scores. These challenging issues might still require solutions from supervised learning rather than the SSL approaches.

4.3.5 Sequence-level vs. note-level evaluation

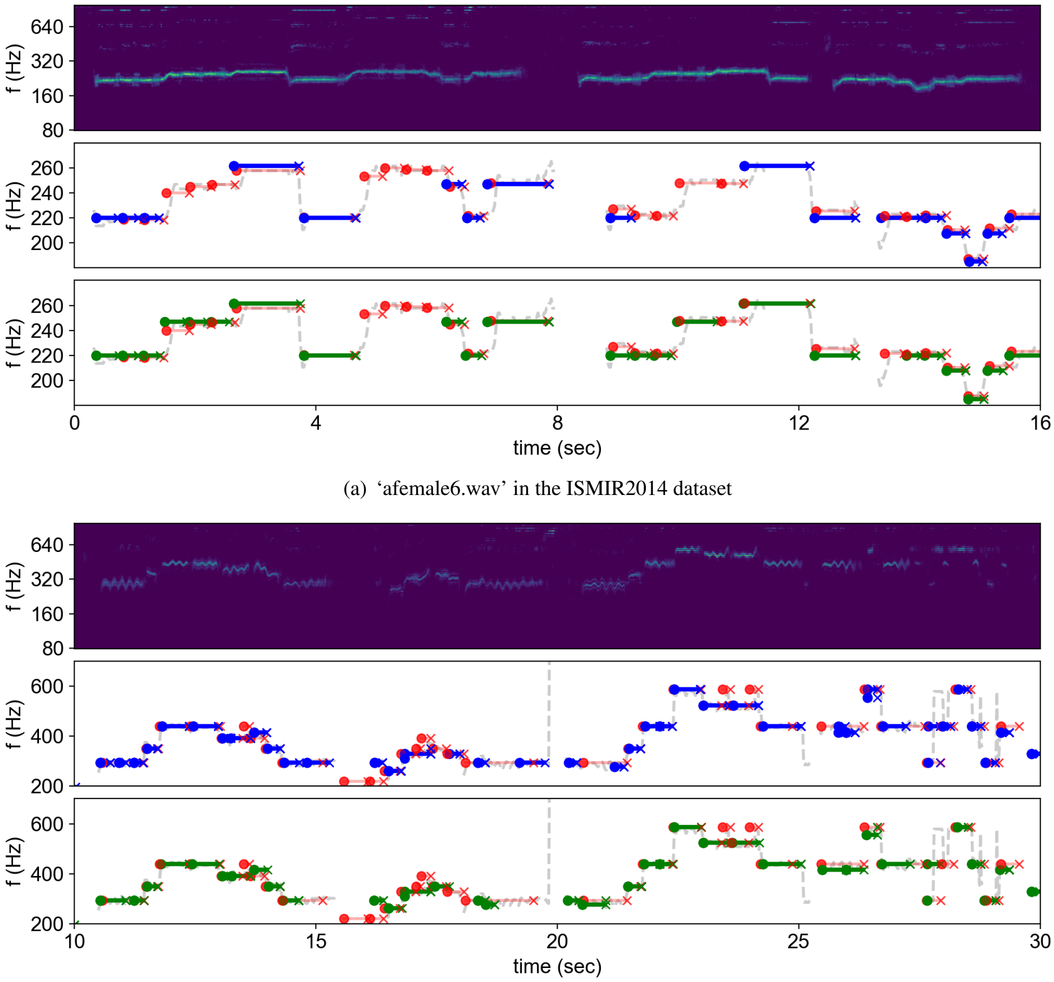
It is worth noting that NAcc exhibits a trend different from other metrics. A high onset-offset-pitch F1-score does not imply a high NAcc. On DALI-test and Cmedia, we observe that using larger scale of $\mathcal{D}_{ul}$ does result in better NAcc using PyramidNet. This can be explained by the high variability of annotation of onset/ offset time. With large-scale and high-diversity data, SSL may not be effective in capturing local event, but can improve the performance in the global scale. Besides, given the fact that the onset/offset annotations are not perfectly reliable, sequence-level metrics such NAcc is worth further investigation when the purpose of SVT is to transcribe the music score rather than replicating the music performance.

Method	P	R	F
[31]	30.4	31.5	30.8
[24]	43.0	37.3	39.8
[32]	39.7	44.0	41.5
[12]	40.9	43.6	42.1
[13]	51.0	53.4	52.0
[8]	62.5	56.9	59.4
ResNet w/o VAT	63.1	60.6	61.7
ResNet w/i VAT	67.7	58.8	62.8
PyramidNet w/o VAT	68.6	58.1	62.7
PyramidNet w/i VAT	**72.2**	**65.3**	**68.4**

Table 2. Performance comparison (in %) of various SVT methods on the ISMIR2014 dataset ($\mathcal{D}_{ul} =$ MIR1K).

4.3.6 Comparison to the state of the arts

Table 2 compares the precision, recall, and F1-score of the proposed method to previous work on the ISMIR2014 dataset, the only public dataset systematically evaluated on note-level SVT (this why a comparison on polyphonic music datasets is not made here). Results of previous work are listed in the upper six rows, while the new results are



(a) ‘afemale6.wav’ in the ISMIR2014 dataset

(b) ‘10.wav’ in the CMedia dataset (10-30 seconds)

Figure 2. Data representations and transcription results using the PyramidNet model. From top to bottom: data representation, transcription results without VAT (blue lines), and transcription results with VAT (green lines). Grey dashed lines are frame-level pitch contours. Red lines are ground truth. Circle dots are onset events, and crosses are offset events.

in the lower four rows. The result of ResNet w/o VAT can be regarded as an imporved version of [8], by re-arranging the channels of the input data representations while following the same output dimensions and temporal decoding processes. Such modification entails 2.3 percentage points of improvement from [8]. Besides, using a model larger than ResNet-18 (i.e. PyramidNet) further improves the resulting F1-score by 1 percentage point. Finally, with the assistance of SSL, the PyramidNet model with MIR1K for VAT achieves 68.5% of F1-score, which outperforms the best previous method [8] by 9.0 percentage points.

4.3.7 Illustration

Figure 2 shows the results of two challenging examples of SVT. The first example is challenging because of the repeated notes (consecutive note with the same pitch), while the main challenge of the second example is its wide pitch range. Figure 2(a) shows that the it is hard to observe the onset and offset events from the data representation. The purely supervised model fails to capture most of the onset and offset events of repeated notes, and this issue can be partly solved by utilizing VAT; see the repeated notes captured at around 2 secs and 9 secs of the example. In Figure 2(b), it can be shown that both models without and with VAT fail to transcribe low-pitch notes and high-pitch ornamentation (around 24 secs), partly due to the fact that these events are less visible on the data representations.

5. CONCLUSION

We have validated the effectiveness of leveraging semi-supervised learning on note segmentation in singing voice transcription. State-of-the-art performance has been reported on public benchmarks. The role of semi-supervised learning is found depending on the model and the size, quality and the diversity of the unlabeled training data. These findings provide insights into future semi-supervised MIR research. The source code is available at the project page.[3] VOCANO is also available as part of the automatic music transcription library Omnizart [33].[4]

[3] https://github.com/B05901022/VOCANO

[4] https://github.com/Music-and-Culture-Technology-Lab/omnizart

6. ACKNOWLEDGMENTS

This work was partially supported by MOST Taiwan under the project *Automatic Music Transcription Algorithms for Interactive Music Systems* (Grant No. 106-2218-E-001-003-MY3).

7. REFERENCES

[1] M. Müller, E. Gómez, and Y.-H. Yang, "Computational methods for melody and voice processing in music recordings (dagstuhl seminar 19052)," in *Dagstuhl Reports*, vol. 9, no. 1, 2019.

[2] E. Anders, "Modeling music: Studies of music transcription, music perception and music production," Ph.D. dissertation, KTH Royal Institute of Technology, 2018.

[3] 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 *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 161–165.

[4] M.-T. Chen, B.-J. Li, and T.-S. Chi, "CNN based two-stage multi-resolution end-to-end model for singing melody extraction," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1005–1009.

[5] F. Rigaud and M. Radenen, "Singing voice melody transcription using deep neural networks," in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2016, pp. 737–743.

[6] T. Nakano, K. Yoshii, Y. Wu, R. Nishikimi, K. W. E. Lin, and M. Goto, "Joint singing pitch estimation and voice separation based on a neural harmonic structure renderer," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 160–164.

[7] S. Kum, J.-H. Lin, L. Su, and J. Nam, "Semi-supervised learning using teacher-student models for vocal melody extraction," in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2020, pp. 93–100.

[8] Z.-S. Fu and L. Su, "Hierarchical classification networks for singing voice segmentation and transcription," in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2019, pp. 900–907.

[9] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5927–5935.

[10] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 41, no. 8, pp. 1979–1993, 2018.

[11] 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 *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2016, pp. 461–467.

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

[13] 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. Sound and Music Computing (SMC)*, 2015.

[14] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian singing transcription based on a hierarchical generative model of keys, musical notes, and F0 trajectories," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 28, pp. 1678–1691, 2020.

[15] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, "An end-to-end framework for audio-to-score music transcription on monophonic excerpts." in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2018, pp. 34–41.

[16] T. Kwon, D. Jeong, and J. Nam, "Polyphonic piano transcription using autoregressive multi-state note model," in *International Society for Music Information Retrieval Conference (ISMIR)*. International Society for Music Information Retrieval, 2020, pp. 454–461.

[17] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[18] R. Schramm and E. Benetos, "Automatic transcription of a cappella recordings from multiple singers," in *AES International Conference on Semantic Audio*. Audio Engineering Society, 2017.

[19] H. Cuesta, B. McFee, and E. Gómez, "Multiple F0 estimation in vocal ensembles using convolutional neural networks," in *International Society for Music Information Retrieval Confence (ISMIR)*, 2020, pp. 302–309.

[20] Y.-T. Wu, B. Chen, and L. Su, "Automatic music transcription leveraging generalized cepstral features and deep learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 401–405.

[21] L. Su, "Vocal melody extraction using patch-based CNN," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 371–375.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] Y. Yamada, M. Iwamura, T. Akiba, and K. Kise, "Shakedrop regularization for deep residual learning," *IEEE Access*, vol. 7, pp. 186 126–186 136, 2019.

[24] 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*, vol. 37, no. 2, pp. 73–90, 2013.

[25] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive mir research." in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2014, pp. 155–160.

[26] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "DALI: a large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm." in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2018, pp. 431–437.

[27] E. Molina, A. M. Barbancho-Perez, L. J. Tardón, I. Barbancho-Perez *et al.*, "Evaluation framework for automatic singing transcription," in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2014.

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

[29] 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 *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2014, pp. 367–372.

[30] G. Meseguer-Brocal, R. M. Bittner, S. Durand, and B. Brost, "Data cleansing with contrastive learning for vocal note event annotations," in *Proc. International Society for Music Information Retrieval Confence (ISMIR)*, 2020, pp. 255–262.

[31] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–86, 2008.

[32] 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 (TASLP)*, vol. 23, no. 2, pp. 252–263, 2015.

[33] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I. Wei, J.-Y. Hsu, Y.-C. Chuang, and L. Su, "Omnizart: A general toolbox for automatic music transcription," *arXiv preprint arXiv:2106.00497*, 2021.