SEMI-SUPERVISED LEARNING USING TEACHER-STUDENT MODELS FOR VOCAL MELODY EXTRACTION

Sangeun Kum[1] **Jing-Hua Lin**[2] **Li Su**[2] **Juhan Nam**[1]

[1] Graduate School of Culture Technology, KAIST, South Korea
[2] Institute of Information Science, Academia Sinica, Taiwan

keums@kaist.ac.kr, jhlin@iis.sinica.edu.tw, lisu@iis.sinica.edu.tw, juhan.nam@kaist.ac.kr

ABSTRACT

The lack of labeled data is a major obstacle in many music information retrieval tasks such as melody extraction, where labeling is extremely laborious or costly. Semi-supervised learning (SSL) provides a solution to alleviate the issue by leveraging a large amount of unlabeled data. In this paper, we propose an SSL method using teacher-student models for vocal melody extraction. The teacher model is pre-trained with labeled data and guides the student model to make identical predictions given unlabeled input in a self-training setting. We examine three setups of teacher-student models with different data augmentation schemes and loss functions. Also, considering the scarcity of labeled data in the test phase, we artificially generate large-scale testing data with pitch labels from unlabeled data using an analysis-synthesis method. The results show that the SSL method significantly increases the performance against supervised learning only and the improvement depends on the teacher-student models, the size of unlabeled data, the number of self-training iterations, and other training details. We also find that it is essential to ensure that the unlabeled audio has vocal parts. Finally, we show that the proposed SSL method enables a baseline convolutional recurrent neural network model to achieve performance comparable to state-of-the-arts.

1. INTRODUCTION

One of the key elements in the success of deep learning is a large amount of labeled data. However, when the labeled data is scarce in a given task, it can be a bottleneck in leveraging the power of deep neural networks. The issue has been found in many music information retrieval (MIR) tasks as well. Among others, melody extraction research has suffered from it as pitch labeling requires experienced annotators to handle the annotation tool and the process is extremely labor-intensive [1].

The lack of labeled data in melody extraction research has been tackled in several different ways. A popular method to alleviate the issue is data augmentation which increases labeled data by transforming the input audio, for example, using pitch-shifting [2–4]. Data augmentation, however, has the limitation in covering the diversity in the input space. Another approach is using multi-track audio data [5–7]. This allows to use monophonic pitch tracking algorithms for the melodic source and therefore it expedites laborious the pitch labeling. However, multi-track recording datasets often maintain individual tracks as stem files where multiple similar sound sources can be mixed (e.g., main vocal and backing vocal). Therefore, obtaining clean pitch labels from multi-track audio can be not straightforward [8, 9]. Recently, melody MIDI files, which are more easily accessible, have been utilized to guide melody extraction from audio with transfer learning techniques from the symbolic to audio domain [3, 10]. MIDI data exhibit greater flexibility than audio on data augmentation, but still face limitations on representing natural pitch contours of singing voice, which usually contain subtle variations such as vibrato and portamento.

Semi-supervised learning (SSL) is another but more general strategy to address the lack of labeled data. SSL uses a large amount of unlabeled data, which is usually easy to collect, jointly with labeled data. A popular class of SSL methods is based on self-training in the teacher-student framework. Recent works have combined random data augmentation with the SSL methods to encourage the model to produce robust output even when input is perturbed. This approach has achieved state-of-the-art performance on image classification [11–13], speech recognition [14], and audio classification [15]. There are a few MIR researches that used the teacher-student framework to address the lack of labeled data, for example, in automatic drum transcription [16] and singing voice detection [17, 18]. However, to the best of our knowledge, recent advances in SSL methods that leverage the power of deep neural networks and random data augmentation in the teacher-student framework have been not studied yet in the music domain.

In this paper, we apply the SSL methods to vocal melody extraction with the following contributions. First, we present the SSL methods for vocal melody extraction leveraging large-scale unlabeled music datasets. This prevents the model from overfitting to small labeled data and improve the performance. Second, we compare three setups of teacher-student models along with various audio

Attribution: Sangeun Kum, Jing-Hua Lin, Li Su, Juhan Nam, “Semi-supervised learning using teacher-student models for vocal melody extraction”, in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

data augmentation techniques. We show the model with the consistency regularization is most effective. Third, we investigate effective SSL strategies by exploring joint training, the size of unlabeled data, and the number of self-training iterations. Fourth, we show that the proposed teacher-student training method enables a baseline convolutional recurrent neural network model to achieve performance comparable to state-of-the-arts. Finally, apart from the SSL method, we propose large-scale testing data artificially generated from unlabeled data using an analysis-synthesis framework, considering the lack of labeled data even at the testing stage. Evaluation on the diverse and sizable test set will reinforce the effectiveness of the proposed method. For reproducibility, the source code and pre-trained model used in this paper are available online [1] .

2. RELATED WORK

The teacher-student framework has been previously studied in several MIR tasks to address the lack of labeled data. Wu and Lerch applied the approach to automatic drum transcription [16]. They used multiple teacher models based on non-negative matrix factorization (NMF) trained with different datasets and a student model based on deep neural network trained with labels from the teachers. They showed that the student model outperforms the teacher models. However, it was not a self-training setting where the teacher model is repeatedly replaced with an improved student model. Schlüter explored the self-training for singing voice detection [17]. They first trained a convolutional neural network (CNN) on the original labels with low-granularity, then a second network on pseudo-labels with high-granularity from the first network, and a third network on the summarized saliency maps from the second network. They showed this self-improvement worked up to the third network. However, they conducted the self-training on weakly-labeled data in the context of multiple-instance learning and did not used any unlabeled data. Recently, Meseguer-Brocal et al. used the teacher-student paradigm for singing voice detection to create a large-scale time-aligned vocal melody and lyrics dataset [18]. They consistently improved the teacher model by increasing the correlation between the prediction of the model and the time-aligned lyrics annotation.

3. METHODS

3.1 Model Architecture

Recent melody extraction algorithms have used CNN [9, 19, 20] and its variants [4, 21, 22] as a standard architecture. Since we focus on the effectiveness of SSL in this paper, we employ a previously proposed convolutional recurrent neural network (CRNN) which was a baseline architecture in [4]. The CRNN architecture consists of 4 ResNet blocks and a bi-directional long short-term memory layer. We first merge the audio waveforms into a mono channel and downsample them to 8 kHz. We then calculate the logarithmic-magnitude spectrogram using short-time Fourier transform with a 1024-point Hann window and an 80-point hop size.The CRNN architecture takes 31 consecutive frames of the spectrogram as input and predicts a pitch label quantized with a resolution of 1/8 semitone and ranged from E2 (82.4 Hz) to B6 (1975.7 Hz). The size of the output layer is 442, including a non-vocal label.

Algorithm 1: Train SSL Models

```
Train a teacher network $T_1$ on labeled data
  $\mathcal{D} = \{(x_d, y_d) : d \in (1, ..., N)\}$;
Generate augmented data
  $\tilde{\mathcal{U}} = \{\tilde{x}_u = RAA(x_u) : u \in (1, ..., M)\}$ from
  unlabeled data $\mathcal{U} = \{x_u : u \in (1, ..., M)\}$;
for $i$ = 1 to $k$ do
    Use $T_i$ to generate pseudo labels for $\mathcal{U}$ (or $\tilde{\mathcal{U}}$);
    Train student network $S_i$ using both $\mathcal{D}$ and $\mathcal{U}$
      (or $\tilde{\mathcal{U}}$) as training data;
    $T_{i+1} = S_i$;
end
```

3.2 SSL in the Teacher-Student Framework

Our SSL method is based on self-training in the teacher-student framework where the teacher model is first trained with labeled data and then the student model is trained with artificial labels generated from the teacher model using unlabeled data. The artificial labels can be the prediction distribution vector [11, 12] or one-hot vector determined by the class with a highest confidence [13, 23]. We formally describe the overall procedure in Algorithm 1. We first train the initial teacher model T_1 using only labeled data $\mathcal{D}$ where x_d are labeled examples and y_d are one-hot reference labels. For unlabeled data $\mathcal{U}$ where x_u are unlabeled examples, we use random data augmentation to generate noisy input data $\tilde{\mathcal{U}}$ where $\tilde{x}_u$ are noisy unlabeled examples. RandAudioAugment (RAA) is an audio version of random data augmentation method which is described in Section 3.4. While it is more effective to use random data augmentation on the student model only in image classification [12], we also try applying it for both teacher and student models for ablation study. Once we train the student model jointly with the labeled data and unlabeled data (with pseudo labels), we replace the teacher model with the student model. We repeat the same pseudo labeling and the training with a new student model.

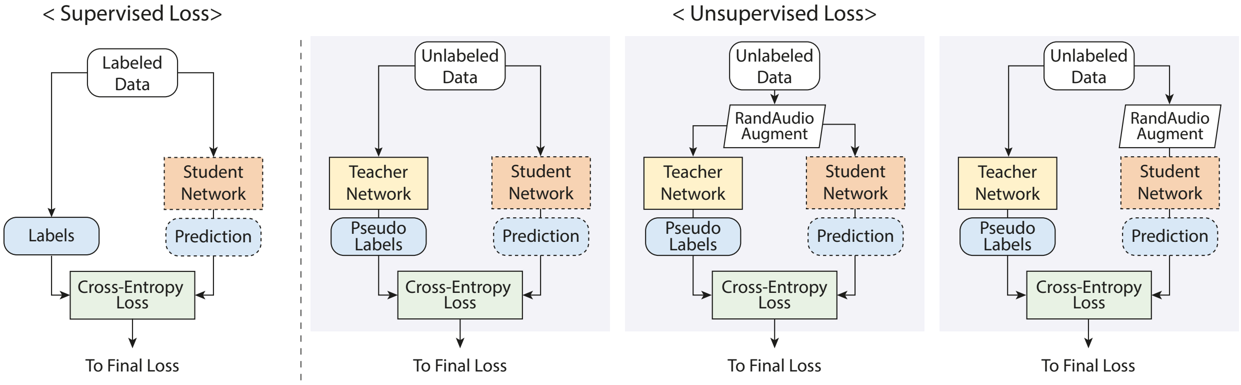
3.3 Proposed Teacher-Student Models

Our proposed Teacher-Student models are illustrated in Figure 1. The supervised loss $\mathcal{L}_D$ is computed with labeled data and defined as:

$$\mathcal{L}_D = \frac{1}{N}\sum_{d=1}^{N} H(y_d, p(y|x_d;\theta_s)) \tag{1}$$

where $H(\cdot)$ denotes the cross-entropy between the pitch label y_d and pitch prediction $p(y|x)$, and θ_s denotes a set of parameters of the student model. The supervised loss is a common loss term of the three investigated

[1] https://github.com/keums/melodyExtraction_SSL



(a) Basic Teacher-Student (b) Noisy Teacher-Student (c) Noisy Student

Figure 1. Diagram of the three Teacher-Student models.

teacher-student models. Each of them are explained below.

Basic Teacher-Student is a fundamental teacher-student framework that uses the unlabeled data $\mathcal{U}$ but trains the student network with the pseudo labels generated from the teacher network. The final loss of Basic Teacher-Student $\mathcal{L}_B$ is defined as

$$\mathcal{L}_B = \mathcal{L}_D + \frac{1}{M}\sum_{u=1}^{M} H(y_u, p(y|x_u;\theta_s)) \tag{2}$$

where y_u is the pseudo labels on $\mathcal{U}$ generated by the teacher network, i.e. $y_u = p(y|x_u;\theta_t)$ where θ_t to denote the parameters of teacher network. The basic teacher-student model is illustrated in Figure 1(a).

Noisy Teacher-Student takes noisy unlabeled data $\tilde{\mathcal{U}}$ for both of the teacher and student networks using RAA and the rest is the same as the basic teacher-student model. The final loss of Noisy Teacher-Student $\mathcal{L}_N$ is defined as

$$\mathcal{L}_N = \mathcal{L}_D + \frac{1}{M}\sum_{u=1}^{M} H(\tilde{y}_u, p(y|\tilde{x}_u;\theta_s)) \tag{3}$$

where $\tilde{y}_u$ is a prediction on $\tilde{\mathcal{U}}$ generated by the teacher network, i.e. $\tilde{y}_u = p(y|\tilde{x}_u;\theta_t)$. The noisy teacher-student model is illustrated in Figure 1(b).

Noisy Student takes noisy unlabeled data $\tilde{\mathcal{U}}$ only for the student network while the teacher network takes unnoised input $\mathcal{U}$ to generate the pseudo labels. The idea is that the student should produce consistent outputs that minimize the difference from the teacher even though the input is perturbed [12]. This notion is also similar to consistency regularization [24, 25]. The final loss of Noisy Student $\mathcal{L}_C$ is defined as

$$\mathcal{L}_C = \mathcal{L}_D + \frac{1}{M}\sum_{u=1}^{M} H(y_u, p(y|\tilde{x}_u;\theta_s)) \tag{4}$$

The noisy student model is illustrated in Figure 1(c).

3.4 Data Augmentation

We conducted pitch-shift by ± 1,2 semitone on the labeled data $\mathcal{D}$ (audio and corresponding labels). In the melody extraction task, it has shown that pitch-shifting can improve the generality and performance of the model by increasing the amount of audio and label pairs for different f_0 [2, 26]. For data augmentation of unlabeled data $\mathcal{U}$, we propose RandAudioAugment (RAA) inspired by RandAugment [27], which is a method of randomly applying different kinds of transformations to increase image data. RAA converts audio by randomly selecting multiple audio effects as follows: audio equalizer (low-shelf, high-shelf), filters (low-pass, high-pass), overdrive, phaser, and reverb. Here, we use *pysndfx* that is a Python library designed for applying effects to audio files[2]. We sampled a random magnitude of each transformation from a predefined range. The implementation details for RAA are also described in the source code.

3.5 Data Selection

The SSL algorithm using large-scale unlabeled data may suffer from labeling noise. Unlabeled data are highly likely to have audio without vocals. Filtering only high-confidence examples or the top-K examples in image classification has demonstrated to be an effective method to handle the labeling noise [12, 28]. Likewise, we performed data selection so that only the tracks with vocal ratios exceeding a threshold were used for training. To estimate the ratio of vocals included in the track, we used our singing voice detector[3] based on CNN based on [29]. Considering the distribution of vocal ratio in the FMA, we set the threshold to 0.3.

4. DATASETS

Table 1 shows the simple statistics of the labeled and unlabeled training datasets and test datasets.

[2] https://github.com/carlthome/python-audio-effects

[3] https://github.com/keums/SingingVoiceDetection

	Dataset	Number of Tracks	Total Length
Training (Labeled)	RWC	100	6h 47m
	MedleyDB	61	2h 39m
	iKala	262	2h 6m
Training (Unlabeled)	In-house	535	6h 21m
	FMA_small	3,521 / 8,000	25h / 60h
	FMA_medium	10,639 / 25,000	89h / 208h
	FMA_large	40,505 / 106,574	337h / 888h
Test	ADC04	12	4m
	MIREX05	9	4m
	MedleyDB	12	43m
	AST218	218	14h 53m

Table 1. Description of datasets. In FMA, The two numbers indicate tracks with vocal (the vocal ratio above 0.3) and all tracks respectively.

4.1 Labeled Data

We used the three labeled datasets (RWC [30], MedleyDB [6], and iKala [7]) and split them into a train and validation set following [9]. We augmented the training data by pitch-shifting with ± 1,2 semitone. The total length of the labeled training data amounts to about 55 hours after the data augmentation.

4.2 Unlabeled Data

As to unlabeled data, we used an in-house dataset crawled from YouTube and the Free Music Archive (FMA) [31]. The in-house dataset is pop songs with vocals recorded in a variety of environments. It includes both public-released and user-uploaded tracks. FMA is a large-scale open dataset containing up to 106,574 tracks and covers 161 genres of music. We used FMA for performance comparison on data scalability. The FMA has three different subsets depending on the number of the track and genre included: FMA_small (FMA$_S$), FMA_medium (FMA$_M$), and FMA_large (FMA$_L$). We selected vocal tracks from them as described in Section 3.5 and denote the selected versions as FMA$_{Sv}$, FMA$_{Mv}$, and FMA$_{Lv}$, respectively. We augmented the unlabeled datasets via RAA during training as described in Section 3.4.

4.3 Test Data

4.3.1 Public Test Sets

We used three public test sets (ADC04 [4], MIREX05 [4], and MedleyDB) to evaluate the performance of vocal melody extraction. In this study, we excluded non-vocal tracks from ADC04 and MIREX05, and used songs not included in training data for MedleyDB. To obtain the ground truth for singing voice in MedleyDB, we adopted its 'MELODY2' annotations. These three datasets have been commonly used to compare the performance of melody extraction. However, the number of tracks and the total length are very limited as shown in Table 1.

4.3.2 Proposed Large-Scale Test Set

To make up the scarcity of testing data for evaluating singing voice extraction algorithms, we propose a new test set composed of DSD100 [32] and MusDB18 [33]. The two multitrack datasets were originally designed for source separation. Each track has four isolated stems: vocals, drums, bass, and others. Following the analysis/synthesis framework [8], the singing melodies for 218 selected tracks [5] were synthesized with automatically generated f_0 contours. In detail, for each song, we extracted the melody of the vocals with five different pitch trackers, and each f_0 information along with the vocal audio was fed into the WORLD [34] (D4C edition [35]) vocoder to reproduce five monophonic variations of the vocal stem. The original vocal audio was parameterized into harmonic and aperiodic spectral envelopes, and then resynthesized with provided pitch contours. Then a mask was applied to filter intervals without f_0 information. For remixing, the amplitude of the synthesized vocal was weighted to that of the original vocal stem, and the rest stems were directly summed up as accompaniments, then mixed with the weighted synthesized vocal that perfectly matched the f_0 annotation. These 1,090 polyphonic mixtures with accurate and automatic annotations constitute the proposed analysis/synthesis test set, AST218 [6].

Each track in AST218 has five variations whose vocal melody was annotated separately with five different pitch estimators: CREPE [36] (with confidence threshold of 0.5 and 0.7), pYIN [37], and Lu&Su [3] (with time step of 10 and 20ms), as they have different merits. Since there is no exact way to pinpoint a common optimal confidence threshold across the entire dataset, we chose two different threshold values for CREPE: one is 0.5, suffering from high false positive (FP) but preserving details; the other threshold is 0.7, acceptable FP though sacrificing some recall. pYIN was chosen for it has even lower FP while producing stable and continuous melodic lines when the vocal stem is monophonic. However, it is not stable in the pholyphonic scenario, which is universal in DSD100 and MusDB18. In need of other polyphonic-based melody estimators to balance the f_0 quality, we chose two time step setups of the Lu&Su model: 20ms, at which this model is optimized; and 10ms, which provides more continuous predictions and offers alternative pitch contours when encountering multiple melodic vocal lines.

The analysis/synthesis framework has been practiced successfully in evaluating monotonic pitch trackers [36]. As a sanity check, we evaluated several patchCNN [19] setups on the original and resynthesized ADC04, MIREX05, and MedleyDB. The differences of OA are within ± 2–5%, which is acceptable, meaning this framework is also applicable for polyphonic test set generation.

When evaluating vocal extraction algorithms on AST218, we averaged the scores from the five variations. Our pilot study shows that these five pitch contours reach

[4] http://labrosa.ee.columbia.edu/projects/melody/

[5] Songs that appear in MedleyDB were excluded for they were part of the training data, but songs in MusDB18 having counterparts in DSD100 were not removed for they are not exactly identical. Additionally, 12 songs that do not have discernible vocal melodies were also excluded.

[6] https://sites.google.com/view/mctl/resource

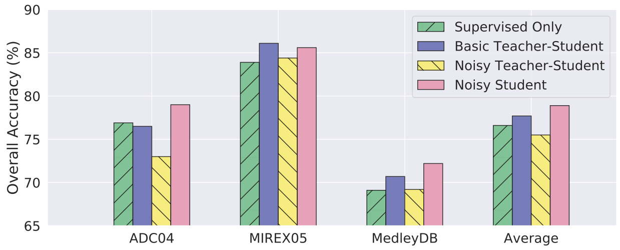


Figure 2. Comparison with supervised-learning model and three student models on three test sets.

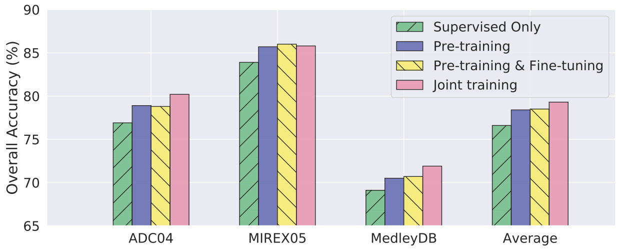


Figure 3. Comparison with pre-training, fine-tuning, and joint training methods on three test sets.

consensus over a majority of frames, while the estimations differ for tricky frames. Rather than manually check on the estimated f_0, we used AST218 in an ensemble manner, fully leveraging the spirit of automatic pitch annotation.

5. EXPERIMENTS

5.1 Experimental Setup

5.1.1 Training Details

We used the CRNN architecture with residual connections and bi-directional long short-term memory in all experiments. The implementation of the model was consistent with that of the main network of [4]. We trained our models using Adam optimizer for 70 epochs on 2 GPUs. The initial learning rate was set to 0.003 in all the experiments. We used a learning rate schedule that reduces the learning rate by 0.7 times if validation accuracy did not increase within three epochs. The model and the training procedures were implemented using Keras [7] [38].

5.1.2 Evaluation

To evaluate the performance of melody extraction, we mainly used overall accuracy (OA) which combines the accuracy of pitch estimation with voice detection. We also used three metrics raw pitch accuracy (RPA) for pitch estimation, and voicing recall (VR) and voicing false alarm (VFA) for voice detection [39]. These metric are computed by *mir_eval* [40] library designed.

5.2 Experiment 1: Teacher-Student Models

Our first experiment is to demonstrate the efficacy of the proposed Teacher-Student models for SSL. In this experiment, we trained three Teacher-Student models described in Section 3.3 using an in-house dataset as unlabeled data. We evaluated the performance of each model on ADC04, MIERX05, and MedleyDB, which have been used as standard test sets for evaluation. As shown in Figure 2, the basic teacher-student model can achieve 1.1% higher average OA than the supervised-only model which has 77.7% average OA. This confirms the possibility of using unlabeled data to improve the performance of melody extraction. Our experiment also shows that the noisy student model outperforms all the others, having 78.9% average OA.

The noisy student model increases OA by 3.1% with respect to the supervised-only model in MedleyDB, which is especially a challenging dataset because it contains tracks that are difficult to distinguish between vocals and background music, or tracks with excessive audio effects. The results indicate that the student network can be trained reliably using the noisy student model, even if the initial teacher network is not robust to diverse noise. Meanwhile, the performance of the noisy teacher-student has deteriorated, being worse than the supervised-only model. This degradation is probably because the noised teacher model is not generating reliable pseudo labels.

5.3 Experiment 2: Joint Training vs. Fine-Tuning

The training methods of the teacher-student framework can be divided into three approaches depending on how $\mathcal{D}$ and $\mathcal{U}$ are used for training: pre-training on only $\mathcal{U}$ and then fine-tuning on $\mathcal{D}$; joint-training on both $\mathcal{U}$ and $\mathcal{D}$ simultaneously. Figure 3 compares the results among pre-training, fine-tuning, and joint training for the noisy student model. The jointly trained model achieves 0.8% higher average OA than the fine-tuned model, with the highest results on MedleyDB. This indicates that joint training on unlabeled data and labeled data would help the networks produce a decision boundary that better reflects real music [41]. Interestingly, the average OA of the pre-trained model only on unlabeled data is higher than that of the supervised learning model. This suggests that the distribution of unlabeled data is similar to that of labeled data. Considering that the in-house dataset consists of pop songs with vocals, the in-house dataset can be seen as having a similar tendency to the labeled data. It provides insight into the data selection in the next experiment.

5.4 Experiment 3: Size of Training Data

We investigated the importance of the size and validity of unlabeled data. To explore the effect of the size of unlabeled data, we started with the in-house dataset as training data for the noisy student model and progressively included larger subsets of FMA. The results can be seen in Figure 4. Although the FMA data set contains more numerous tracks than the in-house dataset, the average OA of FMA_S and FMA_L is lower than that of the model trained only with the in-house dataset. Note that the proposed model focuses only on vocal melodies. As a result, teacher models may suffer from labeling noise generated by numerous instrument tracks included in the FMA. In addition, all labels on the instrumental track are classified as non-vocal pitch, resulting in data imbalance.

[7] We used Keras 2.3.0, Accessed: 15 May 2020

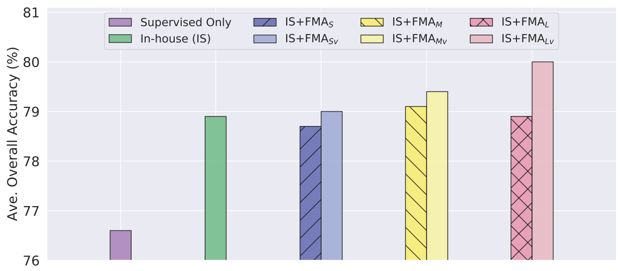


Figure 4. Comparison with Noisy Students on varied sizes of unlabeled datasets. The subscript 'v' denotes a selected subset of FMA whose vocal ratio exceeds a threshold. We use the average of OA for three public test sets.

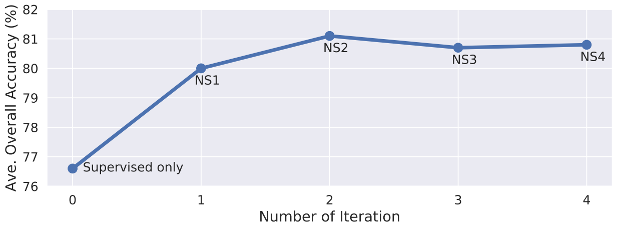


Figure 5. Effect of iteration training for Noisy Students.

To confirm the validity of the dataset, we performed data selection for each FMA subset as mentioned in Section 3.5 and used them to train each student model. Interestingly, as the size of the $\mathcal{U}$ increases, the performance of each model tends to be significantly improved. For example, FMA_{Lv} achieves an average OA of 80.2%, which is 3.6% higher than the supervised-only model. This indicates that effective SSL requires a large amount of $\mathcal{U}$ with a similar distribution for $\mathcal{D}$.

5.5 Experiment 4: Iterative Training

We iterated the self-training 4 times for the noisy student model using the in-house dataset and FMA$_{Lv}$. The results are illustrated in Figure 5. We observe that the performance continuously increases up to 2 iterations achieving the highest average OA of 81.1%. Generally, self-training tends to amplify the error caused by labelling noise during training. However, the noisy student model trained on large-scale unlabeled data can help overcome this difficulty. Nevertheless, increasing the number of training iterations three or more times does not improve performance, and rather slightly lower the accuracy.

5.6 Comparison with State-of-the-Arts

We compared the supervised-only model (as a baseline) and proposed the noisy student model (NS) with four recent melody extraction algorithms based on deep neural networks: the patch-based CNN (patchCNN) [19], the deep salience map (DSM) [9], the streamlined encoder/decoder network (segNet) [21], and the joint detection and classification model (JDC) [4], which have open-sourced codes with vocal mode. Each method was run with its default parameters, and then evaluated on the three conventional test sets and the newly introduced AST218. Besides, we report the frame-level scores instead of song-level ones to settle uneven song lengths.

Methods	ADC04	MIREX05	MedleyDB	AST218
PatchCNN [19]	76.9 / 72.9	69.7 / 73.8	44.0 / 59.3	42.3 / 59.7
DSM [9]	89.2 / 72.2	87.7 / 80.1	80.6 / 75.4	38.9 / 68.3
SegNet [21]	88.7 / 83.3	82.6 / 80.0	70.6 / 75.5	41.5 / 68.1
JDC [4]	**90.6 / 83.5**	**91.4 / 87.4**	72.7 / 78.1	55.8 / **75.4**
Baseline	78.7 / 76.8	79.9 / 81.5	57.2 / 70.7	**56.3** / 69.7
Proposed (NS)	90.4 / 82.2	90.4 / 85.9	**76.3 / 79.2**	54.2 / 74.2

Table 2. Vocal melody extraction results in terms of (RPA / OA) of the proposed and other methods on various test sets. The proposed model is iterated the self-training two times using the in-house dataset and FMA$_{Lv}$.

Methods	ADC04	MIREX05	MedleyDB	AST218
PatchCNN	91.8 / 46.1	80.3 / **11.6**	60.1 / 22.4	61.6 / 26.0
DSM	95.7 / 61.1	93.9 / 29.4	**85.4** / 26.6	44.6 / 7.7
SegNet	95.2 / 38.5	92.2 / 24.0	78.8 / 21.7	51.7 / 10.0
JDC	96.7 / 40.2	**97.5** / 18.5	80.5 / 18.3	64.7 / **8.6**
Baseline	92.6 / **33.8**	89.1 / 15.2	71.0 / **16.7**	**72.0** / 19.2
Proposed (NS)	**97.4** / 42.1	97.3 / 20.4	83.3 / 19.1	61.6 / 9.4

Table 3. Voicing detection results in terms of (VR / VFA) of the proposed and other methods on various test sets.

Table 2 and Table 3 list the results of each method on the four test sets. In general, performances of the proposed NS model are comparable to other supervised-learning-based methods and even outperforms others in MedleyDB, and it effectively improves the OA of the baseline by 4.5–8.5%. The overall rankings of VR and VFA vary across the test sets, but the behavior converges in terms of OA. One can also observe that the AST218 is the most challenging in the majority of cases. In such a dataset, the performance of the NS model shows that the proposed method is robust to large-scale evaluation. However, the NS model improves the baseline except for VR and RPA in AST218. This result might be because the simple rule-based remixing of vocal and accompaniment tracks in AST218 is different from the artistic practice of mixing engineers, which can affect voicing detection and, in turn, RPA.

6. CONCLUSION

This study provides a framework of semi-supervised learning using the teacher-student model for vocal melody extraction. We compared three setups of teacher-student models and revealed that the NS model is the most effective and robust to real-world music where various noises can be present. We showed that large-scale unlabeled data is effective when they are properly selected. We found that iterative training for the teacher-student model helps improve performance. We also confirmed the effectiveness of the proposed method by evaluating it on artificial large-scale test data generated from automatically annotated multitrack data. Although these findings are based only on vocal melody extraction, we believe our method can be extended to other MIR tasks that suffer from the lack of labeled data such as automatic music transcription and chord recognition.

7. ACKNOWLEDGEMENT

This research was supported by BK21 Plus Postgraduate Organization for Content Science (or BK21 Plus Program) and Basic Science Research Program through the National Research Foundation of Korea (2015R1C1A1A02036962).

8. REFERENCES

[1] J. Salamon, "What's broken in music informatics research? three uncomfortable statements," in *Proc. of the 36th International Conference on Machine Learning, PMLR 97*, 2019.

[2] S. Kum, C. Oh, and J. Nam, "Melody extraction on vocal segments using multi-column deep neural networks," in *Proc. ISMIR*, 2016, pp. 819–825.

[3] W.-T. Lu and L. Su, "Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning," in *Proc. ISMIR*, 2018, pp. 521–528.

[4] S. Kum and J. Nam, "Joint detection and classification of singing voice melody using convolutional recurrent neural networks," *Applied Sciences*, vol. 9, no. 7, p. 1324, 2019.

[5] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the mir-1k dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2009.

[6] 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. ISMIR*, 2014, pp. 155–160.

[7] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *Proc. ICASSP*, 2015, pp. 718–722.

[8] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez, and J. P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets." in *Proc. ISMIR*, 2017, pp. 71–78.

[9] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music," in *Proc. ISMIR*, 2017.

[10] Y. Gao, B. Zhu, W. Li, K. Li, Y. Wu, and F. Huang, "Vocal melody extraction via dnn-based pitch estimation and salience-based pitch refinement," in *Proc. ICASSP*, 2019, pp. 1000–1004.

[11] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 5050–5060.

[12] Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *arXiv preprint arXiv:1911.04252*, 2019.

[13] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[14] L. Mošner, M. Wu, A. Raju, S. H. K. Parthasarathi, K. Kumatani, S. Sundaram, R. Maas, and B. Hoffmeister, "Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning," in *Proc. ICASSP*. IEEE, 2019, pp. 6475–6479.

[15] K. Lu, C.-S. Foo, K. K. Teh, H. D. Tran, and V. R. Chandrasekhar, "Semi-supervised audio classification with consistency-based regularization," *Proc. Interspeech*, pp. 3654–3658, 2019.

[16] C.-W. Wu and A. Lerch, "Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data," in *Proc. ISMIR*, 2017, pp. 613–620.

[17] J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples." in *Proc. ISMIR*, 2016, pp. 44–50.

[18] 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. ISMIR*, 2018.

[19] L. Su, "Vocal melody extraction using patch-based CNN," in *Proc. ICASSP*, 2018, pp. 371–375.

[20] 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. ICASSP*, 2019, pp. 1005–1009.

[21] T.-H. Hsieh, L. Su, and Y.-H. Yang, "A streamlined encoder/decoder architecture for melody extraction," in *Proc. ICASSP*. IEEE, 2019, pp. 156–160.

[22] H. Chou, M.-T. Chen, and T.-S. Chi, "A hybrid neural network based on the duplex model of pitch perception for singing melody extraction," in *Proc. ICASSP*, 2018, pp. 381–385.

[23] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, 2013, p. 2.

[24] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Advances in neural information processing systems*, 2016, pp. 1163–1171.

[25] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *arXiv preprint arXiv:1904.12848*, 2019.

[26] R. M. Bittner, B. McFee, and J. P. Bello, "Multitask learning for fundamental frequency estimation in music," *arXiv preprint arXiv:1809.00381*, 2018.

[27] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical data augmentation with no separate search," *arXiv preprint arXiv:1909.13719*, 2019.

[28] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan, "Billion-scale semi-supervised learning for image classification," *arXiv preprint arXiv:1905.00546*, 2019.

[29] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks." in *Proc. ISMIR*, 2015, pp. 121–126.

[30] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: popular, classical and jazz music databases." in *Proc. ISMIR*, 2002, pp. 287–288.

[31] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "Fma: A dataset for music analysis," in *Proc. ISMIR*, 2017.

[32] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation - 12th International Conference*. Springer, 2017, pp. 323–332.

[33] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[34] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[35] M. Morise, "D4c, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[36] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proc. ICASSP*. IEEE, 2018, pp. 161–165.

[37] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. ICASSP*. IEEE, 2014, pp. 659–663.

[38] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[39] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 2, pp. 118–134, 2014.

[40] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. Ellis, "mir_eval: A transparent implementation of common mir metrics," in *Proc. ISMIR*, 2014.

[41] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, 2018, pp. 3235–3246.