MULTITASK LEARNING FOR FRAME-LEVEL INSTRUMENT RECOGNITION

Yun-Ning Hung[1], *Yi-An Chen*[2] *and Yi-Hsuan Yang*[1]

[1] Research Center for IT Innovation, Academia Sinica, Taiwan
[2] KKBOX Inc., Taiwan
{biboamy,yang}@citi.sinica.edu.tw, annchen@kkbox.com

ABSTRACT

For many music analysis problems, we need to know the presence of instruments for each time frame in a multi-instrument musical piece. However, such a frame-level instrument recognition task remains difficult, mainly due to the lack of labeled datasets. To address this issue, we present in this paper a large-scale dataset that contains synthetic polyphonic music with frame-level pitch and instrument labels. Moreover, we propose a simple yet novel network architecture to jointly predict the pitch and instrument for each frame. With this multitask learning method, the pitch information can be leveraged to predict the instruments, and also the other way around. And, by using the so-called pianoroll representation of music as the main target output of the model, our model also predicts the instruments that play each individual note event. We validate the effectiveness of the proposed method for frame-level instrument recognition by comparing it with its single-task ablated versions and three state-of-the-art methods. We also demonstrate the result of the proposed method for multi-pitch streaming with real-world music. For reproducibility, we will share the code to crawl the data and to implement the proposed model at: `https://github.com/biboamy/instrument-streaming`.

Index Terms— Instrument recognition, pitch streaming

1. INTRODUCTION

Pitch and timbre are two fundamental properties of musical sounds. While the pitch decides the notes sequence of a musical piece, the timbre decides the instruments used to play each note. Since music is an art of time, for detailed analysis and modeling of the information of a musical piece, we need to build a computational model that predicts the pitch and instrument labels for each time frame. With the release of several datasets [1, 2] and the development of deep learning techniques, recent years have witnessed great progress in frame-level pitch recognition, a.k.a., multi-pitch estimation (MPE) [3, 4]. However, this is not the case for the instrument part, presumably due to the following two reasons.

First, manually annotating the presence of instruments for each time frame in a multi-instrument musical piece is a time-consuming and labor-intensive process. As a result, most datasets available to the public only provide instrument labels on the *clip level*, namely, labeling which instruments are present over an entire audio clip of possibly multi-second long [5–8]. Such clip-level labels do not specify the presence of instruments for each short-time frame (e.g., multiple milliseconds, or for each second). Datasets with frame-level instrument labels emerge only over the recent few years [1,2,9,10]. However, as listed in Table 1 (and will be discussed at length in Section 2), these datasets contain at most a few hundred songs and some of them contain only classical musical pieces. The musical diversity found in these datasets might therefore not be sufficient to train a deep learning model that performs well for different musical pieces.

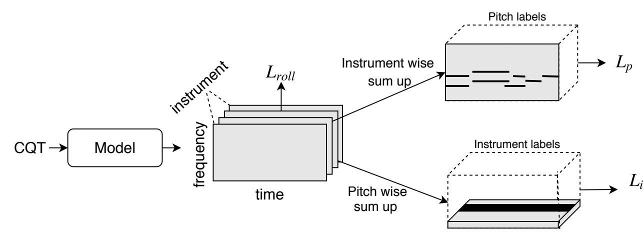

Fig. 1. Architecture of the proposed model, which employs three loss functions for predicting the (multitrack) pianoroll, the pitch roll, and the instrument roll. The pitch and instrument predictions are computed directly from the predicted pianoroll, which is a tensor of {frequency, time, instrument}.

Second, we note that most recent work that explores deep learning techniques for frame-level instrument recognition focuses only on the instrument recognition task itself and adopts the *single-task* learning paradigm [13, 14, 16]. This has the drawback of neglecting the strong relations between pitch and instruments. For example, different instruments have their own pitch ranges and tend to play different parts in a polyphonic musical composition. Proper modeling of the onset and offset of musical notes may also make it easier to detect the presence of instruments [14]. From a methodological point of view, we see a potential gain to do better than these prior arts by using a *multitask* learning paradigm that models timbre and pitch jointly. This requires a dataset that contains

	Pitch labels	Instrument labels	Real or Synth	Genre	Number of Songs
MedleyDB [1]	△ [3, 11]	√ [12, 13]	Real	Variety	122
MusicNet [2]	√ [4]	√ [14]	Real	Classical	330
Bach10 [9]	√ [9]	√ [15]	Real	Classical	10
Mixing Secret [10]		√ [13]	Real	Variety	258
MuseScore (this paper)	√	√	Synthetic	Variety	344,166

Table 1. This table provides information regarding some datasets that provide frame-level labels for either pitch or instrument: whether the audio is real or synthetic, the genre and the number of songs. We also cite some papers (after the symbols √ or △) that employed these datasets for training either pitch or instrument recognition models. And, we use △ to denote 'part of it.'

both frame-level pitch and instrument labels.

In this paper, we introduce a new large-scale dataset called *MuseScore* to address these needs. The dataset contains the audio and MIDI pairs for 344,166 musical pieces downloaded from the official website (`https://musescore.org/`) of MuseScore, an open source and free music notation software licensed under GPL v2.0. The audio is synthesized from the corresponding MIDI file, usually using the sound font of the MuseScore synthesizer. Therefore, it is not difficult to temporally align the audio and MIDI files to get the frame-level pitch and instrument labels for the audio. Although the dataset only contains synthesized audio, it includes a variety of performing styles in different musical genres.

Moreover, we propose to transform each MIDI file to the *multitrack pianoroll* representation of music (see Fig. 1 for an illustration) [17], which is a binary tensor representing the presence of notes over different time steps for each instrument. Then, we propose a multitask learning method that learns to predict from the audio of a musical piece its (multitrack) pianoroll, frame-level pitch labels (a.k.a., the *pitch roll*), and the instrument labels (a.k.a., the *instrument roll*). While the latter two can be obtained by directly summing up the pianoroll along different dimensions, the three involved loss functions would work together to force the model learn the interactions between pitch and timbre. Our experiments show that the proposed model can not only perform better than its task-specific counterparts, but also existing methods for frame-level instrument recognition [13, 14, 16].

2. BACKGROUND

To our knowledge, there are four public-domain datasets that provide frame-level instrument labels, as listed in Table 1. Among them, MedleyDB [1], MusicNet [2] and Bach10 [9] are collected originally for MPE research, while Mixing Secret [10] is meant for instrument recognition. When it comes to building "clip-level" instrument recognizers, there are other more well-known datasets such as the ParisTech [5] and IRMAS [6] datasets. Still, there are previous work that uses these datasets for building either clip-level [12, 15] or frame-level [13, 14] instrument recognizers.

There are three recent works on frame-level instrument recognition. The model proposed by Hung and Yang [14] is trained and evaluated on different subsets of MusicNet [2], which consists of only classical music. This model considers the pitch labels estimated by a pre-trained model (i.e. [3]) as an additional input to predict instrument, but the pre-trained model is fixed and not further updated. The model presented by Gururani *et al.* [13] is trained and evaluated on the combination of MedleyDB [1] and Mixing Secrets [10]. Both [14] and [13] use frame-level instrument labels for training. In contrast, the model presented by Liu *et al.* [16] uses only clip-level instrument labels associated with YouTube videos for training, using a weakly-supervised approach. Both [16] and [13] do not consider pitch information.

As the existing datasets are limited in genre coverage or data size, prediction models trained on these datasets may not generalize well, as shown in [3] for pitch recognition. Unlike these prior arts, we explore the possiblity to train a model on large-scale synthesized audio dataset, using a multitask learning method that considers both pitch and timbre.

OpenMIC-2018 [7] is a new large-scale dataset for training clip-level instrument recognizers. It contains 20,000 10-second clilps of Creative Commons-licensed music of various genres. But, there is no frame-level labels.

Multi-pitch streaming has been referred to as the task that assigns instrument labels to note events [18]. Therefore, it goes one step closer to full transcription of musical audio than MPE. However, as the task involves both frame-level pitch and instrument recognition, it is only attempted sporadically in the literature (e.g., [18, 19]). By predicting the pianorolls, the proposed model actally performs multi-pitch streaming.

3. PROPOSED DATASET

The MuseScore dataset is collected from the online forum of the MuseScore community. Any user can upload the MIDI and the corresponding audio for the music pieces they create using the software. The audio is therefore usually synthesized by the MuseScore synthesizer, but the user has the freedom to use other synthesizers. The audio clips have diverse musical genres and are about two mins long on average. More statistics of the dataset can be found from our GitHub repo.

While the collected audio and MIDI pairs are usually well

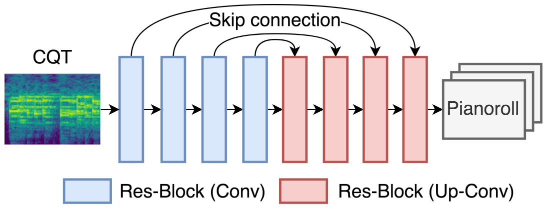

Fig. 2. The network architecture of the proposed model. It has a simple U-net structure [23] with four residual convolution layers and four residual up-convolution layers.

aligned, to ensure the data quality we further run the dynamic time warping (DTW)-based alignment algorithm proposed by Raffel [20] over all the data pairs. We then compute from each MIDI file the groundtruth pianoroll, pitch roll and instrument roll using `Pypianoroll` [17].

The dataset contains 128 different instrument categories as defined in the MIDI spec. A main limitation is that there is no singing voice. This can be made up by datasets with labels of vocal activity [21], such as the Jamendo dataset [22].

Due to copyright issues, we cannot share the dataset itself but the code to collect and process the data.

4. PROPOSED MODEL

As Fig. 1 shows, the proposed model learns a mapping $f(\cdot)$ (i.e., the 'Model' block in the figure) between an audio representation $\mathbf{X}$, such as the constant-Q transform (CQT) [24], and the pianoroll $\mathbf{Y}_{roll} \in \{0,1\}^{F\times T\times M}$, where F, T and M denote the number of pitches, time frames and instruments, respectively. Namely, the model can be viewed as a multi-pitch streaming model. The model has two by-products, the pitch roll $\mathbf{Y}_p \in \{0,1\}^{F\times T}$ and the instrument roll $\mathbf{Y}_i \in \{0,1\}^{M\times T}$. As Fig. 1 shows, from an input audio, our model computes $\widehat{\mathbf{Y}}_p$ and $\widehat{\mathbf{Y}}_i$ directly from the pianoroll $\widehat{\mathbf{Y}}_{roll}$ predicted by the model. Therefore, $f(\cdot)$ contains all the learnable parameters of the model.

We train the model $f(\cdot)$ with a multitask learning method by using three cost functions, L_{roll}, L_p and L_i, as shown in Fig. 1. For each of them, we use the binary cross entropy (BCE) between the groundtruth and the predicted matrices (tensors). The BCE is defined as:

$$L_* = -\sum \left[\mathbf{Y}_* \cdot \ln \sigma(\widehat{\mathbf{Y}_*}) + (1-\mathbf{Y}_*)\cdot \ln(1-\sigma(\widehat{\mathbf{Y}_*}))\right], \quad (1)$$

where σ is the sigmoid function that scales its input to $[0,1]$. We weigh the three cost terms so that they have the same range, and use their weighted sum to update $f(\cdot)$.

In sum, pitch and timbre are modeled jointly with a shared network by our model. This learning method is designed for music and, to our knowledge, has not been used elsewhere.

Method	Instrument	Pitch	Pianoroll
L_{roll} only (ablated)	—	—	0.623
L_i only (ablated)	0.896	—	—
L_p only (ablated)	—	0.799	—
all (proposed)	0.947	0.803	0.647

Table 2. Performance comparison of the proposed multitask learning method ('all') and 3 single-task ablated versions, for frame-level instrument recognition (in F1-score), frame-level pitch recognition (Acc), and pianoroll prediction (Acc) using the triaining and test subsets of MuseScore, for 9 instruments.

4.1. Network Structure

The network architecture of our model is shown in Fig. 2. It is a simple convolutional encoder/decoder network with symmetric skip connections between the encoding and decoding layers. Such a "U-net" structure has been found useful for image segmentation [23], where the task is to learn a mapping function between a dense, numeric matrix (i.e., an image) and a sparse, binary matrix (i.e., the segment boundaries). We presume that the U-net structure can work well for predicting the pianorolls, since it also involves learning such a mapping function. In our implementation, the encoder and decoder are composed of four residual blocks for convolution and up-convolution. Each residual block has three convolution, two batchNorm and two leakyReLU layers. The model is trained with stochastic gradient descent with 0.005 learning rate. More details can be found from our GitHub repo.

4.2. Model Input

We use CQT [24] to represent the input audio, since it adopts a log frequency scale that better aligns with our perception of pitch. CQT also provides better frequency resolution in the low-frequency part, which helps detect the fundamental frequencies. For the convenience of training with mini-batches, each audio clip in the training set is divided into 10-second segments. We compute CQT by `librosa` [25], with 16 kHz sampling rate, 512-sample hop size, and 88 frequency bins.

5. EXPERIMENT

5.1. Ablation Study

We report two sets of experiments for frame-level instrument recognition. In the first experiment, we compare the proposed multitask learning method with its single-task versions, using two non-overlapping subsets of MuseScore as the training and test sets. Specifically, we consider only the 9 most popular instruments[1] and run a script to pick for each instrument 5,500 clips as the training set and 200 clips as the test set. We consider three ablated versions here: using the U-net architecute

[1]Piano, acoustic guitar, electric guitar, trumpet, sax, violin, cello & flute.

Method	Training set	Piano	Guitar	Violin	Cello	Flute	Avg
[16]	YouTube-8M [26]	0.766	0.780	0.787	0.755	0.708	0.759
[13]	Training split of 'MedleyDB+Mixing Secrets' [13]	0.733	0.783	0.857	0.860	0.851	0.817
[14]	MuseScore training subset	0.690	0.660	0.697	0.774	0.860	0.736
Ours	MuseScore training subset	0.718	0.819	0.682	0.812	0.961	0.798

Table 3. AUC scores of per-second instrument recognition on the test split of 'MedleyDB+Mixing Secrets', for 5 instruments.

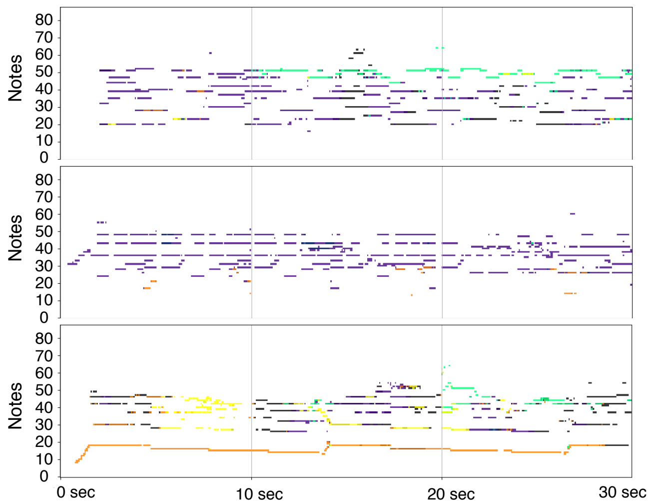

Fig. 3. The predicted pianoroll (best viewed in color) for the first 30 seconds of three real-world music. We paint different instruments with different colors: *Black*—piano, *Purple*—guitar, *Green*—violin, *Orange*—cello, *Yello*—flute.

shown in Fig. 1 to predict the pianoroll with only L_{roll}, to predict directly the instrument roll (i.e. only considering L_i), and to preidct directly the pitch roll (i.e. only L_p).

Result shown in Table 2 clearly demonstrates the superiority of the proposed multitask learning method over the single-task counterparts, especially for instrument prediction. Here, we use `mir_eval` [27] to calculate the 'pitch' and 'pianoroll' accuracies. For 'instrument', we report the F1-score.

5.2. Comparison with Existing Methods

In the second experiment, we compare our method with three existing methods [13, 14, 16]. Following [13], we take 15 songs from MedleyDB and 54 songs from Mixing Secret as the test set, and consider only 5 instruments (see Table 3). The test clips contain instruments (e.g., singing voice) that are beyond these five. We evaluate the result for per-second instrument recognition in terms of area under the curve (AUC).

As shown in Table 3, these methods use different training sets. Specifically, we retrain model [14] using the same training subset of MuseScore as the proposed model. The model [16] is trained on the YouTube-8M dataset [26]. The model [13] is trained on a training split of 'MedleyDB+Mixing Secret', with 100 songs from each of the two datasets. The model [13] therefore has some advantages since the training set is close to the test set. The result of [16] and [13] are from the authors of the respective papers.

Table 3 shows that our model outperforms the two prior arts [14, 16] and is behind model [13]. We consider our model compares favorably with [13], as our training set is quite different from the test set. Interestingly, our model is better at the flute, while [13] is better at the violin. This might be related to the difference between the real and synthesized sounds for these instruments, but future work is needed to clarify.

5.3. Multi-pitch Streaming

Finally, Fig. 3 demonstrates the predicted pianorolls for the first 30 seconds of three randomly-selected real-world songs.[2] In general, the proposed model can predict the notes and instruments pretty nicely, especially for the second clip, which contains only a guitar solo. This is promising, since the model is trained with synthetic audio only. Yet, we also see two limitations of our model. First, it cannot deal with sounds that are not included in the training data—e.g., for the 5th–10th seconds of the third clip, our model mistakes the piano for the flute, possibly because the singer hums in the meanwhile. Second, it cannot predict the onset times accurately—e.g., the violin melody of the first clip actually plays the same note for several times, but the model mistakes them for long notes.

6. CONCLUSION

In this paper, we have presented a new synthetic dataset and a multitask learning method that models pitch and timbre jointly. It allows the model to predict instrument, pitch and pianorolls representation for each time frame. Experiments show that our model generalizes well to real music.

In the future, we plan to improve the instrument recognition by re-synthesizing the MIDI files from Musescore dataset to produce more realistic instrument sound. Moreover, we also plan to mix the singing voice clips from [1] with our training data (for data augmentation) to deal with singing voices.

[2]The three songs are, from top to bottom: *All of Me* violin & guitar cover (https://www.youtube.com/watch?v=YpYQh7eQULc), *Ocean* by Purdull (https://www.youtube.com/watch?v=5Lb9GvEO-sA) and *Beautiful* by Christina Aguilera (https://www.youtube.com/watch?v=eAfyFTzZDMM).

7. REFERENCES

[1] Rachel M. Bittner et al., "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proc. ISMIR*, 2014, [Online] http://medleydb.weebly.com/.

[2] John Thickstun, Zaid Harchaoui, and Sham M. Kakade, "Learning features of music from scratch," in *Proc. Int. Conf. Learning Representations*, 2017, [Online] https://homes.cs.washington.edu/∼thickstn/musicnet.html.

[3] Rachel M. Bittner et al., "Deep salience representations for $f0$ estimation in polyphonic music," in *Proc. ISMIR*, 2017.

[4] John Thickstun et al., "Invariances and data augmentation for supervised music transcription," *Proc. ICASSP*, pp. 2241–2245, 2018.

[5] Cyril Joder, Slim Essid, and Gaël Richard, "Temporal integration for audio classification with application to musical instrument classification," *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, no. 1, pp. 174–186, 2009.

[6] Juan J. Bosch et al., "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," in *Proc. ISMIR*, 2012.

[7] Eric J. Humphrey, Simon Durand, and Brian McFee, "OpenMIC-2018: An open dataset for multiple instrument recognition," in *Proc. ISMIR*, 2018, [Online] https://github.com/cosmir/openmic-2018.

[8] Jort F. Gemmeke et al., "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.

[9] Zhiyao Duan, Bryan Pardo, and Changshui Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, pp. 2121–2133, 2010.

[10] Siddharth Gururani and Alexander Lerch, "Mixing secrets: a multi-track dataset for instrument recognition in polyphonic music," in *Proc. ISMIR-LBD*, 2017.

[11] Jong Wook Kim et al., "Crepe: A convolutional representation for pitch estimation," in *Proc. ICASSP*, 2018.

[12] Peter Li et al., "Automatic instrument recognition in polyphonic music using convolutional neural networks," *CoRR*, vol. abs/1511.05520, 2015.

[13] Siddharth Gururani, Cameron Summers, and Alexander Lerch, "Instrument activity detection in polyphonic music using deep neural networks," in *Proc. ISMIR*, 2018.

[14] Yun-Ning Hung and Yi-Hsuan Yang, "Frame-level instrument recognition by timbre and pitch," in *Proc. ISMIR*, 2018, pp. 135–142.

[15] Dimitrios Giannoulis, Emmanouil Benetos, Anssi Klapuri, and Mark D. Plumbley, "Improving instrument recognition in polyphonic music through system integration," *Proc. ICASSP*, pp. 5222–5226, 2014.

[16] Jen-Yu Liu, Yi-Hsuan Yang, and Shyh-Kang Jeng, "Weakly-supervised visual instrument-playing action detection in videos," *IEEE Trans. Multimedia*, in press.

[17] Hao-Wen Dong, Wen-Yi Hsiao, and Yi-Hsuan Yang, "Pypianoroll: Open source Python package for handling multitrack pianoroll," in *Proc. ISMIR-LBD*, 2018, [Online] https://github.com/salu133445/pypianoroll.

[18] Zhiyao Duan, Jinyu Han, and Bryan Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 138–150, 2014.

[19] Vipul Arora and Laxmidhar Behera, "Multiple f0 estimation and source clustering of polyphonic music audio using PLCA and HMRFs," *IEEE/ACM Trans. Audio, Speech and Language Processing*, vol. 23, no. 2, pp. 278–287, 2015.

[20] Colin Raffel, *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*, Ph.D. thesis, Columbia U., 2016, [Online] https://github.com/craffel/alignment-search.

[21] Kyungyun Lee, Keunwoo Choi, and Juhan Nam, "Revisiting singing voice detection: A quantitative review and the future outlook," in *Proc. ISMIR*, 2018.

[22] Mathieu Ramona, G. Richard, and B. David, "Vocal detection in music with support vector machines," in *Proc. ICASSP*, 2008, pp. 1885–1888.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015.

[24] Christian Schörkhuber and Anssi Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. Sound and Music Computing Conf.*, 2010.

[25] Brian McFee et al., "librosa: Audio and music signal analysis in Python," in *Proc. Python in Science Conf.*, 2015, [Online] https://librosa.github.io/librosa/.

[26] "YouTube-8M," https://research.google.com/youtube8m/.

[27] Colin Raffel et al., "mir_eval: A transparent implementation of common MIR metrics," in *Proc. ISMIR*, 2014, [Online] https://craffel.github.io/mir_eval/.