End-to-End Musical Key Estimation Using a Convolutional Neural Network

Filip Korzeniowski and Gerhard Widmer
Department of Computational Perception,
Johannes Kepler University, Linz, Austria
Email: filip.korzeniowski@jku.at

***Abstract*—We present an end-to-end system for musical key estimation, based on a convolutional neural network. The proposed system not only out-performs existing key estimation methods proposed in the academic literature; it is also capable of learning a unified model for diverse musical genres that performs comparably to existing systems specialised for specific genres. Our experiments confirm that different genres do differ in their interpretation of tonality, and thus a system tuned e.g. for pop music performs subpar on pieces of electronic music. They also reveal that such cross-genre setups evoke specific types of error (predicting the relative or parallel minor). However, using the data-driven approach proposed in this paper, we can train models that deal with multiple musical styles adequately, and without major losses in accuracy.**

I. Introduction

The key of a piece of (Western) music defines its harmonic center and thus plays a vital role in the piece's tonality. It gives meaning to the harmonic progression of a piece, and provides the backdrop for the build-up and release of harmonic tension. Thus, if we want to automatically detect tension and its release in musical audio, if we want to find similarities in harmonic structure in recordings of songs, or if we want to assist disc jockeys or creators of electronic music in finding appropriate musical samples to work with, we need reliable methods that extract the key from a musical piece. It is therefore of no surprise that the automatic key estimation is a long-standing task in the music information retrieval community.

Many key estimation systems proposed in the academic literature rely on the same fundamental pipeline. First, a time-frequency representation, like the spectrogram or the constant-q-transform, is computed from the audio. Then, a variant of a pitch-class-profile, also called chroma feature, is extracted for every time frame. Here, the goal is to get an octave-independent representation of pitch classes that is robust to timbre. Finally, these features are aggregated over time and matched with feature templates for each key, where the key with the best matching template is selected. Examples of such systems are [1]–[3], and, most recently, [4]. Such systems typically report a single *global key* for a piece. This limitation is reasonable for a variety of genres (like pop/rock or electronic music), but fails to cope with pieces that contain key modulations (as common in classical music).

Another line of work considers estimating key and chords simultaneously, e.g. [5]–[8]. Such systems aim at exploiting the musicological relationship between key and chords in order to improve the accuracy of both. They typically estimate *local keys*, and are thus able to cope with key modulations. However, while they bear the potential to explain the harmonic content of musical audio more holistically, dedicated systems currently seem to achieve better results[1].

This paper targets the estimation of a *single global key* for pieces of musical audio. In contrast to previous works, we abandon hand-crafting or tuning elements in the key estimation pipeline. Instead, we employ a convolutional neural network that encompasses the three stages of pre-processing, feature extraction, and classification. Although neural networks have been used for key detection (e.g. [9], [10]), the existing approaches rely on a hand-crafted feature extraction stage: in [9], a pitch class distribution matrix is fed into a neural network classifier; in [10], beat-aligned chroma and timbre features are used as input. Our system operates directly on the spectrogram, and it can estimate all its parameters from the data. To our knowledge, this is the first work that replaces the complete key estimation pipeline with a model that can be optimised in an end-to-end manner[2].

II. Method

Our system consists of two steps: first, we compute from the audio a logarithmically filtered log-magnitude spectrogram. This process is detailed in Sec. II-A. Then, we feed this time-frequency representation to the convolutional neural network, described in Sec. II-B, for classification.

A. Input Processing

We input a spectral representation of the audio to our models. Based on our previous work on extracting harmonic information from audio [11], [12], we first compute from the audio the magnitude spectrogram $|\mathbf{S}|$ (frame size of 8192 at 5 frames per second, where the sample rate is 44.1 kHz); then, we apply a filterbank $\mathbf{B}_{\text{Log}}^{\triangle}$ with logarithmically spaced triangular filters (24 bands per octave, from 65 Hz to 2100 Hz), which results in a time-frequency representation in which the fundamental frequencies of notes are spaced linearly; finally,

[1] See results of the yearly MIREX challenges at www.music-ir.org/mirex.

[2] One might argue that [7] is also an "end-to-end" system that detects both chords and keys. However, they used feature extraction methods heavily based on expert knowledge (tuning correction, harmonic-percussive source separation, beat synchronisation, frequency-split chroma computation), and trained a dynamic Bayesian network whose structure is based on domain knowledge. Also, its key estimation performance was never evaluated.

ISBN 978-0-9928626-7-1 © EURASIP 2017

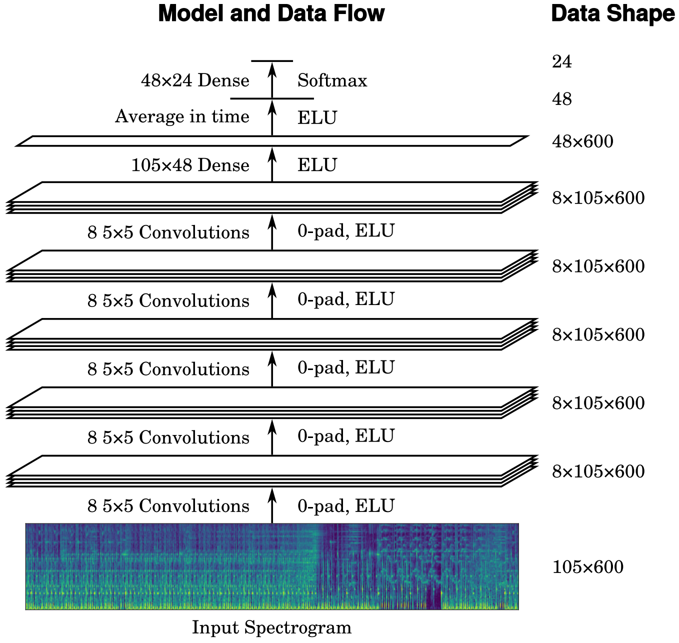

Fig. 1. Neural network for musical key estimation. Convolutions are 0-padded, i.e. the feature maps keep the input shape. All layers are followed by exponential-linear activations [15], except the last, which is followed by a softmax.

we logarithmise the magnitudes of the filtered spectrogram to compress the value range, resulting in the logarithmically filtered log-magnitude spectrogram

$$\mathbf{L} = \log\left(1 + \mathbf{B}_{\text{Log}}^{\triangle}\left|\mathbf{S}\right|\right).$$

This representation is similar to a constant-q transform, but is much cheaper to compute. Additionally, as shown in [13], the constant-q transform does not necessarily lead to better results in tasks relying on pitch information. All computations are done using the madmom library [14].

B. Model

The proposed neural network is designed to encompass all stages of the classic key estimation pipeline: a pre-processing stage of convolutional layers, a dense layer that projects the feature maps into a short representation at the time-frame level, a global averaging layer that aggregates this representation over time, and a softmax classification layer that predicts the global key of a piece. Figure 1 shows our model's architecture: five convolutional layers with 8 feature maps computed by 5×5 kernels, followed by a dense layer with 48 units applied frame-wise; this projection is then averaged over time and classified using a 24-way softmax layer. All layers (except the softmax layer) use the exponential-linear activation function [15].

The convolutional layers constitute the first part of the "feature extraction" equivalent in traditional key estimation systems. They are intended to process the input spectrogram, deal with detrimental factors such as noise or slight detuning, and, together with the projection layer, compute a short frame-wise description of harmonic content. This part of the network can process inputs of arbitrary lengths. Its output is aggregated in the following layers.

Before classification, an averaging layer reduces the extracted representation to a fixed-length vector. We could employ other, more powerful methods (like recurrent layers), but we found in preliminary experiments that they fail to achieve better results.

Finally, a softmax classification layer predicts the global key for the audio. We restrict ourselves to major and minor modes only, resulting in 24 possible classes (12 tonics $\times$ $\{\text{major}, \text{minor}\}$) as output. This is a common restriction, since most musical pieces are in either major or minor, and as of now, there are no datasets with reliable song-level annotations of other modes.

C. Training

We train the model using stochastic gradient descent with momentum, back-propagating through the network the categorical cross-entropy error between true key label y_i and network output $\hat{y}_i$, and apply weight decay with a factor of 10^{-4} for regularisation. The initial learning rate is 0.001, with a momentum factor of 0.9. If validation accuracy did not increase within 10 epochs, we halve the learning rate and continue training with the parameters that gave the best results until then. After 100 epochs, we select the model that achieved the best validation accuracy.

III. Experiments

Our experiments aim at 1) comparing the proposed system to reference systems, and 2) examining the effect that the type of training data (in our case, the musical genre) has on the results. Template-based algorithms require specialised key templates for genres like electronic dance music in order to perform well [4]. We want to see if, and how strongly, our system is affected by this. In the following, we discuss the data, the evaluation metrics, the reference systems, and the different set-ups of our system that we used in the experiments.

A. Data

We use three datasets in the course of our experiments: the GiantSteps key dataset [16], the GiantSteps-MTG key dataset, and a subset of the McGill Billboard dataset [17].

The **GiantSteps Key Dataset**[3] comprises 604 two-minute audio previews from www.beatport.com, with key ground truth for each excerpt. It consists of various sub-genres of electronic music. We use this dataset for testing purposes only, and will refer to it as *GS*.

The **GiantSteps MTG Key dataset**[4], collected by Ángel Faraldo from the Music Technology Group at Universitat Pompeu Fabra, comprises 1486 two-minute audio previews from the same source. These excerpts are distinct from the ones in the GS dataset. From this set, we only use excerpts

[3] https://github.com/GiantSteps/giantsteps-key-dataset

[4] https://github.com/GiantSteps/giantsteps-mtg-key-dataset

labelled with a single key and a high confidence for training (1077 pieces). We will refer to this dataset as *GS^{MTG}*.

The **McGill Billboard Dataset**[5] comprises 742 unique songs sampled from the American Billboard charts between 1958 and 1991, and thus consists of mostly pop and rock music. Unfortunately, only the tonic, and not the mode (major or minor), is annotated for each piece. We therefore estimate the mode using the tonic and chord annotations, following a simple procedure for each piece: 1) select all chords whose root is the tonic; 2) if more than 90% of these chords are major, the key mode is assumed to be major, and vice-versa for minor; 3) else, discard the song, because we cannot confidently estimate the mode. Similarly, we discard songs with multiple annotated tonics. This leaves us with 625 songs with key annotations. We then divide the set into subsets of 62.5% for training, 12.5% for validation, and 25% for testing. The exact division and key ground truths are available online[6]. We will refer to this dataset as *BB^{TV}* and *BB^{TE}* for the train/validation and test sub-sets, respectively.

B. Data Augmentation

The datasets provide only few training data (1077 in GS$^{\text{MTG}}$, 391 in BB$^{\text{TV}}$), compared to datasets used in computer vision (e.g. 40000 in CIFAR-10 or 60000 in MNIST). The generalisation capability of deep neural networks, however, depends on a large number of training samples. We thus have to rely on data augmentation to increase the number of training examples artificially.

Several augmentation techniques for audio input have been explored [18], with pitch shifting being particularly popular in harmony-related tasks [11], [19], [20]. Since pitch shifting is an expensive time-domain operation, most works manipulate the time-frequency representation to emulate it. In this work, however, we found that using a time-domain pitch shifting algorithm[7] directly on the audio gave better results in terms of classification accuracy. We therefore shift each training song in the range of -4 to +7 semitones (and adjust the target key accordingly), which increases the amount of training data by a factor of 12.

C. Metrics

Evaluating key estimation results requires a more detailed quantitative analysis than computing accuracy scores. In particular, although we consider the task to be a simple 24-way classification problem when designing the system, some classes are semantically closer to each other than others. For example, the key of A-minor is called the "relative minor" to the key of C-major, as they share all pitch classes, and differ only in the tonic. Therefore, it is reasonable to consider some errors to be more severe than others.

The MIREX evaluation campaign[8] developed an evaluation strategy and introduced a single weighted measure that reflects the above considerations. Following their guidelines, key predictions can fall into the following categories:

- **Correct:** if the tonic and the mode (major/minor) of prediction and target correspond.
- **Fifth:** if the tonic of the prediction is the fifth of the target (or vice versa), and modes correspond.
- **Relative Minor/Major:** if modes differ and either a) the predicted mode is minor and the predicted tonic is 3 semitones below the target, or b) the predicted mode is major and the predicted tonic is 3 semitones above the target.
- **Parallel Minor/Major:** if modes differ but the predicted tonic matches the target.
- **Other:** Prediction errors not caught by any category, i.e. the most severe errors.

We first compute the ratio of predictions that fall into each category. We then calculate the MIREX **weighted** score as $w = r_c + 0.5 \cdot r_f + 0.3 \cdot r_r + 0.2 \cdot r_p$, where r_c, r_f, r_r, and r_p are the ratios of the correct, fifth, relative minor/major, and parallel minor/major, respectively. These ratios reveal more about the capability of the algorithms than accuracy (i.e., the "correct" ratio) alone. They allow us to see the kind of mistakes the system makes, and at the same time, assign a single number for comparing its performance with others.

The ratios of the individual error categories cannot be compared in isolation, but only in context with the other ratios. The only numbers that can be compared individually are the weighted score (because it aggregates all error types), percentage of correct classifications (because it corresponds to classification accuracy), and the "other" error ratio (because it tells us how often a system predicts unrelated keys). That is why, in Table I, we will highlight the best results only for these categories.

D. Setups and Reference Systems

We train our method in three configurations: CK^1, trained on GS$^{\text{MTG}}$; CK^2, trained on BB$^{\text{TV}}$; and CK^3, trained on both GS$^{\text{MTG}}$ and BB$^{\text{TV}}$. We evaluate each of the trained models on GS and on BB$^{\text{TE}}$. This way, we observe the system's performance when trained on the *same* genres as it is tested on, when trained on a *different* genre (the cross-genre setup), and when trained on multiple genres (to see if it can learn a unified model for different genres).

We compare our method against the Queen Mary Key Detector (QM) [21] and three variations of the method presented in [4] (EDM$^{\text{A}}$, EDM$^{\text{M}}$, EDM$^{\text{T}}$). For both systems, open source implementations are available[9]. QM consists of a hand-crafted pre-processing stage and correlates the obtained chromagrams with key profiles based on Bach's Well Tempered Klavier. The three EDM systems also use a hand-crafted pre-processing stage, but use different key profiles for classification: EDM$^{\text{A}}$ uses key profiles automatically derived from a set of electronic music; EDM$^{\text{M}}$ uses hand-tuned profiles

[5] http://ddmal.music.mcgill.ca/research/billboard

[6] http://www.cp.jku.at/people/korzeniowski/bb.zip

[7] We used the SoX software available at http://sox.sourceforge.net/.

[8] http://www.music-ir.org/mirex

[9] http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html
https://github.com/angelfaraldo/edmkey

ISBN 978-0-9928626-7-1 © EURASIP 2017

TABLE I
Results of various training configurations of our proposed model and of reference systems. Boldface indicates best results. GS and GS$^{\text{MTG}}$refer to the GiantSteps datasets (electronic music), BB* to various subsets of the Billboard dataset (pop/rock music). CK* denote the proposed model trained on different data sets, EDM* and QM denote reference systems by [4], [21].

Test Set	Method	Train Set	Weighted	Correct	Fifth	Relative	Parallel	Other
GS	CK^{1}	GS$^{\text{MTG}}$	**74.3**	**67.9**	6.8	7.1	4.3	**13.9**
	CK^{2}	BB$^{\text{TV}}$	57.3	47.0	6.5	12.6	16.6	17.4
	CK^{3}	GS$^{\text{MTG}}$, BB$^{\text{TV}}$	69.2	61.9	6.8	8.6	6.3	16.4
	EDM$^{\text{A}}$		65.6	57.8	7.3	6.6	10.8	17.6
	EDM$^{\text{M}}$		70.1	63.7	8.6	2.7	6.5	18.5
	EDM$^{\text{T}}$		44.6	33.6	8.8	15.4	9.9	32.3
	QM		50.4	39.6	11.9	13.2	4.3	31.0
BB$^{\text{TE}}$	CK^{1}	GS$^{\text{MTG}}$	72.8	62.5	7.6	13.2	12.5	**4.2**
	CK^{2}	BB$^{\text{TV}}$	**83.9**	**77.1**	9.0	4.9	4.2	4.9
	CK^{3}	GS$^{\text{MTG}}$, BB$^{\text{TV}}$	79.7	70.8	9.7	9.0	6.3	**4.2**
	EDM$^{\text{A}}$		78.7	70.8	11.8	2.8	5.6	9.0
	EDM$^{\text{M}}$		28.9	14.6	2.1	16.0	42.4	25.0
	EDM$^{\text{T}}$		75.4	66.7	12.5	6.3	2.8	11.8
	QM		60.9	52.1	11.8	4.2	8.3	23.6

based on the automatically derived ones (effectively disabling the prediction of major keys); EDM$^{\text{T}}$ uses profiles based on European classical music. Of all submissions to MIREX, EDM$^{\text{M}}$ and QM achieved the best results on the electronic and classical music datasets used for evaluation, respectively. We thus consider both to be state of the art.

E. Results

Table I shows the evaluation results of all training configurations of our proposed model, and of the reference systems. We determine the statistical significance of the results using a Wilcoxon signed rank test, with the error types representing the ranks. If trained on the correct genre, our model clearly outperforms the reference systems: 74.3 vs. 70.1 ($\alpha = 0.001$) for the GiantSteps dataset, and 83.9 vs. 78.7 ($\alpha = 0.014$) for the Billboard dataset.

Examining the cross-genre setups, we observe a significant drop in key estimation accuracy: tested on GS (electronic music), a model trained on BB$^{\text{TV}}$ (pop/rock) achieves a weighted score of only 57.3, compared to 74.3 when trained on electronic music from GS$^{\text{MTG}}$. However, we also see that the number of severe mistakes (category "other") that our system commits in this setup is not higher than those of the reference systems: the model only predicts a completely unrelated key 17.4% of the time—similar to the reference systems specialised on this genre (17.6% and 18.5% for EDM$^{\text{A}}$ and EDM$^{\text{M}}$, respectively); vice-versa, when trained on GS$^{\text{MTG}}$ and evaluated on BB$^{\text{TE}}$, it achieves the lowest rate of severe mistakes (4.2%).

The most common error occurring in these cross-genre setups is predicting the wrong mode (resulting in parallel minor/major) and predicting the relative minor/major key. This suggests that while the model is still able recognise some fundamental concepts of tonality, finer characteristics vary too much between pieces of different genres.

We wanted to see if the proposed model can be trained to provide a good unified key estimator for multiple genres by combining training data. The resulting system CK^{3} does not reach the performance of the specialised ones (69.2 vs 74.3 on GS, 79.7 vs. 83.9 on BB$^{\text{TE}}$); however, on GS, it performs as well as EDM$^{\text{M}}$, which is tuned manually to give good results on electronic music datasets (69.2 vs. 70.1, $\alpha = 0.94$). It still performs better than EDM$^{\text{A}}$, which is also trained on electronic music, but without manual post-training adaptations (69.2 vs. 65.5, $\alpha = 0.02$).

The numbers presented for the EDM* systems for the GiantSteps dataset differ from the ones originally reported in [4]. This is mainly because we applied a stricter criterion for the "fifth" category: we require the predicted mode to match the target mode, while [4] ignores the mode for this category. Also, according to personal correspondence with the author, changes in the library used in the original implementation worsened the results relative to the original ones.

IV. Conclusion and Future Work

We have presented a global key estimation system based on a convolutional neural network. Compared to previous work, this model can be automatically trained end-to-end, without the need of expert knowledge in feature design or specific pre-processing steps such as tuning correction or spectral whitening.

We have shown experimentally that the model performs state-of-the-art on datasets of electronic music and pop/rock music. Additionally, we are planning to evaluate the proposed model on more genres, e.g. classical music.

Another feature of the proposed model is its ability to adapt to multiple types of music without changing the model itself; it just needs to be re-trained with a training set extended to the type of music of interest. While it still showed good performance in such a scenario, it did not reach the level of its specialised counterparts.

A clear limitation of the proposed method is that it only estimates a global key for a complete piece. While this is adequate for certain types of music, other types (e.g. classical music) involve key modulations that our method currently cannot capture. A possible easy fix could be to apply our model using a sliding window over the spectrogram. Extending the proposed method in such a way is left to future work.

Finally, we have to keep in mind that even with data augmentation, we are still working with small datasets. Although we increase the number of training samples by a factor of 12 using pitch-shifting, this is not equivalent to having available 12 times as many musical pieces: the musical content of the artificial data points is still the same as in the seed data point, just in a different key. We expect the system's performance to improve once more training data is available.

Acknowledgments

This work is supported by the European Research Council (ERC) under the EUs Horizon 2020 Framework Programme (ERC Grant Agreement number 670035, project "Con Espressione"). The Tesla K40 used for this research was donated by the NVIDIA Corporation.

References

[1] S. Pauws, "Musical Key Extraction From Audio," in *Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR)*, Barcelona, Spain, Oct. 2004.

[2] D. Temperley, "What's Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered," *Music Perception: An Interdisciplinary Journal*, vol. 17, no. 1, pp. 65–100, Oct. 1999.

[3] K. Noland and M. Sandler, "Signal Processing Parameters for Tonality Estimation," in *Audio Engineering Society Convention 122*. Audio Engineering Society, May 2007.

[4] Á. Faraldo, E. Gómez, S. Jordà, and P. Herrera, "Key Estimation in Electronic Dance Music," in *Advances in Information Retrieval*, N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, and G. Silvello, Eds. Cham: Springer International Publishing, 2016, vol. 9626, pp. 335–347.

[5] M. Mauch and S. Dixon, "Simultaneous Estimation of Chords and Musical Context From Audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1280–1289, Aug. 2010.

[6] J. Pauwels and J.-P. Martens, "Combining Musicological Knowledge About Chords and Keys in a Simultaneous Chord and Local Key Estimation System," *Journal of New Music Research*, vol. 43, no. 3, pp. 318–330, Jul. 2014.

[7] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "An End-to-End Machine Learning System for Harmonic Analysis of Music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1771–1783, Aug. 2012.

[8] B. Di Giorgi, M. Zanoni, A. Sarti, and S. Tubaro, "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," in *Proceedings of the 8th International Workshop on Multidimensional Systems*, Erlangen, Germany, 2013.

[9] J. Sun, H. Li, and L. Lei, "Key detection through pitch class distribution model and ANN," in *2009 16th International Conference on Digital Signal Processing*, Jul. 2009, pp. 1–6.

[10] S. Dieleman, P. Brakel, and B. Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*. University of Miami, 2011, pp. 669–674.

[11] F. Korzeniowski and G. Widmer, "A Fully Convolutional Deep Auditory Model for Musical Chord Recognition," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Salerno, Italy, Sep. 2016.

[12] ——, "Feature Learning for Chord Recognition: The Deep Chroma Extractor," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, Aug. 2016.

[13] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the Potential of Simple Framewise Approaches to Piano Transcription," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, Aug. 2016.

[14] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new Python Audio and Music Signal Processing Library," in *Proceedings of the 24th ACM International Conference on Multimedia (ACMMM)*, Amsterdam, Netherlands, Oct. 2016.

[15] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *International Conference on Learning Representations (ICLR), arXiv:1511.07289*, San Juan, Puerto Rico, Feb. 2016.

[16] P. Knees, A. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. 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)*, Málaga, Spain, Oct. 2015.

[17] J. A. Burgoyne, J. Wild, and I. Fujinaga, "An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis." in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, Oct. 2011.

[18] J. Schlüter and T. Grill, "Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks." in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, Oct. 2015.

[19] T. Cho, "Improved Techniques for Automatic Chord Recognition from Music Audio Signals," Dissertation, New York University, New York, 2014.

[20] E. J. Humphrey and J. P. Bello, "Rethinking Automatic Chord Recognition with Convolutional Neural Networks," in *11th International Conference on Machine Learning and Applications (ICMLA)*. Boca Raton, USA: IEEE, Dec. 2012.

[21] C. Cannam, M. Mauch, M. E. Davies, S. Dixon, C. Landone, K. Noland, M. Levy, M. Zanoni, D. Stowell, and L. A. Figueira, "MIREX 2016 Entry: Vamp Plugins from the Centre for Digital Music," MIREX, Tech. Rep., 2016.