A HARMONIC STRUCTURE-BASED NEURAL NETWORK MODEL FOR MUSICAL PITCH DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper we design a harmonic acoustic model for pitch detection. This model arranges conventional convolution and sparse convolution in a way such that the global harmonic patterns captured by sparse convolution are composed of the large number of local patterns captured by layers of conventional convolution. When trained on the MAPS dataset, the harmonic model outperforms all existing pitch detection systems trained on the same dataset. Most impressively, when trained on MAPS with simple data augmentation, the harmonic model with an LSTM layer on top surpasses an up-to-date, more complex pitch detection system trained on the MAESTRO dataset to which complicated data augmentation is applied and whose training split is an order-of-magnitude larger than the training split of MAPS. The harmonic model has demonstrated potential to be used for advanced automatic music transcription (AMT) systems.

1 INTRODUCTION

1.1 BACKGROUND

In recent years, deep learning has emerged as a promising approach to pitch detection. Pitch detection is the process of detecting the pitches present in a frame, i.e., a short snippet of musical waveform. The results of pitch detection can be post-processed to extract note contours, i.e., note onsets and offsets. The whole process of pitch detection and note extraction is called automatic music transcription (AMT). This paper is devoted to pitch detection for solo piano music. For AMT, of first importance is an acoustic model that can predict the active pitches in a frame. Using acoustic models as building blocks, more advanced architectures can be constructed for various purposes.

Kelz et al. (2016) designed an acoustic model for pitch detection, which is referred to as *the Kelz model* hereafter. This model was modified from the one developed in Schlüter & Böck (2014) for onset detection and resembles the LeNet-5 model (Lecun et al., 1998). The Kelz model treats pitch detection simply as image-related tasks by using convolution layers to capture local frequency patterns. It does not explicitly capture the harmonic patterns of pitched music. Instead, it relies on fully connected layers to this end. The above handling of harmonic patterns weakens the model's generalisation capability. This problem has been partially studied in Kelz & Widmer (2017).

Hawthorne et al. (2018) designed an AMT system. This system consists of an onset detector and a frame detector that detects the pitches in each frame. The two detectors have similar structures by topping the Kelz model with a bi-directional LSTM. Skip connections are featured by feeding the onset detector's output into the other detector to serve as additional intermediate features. Hawthorne et al. (2019) used a similar AMT system as a sub-system to build a bigger system for piano sound generation. The AMT system in Hawthorne et al. (2019) uses more features and introduces a separate detector for offset detection. The dataset used in Hawthorne et al. (2019) is far larger than the one used in Hawthorne et al. (2018).

Kelz et al. (2019) designed an AMT system consisting of three separate detectors for pitch, onset and offset detection. The pitch detector is a Kelz model. Some intermediate features of the pitch detector are used as the sole inputs to the other two detectors. Both the onset and offset detectors only have a convolution and a fully connected layer. Finally, note contours are extracted by fusing the predictions from the three detectors with a hidden Markov chain model (HMM).

Bittner et al. (2017) designed a fully convolutional acoustic model named the harmonic constant-q transform (HCQT) model. It uses HCQTs as input representation. HCQTs are constructed from CQTs, which like any spectrograms have a time and a frequency dimension. Besides the above two dimensions, HCQTs have a newly added dimension called the harmonic dimension. The harmonic dimension consists of the fundamental frequency/pitch (denoted by f_0), a sub-harmonic at $\frac{1}{2}f_0$, and four harmonics ranging from $2f_0$ to $5f_0$. Rearranging CQTs into HCQTs enables the capture of harmonic patterns/structures specific to pitched music, whereas the convolutional architecture enables the capture of local patterns as in most image-related problems.

Elowsson (2018) designed an AMT system consisting of six cascaded networks that are trained one after another. Skip connections are used among these networks in a similar fashion as residual networks do. Except the first network, all other networks are multilayer perceptrons (MLPs) that have two to three layers. The first network N1 takes variable-q transforms (VQTs) as inputs and detects tentative pitches by linearly combing 50 frequency bins for each pitch that include harmonics and non-harmonics. N1 is essentially a convolution layer with a single sparse kernel. A second network N2 performs more accurate pitch estimation from the output of N1. The overall effect of N1 and N2 is equivalently an acoustic model that resembles the HCQT model. The difference is that N1 and N2 take into account more frequency bins when capturing harmonic patterns, whereas the HCQT model pays more attention to local patterns. The remaining four networks estimate onsets, offsets and tentative notes, and compute probabilities for each note, respectively.

1.2 Motivation and Contribution

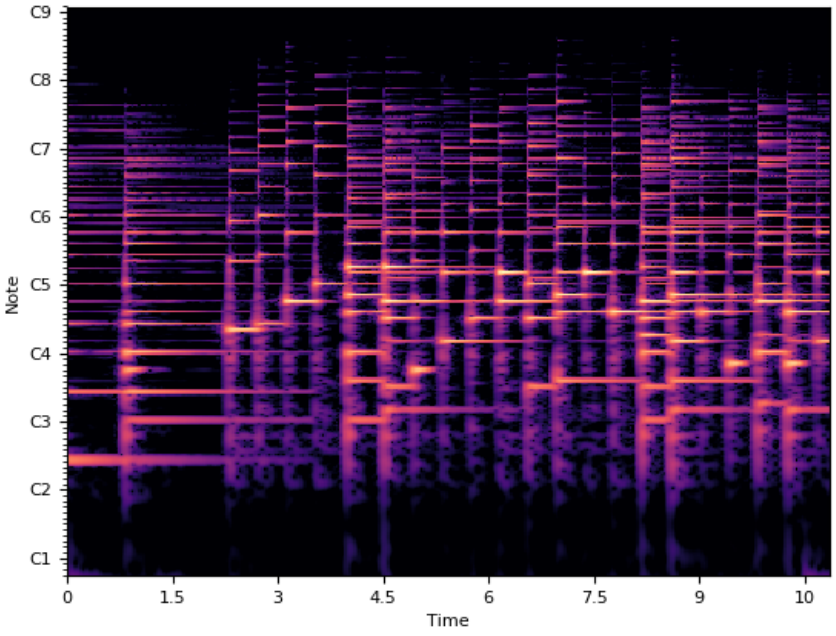

Figure 1: An example VQT spectrogram of 10 seconds.

Loosely speaking, the waveform of a music note is composed of a number of monotones, among which the lowest frequency is called the fundamental frequency or *the pitch*, and all the other are integer multiples of the pitch referred to as *the harmonics*. For polyphonic music, there can be multiple notes active at the same time; and their pitches and harmonics often overlap. This leads to the challenge for pitch detection. Figure 1 shows an example VQT spectrogram. This spectrogram features a grid structure composed of frequency stripes. These stripes are a result of non-perfect pitches

and the windowing effect (also known as power leakage) in the calculation of the discrete Fourier transform (DFT). We call these stripes *the local patterns*, as they only involve a small number of local neighbouring frequencies.

Next let us talk about harmonic patterns. To tell if a specific note is active, intuitively we will first check if its pitch (denoted by f_0) and harmonics are light (i.e., have energy). If it is the case, then it is likely the note is active. On the other hand, if it is also light at $\frac{1}{2}f_0$ or $\frac{1}{3}f_0$, then the likelihood drops, because now f_0 could be the second or third harmonic of another note. When the degree of polyphony increases, this logical reasoning is definitely over our heads and deep learning comes in handy. We call the interaction patterns among the pitches and harmonics of different notes *the harmonic patterns*.

Since the constituent frequencies of harmonic patterns are sparsely distributed, it is inappropriate to capture harmonic patterns with the conventional convolution whose receptive field can cover only contiguous frequencies. Instead, we need a type of convolution whose receptive field in the frequency dimension corresponds to the above sparsely distributed frequencies. We term this type of convolution *sparse convolution*.

For pitch detection, it is critical as well as challenging to capture both of the above two types of frequency patterns. The existing acoustic models focused only on either of them. The existing AMT systems concentrated on more advanced, complex network structures. The contribution of this paper is as follows.

1. We design a harmonic acoustic model for pitch detection. This model takes VQTs as inputs. In the first part of this model, the stripe-shaped local frequency patterns are captured with layers of conventional convolution. Then in the second part, the global harmonic patterns are captured with sparse convolution.
2. When all the systems participating in the comparison are trained on MAPS, the harmonic model achieves the best performance, leading the runner-up by 3.5%.
3. When trained on MAPS with simple data augmentation, the harmonic model enhanced by an LSTM layer outperforms an up-to-date, more complex system trained on the MAESTRO dataset to which complicated data augmentation is applied and whose training split is 15 times as large as the training split of MAPS, demonstrating the potential of the harmonic model to be used for building advanced AMT systems.

2 DATASETS

2.1 MAPS

MAPS (Emiya et al., 2010) is a piano dataset generated by a software synthesizer and a real piano, namely, a Yamaha Disklavier upright piano, from standard MIDI files. Disklavier can be controlled by computer, and accept MIDI files as inputs and output MIDI files that are strictly synchronized with the sound generated. These output MIDI files can be used to generate ground-truth labels. The sound from Disklavier was recorded under two settings, namely, a close and an ambient setting. The synthesizer has 7 settings that differ in soundfont and reverberation configuration. Thus, there are 9 settings in total. Each setting has 30 recordings, resulting in a total of 270 recordings. Since there are only 160 musical compositions, these 9 settings have overlap in composition.

Next we need to partition the dataset into three splits, namely, a training, a validation and a test split. In this process, the general criterion is that the training and test splits should have no instrument and composition overlap so as to fairly compare the generalisation capability of different models. We choose the 60 recordings from Disklavier as the test split. We exclude the 71 recordings whose compositions appear in the test split from the 210 synthesized recordings and use the remaining 139 recordings as the training split. We use the above 71 recordings as the validation split.

2.2 MAESTRO

MAESTRO (Hawthorne et al., 2019) is a piano dataset generated by Yamaha Disklavier grand pianos from 9 years of an international piano competition. It has 1184 recordings in total that have a total

duration of about 172 hours and a total size of about 103 GB. By contrast, MAPS only has a total duration of about 18 hours and a total size of about 11 GB. When partitioning MAESTRO into training, validation and test splits, we simply follow the recommendation given in Hawthorne et al. (2019). In particular, the training split has 954 recordings, the validation split has 105 recordings, and the test split has 125 recordings. The training split of MAESTRO is about 15 times as large as the training split of MAPS.

3 MODEL CONFIGURATION

3.1 INPUT REPRESENTATION

The harmonic model uses VQT as input representation. The procedure for computing VQT is given in appendix A.1. The setting for VQT computation is as follows. The sampling rate is 44.1 kHz. The number of frequency bins per octave (denoted by B) is 36. The minimum frequency is the frequency of MIDI note 21 multiplied by a factor of $2^{-1/B}$. The maximum frequency is the frequency of MIDI note 132 multiplied by a factor of $2^{1/B}$. Thus, we have 336 frequency bins in total. The bandwidth for each frequency bin is set to

$$\Omega_k = \max\left(f_k(2^{1/B}–2^{-1/B}), 14.1\right) \quad \text{Hz} \tag{1}$$

where f_k is the centre frequency. The hop size (denoted by h) is 64 samples. For pitch detection, we do not like the hop size to be too small, because in this case adjacent frames in the spectrogram will be highly correlated. So we down-sample the resulting spectrogram by a factor of 22.

3.2 LABELLING

We formulate frame-wise pitch detection as a multi-label classification problem. Frame labels are determined from the MIDI file. First, we translate the use of the sustain pedal into extended note duration as in Hawthorne et al. (2018). Next, we express the onset and offset for a specific note in terms of samples as

$$\begin{cases} s_{\text{on}} = \lfloor t_{\text{on}} \times \text{sr} \rfloor, \\ s_{\text{off}} = \lfloor t_{\text{off}} \times \text{sr} \rfloor, \end{cases} \tag{2}$$

where t_{on} and t_{off} are the onset and offset times in seconds, respectively. Finally, the start and end frames of this note can be expresses as

$$\begin{cases} f_{\text{on}} = \lfloor (s_{\text{on}} + h/2)/h \rfloor, \\ f_{\text{off}} = \lfloor (s_{\text{off}} + h/2 - 1)/h \rfloor. \end{cases} \tag{3}$$

3.3 NETWORK STRUCTURE

We propose a harmonic acoustic model for frame-wise pitch detection. The structure of this model is given in Table 1. In this table, we use – to mean that the output shape of a layer is the same as the output shape of the above layer. Layer 0 is the input with shape (none $\times$ none $\times$ 336 $\times$ 1) where the dimension order is batch, frame, frequency and channel. Here we use none to denote a dynamic dimension. The abbreviations used in this table are defined as follows. 1) conv($32 \times 3 \times 3$) stands for a convolution layer with 32 kernels of receptive field 3×3. 2) dropout(0.8) is a dropout layer with keeping probability 0.8. 3) sparse-conv($256 \times 1 \times 79$) is a sparse convolution layer with 256 kernels of receptive field 1×79. 4) maxpool(1×3) is a max-pooling layer with receptive 1×3 , frame stride 1, and frequency stride 3. 5) FC(n) is a fully connected layer with n features.

This structure can be divided into three parts. Part 1 consists of layers 1 through 7. This part uses four consecutive convolution layers to capture local frequency patterns. The overall receptive field of these layers in the frequency domain is 9 frequency bins. Part 2 consists of layers 8, 9 and 10. Layer 8 is a sparse convolution layer. This layer does the same job as network N1 of Elowsson (2018). In particular, for each pitch f_0 we select 50 frequency bins relative to f_0 as the input features for detecting the presence of f_0. These 50 frequency bins include bins over and under f_0,

Table 1: Structure of the harmonic acoustic model

LAYER NO.	OPERATION	OUTPUT SHAPE
0	input(none $\times$ none $\times$ 336 $\times$ 1)	none $\times$ none $\times$ 336 $\times$ 1
1	conv(32 $\times$ 3 $\times$ 3)	none $\times$ none $\times$ 336 $\times$ 32
2	conv(32 $\times$ 3 $\times$ 3)	–
3	dropout(0.8)	–
4	conv(32 $\times$ 3 $\times$ 3)	–
5	dropout(0.8)	–
6	conv(32 $\times$ 3 $\times$ 3)	–
7	dropout(0.8)	–
8	sparse-conv(256 $\times$ 1 $\times$ 79)	none $\times$ none $\times$ 264 $\times$ 256
9	maxpool(1 $\times$ 3)	none $\times$ none $\times$ 88 $\times$ 256
10	dropout(0.8)	–
11	FC(64)	none $\times$ none $\times$ 88 $\times$ 64
12	dropout(0.8)	–
13	FC(1)	none $\times$ none $\times$ 88

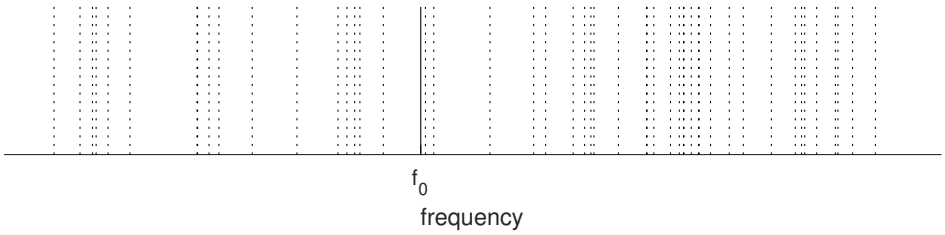

Figure 2: The receptive field of the sparse convolution in the frequency dimension where the solid vertical line is the pitch and the dotted vertical lines are the input features for the pitch (Elowsson, 2018).

and harmonics and non-harmonics. Please refer to Elowsson (2018) for a complete list of these bins. Figure 2 shows the distribution of these bins. The original 50 bins were given when the number of bins per octave is 240. And when converted to 36 bins per octave used by the harmonic model, some bin indices become non-integers. For these non-integers, we take both their floors and ceils. Thus, for each f_0 we have a total of 79 input features. For an f_0, some of its input features could be out of the VQT's bin range. In this case, we assume that the out-of-range input features are zeros. Since a typical piano only has 88 keys ranging from MIDI notes 21 to 108, among the VQT's 336 frequency bins only the first 264 are pitches to detect. Therefore, after the sparse convolution layer, we get an output of shape (none $\times$ none $\times$ 264 $\times$ 256) where 256 is the number of output features for each pitch. This output is at pitch level. However, the datasets only allow labels at note level. So we use max-pooling of receptive field 1 $\times$ 3 to down-sample the output to note level, resulting in an output of shape (none $\times$ none $\times$ 88 $\times$ 256). Part 3 consists of layers 11, 12 and 13, after which we get predictions for the 88 notes. In the above structure except the last layer, when it is applicable, ReLU is used as activation function and the output is batch-normalized. The last layer uses sigmoid as activation function. We use dropout at different places to control overfitting.

The difference between the harmonic model and the Kelz model is that the former explicitly captures harmonic frequency patterns with sparse convolution, whereas the latter does this implicitly by letting the network to learn them on itself. This implicit handling of harmonic patterns makes the trained model overfit the timbres and composition styles of the training split. However, the training and test splits often have different timbres and composition styles so that this implicit handling will impact the generalisation capability of the trained model.

The harmonic model differs from network N1 of Elowsson (2018) in two aspects. First, the harmonic model captures enough number of local frequency patterns with part 1, whereas N1 did not consider

Table 2: Comparison of pitch detection systems tested on MAPS (ensemble/average in percentage)

		F	P	R
the Kelz model (Kelz et al., 2016)		/71.60	/81.18	/65.07
Elowsson (2018)		72.9/	84.1/	64.4/
Hawthorne et al. (2018)		/78.30	/88.53	/70.89
Kelz et al. (2019)		/77.16	/90.73	/67.85
Hawthorne et al. (2019)	w/o data aug.	/82.02	N/A	N/A
	w/ data aug.	/84.91	/92.86	/78.46
the harmonic model	w/o data aug. & LSTM	81.33/81.78	86.86/86.68	76.47/77.89
	w/ data aug., w/o LSTM	83.16/83.42	85.52/85.44	80.93/81.96
	w/ data aug. & LSTM	85.85/85.82	87.33/87.80	84.41/84.42

these patterns. Second, the harmonic model has 256 output features for each pitch, whereas N1 only has one.

Compared with the HCQT model, the harmonic model is able to capture more complex frequency patterns by using more features for each pitch and placing sparse convolution after conventional convolution. The overall receptive field of the convolution layers in the HCQT model is over one octave and thus could lead to overfitting of the composition styles. This problem has been advertently avoided in both the Kelz and the harmonic model by using a relatively small overall receptive filed when capturing local frequency patterns.

3.4 Loss Function

We use the binary cross entropy loss. Denote by $p \in \{0, 1\}$ the ground-truth label for a note in a frame and by $\hat{p} \in [0, 1]$ the predicted probability for this note. The loss for this note is formulated as

$$l \triangleq -p \ln(\hat{p}) - (1 - p) \ln(1 - \hat{p}). \tag{4}$$

3.5 Performance Measure

As a convention, the performance of pitch detection is solely measured by the f-measure defined as

$$F \triangleq \frac{2PR}{P + R}. \tag{5}$$

In the above equation P and R are the precision and recall defined, respectively, as

$$\begin{cases} P \triangleq \dfrac{\text{TPs}}{\text{TPs} + \text{FPs}}, \\ R \triangleq \dfrac{\text{TPs}}{\text{TPs} + \text{FNs}}, \end{cases} \tag{6}$$

where TPs is the number of true positives, FPs is the number of false positives, and FNs is the number of false negatives. When there is more than one recording, the above metrics can be calculated in two ways. We can first calculate them for individual recordings and then average these results. We call metrics obtained this way the average results. We can alternatively treat all the recordings as an ensemble and directly calculate the metrics. We refer to metrics obtained this way the ensemble results.

4 Experiments

In this section, we will compare performance of the harmonic model, the Kelz model and other more complex pitch detection systems built upon acoustic models. To enhance the performance, the existing systems exploited various techniques/tricks, such as data augmentation (Hawthorne et al., 2019), RNN (Hawthorne et al., 2018; 2019), HMM (Kelz et al., 2019), joint-task training (Hawthorne et al.,

2018; 2019; Kelz et al., 2019), and larger training split (Elowsson, 2018). Therefore, for a fair comparison we will apply two tricks to the harmonic model, namely, data augmentation and RNN. The first type of data augmentation is pitch shift whose procedure is detailed in appendix A.2. The second type is to change the powers of different frequencies by multiplying the amplitude of each frequency bin by a random number ranging from 0.9 to 1.1. The third type is to use the two sound channels of each recording in the training split as independent training examples. In this case, for testing we will average the logits of the two sound channels. To enhance the harmonic model, we can top the harmonic model with an LSTM layer. Specifically, in this case we first train the harmonic model. Next, we remove layers 11, 12 and 13 and keep the parameters of the remaining layers fixed and untrainable. Then we replace these removed layers with an LSTM of 64 hidden units, which is in turn followed by an FC(1) layer (i.e., a fully connected layer that converts the number of features for each note from 64 to 1).

We use Tensorflow $1.13.1$ (Abadi et al., 2016) as the neural network framework and implement sparse convolution ourselves. The optimizer is the Adam optimizer. For the harmonic model without LSTM on top, each example has 300 frames; the batch size is 2; and the learning rate is 10^{-4}. For the harmonic model with LSTM on top, each example has 600 frames; the batch size is 1; and the learning rate is 10^{-3}.

Table 2 compares the performance of different pitch detection systems. The system in Elowsson (2018) was trained on a self-made training dataset that was purely synthesized and does not overlap in musical composition with the test split of MAPS. The system in Hawthorne et al. (2019) was trained on MAESTRO. Hawthorne et al. (2019) also augmented the training split by pitch shift, compressing, equalising, and adding reverberation and pink noise. The results of the Kelz model are cited from Hawthorne et al. (2018) which implemented this model and tested it on MAPS by following the training-test split partition given in section 2.1. Note that MAESTRO cannot be used both for training and for testing, because this has the problem of instrument overlap. Therefore, to objectively access a system's generalisation capability, we only compare the test performance of different systems on the test split of MAPS. Among the existing systems, some used the ensemble results and some used the average results. For the system of Hawthorne et al. (2019) trained without data augmentation, only the f-measure is available.

In its pure form without data augmentation and LSTM, the harmonic model defeats all the existing systems except the system in Hawthorne et al. (2019). The system in Hawthorne et al. (2019) trained without data augmentation leads the pure harmonic model by 0.24. This lead is attributed to the MAESTRO's far larger training split and joint-task training with more complex network structure, as evidenced and contrasted by the results of Hawthorne et al. (2018) that were obtained on MAPS with a less complex network structure. The system of Hawthorne et al. (2019) with data augmentation surpassed the harmonic model with data augmentation but without LSTM by 1.49. However, when data augmentation and LSTM are both applied, the harmonic model outperforms the system of Hawthorne et al. (2019) with data augmentation by 0.91 and reaches a record high of 85.82. Note that, even when LSTM is applied to the harmonic model, none of the existing systems except the Kelz model is simpler than the harmonic model.

5 CONCLUSIONS

In this paper we designed a harmonic acoustic model for pitch detection. This model effectively captures the complex frequency interactions characterizing polyphonic pitched music through conventional convolution and sparse convolution inspired by the harmonic structure of pitched music. In its pure form without RNN and data augmentation, the harmonic model outperformed most of the existing pitch detection systems. Most noticeably, when trained on MAPS and data augmentation is done, the harmonic model with an LSTM layer on top outdid the complex system in Hawthorne et al. (2019) trained on MAESTRO whose training split 15 times as large as the training split of MAPS. Thus, the harmonic model has shown great potential to be used for building advanced AMT systems.

A possible future direction is to make more potential of complex spectrograms, instead of using only amplitude spectrograms. A mixture of signal can be inseparable in the real number domain but could be separable in the complex number domain. Trabelsi et al. (2018) has done some preliminary study in this direction. However, our own study showed that the technique of deep complex network

proposed in Trabelsi et al. (2018) did not yield a performance comparable with that of real networks. Therefore, definitely more can be done.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, and *et al*. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprintarXiv:1603.04467*, 2016.

Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello. Deep salience representations for F0 estimation in polyphonic music. In Sally Jo Cunningham, Zhiyao Duan, Xiao Hu, and Douglas Turnbull (eds.), *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pp. 63–70, 2017. URL https://ismir2017.smcnus.org/wp-content/uploads/2017/10/85_Paper.pdf.

Judith C. Brown. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, 1(89):425–434, January 1991.

Anders Elowsson. Polyphonic pitch tracking with deep layered learning. *CoRR*, abs/1804.02918, 2018. URL http://arxiv.org/abs/1804.02918.

Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.

Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. In Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos (eds.), *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pp. 50–57, 2018. URL http://ismir2018.ircam.fr/doc/pdfs/19_Paper.pdf.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse H. Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=r1lYRjC9F7.

Nicki Holighaus, Monika Dörfler, Gino Angelo M. Velasco, and Thomas Grill. A framework for invertible, real-time constant-q transforms. *IEEE Trans. Audio, Speech & Language Processing*, 21(4):775–785, 2013. doi: 10.1109/TASL.2012.2234114.

R. Kelz, S. Böck, and G. Widmer. Deep polyphonic ADSR piano note transcription. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pp. 246–250, May 2019. doi: 10.1109/ICASSP.2019.8683582.

Rainer Kelz and Gerhard Widmer. An experimental analysis of the entanglement problem in neural-network-based music transcription systems. In Christian Dittmar, Jakob Abeßer, and Meinard Müller (eds.), *AES International Conference Semantic Audio 2017, Erlangen, Germany, June 22-24, 2017*. Audio Engineering Society, 2017. URL http://www.aes.org/e-lib/browse.cfm?elib=18761.

Rainer Kelz, Matthias Dorfer, Filip Korzeniowski, Sebastian Böck, Andreas Arzt, and Gerhard Widmer. On the potential of simple framewise approaches to piano transcription. In Michael I. Mandel, Johanna Devaney, Douglas Turnbull, and George Tzanetakis (eds.), *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, pp. 475–481, 2016. URL https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/179_Paper.pdf.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. doi: 10.1109/5.726791.

Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pp. 6979–6983. IEEE, 2014. doi: 10.1109/ICASSP.2014.6854953.

Christian Schörkhuber and Anssi Klapuri. Constant-Q transform toolbox for music processing. In *Proceedings of 7th Sound and Music Computing Conference*, Barcelona, Spain, 2010.

C. Schörkhuber, A. Klapuri, and A. Sontacchi. Audio pitch shifting using the constant-q transform. *Journal of the Audio Engineering Society*, 61(7/8):562–572, 2013.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J. Pal. Deep complex networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1T2hmZAb.

A Appendix

A.1 Computation of VQT

Since there is no clear, standard procedure for computing VQT, here we will sketch such a procedure by ignoring the underlying details. CQT is arguably the only choice for characterizing harmonic frequency patterns due to the following advantages. First, it was born for processing musical signal of equal tempered scale, because its frequency bins are strictly log-linearly spaced and the bandwidth of each filter is proportional to the centre frequency (Brown, 1991). Second, it is preferable over the traditional wavelet transform, because the latter cannot provide sufficient frequency resolution for musical signal processing. Third, there exists a fast algorithm for CQT (Schörkhuber & Klapuri, 2010). The drawback with CQT is the low time resolution for lower frequencies. For example, when the number of frequency bins per octave is 36, the frame length for MIDI note 21 is 1.87 seconds. Therefore, CQT is not ideal for detecting lower notes. Hence there comes VQT (Holighaus et al., 2013).

In VQT the bandwidth of a filter within the filter bank can be expresses as

$$\Omega_k = f_k(2^{1/B}–2^{-1/B}) + \gamma \tag{7}$$

where f_k is the filter's centre frequency, B is the number of frequency bins per octave, and γ is a non-negative constant. When γ is zero, VQT reduces to CQT. The rationale behind VQT is that by enlarging the bandwidth we can shrink the frame length and thus get better time resolution. In order to properly apply VQT, we need to understand what the bandwidth means. In VQT, filters have strictly finite bandwidth therefore infinite length in the time domain. That is, the frame length for each frequency is infinite. We can only consider the frame length from an engineering perspective. Let us define the frame length as the length of the zone within the infinite frame that contains a major proportion of the frame's energy. If this proportion is to be over 99%, then it can be proved that the frame length is $2.88/\Omega_k$ seconds, or $2.88 \times \text{sr}/\Omega_k$ samples where sr is the sampling rate. Thus, if we like a maximum frame length of 20000 samples when sr is 44.1 kHz, we should set the minimum bandwidth to be $2.88 \times 44100/20000 = 6.35$ Hz.

Next, let us talk about zero padding. In VQT, when computing the spectral coefficients for each frequency at different times, the signals involved are cyclic shifts of the original recording. Thus, when computing the coefficient at sample zero, half the data comes from the end of the recording. We can get around this undesirable effect by zero padding. To be specific, we can pad

$$\lceil 1.44/\Omega_{\text{min}} \times \text{sr}/h \rceil \times h \tag{8}$$

zeros at each end of the recording, where h is the hop size, and then compute the VQT. After that we throw away

$$\lceil 1.44/\Omega_{\text{min}} \times \text{sr}/h \rceil \tag{9}$$

coefficients at each end of the spectrogram. Finally, let us talk about the hop size in terms of samples. In VQT, the hop size is any number such that

$$\text{sr}/h \geq \Omega_{\text{max}} \tag{10}$$

where Ω_{max} is the maximum bandwidth. After getting the spectrogram, we convert it to dB scale according to

$$20 \times \log_{10}\left(|\text{VQT}| + 10^{-10}\right) + 200 \tag{11}$$

where $|\cdot|$ is the absolute operator to get the amplitude. Adding 10^{-10} is to make the computation numerically stable. When the amplitude is zero, we get the minimum power of -200 dB. When padding values for the inputs to the first neural network layer, it is desirable that the padded values are physically meaningful. In our case, we would like these values to stand for a zero power level. Thus, these values should be -200. For convenience we alternatively shift the dB scaled spectrogram by 200 so that we can keep on using zeros for padding.

A.2 PITCH SHIFT

Step 1 Setting the number of frequency bins per octave (B) to a higher value of 84, calculate the VQT.

Step 2 As per the procedure given in Schörkhuber et al. (2013), shift the pitch of the above VQT by 0, ± 1 and ± 2 frequency bins.

Step 3 For each pitch-shifted VQT, recover the discrete Fourier transform (DFT) of the signal.

Step 4 From the above DFT, calculate the VQT for $B = 36$.