FRAME-LEVEL INSTRUMENT RECOGNITION BY TIMBRE AND PITCH

Yun-Ning Hung and Yi-Hsuan Yang

Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan

{biboamy,yang}@citi.sinica.edu.tw

ABSTRACT

Instrument recognition is a fundamental task in music information retrieval, yet little has been done to predict the presence of instruments in multi-instrument music for each time frame. This task is important for not only automatic transcription but also many retrieval problems. In this paper, we use the newly released MusicNet dataset to study this front, by building and evaluating a convolutional neural network for making frame-level instrument prediction. We consider it as a multi-label classification problem for each frame and use frame-level annotations as the supervisory signal in training the network. Moreover, we experiment with different ways to incorporate pitch information to our model, with the premise that doing so informs the model the notes that are active per frame, and also encourages the model to learn relative rates of energy buildup in the harmonic partials of different instruments. Experiments show salient performance improvement over baseline methods. We also report an analysis probing how pitch information helps the instrument prediction task. Code and experiment details can be found at https://biboamy. github.io/instrument-recognition/.

1. INTRODUCTION

Progress in pattern recognition problems usually depends highly on the availability of high-quality labeled data for model training. For example, in computer vision, the release of the ImageNet dataset [11], along with advances in algorithms for training deep neural networks [26], has fueled significant progress in *image-level* object recognition. The subsequent availability of other datasets, such as the COCO dataset [30], provide bounding boxes or even *pixel-level* annotations of objects that appear in an image, facilitating research on localizing objects in an image, semantic segmentation, and instance segmentation [30]. Such a move from image-level to pixel-level prediction opens up many new exciting applications in computer vision [16].

Analogously, for many music-related applications, it is desirable to have not only *clip-level* but also *frame-level* predictions. For example, expert users such as music composers may want to search for music with certain attributes

and require a system to return not only a list of songs but also indicate the time intervals of the songs that have those attributes [3]. Frame-level predictions of music tags can be used for visualization and music understanding [31,45]. In automatic music transcription, we want to know the musical notes that are active per frame as well as figure out the instrument that plays each note [13]. Vocal detection [40] and guitar solo detection [36] are another two examples that requires frame-level predictions.

Many of the aforementioned applications are related to the classification of sound sources, or instrument classification. However, as labeling the presence of instruments in multi-instrument music for each time frame is labor-intensive and time-consuming, most existing work on instrument classification uses either datasets of solo instrument recordings (e.g., the ParisTech dataset [24]), or datasets with only clip- or excerpt-level annotations (e.g., the IRMAS dataset [7]). While it is still possible to train a model that performs frame-level instrument prediction from these datasets, it is difficult to evaluate the result due to the absence of frame-level annotations. [1] As a result, to date little work has been done to specifically study frame-level instrument recognition, to the best of our knowledge (see Section 2 for a brief literature survey).

The goal of this paper is to present such a study, by taking advantage of a recently released dataset called Music-Net [44]. The dataset contains 330 freely-licensed classical music recordings by 10 composers, written for 11 instruments, along with over 1 million annotated labels indicating the precise time of each note in every recording and the instrument that plays each note. Using the *pitch* labels available in this dataset, Thickstun *et al.* [43] built a convolutional neural network (CNN) model that establishes a new state-of-the-art in multi-pitch estimation. We propose that the frame-level *instrument* labels provided by the dataset also represent a valuable information source. And, we try to realize this potential by using the data to train and evaluate a frame-level instrument recognition model.

Specifically, we formulate the problem as a multi-label classification problem for each frame and use frame-level annotations as the supervisory signal in training a CNN model with three residual blocks [21]. The model learns

---

[1] Moreover, these datasets may not provide high-quality labeled data for frame-level instrument prediction. To name a few reasons: the Paris-Tech dataset [24] contains only instrument solos and therefore misses the complexity seen in multi-instrument music; the IRMAS dataset [7] labels only the "predominant" instrument(s) rather than all the active instruments in each excerpt; moreover, an instrument may not be always active throughout an excerpt.

to predict instruments from a spectral representation of audio signals provided by the constant-Q transform (CQT) (see Section 4.1 for details). Moreover, as another technical contribution, we investigate several ways to incorporate pitch information to the instrument recognition model (Sections 4.2), with the premise that doing so informs the model the notes that are active per frame, and also encourages the model to learn the energy distribution of partials (i.e., fundamental frequency and overtones) of different instruments [2,4,14,15]. We experiment with using either the ground truth pitch labels from MusicNet, or the pitch estimates provided by the CNN model of Thickstun *et al.* [43] (which is open-source). Although the use of pitch features for music classification is not new, to our knowledge few attempts have been made to jointly consider timbre and pitch features in a deep neural network model. We present in Section 5 the experimental results and analyze whether and how pitch-aware models outperform baseline models that take only CQT as the input.

2. RELATED WORK

A great many approaches have been proposed for (clip-level) instrument recognition. Traditional approaches used domain knowledge to engineer audio feature extraction algorithms and fed the features to classifiers such as support vector machine [25, 32]. For example, Diment *et al.* [12] combined Mel-frequency cepstral coefficients (MFCCs) and phase-related features and trained a Gaussian mixture model. Using the instrument solo recordings from the RWC dataset [17], they achieved 96.0%, 84.9%, 70.7% accuracy in classifying 4, 9, 22 instruments, respectively. Yu *et al.* [47] used sparse coding for feature extraction and support vector machine for classifier training, obtaining 96% accuracy in 10-instrument classification for the solo recordings in the ParisTech dataset [24]. Recently, Yip and Bittner [46] made open-source a solo instrument classifier that uses MFCCs in tandem with random forests to achieve 96% frame-level test accuracy in 18-instrument classification using solo recordings from the MedleyDB multi-track dataset [5]. Recognizing instruments in multi-instrument music has been proven more challenging. For example, Yu *et al.* [47] achieved 66% F-score in 11-instrument recognition using a subset of the IRMAS dataset [7].

Deep learning has been increasingly used in more recent work. Deep architectures can "learn" features by training the feature extraction module and the classification module in an end-to-end manner [26], thereby leading to better accuracy than traditional approaches. For example, Li *et al.* [27] showed that feeding raw audio waveforms to a CNN achieves 72% (clip-level) F-micro score in discriminating 11 instruments in MedleyDB, which MFCCs and random forest only achieves 64%. Han *et al.* [19] trained a CNN to recognize predominant instrument in IRMAS and achieved 60% F-micro, which is about 20% higher than a non-deep learning baseline. Park *et al.* [35] combined multi-resolution recurrence plots and spectrogram with CNN to achieved 94% accuracy in 20-instrument classification using the UIOWA solo instrument dataset [18].

Number of instruments used	Number of clips		Pitch est. accuracy
	Train set	Test set	
0	3	0	—
1	172	5	62.9%
2	33	1	56.2%
3	95	4	60.5%
4	15	0	56.6%
6	2	0	49.6%

Table 1: The number of clips in the training and test sets of MusicNet [44], divided according to the number of instruments used (among the seven instruments we consider in our experiment) per clip (e.g., a piano trio uses 3 instruments). We also show the average frame-level multi-pitch estimation accuracy (using mir_eval [38]) achieved by the CNN model proposed by Thickstun *et al.* [43].

Due to the lack of frame-level instrument labels in many existing datasets, little work has focused on frame-level instrument recognition. The work presented by Schlüter for vocal detection [40] and by Pati and Lerch for guitar solo detection [36] are exceptions, but they each addressed one specific instrument, rather than general instruments. Liu and Yang [31] proposed to use clip-level annotations in a weakly-supervised setting to make frame-level predictions, but the model is for general tags. Moreover, due to the assumption that CNN can learn high-level features on its own, domain knowledge of music has not been much used in prior work on deep learning based instrument recognition, though there are some exceptions [33, 37].

Our work differentiates itself from the prior arts in two aspects. First, we focus on frame-level instrument recognition. Second, we explicitly employ the result of multi-pitch estimation [6, 43] as additional inputs to our CNN model, with a design that is motivated by the observation that instruments have different pitch range and have unique energy distributions in the partials [14].

3. DATASET

Training and evaluating a model for frame-level instrument recognition is possible due to the recent release of the MusicNet dataset [44]. It contains 330 freely-licensed music recordings by 10 composers with over 1 million annotated pitch and instrument labels on 34 hours of chamber music performances. Following [43], we use the pre-defined split of training and test sets, leading to 320 and 10 clips in the training and test sets, respectively. As there are only seven different instruments in the test set, we only consider the recognition of these seven instruments in our experiment. They are *Piano, Violin, Viola, Cello, Clarinet, Bassoon* and *Horn*. For the training set, we do not exclude the sounds from the instruments that are not on the list, but these instruments are not labeled. Different clips use different number of instruments. See Table 1 for some statistics. For convenience, each clip is divided into 3-second segments. We use these segments as the input to our model. We zero-pad (i.e., adding silence) the last seg-

ment of each clip so that it is also 3 seconds. Due to space limit, for details we refer readers to the MusicNet website (check reference [44] for the URL) and also our project website (see the abstract for the URL).

We note that the MedleyDB dataset [5] can also be used for frame-level instrument recognition, but we choose MusicNet for two reasons. First, MusicNet is more than three times larger than MedleyDB in terms of the total duration of the clips. Second, MusicNet has pitch labels for each instrument, while MedleyDB only annotates the melody line. However, as MusicNet contains only classical music and MedleyDB has more Pop and Rock songs, the two datasets feature fairly different instruments and future work can be done to consider they both.

4. INSTRUMENT RECOGNITION METHOD

4.1 Basic Network Architectures that Uses CQT

To capture the timbral characteristics of each instrument, in our basic model we use CQT as the feature representation of music audio. CQT is a spectrographic representation that has a musically and perceptual motivated frequency scale [41]. We compute CQT by `librosa` [34], with sampling rate 44,100 and 512-sample window size. 88 frequency notes are extracted with 12 bins per octave, which forms a matrix $\mathbf{X} \in \mathcal{R}^{258 \times 88}$ as the input data, for each inputting 3-second audio segment.

We experiment with two baseline models. The first one is adapted from the CNN model proposed by Liu and Yang [31], which has been shown effective for music auto-tagging. Instead of using 6 feature maps as the input to the model as they did, we just use CQT as the input. Moreover, we use frame-level annotations as the supervisory signal in training the network, instead of training the model in a weakly-supervised fashion as they did. A batch normalization layer [23] is added after each convolution layer. Figure 1a shows the model architecture.

The second one is adapted from a more recent CNN model proposed by Chou *et al.* [10], which has been shown effective for large-scale sound event detection. Its design is special in two aspects. First, it uses 1D convolutions (along time) instead of 2D convolutions. While 2D convolutions analyze the input data as a chunk and convolve on both spectral and temporal dimensions, the 1D convolutions (along time) might better capture frequency and timbral information in each time frame [10, 29]. Second, it uses the so-called residual (Res) blocks [21,22] to help the model learn deeper. Specifically, we employ three Res-blocks in between an *early* convolutional layer and a *late* convolutional layer. Each Res-block has three convolutional layers, so the network has a stack of 11 convolutional layers in total. We expect such a deep structure can learn well for a large-scale dataset such as MusicNet. Figure 1b shows its model architecture.

4.2 Adding Pitch

Although people usually expect neural networks can learn high-level feature such as pitch, onset and melody, our pi-

lot study shows that with the basic architecture the network still confuses some instruments (e.g., clarinet, bassoon and horn), and that onset frames for each instrument are not nicely located (see the second row of Figure 3). We propose to remedy this with a pitch-aware model that explicitly takes pitch as input, in a hope that doing so can amplify onset and timbre information. We experiment with several methods for inviting pitch to join the model.

4.2.1 Source of Frame-level Pitch Labels

We consider two ways of getting pitch labels in our experiment. One is using human-labeled ground truth pitch labels provided by MusicNet. However, in real-word applications, it is hard to get 100% correct pitch labels. Hence, we also use pitch estimation predicted by a state-of-the-art multi-pitch estimator proposed by Thickstun *et al.* [43]. The author proposed a translation-invariant network which combines traditional filterbank with a convolutional neural network. The model shares parameters in the log-frequency domain, which exploits the frequency invariance of music to reduce the number of model parameters and to avoid overfitting to the training data. The model reaches the top performance in the 2017 MIREX Multiple Fundamental Frequency Estimation evaluation [1]. The average pitch estimation accuracy, evaluated using mir_eval [38], is shown in Table 1.

4.2.2 Harmonic Series Feature

Figure 1c depicts the architecture of a proposed pitch-aware model. In this model, we aim to exploit the observation that the energy distribution of the partials constitutes a key factor in the perception of instrument timbre [14]. Being motivated by [6], we propose the *harmonic series feature* (HSF) to capture the harmonic structure of music notes, calculated as follows. We are given the input pitch estimate (or ground truth) $\mathbf{P}_0 \in \mathcal{R}^{258 \times 88}$, which is a matrix with the same size as the CQT matrix. The entries in $\mathbf{P}_0$ take the value of either 0 or 1 in the case of ground truth pitch labels, and the value in $[0, 1]$ in the case of estimated pitches. If the value of an entry is close to 1, we know that likely a music note with the fundamental frequency is active on that time frame.

First, we construct a *harmonic map* that shifts the active entries in $\mathbf{P}_0$ upwards by a multiple of the corresponding fundamental frequency (f_0). That is, the (t, f)-th entry in the resulting harmonic map $\mathbf{P}_n \in \mathcal{R}^{258 \times 88}$ is nonzero only if that frequency is $(n + 1)$ times larger than an active f_0 that frame, i.e., $f = f_0 \cdot (n + 1)$.

Then, a harmonic series feature up to the $(n+1)$-th harmonics,[2] denoted as $\mathbf{H}_n \in \mathcal{R}^{258 \times 88}$, is computed by an element-wise sum of $\mathbf{P}_0, \mathbf{P}_1, \ldots$ up to $\mathbf{P}_n$, as illustrated in Figure 1c. In that follows, we also refer to $\mathbf{H}_n$ as HSF–n.

When using HSF–n as input to the instrument recognition model, we concatenate CQT $\mathbf{X}$ and $\mathbf{H}_n$ along the channel dimension, to the effect that emphasizing the partials in the input audio. The resulting matrix is then used as the input to a CNN model depicted in Figure 1c. The CNN

---

[2] We note that the first harmonic is the fundamental frequency.

(a) Baseline CNN [31]   (b) CNN + ResBlocks [10]   (c) Pitch-aware model (CQT+HSF)
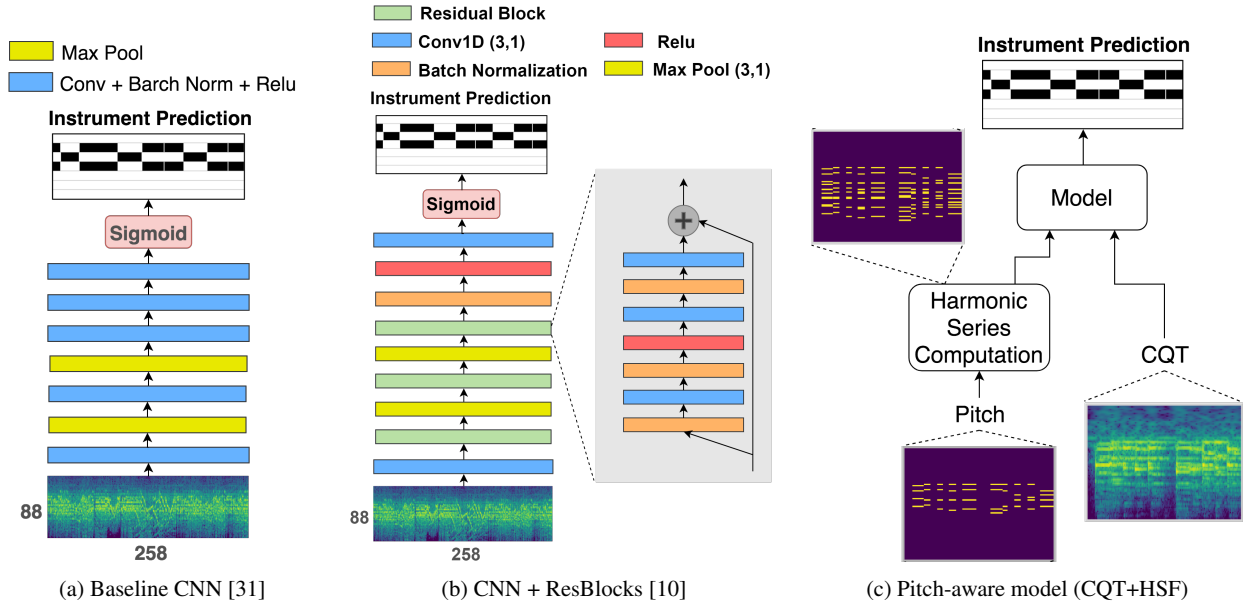
Figure 1: Three kinds of model structure used in this instrument recognition experiment.

model used here is also adapted from [10], using 1D convolutions, ResBlocks, and 11 convolutional layers in total. We call this model '*CQT+HSF–n*' hereafter.

4.2.3 Other Ways of Using Pitch

We consider another two methods to use pitch information.

First, instead of stressing the overtones, the matrix $\mathbf{P}_0$ already contains information regarding which pitches are active per time frame. This information can be important because different instruments (e.g., violin, viola and cello) have different pitch ranges. Therefore, a simple way of taking pitch information into account is to concatenate $\mathbf{P}_0$ with the input CQT $\mathbf{X}$ along the frequency dimension (which is fine since we use 1D convolutions), leading to a 258×176 matrix, and then feed it to the early convolutional layer. This method exploits pitch information right from the beginning of the feature learning process. We call it the '*CQT+pitch (F)*' method for short.

Second, we can also concatenate $\mathbf{P}_0$ with the input CQT $\mathbf{X}$ along the channel dimension, to allow the pitch information to directly influence the input CQT $\mathbf{X}$. It can tell us the pitch note and onset timing, which is critical in instrument recognition. We call this method '*CQT+pitch (C)*'.

4.3 Implementation Details

All the networks are trained using stochastic gradient descend (SGD) with momentum 0.9. The initial learning rate is set to 0.01. The weighted cross entropy, as defined below, is used as the cost function for model training:

$$l_n = -y_n[t_n \cdot \log \sigma(\hat{y}_n) + (1 - y_n) \cdot \log(1 - \sigma(\hat{y}_n))], \quad (1)$$

where y_n and $\hat{y}_n$ are the ground truth and predicted label for the n-th instrument per time frame, $\sigma(\cdot)$ is the sigmoid function to reduce the scale of $\hat{y}_n$ to $[0, 1]$, and w_n is a weight computed to emphasize positive labels and

counter class imbalance between the instruments, based on the trick proposed in [39]. Code and model are built with the deep learning framework PyTorch.

Due to the final sigmoid layer, the output of the instrument recognition model is a continuous value in $[0, 1]$ for each instrument per frame, which can be interpreted as the likelihood of the presence for each instrument. To decide the existence of an instrument, we need to pick a threshold to binarize the result. Simply setting the threshold to 0.5 equally for all the instruments may not work well. Accordingly, we implement a simple threshold picking algorithm that selects the threshold (from 0.01, 0.02, ... to 0.99, in total 99 candidates) per instrument by maximizing the F1-score on the training set.

F1-score is the harmonic mean of precision and recall. In our experiments, we compute the F1-score independently (by concatenating the result for all the segments) for each instrument and then report the average result across instruments as the performance metric.

We do not implement any smoothing algorithm to post-process the recognition result, though this may help [28].

5. PERFORMANCE STUDY

The evaluation result is shown in Table 2. We first examine the result between two models without pitch information. From the first and second rows, we see that adding Res-blocks indeed leads to a more accurate model. Therefore, we also use Res-blocks for the pitch-aware models.

We then examine the result when we use ground truth pitch labels to inform the model. From the upper half of Table 2, pitch-aware models (i.e., CQT+HSF) indeed outperform the models that only use CQT. While the CQT-only model based on [10] attains 0.887 average F1-score, the best model CQT+HSF-3 reaches 0.933. Salient improvement is found for *Viola*, *Clarinet*, and *Bassoon*.

Pitch source	Method	Piano	Violin	Viola	Cello	Clarinet	Bassoon	Horn	Avg.
none	CQT only (based on [31])	0.972	0.934	0.798	0.909	0.854	0.816	0.770	0.865
	CQT only (based on [10])	**0.982**	**0.956**	**0.830**	**0.933**	**0.894**	**0.822**	**0.789**	**0.887**
groud-truth pitch	CQT+HSF–1	**0.999**	**0.986**	**0.916**	**0.972**	**0.945**	0.909	0.776	0.929
	CQT+HSF–2	0.997	0.984	0.912	0.968	0.941	0.906	0.799	0.930
	CQT+HSF–3	0.997	0.985	0.914	0.971	0.944	0.907	0.810	**0.933**
	CQT+HSF–4	0.997	**0.986**	0.909	0.969	0.944	0.904	**0.815**	0.932
	CQT+HSF–5	0.998	0.975	0.902	0.968	0.942	**0.912**	0.803	0.928
estimated pitch by [43]	CQT+HSF–1	0.983	0.955	**0.841**	0.935	0.901	0.822	0.793	0.890
	CQT+HSF–2	0.983	0.954	0.830	0.933	0.899	0.820	0.800	0.889
	CQT+HSF–3	0.983	0.955	0.829	0.934	0.903	0.818	**0.805**	0.890
	CQT+HSF–4	0.981	0.955	0.833	**0.937**	0.903	0.831	0.793	0.890
	CQT+HSF–5	**0.984**	0.956	0.835	0.935	**0.915**	**0.839**	0.805	**0.896**
	CQT+Pitch (F)	0.983	0.955	0.829	0.936	0.887	0.819	0.791	0.886
	CQT+Pitch (C)	0.982	**0.958**	0.819	0.921	0.898	0.827	0.794	0.886

Table 2: Recognition accuracy (in F1-score) of model with and without pitch information, using either ground truth pitches or estimated pitches. We use bold font to highlight the best result per instrument for the three groups of results.
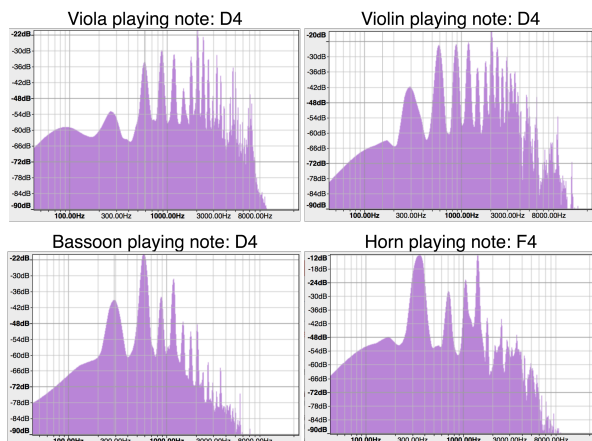

Figure 2: Harmonic spectrum of Viola (top left), Violin (top right), Bassoon (bottom left) and Horn (bottom right), created by the software Audacity [42] for real-life recordings of instruments playing a single note.

Moreover, a comparison among the pitch-aware models shows that different instruments seem to prefer different numbers of harmonics n. *Horn* and *Bassoon* achieve best F1-score with larger n (i.e., using more partials), while *Viola* and *Cello* achieves best F1-score with smaller n (using less partials). This is possibly because string instruments have similar amplitudes for the first five overtones, as Figure 2 exemplifies. Therefore, when more overtones are emphasized, it may be hard for the model to detect those trivial difference, and this in turn causes confusion between similar string instruments. In contrast, there is salient difference in the amplitudes of the first five overtones for *Horn* and *Bassoon*, making HSF–5 effective.

Figure 3 shows qualitative result demonstrating the prediction result for four clips in the test set. By comparing the result of the first two rows and the last row, we see that onset frames are clearly identified by the HSF-based model.

Furthermore, when adding HSF, it seems easier for a model to distinguish between similar instruments (e.g., violin versus viola). These examples show that adding HSF helps the model learn onset and timbre information.

Next, we examine the result when we use pitch estimation provided by the model of Thickstun *et al.* [43]. We know already from Table 1 that multi-pitch estimation is not perfect. Accordingly, as shown in the last part of Table 2, the performance of the pitch-aware models degrades, though still better than the model without pitch information. The best result is obtained by CQT+HSF–5, reaching 0.896 average F1-score. Except for *Violin*, CQT+HSF–5 outperforms CQT-only for all the instruments. We see salient improvement for *Viola*, *Clarinet*, *Bassoon* and *Horn*, for which the CQT-only model performs relatively worse. This shows that HSF helps highlight differences in the spectral patterns of the instruments.

Besides, similar to the case when using ground truth pitch labels, when using the estimated pitches, we see that *Viola* still prefers using fewer harmonic maps, whereas *Bassoon* and *Horn* prefer more. Given the observation that different instruments prefer different number of harmonics, it may be interesting to design an automatic way to dynamically decide the number of harmonic maps per frame, to further improve the result.

The fourth row of Figure 3 gives some result for CQT+HSF–5 based on estimated pitches. Compared to the result of CQT only (second row), we see that CQT+HSF–5 nicely reduces the confusion between *Violin* and *Viola* for the solo violin piece, and reinforces the onset timing for the string quartet piece.

Moving forward, we examine the result of the other two pitch-based methods, CQT+Pitch (F) and CQT+Pitch (C), using again estimated pitches. From the last two rows of Table 2, we see that these two methods do not perform better than even the second CQT-only baseline. As these two pitch-based methods take the pitch estimates directly as the model input, we conjecture that they are more sensitive

Figure 3: Prediction results of different methods for four test clips. The first row shows the ground truth frame-level instrument labels, where the horizontal axis denotes time. The other rows show the frame-level instrument recognition result for a model that only uses CQT ('CQT only'; based on [10]) and three pitch-aware models that use either ground truth or estimated pitches. We use black shade to indicate the instrument(s) that are considered active in the labels or in the recognition result in each time frame.
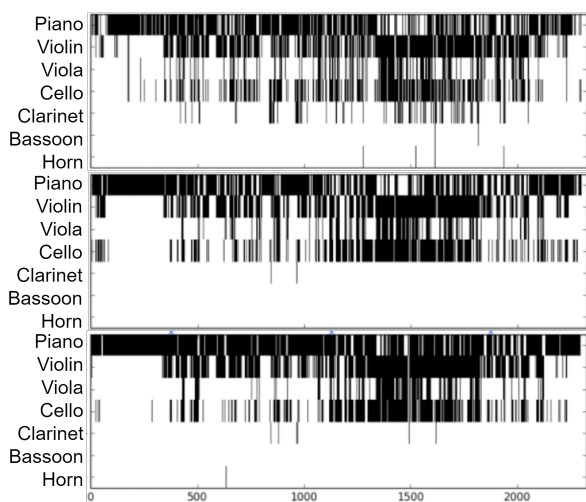

Figure 4: Frame-level instrument recognition result for a pop song, *Make You Feel My Love* by Adele, using the baseline CNN [31] (top), CNN + Res-blocks [10] (middle) and CQT+HSF–5 using estimated pitches (bottom).

to errors in multi-pitch estimation and accordingly cannot perform well. From the recognition result of the string quartet clip in the third row of Figure 3, we see that the CQT+Pitch (F) method cannot distinguish between similar instruments such as *Violin* and *Viola*. This suggests that HSF might be a better way to exploit pitch information.

Finally, out of curiosity, we test our models on a famous pop music (despite that our models are trained on classical music). Figure 4 shows the prediction result for the song *Make You Feel My Love* by Adele. It is encouraging to see

that our models correctly detect the *Piano* used throughout the song and the string instruments used in the middle solo part. Moreover, they correctly give almost zero estimate for the wind and brass instruments. Moreover, when using the Res-blocks, the prediction errors on *clarinet* are reduced. When using the pitch-aware model, the prediction errors on *Violin* and *Cello* at the beginning of the song are reduced. Besides, *Piano* timbre can also be strengthened when *Piano* and the strings play together at the bridge.

6. CONCLUSION

In this paper, we have proposed several methods for frame-level instrument recognition. Using CQT as the input feature, our model can achieve 88.7% average F1-score for recognizing seven instruments in the MusicNet dataset. Even better result can be obtained by the proposed pitch-aware models. Among the proposed methods, the HSF-based models achieve the best result, with average F1-score 89.6% and 93.3% respectively when using estimated and ground truth pitch information.

In future work, we will include MedleyDB to our training set to cover more instruments and music genres. We also like to explore joint learning frameworks and recurrent models (e.g., [8, 9, 20]) for better accuracy.

7. ACKNOWLEDGEMENT

8. REFERENCES

[1] MIREX multiple fundamental frequency estimation evaluation result, 2017. [Online] http:

```
//www.music-ir.org/mirex/wiki/2017:
Multiple_Fundamental_Frequency_
Estimation_%26_Tracking_Results_
-_MIREX_Dataset.
```

[2] Giulio Agostini, Maurizio Longari, and Emanuele Pollastri. Musical instrument timbres classification with spectral features. *EURASIP Journal on Applied Signal Processing*, 1:5–14, 2003.

[3] Kristina Andersen and Peter Knees. Conversations with expert users in music retrieval and research challenges for creative MIR. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 122–128, 2016.

[4] Jayme Garcia Arnal Barbedo and George Tzanetakis. Musical instrument classification using individual partials. *IEEE Trans. Audio, Speech, and Language Processing*, 19(1):111–122, 2011.

[5] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello1. MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2014. [Online] http://medleydb.weebly.com/.

[6] Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan P. Bello. Deep salience representations for $f0$ estimation in polyphonic music. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 63–70, 2017.

[7] Juan J. Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 559–564, 2012. [Online] http://mtg.upf.edu/download/datasets/irmas/.

[8] Ning Chen and Shijun Wang. High-level music descriptor extraction algorithm based on combination of multi-channel CNNs and LSTM. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 509–514, 2017.

[9] Keunwoo Choi, Gyorgy Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2017.

[10] Szu-Yu Chou, Jyh-Shing Jang, and Yi-Hsuan Yang. Learning to recognize transient sound events using attentional supervision. In *Proc. Int. Joint Conf. Artificial Intelligence*, 2018.

[11] Jia Deng et al. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2009.

[12] Aleksandr Diment, Padmanabhan Rajan, Toni Heittola, and Tuomas Virtanen. Modified group delay feature for musical instrument recognition. In *Proc. Int. Symp. Computer Music Multidisciplinary Research*, 2013.

[13] Zhiyao Duan, Jinyu Han, and Bryan Pardo. Multi-pitch streaming of harmonic sound mixtures. *IEEE/ACM Trans. Audio, Speech, and Language Processing*, 22(1):138–150, 2014.

[14] Zhiyao Duan, Yungang Zhang, Changshui Zhang, and Zhenwei Shi. Unsupervised single-channel music source separation by average harmonic structure modeling. *IEEE Trans. Audio, Speech, and Language Processing*, 16(4):766 – 778, 2008.

[15] Slim Essid, Gaël Richard, and Bertrand David. Musical instrument recognition by pairwise classification strategies. *IEEE Trans. Audio, Speech, and Language Processing*, 14(4):1401–1412, 2006.

[16] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[17] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC Music Database: Popular, classical and jazz music databases. In *Proc. Int. Society of Music Information Retrieval Conf.*, pages 287–288, 2002. [Online] https://staff.aist.go.jp/m.goto/RWC-MDB/rwc-mdb-i.html.

[18] Matt Hallaron et al. University of Iowa musical instrument samples. University of Iowa, 1997. [Online] http://theremin.music.uiowa.edu/MIS.html.

[19] Yoonchang Han, Jaehun Kim, and Kyogu Lee. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Trans. Audio, Speech, and Language Processing*, 25(1):208 – 221, 2017.

[20] Curtis Hawthorne, Erich Elsen adn Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *Proc. Int. Soc. Music Information Retrieval Conf.*, 2018.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2016.

[22] Shawn Hershey et al. CNN architectures for large-scale audio classification. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2017.

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conf. Machine Learning*, pages 448–456, 2015.

[24] 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*, 17(1):174–186, 2009.

[25] Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Instrument identification in polyphonic music: Feature weighting with mixed sounds, pitch-dependent timbre modeling, and use of musical context. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 558–563, 2005.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[27] Peter Li, Jiyuan Qian, and Tian Wang. Automatic instrument recognition in polyphonic music using convolutional neural networks. *arXiv preprint arXiv:1511.05520*, 2015.

[28] Dawen Liang, Matthew D. Hoffman, and Gautham J. Mysore. A generative product-of-filters model of audio. In *Proc. Int. Conf. Learning Representations*, 2014.

[29] Hyungui Lim, Jeongsoo Park, Kyogu Lee, and Yoon-chang Han. Rare sound event detection using 1D convolutional recurrent neural networks. In *Proc. Int. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[30] Tsung-Yi Lin et al. Microsoft COCO: Common objects in context. In *Proc. European Conf. Computer Vision*, pages 740–755, 2014.

[31] Jen-Yu Liu and Yi-Hsuan Yang. Event localization in music auto-tagging. *Proc. ACM Int. Conf. Multimedia*, pages 1048–1057, 2016.

[32] Arie Livshin and Xavier Rodet. The significance of the non-harmonic "noise" versus the harmonic series for musical instrument recognition. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2006.

[33] Vincent Lostanlen and Carmine-Emanuele Cella. Deep convolutional networks on the pitch spiral for musical instrument recognition. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 612–618, 2016.

[34] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proc. Python in Science Conf.*, pages 18–25, 2015. [Online] https://librosa.github.io/librosa/.

[35] Taejin Park and Taejin Lee. Musical instrument sound classification with deep convolutional neural network using feature fusion approach. *arXiv preprint arXiv:1512.07370*, 2015.

[36] Kumar Ashis Pati and Alexander Lerch. A dataset and method for electric guitar solo detection in rock music. In *Proc. Audio Engineering Soc. Conf.*, 2017.

[37] Jordi Pons, Thomas Lidy, and Xavier Serra. Experimenting with musically motivated convolutional neural networks. In *Proc. Int. Workshop on Content-based Multimedia Indexing*, 2016.

[38] Colin Raffel, Brian Mcfee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: a transparent implementation of common mir metrics. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2014. [Online] https://github.com/craffel/mir_eval.

[39] Rif A. Saurous et al. The story of audioset, 2017. [Online] http://www.cs.tut.fi/sgn/arg/dcase2017/documents/workshop_presentations/the_story_of_audioset.pdf.

[40] Jan Schlüter. Learning to pinpoint singing voice from weakly labeled examples. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2016.

[41] Christian Schoerkhuber and Anssi Klapuri. Constant-Q transform toolbox for music processing. In *Proc. Sound and Music Computing Conf.*, 2010.

[42] Audacity Team. Audacity. https://www.audacityteam.org/, 1999-2018.

[43] John Thickstun, Zaid Harchaoui, Dean P. Foster, and Sham M. Kakade. Invariances and data augmentation for supervised music transcription. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2018. [Online] https://github.com/jthickstun/thickstun2018invariances.

[44] 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.

[45] Ju-Chiang Wang, Hsin-Min Wang, and Shyh-Kang Jeng. Playing with tagging: A real-time tagging music player. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pages 77–80, 2014.

[46] Hanna Yip and Rachel M. Bittner. An accurate open-source solo musical instrument classifier. In *Proc. Int. Soc. Music Information Retrieval Conf., Late-Breaking Demo Paper*, 2017.

[47] Li-Fan Yu, Li Su, and Yi-Hsuan Yang. Sparse cepstral codes and power scale for instrument identification. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pages 7460–7464, 2014.