MELODY EXTRACTION USING A HARMONIC CNN

Jiří Balhar
Charles University
Institute of Formal and Applied Linguistics

Jan Hajič jr.
Charles University
Institute of Formal and Applied Linguistics

EXTENDED ABSTRACT

In the recent years the approaches to melody extraction have shifted to the use of deep learning (DL) [1, 2]. These new methods transform the input signal to a time-frequency representation and then use DL techniques to sequentially process the transformed input. The result is a saliency map from which they select the melody f_0 trajectory and determine voicing. Among new DL-based works we can find variety of approaches that try new signal transformations or network topologies. On the other hand the building blocks out of which these systems are built are standard (fully connected layers, CNNs and RNNs). In this paper we propose a new kind of CNN architecture specialized for processing harmonic signals in audio.

A harmonic signal is usually comprised of a fundamental frequency and overtones that are spaced apart by constant ratios, these features are therefore non-local on a input time-frequency representation. In practice CNN kernels are usually small (previous works use usually a size of a semitone in the frequency axis [1, 2]) and therefore cannot process the whole harmonic structure in one layer. To make up for this, existing works also include a final "big" convolutional layer to "capture relationships between frequency content within a octave" [2]. This is only a partial solution because only one layer of the whole network can exploit the defining characteristic of a harmonic signal.

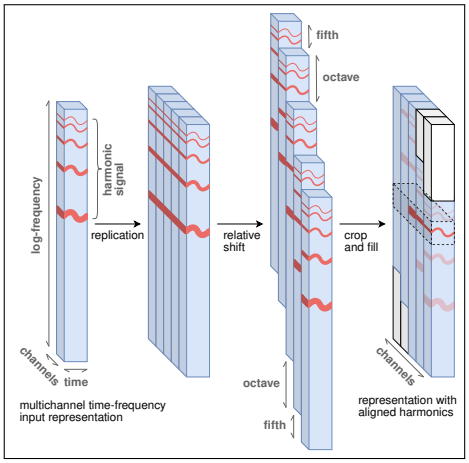

Figure 1. Diagram of the input transformation for convolution layers.

Our proposed Harmonic Convolutional Neural Network (HCNN) overcomes this limitation.[1] The convolutional layer in our network is able to use all the relevant harmonic information in each layer. Our architecture has the standard structure of processing an input through a series of convolutional layers similar to [2]. Crucially before each convolutional layer we add an additional transformation of the input (see Figure 1).

The transformation first creates copies of the input and stacks them in the channel axis. We then shift these copies relative to the original in the frequency axis so that the information about harmonically related peaks

[1] For the source code please see the accompanying repository at `https://github.com/kukas/music-transcription`

are positioned above the original fundamental frequency. For a log-frequency spectrogram, the offsets are constant. This transformed input is then fed into the next convolutional layer. Since the convolutional filter has the access to all channels, it follows that it has access to the provided harmonically related information for every time-frequency position in the matrix.

We train two architectures on a subset of MedleyDB. HCNN noctx which uses only $\approx 5.5\,\mathrm{ms}$ window of input audio for the melody estimation and HCNN ctx which uses $\approx 162.4\,\mathrm{ms}$ (comparable to Bittner et al. [2]). We compare these with state-of-the-art baselines: "SAL" [6], "BIT" [2] and "BAS" [1]. In case of BIT and BAS, we ran the algorithms using the source code obtained from the links provided in the papers keeping default parameters. For the SAL algorithm we used the implementation in Essentia library [2].

As testing data we use MedleyDB test set, ADC04 dataset, MIREX05 training set [3], Orchset [3], a subset of MDB-melody-synth [5] and a subset of WJAZZD [4]. For the complete list of selected testing tracks please see our code repository.

Method	ADC04	MDB-m-s test	MIREX05 train.	MDB test	ORCH-SET	WJazzD test
SAL	0.714	0.527	0.715	0.519	0.235	0.667
BIT	0.716	0.633	0.702	0.611	0.407	0.692
BAS	0.669	**0.689**	0.734	0.640	**0.483**	0.700
HCNN noctx	**0.737**	0.626	0.723	0.635	0.439	0.715
HCNN ctx	0.726	0.661	**0.755**	**0.652**	0.459	**0.725**

Table 1. Overall Accuracy of the selected methods. Highlighted values are the highest across the dataset.

Our methods outperform all selected baselines on four out of six testing datasets (see Table 1). Compared to the most similar method BIT we achieve better results on all datasets with a gain of +3.6 percentage points in average on the Overall Accuracy (OA) metric while using almost 20 times less trainable parameters in the model. Based on these results we believe that the architecture has a potential in other related tasks such as multi-f_0 tracking.

ACKNOWLEDGMENTS

We would like to thank the Jazzomat Research Project for kindly providing the WJazzD testing data.

REFERENCES

[1] Dogac Basaran, Slim Essid, and Geoffroy Peeters. Main Melody Estimation with Source-Filter NMF and CRNN. *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 82–89, 2018.

[2] Rachel M. Bittner, Brian Mcfee, Justin Salamon, Peter Li, and Juan P. Bello. Deep Salience Representations for F0 Estimation in Polyphonic Music. *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, pages 63–70, 2017.

[3] Juan J. Bosch, Ricard Marxer, and Emilia Gómez. Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music. *Journal of New Music Research*, 45(2):101–117, 2016.

[4] Martin Pfleiderer, Klaus Frieler, Jakob Abeßer, Wolf-Georg Zaddach, and Benjamin Burkhart, editors. *Inside the Jazzomat*. Schott Campus, 2017.

[5] Justin Salamon, Rachel M. Bittner, Jordi Bonada, Juan J. Bosch, Emilia Gomez, and Juan Pablo Bello. An Analysis/Synthesis Framework for Automatic F0 Annotation of Multitrack Datasets. *18th International Society for Music Information Retrieval Conference*, pages 71–78, 2017.

[6] Justin Salamon and Emilia Gomez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1759–1770, 2012.

[2] `https://essentia.upf.edu`

[3] We downloaded ADC04 and MIREX05 datasets from `https://labrosa.ee.columbia.edu/projects/melody/`