MIREX 2017
CNN-BASED AUTOMATIC MUSICAL KEY DETECTION SUBMISSIONS HS1/HS2/HS3

Hendrik Schreiber
tagtraum industries incorporated
hs@tagtraum.com

ABSTRACT

End-to-end global key estimation using deep convolutional networks is probably the most promising approach to solving this classic music information retrieval (MIR) problem. We propose a deep, yet relatively compact network architecture as our submission to MIREX 2017. The network has been trained on multiple groundtruths featuring electronic dance music (EDM) as well as a subset of the Million Song Dataset (MSD) with key annotations derived from MIDI files contained in the Lakh MIDI Dataset (LMD).

1. INTRODUCTION

Key estimation is a classic task in music information retrieval (MIR). The goal is to determine the key of a given Western piece of music using just the audio. While key changes are certainly possible, often algorithms are only concerned with estimating the global key, not taking modulations into account. Many published approaches to global key estimation follow the same steps: transform the audio signal to the frequency domain, map the energies of the resulting spectrogram to pitch classes, and then correlate the result with key-specific templates. Additional techniques for improving results are background noise removal/spectral whitening and tuning correction [5, 8]. Furthermore, it has been shown that key-templates are genre-specific [4]. The best performing algorithm on the MIREX 2005 dataset by Cannam and Noland scored 86.83 while only reaching 46.97 for the GiantSteps dataset [6]. At the same time, the algorithm by Faraldo et al. scored an exceptional 68.26 for GiantSteps, but only 74.91 for MIREX 2005. Since MIREX 2005 exclusively contains classical music and GiantSteps only *electronic dance music* (EDM), genre-specific templates are a reasonable conclusion. Even when using a non-template approach, genre-specific tonality remains a challenge. Korzeniowski and Widmer have recently shown, that end-to-end key estimation using a convolutional neural network performs as well or better than the previous state of the art [7]. But when trained with examples from multiple genres, performance always stayed well below the results achieved when trained with examples of the same genre as the test examples.

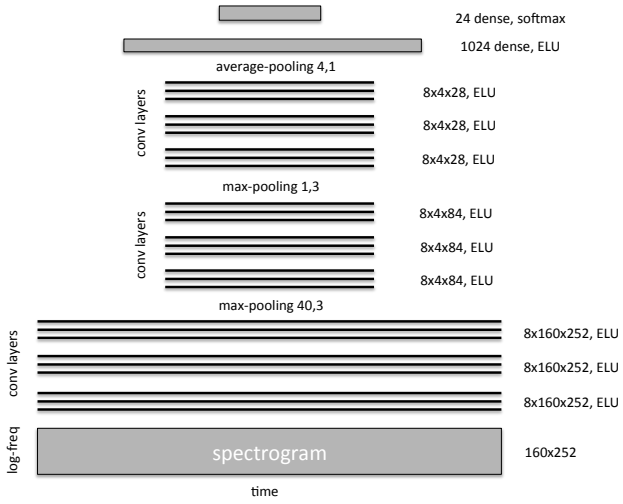

Figure 1. Schematic overview of the network architecture.

2. DESCRIPTION

Our proposed method also employs end-to-end machine learning using a convolutional neural network. As feature, we use a magnitude spectrogram, which we obtain by first downsampling the signal to $11,025$ Hz, performing an STFT with window length $16,386$ and hop size $8,192$, and then resampling the frequency axis to a logarithmic scale with a resolution of 3 bins per semitone. This oversampling is inspired by tuning correction approaches like [3]. It also helps with distinguishing adjacent harmonic peaks, as there are usually at least two (lower energy) frequency bins located in between. Pitches below C1 and above B7 are disregarded. As input for the network we use a spectrogram with the time/frequency dimensions $(160, 252)$, roughly equivalent to 2 minutes of audio.

The network consists of 9 zero-padded convolutional layers, each with 8 filters and a kernel size of 3×3. The convolutional layers are followed by a fully connected layer of size 1024 and an output layer of size 24 (one for each key, only considering 12 major and minor keys). Figure 1 gives a schematic overview of the architecture.

Between layers 3 and 4 we use a max-pooling layer with shape $(40, 3)$ to reduce the input by a factor of 40 along

the time axis and 3 along the frequency axis. The intuition here is that the key does not change rapidly in time and we should be able to disregard much of the information about *when* notes are played. Because we effectively oversample by a factor of 3 along the frequency axis, max-pooling with a stride of 3 in the frequency direction allows us to remove superfluous data, thus sharpen the pitch related information. After pooling we apply a dropout of 0.2. Between layers 6 and 7 we use a max-pooling layer with shape $(1, 3)$. Here the assumption is that we can reduce the signal to the most dominant complex information spanning multiple frequency bins. After pooling, we again apply a dropout of 0.2 to avoid overfitting. The last convolutional layer is followed by average-pooling along the time axis with the shape $(40, 1)$, followed by a dropout of 0.5, the fully connected layer and another dropout layer (0.5). All activation functions are ELUs [2], except the last one, which is a softmax function.

The proposed network has a total of $259,752$ model parameters. Informal experiments have shown that reducing the size of the large dense layer decreases accuracy only modestly.

3. TRAINING DATASETS

For the MIREX submission we trained the network with the EDM *GiantSteps MTG key dataset* [1] collected by Ángel Faraldo. Of the $1,486$ annotations, we used only those that have a confidence of 2 and an unambiguous key, resulting in a dataset of size $1,159$. The spectra were extracted from the 2 minutes audio previews. Our submission trained with this dataset is named `HS1`.

To create a second, non-EDM dataset, we parsed the MIDI files contained in the *LMD-matched* subset of Colin Raffel's *Lakh MIDI Dataset* (LMD) [9] to extract key change events. We assumed that those files that contain only a single key change event that is not C major are accurately described by the given key. In those cases where multiple MIDI files are mapped to a single *Million Song Dataset* [1] id, we used the key annotation if it was contained in at least half the MIDI files. Otherwise we discarded the track. This resulted in a dataset of size $6,981$. We were able to match almost 70% of those tracks to MSD genre annotations [10]. 31% of the matched tracks were labeled `rock`, 22% `pop`, 9% `country`, 5% `r&b`, 5% `classical`, and 3% `jazz`. Less than 2% of the tracks were labeled `dance` or `electronica/dance`. The audio features were extracted from the 7digital audio previews. To create spectrograms of size $(160, 252)$ we repeated shorter spectrograms when necessary. The submission trained with this dataset is named `HS2`.

Additionally, we created a combined dataset (submission `HS3`).

[1] `https://github.com/GiantSteps/GiantSteps-mtg-key-dataset`

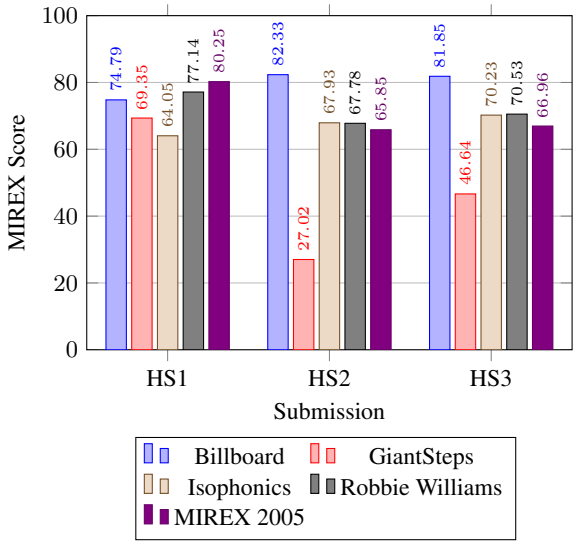

Figure 2. MIREX scores for all three submissions for the five test datasets.

4. DATA AUGMENTATION

To increase the number of samples and ensure uniform distribution of keys per mode, we shifted the pitch of each song by -4 to $+7$ semitones [7]. Furthermore, during training we randomly flipped the spectrum along the time axis.

5. RESULTS

Even though the same network architecture was used in all three submissions, results vary significantly depending on training and test datasets (Figure 2). Therefore both low and high scores must be credited more to the training datasets than to the approach itself. This is most evident with *GiantSteps* as test set. When trained on similar data, our approach reaches a relatively high MIREX score of 69.35 (`HS1`), and when trained on very different data, only a very low 27.02 (`HS2`). Trained with the combined dataset we reach a value right in the middle: 46.64 (`HS3`). This confirms the findings by Korzeniowski and Widmer [7].

Compared with other contest participants, our submissions performed well. They produced the top MIREX scores for two out of five test datasets. `HS1` just barely reaches the overall highest score for the *Robbie Williams* dataset with 77.14. And by a margin of 3.49 points `HS2` reaches the top score for the *Billboard* dataset with 82.33. Since `HS2` was trained exclusively on LMD-derived data, this must be seen as validation for our approach of extracting training data from the LMD. It also suggests a strong similarity between the two datasets.

6. REFERENCES

[1] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Society*

for Music Information Retrieval Conference (ISMIR), pages 591–596, Miami, USA, 2011.

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, Feb. 2015.

[3] Karin Dressler and Sebastian Streich. Tuning frequency estimation using circular statistics. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 357–360, Vienna, Austria, 2007.

[4] Ángel Faraldo, Emilia Gómez, Sergi Jordà, and Perfecto Herrera. Key estimation in electronic dance music. In *European Conference on Information Retrieval*, pages 335–347. Springer, 2016.

[5] Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, UPF Barcelona, 2006.

[6] Peter Knees, Ángel Faraldo, Perfecto Herrera, Richard Vogl, Sebastian Böck, Florian Hörschläger, and Mickael Le Goff. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 364–370, Málaga, Spain, October 2015.

[7] Filip Korzeniowski and Gerhard Widmer. End-to-end musical key estimation using a convolutional neural network. In *In Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, Kos, Greece, 2017.

[8] Katy Noland. *Computational Tonality Estimation: Signal Processing and Hidden Markov Models*. PhD thesis, Queen Mary, University of London, 2009.

[9] Colin Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD thesis, 2016.

[10] Hendrik Schreiber. Improving genre annotations for the million song dataset. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 241–247, Málaga, Spain, 2015.