LOCAL KEY ESTIMATION IN CLASSICAL MUSIC RECORDINGS: A CROSS-VERSION STUDY ON SCHUBERT'S WINTERREISE

Hendrik Schreiber, Christof Weiß, Meinard Müller

International Audio Laboratories Erlangen, Germany

{hendrik.schreiber,christof.weiss,meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

While global key and chord estimation for both popular and classical music recordings have received a lot of attention, little research has been devoted to estimating the local key for classical music. In this work, we approach local key estimation on a unique cross-version dataset comprising nine performances (versions) of Schubert's song cycle *Winterreise*—a challenging scenario of high musical ambiguity and subjectivity. We compare an HMM-based system with a CNN-based approach. For both models, we employ a similar training procedure including the optimization of hyperparameters on a validation split. We systematically evaluate the model predictions and provide musical explanations for key confusions. As our main contribution, we explore how different training–test splits affect the models' efficacy. Splitting along the song axis, we find that both methods perform similarly well. Splitting along the version axis leads to clearly higher results, especially for the CNN, which seems to effectively learn the harmonic progressions of the songs ("cover song effect") and successfully generalizes to unseen versions.

Index Terms— music information retrieval, local key estimation, harmony analysis, evaluation, deep neural networks

1. INTRODUCTION

The tonal analysis of music audio recordings is of high relevance for both musicologists and music listeners and therefore constitutes a central task in music information retrieval (MIR) research. Notions of tonal structures relate to different temporal scales. Many researchers have focused on local (i. e., temporally concentrated) structures such as *chords* [1–5]—loosely defined as sets of pitches that are perceived as an entity. In contrast, the *global key* describes the tonality of a whole song, piece, or movement. It can be defined as a set of pitch relationships that establishes a particular major or minor chord as a tonal center [6], attaining a subjective sense of arrival and rest [7]. In this paper, we consider the intermediate notion of *local key*, which relates to mid- and large-scale segments of a piece.

For the global key in Western classical music, the beginning and ending sections [8] and the final chord [9] play an important role, and the key label is often provided by the composer as part of the title. Contrasting this global view, the musical key may also change over the course of a piece, thus calling for a *local* key analysis. When the harmonic structure prepares the arrival of the new key, we speak of a modulation [6]. Modulations often proceed gradually over a certain time span leading to ill-defined segment boundaries. Furthermore, some keys are closely related to each other such as *relative keys* (e. g., C major ↔ A minor), which share the same underlying diatonic scale. Some researchers therefore focus on the 12-class problem of diatonic scale detection [10,11]. There is also a high similarity between parallel keys (C major ↔ C minor) or fifth-related keys (C major ↔ G major), whose associated scales largely overlap by sharing many pitch classes. Due to these issues, local key estimation (LKE) is a challenging task where annotations are often ambiguous and highly subjective by nature. Several approaches therefore avoid the "hard" detection of keys and boundaries and propose multi-scale [12, 13], self-referential [14], or probabilistic [11, 15, 16] visualization techniques instead. Figure 1 shows a visualization of LKE results with an arrangement of keys according to the circle of fifths—thus showing closely related keys next to each other. At second 40, we observe a confusion with the relative key and around second 90, a confusion with a fifth-related key.

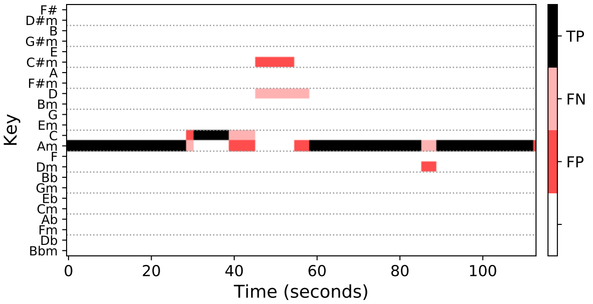


Fig. 1: Local key predictions of the CNN model for song 15 "Die Krähe" from Schubert's song cycle *Winterreise*, performed by T. Quasthoff and C. Spencer (1998). Dark red bars indicate false positives, brighter red bars false negatives, black bars true positives.

To address automatic LKE from audio recordings, different methods have been proposed. Traditional approaches combine chroma features with template-based recognition [17, 18]. For segmentation and post-filtering, many researchers used Hidden Markov Models (HMMs) [10, 19], or non-negative matrix factorization (NMF) as an alternative [17]. As we know from chord estimation research [1, 2], HMMs are useful mainly due to the context-sensitive smoothing effect and less due to their quality as a language model for key transitions. Several methods simultaneously address the estimation of chords, local keys, and (down-)beats [18, 20, 21]. Recently, deep-learning techniques have become popular for chord estimation [4, 5] and global key estimation [22–24] in music recordings. Korzeniowski et al. [23] successfully used convolutional neural net-

This work was supported by the German Research Foundation (DFG MU 2686/10-1, DFG MU 2686/7-2). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE). Moreover, we thank Junseong Park and Vlora Arifi-Müller for their help in preparing data and annotations.

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

ID	Singer	Pianist	Year	Duration
AL98	Thomas Allen	Roger Vignoles	1998	1:13:33
FI55	Dietrich Fischer-Dieskau	Gerald Moore	1955	1:14:35
FI66	Dietrich Fischer-Dieskau	Jörg Demus	1966	1:11:23
FI80	Dietrich Fischer-Dieskau	Daniel Barenboim	1980	1:13:07
HU33	Gerhard Hüsch	Hanns-Udo Müller	1933	1:07:31
OL06	Thomas Oliemans	Bert van den Brink	2006	1:14:42
QU98	Thomas Quasthoff	Charles Spencer	1998	1:12:24
SC06	Randall Scarlata	Jeremy Denk	2006	1:06:45
TR99	Roman Trekel	Ulrich Eisenlohr	1999	1:15:21

Table 1: Cross-version dataset of Franz Schubert's *Winterreise*.

works (CNN) to estimate the global key for music recordings across different genres. Though we are not aware of any research using deep neural networks for LKE, this is an obvious endeavor due to the task's similarity to chord and global key estimation—both of which have been tackled successfully using CNNs.

While most audio-based LKE systems were developed and tested on popular music, Western classical music has rarely been approached. Mearns et al. [25] analyze modulations in synthesized recordings of twelve chorales by J. S. Bach. Papadopoulos and Peeters [18] consider recordings of Mozart's piano sonatas. Weiss et al. [16] provide visualizations of local key regions in Wagner's operas. Compared to popular music, changes between closely related keys and gradual modulations are particularly prominent in classical music. Moreover, many classical music styles involve altered chords featuring non-scale tones that make LKE even harder. As a peculiarity of classical music, there are usually many recorded performances (interpretations) available. Together with other representations, such as symbolic scores, we consider these as individual *versions* of an abstract musical work. Exploiting several such versions in a *cross-version scenario* allows for studying and improving the robustness and generalization for various tasks such as chord [3] and scale analysis [16] or singing voice detection [26, 27].

In this paper, we study LKE within a cross-version scenario. We make use of a dataset comprising nine recorded performances (referred to as *versions*) of Franz Schubert's 24-song cycle *Winterreise* [13]. Using measure annotations as anchor points, we semi-automatically generate local key annotations [13, 28]. We propose a straightforward LKE approach based on a CNN and compare it to a traditional method using chroma features and HMMs. In our experiments, we evaluate the efficacy of both methods and systematically assess their robustness. As our main contribution, we investigate the effect of using different training–test splits that require generalization across versions, songs, or both. Furthermore, we conduct an in-depth analysis and investigate musical reasons for key confusions.

The paper is organized as follows. In Section 2, we start with the description of the dataset and our training–test scenarios. We proceed in Section 3 with introducing the technical approaches (HMM and CNN). In Section 4, we then discuss our results in detail. We draw our conclusions in Section 5.

2. CROSS-VERSION DATASET

In this section, we describe our dataset and annotation procedure followed by the different splits used for training, validation, and testing.

2.1. Dataset

Franz Schubert's song cycle *Winterreise* (Winter Journey, published 1828) consists of 24 songs for voice (originally tenor) and piano.

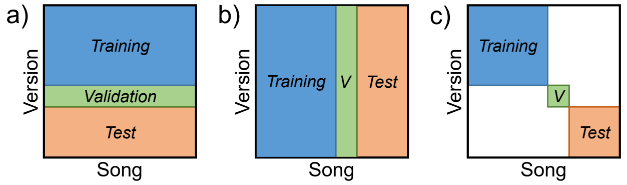


Fig. 2: Dataset splitting into training, validation, and test sets. **(a)** Version split V, **(b)** Song split S, **(c)** Neither song nor version split N.

The individual songs differ in length and complexity. Some songs are harmonically unambiguous showing distinct key regions of diatonic pitch content (No. 2) or being based on a single tonic chord (No. 24). Other songs involve many altered chords (No. 10) and ambiguous key regions (No. 16). Inspired by previous analyses [13, 28], each song has been annotated on score level (musical time axis) with continuous local key segments by a professionally trained musician.[1] Since the local key is sometimes ambiguous, our annotations differ from [13, 28] in several respects: We did not label unclear or transitional passages with "no key" but decided on the most likely key. Furthermore, we ensured a certain continuity of the key segments.

Our dataset [13] comprises nine complete performances by different duos, recorded in a studio setting (Table 1). While all versions realize the same musical scores, tempi, dynamics, and acoustic conditions such as reverb and timbre can vary greatly. On average, a song lasts 3 min (σ=1:10 min), ranging from 0:44 min (No. 18, SC06) to 6:18 min (No. 1, OL06). We manually annotated measure positions for two recordings (HU33, SC06) and automatically transferred these to the other recordings using synchronization techniques as proposed in [29]. Using the measure positions as anchor points, we semi-automatically transferred the local key regions from the score level to the nine recordings (physical time axis).[2]

2.2. Splits

To train our models and optimize their hyperparameters, we split our dataset into training, validation, and test subsets so that each song in each version is analyzed exactly once in a cross-validation procedure. Since our dataset has a specific structure, we can split along two axes—the "version axis" and the "song axis" (see Figure 2). In order to systematically investigate the models' efficacy when trained in different ways, we create three different splits:

- **Version split** V (Figure 2a). The training subset contains all songs in five versions, the validation subset all songs of one version, and the test subset all songs of three versions. In this case, the models can exploit their knowledge of the abstract musical structure (harmonic progressions), but have to generalize to unseen *acoustic conditions* and different interpretations, which is not trivial.
- **Song split** S (Figure 2b). The training subset contains recordings of 13 songs in all nine versions, the validation subset three songs in all versions, and the test subset eight songs in all versions. The models have to generalize to unseen *musical pieces* with different harmonic properties but can adapt to the acoustic conditions of each version during training.
- **Neither split** N (Figure 2c). In this strict split, the training subset contains 19 songs in four versions, the validation subset two other songs in two other versions, and the test subset

[1] The 2nd author.

[2] Annotations are at `https://github.com/hendriks73/key-cnn`.

three other songs in three other versions. Thus, the model knows neither song nor version and has to generalize across both axes. This is the only split where not all data is used in one run, and it is the most realistic one.

To ensure comparability, we fix the exact versions and songs in each of the splits (no randomization) for both models.

3. METHODS

We present two approaches for LKE. The first one is a classical system relying on HMMs, the second one a typical CNN-based system.

3.1. HMM-Based Method

Our first system, denoted as HMM, relies on the extraction of chroma features using the filter-bank method proposed in [30].[3] We post-process the filter-bank output (pitch features) by applying logarithmic compression with a parameter $\gamma \in \{100, 1000, 10\,000\}$ and apply pitch weighting to emphasize the mid range centered at C4 [1]. We smooth the resulting 10 Hz chroma features with a median filter of length $\lambda \in \{81, 85, \ldots, 157\}$. On the training subset, we learn Gaussian models for the 24 keys (assuming enharmonic equivalence) in the chroma space $\mathbb{R}^{12}$. We cyclically average the major and the minor key model over the chroma dimension in order to achieve transposition-blind models, which we then use for generating the HMM's emission probabilities. Inspired by [1], we apply a uniform, diagonal-enhanced transition matrix with a self-transition probability of $1 - \sigma$ where $\sigma \in \{10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$. Using these HMM parameters, we run Viterbi decoding to predict a key label for every 10 Hz frame. We optimize the parameters γ, λ, and σ on the validation subset. That way, we exploit the available data in a similar way as the CNN-based system described next.

3.2. CNN-Based Method

The second system, denoted as CNN, is identical to the global key estimation network DeepSquare [24].[4] It is a convolutional neural network in the style of Oxford's Visual Geometry Group's (VGG) image recognition networks [31]. For feature extraction, we use twelve 2D convolutional layers with square kernels of sizes 5×5 and 3×3, batch normalization [32] after every convolutional layer, and 2×2 max pooling with a subsequent dropout layer ($p = 0.3$) after every second convolutional layer. This is followed by a fully convolutional classification stage using a "bottleneck" layer (1×1 convolution), 2D global average pooling, and the softmax activation function. Overall, the network has 293 296 trainable parameters. The employed training procedure is similar to [24]. We first convert the audio to constant-Q magnitude spectrograms. Then, we use samples of dimension $F \times T$ as input to the network. $F = 168$ is the number of frequency bins covering a frequency range of seven octaves with a frequency resolution of two bins per semitone. $T = 60$ is the number of time frames with a resolution of 0.19 s per frame, i.e., 60 frames correspond to 11.1 s. To account for class imbalances within the major or minor keys, we randomly shift each spectrogram along the frequency axis by $\{-4, -3, \ldots, 6, 7\}$ semitones and adjust the ground truth labels accordingly. Since key estimation is a single-label, multi-class problem, we use categorical cross-entropy as loss function. Adam [33] is used as optimizer with a batch size of 32 and an initial learning rate of 0.001. Once the validation loss plateaus, we halve the learning rate and continue training with the best performing model up to that point (stepwise annealing). We repeat this at most ten times. If reduction does not lead to a lower validation loss three times in a row, we stop training.[5]

4. RESULTS

In order to evaluate LKE on classical music, we trained both systems on recorded songs from Schubert's *Winterreise* using different data splits. As evaluation measure, we compute the accuracy while ignoring "no key" regions (which only occur at the beginning and ending of a piece). Moreover, we analyze musically explainable key confusions such as relative, parallel, and fifth-related keys. We discuss the results with a focus on the data splits and musical key confusions.

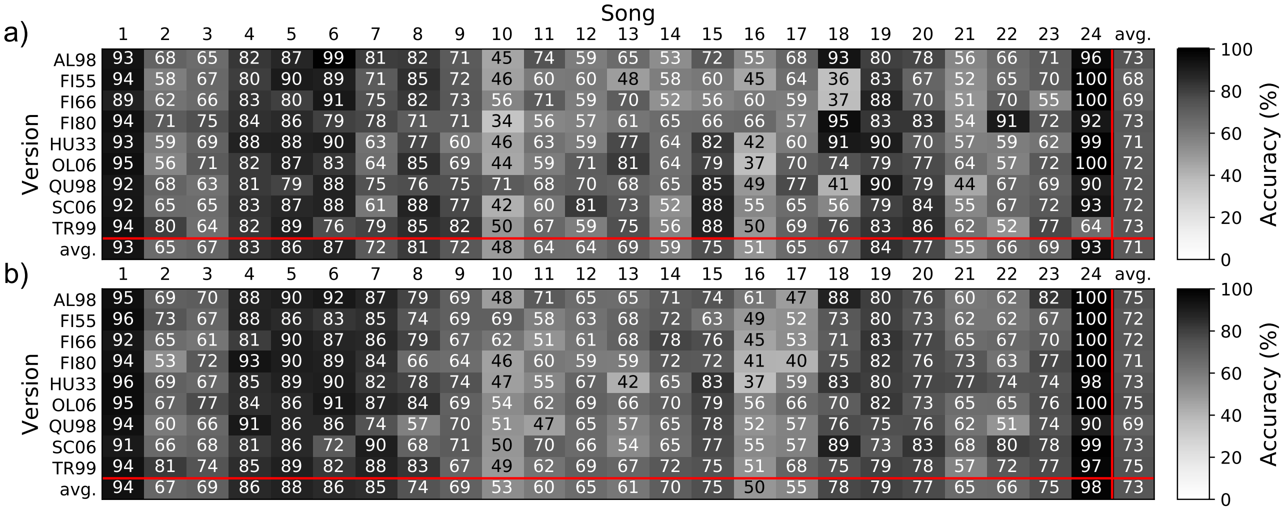
Detailed Results. We first consider the realistic split N, where neither test songs nor test versions are seen during training or validation. Figure 3a depicts the HMM's results. With most songs, we observe a similar accuracy for the different versions. However, the accuracy varies greatly between songs. For example, song no. 1 reaches high accuracies around 93% for all versions, which is expected due to its clear harmonic structure. In contrast, song no. 10, which is highly chromatic, shows low accuracies around 50%. For few songs, we observe higher variance along the version axis. An example for such an outlier is song No. 18, whose accuracy is strongly version-dependent. Investigating these results in detail, we find that this is a very short song of approx. 45 seconds, whose beginning and ending sections are monophonic (unisono) without any chords in the piano, thus posing a particular challenge. The HMM's tendency to stay in a key reinforces the impact of such errors on the overall accuracy. Comparing the HMM's results with the CNN's (Figure 3b), we observe similar tendencies in both plots. With an average of 73%, the CNN performs only marginally better than the HMM trained on split N (71%). For the CNN, accuracies are also similar across different versions of a song. Moreover, the variation across songs is similar to the HMM's results, which indicates that *musical* properties of the individual songs may pose the main challenge for both systems.

Data Splits. To statistically summarize these results, we report the average per-song accuracy values in Figure 4. For the "neither" split N, the two right-most bars correspond to the overall averages (lower-right values) in Figure 3a+b. Black errorbars indicate the average standard deviation over all versions of a song ("vertical direction" in Figure 3). Red errorbars denote the average standard deviation over all songs of a version ("horizontal direction" in Figure 3). The standard deviation across songs (14.4% for CNN) is substantially greater than across versions (5.2% for CNN), which confirms our observation that the accuracy variance can be traced back more to differences in songs than in versions. This also holds for the other splits V and S depicted in Figure 4. Comparing the average accuracy between splits, we find that the "song split" S leads to similar results (69% for HMM, 72% for CNN). Interestingly, accuracies are a bit lower than for N, despite having more training data available in each step. In N, the split between training and validation is stricter, which leads to higher generalization and robustness of the trained systems. Contrary to findings for genre classification [34], we found no advantage for either system when being exposed to other versions from the same CD recording, i.e., no observable "album effect." Looking at the "version split" V, we find a remarkable result. Accuracies are considerably higher with 76% for HMM and 96% for CNN. Both systems apparently have a capacity to learn the specific *musical* characteristics of the individual songs (resp. their specific annotations),

[3] We use the librosa implementation: https://librosa.github.io/

[4] Scaled with model sizing parameter $k = 8$, see [24] for details.

[5] Trained CNN models: https://github.com/hendriks73/key-cnn



a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	avg.
AL98	93	68	65	82	87	99	81	82	71	45	74	59	65	53	72	55	68	93	80	78	56	66	71	96	73
FI55	94	58	67	80	90	89	71	85	72	46	60	60	48	58	60	45	64	36	83	67	52	65	70	100	68
FI66	89	62	66	83	80	91	75	82	73	56	71	59	70	52	56	60	59	37	88	70	51	70	55	100	69
FI80	94	71	75	84	86	79	78	71	71	34	56	57	61	65	66	66	57	95	83	83	54	91	72	92	73
HU33	93	59	69	88	88	90	63	77	60	46	63	59	77	64	82	42	60	91	90	70	57	59	62	99	71
OL06	95	56	71	82	87	83	64	85	69	44	59	71	81	64	79	37	70	74	79	77	64	57	72	100	72
QU98	92	68	63	81	79	88	75	76	75	71	68	70	68	65	85	49	77	41	90	79	44	67	69	90	72
SC06	92	65	65	83	87	88	61	88	77	42	60	81	73	52	88	55	65	56	79	84	55	67	72	93	72
TR99	94	80	64	82	89	76	79	85	82	50	67	59	75	56	88	50	69	76	83	86	62	52	77	64	73
avg.	93	65	67	83	86	87	72	81	72	48	64	64	69	59	75	51	65	67	84	77	55	66	69	93	71

b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	avg.
AL98	95	69	70	88	90	92	87	79	69	48	71	65	65	71	74	61	47	88	80	76	60	62	82	100	75
FI55	96	73	67	88	86	83	85	74	69	69	58	63	68	72	63	49	52	73	80	73	62	62	67	100	72
FI66	92	65	61	81	90	87	86	79	67	62	51	61	68	78	76	45	53	71	83	77	65	67	70	100	72
FI80	94	53	72	93	90	89	84	66	64	46	60	59	59	72	72	41	40	75	82	76	73	63	77	100	71
HU33	96	69	67	85	89	90	82	78	74	47	55	67	42	65	83	37	59	83	80	77	77	74	74	98	73
OL06	95	67	77	84	86	91	87	84	69	54	62	69	66	70	79	56	66	70	82	73	65	65	76	100	75
QU98	94	60	66	91	86	86	74	57	70	51	47	65	57	65	78	52	57	76	75	76	62	51	74	90	69
SC06	91	66	68	81	86	72	90	68	71	50	70	66	54	65	77	55	57	89	73	83	68	80	78	99	73
TR99	94	81	74	85	89	82	88	83	67	49	62	69	67	72	75	51	68	75	79	78	57	72	77	97	75
avg.	94	67	69	86	88	86	85	74	69	53	60	65	61	70	75	50	55	78	79	77	65	66	75	98	73

Fig. 3: Individual accuracy values (in percent) per song and version for the strict neither split N. **(a)** Results for HMM. **(b)** Results for CNN.

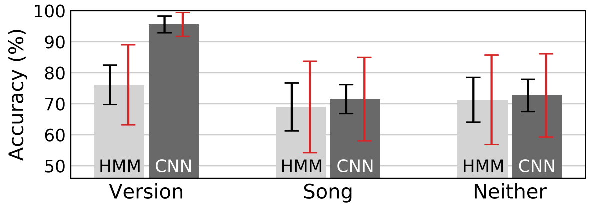


Fig. 4: Average accuracies for songs based on different splits (V, S, N) and models (HMM, CNN). Black errorbars denote standard deviations *across versions* (averaged over all songs), red errorbars denote standard deviations *across songs* (averaged over all versions).

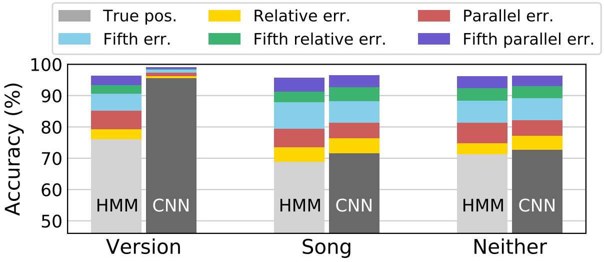


Fig. 5: Accumulated accuracy including different types of musically explainable key confusions.

with the CNN's capacity being greater than the HMM's. The CNN system is effectively (over)fitting to harmonic progressions in the songs. Therefore, we cannot expect it to perform similarly well for other classical songs—we might call this a "cover song effect." Interestingly, generalization to unseen acoustic conditions works well, especially for the CNN. Exploiting several versions for training (and validation) seems to build up the model's robustness against version differences and thus, is sufficient for avoiding the "album effect."

Musical Key Confusions. Finally, we want to discuss the specific types of confusions in the models' predictions. Figure 5 shows the percentage of frames with certain prediction errors on top of the accuracy (true positives). We report confusions with the relative key (e. g., C major ↔ A minor), the parallel key (C major ↔ C minor), fifth-related keys (C major ↔ G major, or C major ↔ F major), as well as the relative of fifth-related keys (C major ↔ E minor, or C major ↔ D minor) and the parallel of fifth-related keys (C major ↔ G minor, or C major ↔ F minor). For both the S- and the N-split, we see that the most common errors are fifth errors, parallel key errors, relative key errors, relative fifth errors, and parallel fifth errors (roughly in this order). Together, these errors explain most of the performance gap between the CNN trained on the V-split and either system trained on one of the other splits (Figure 5). Gray and yellow bars together constitute the accuracy for estimating the correct diatonic scale [11, 16]. Correct predictions and all musical errors together comprise ≈ 95% of all frames. Based on these observation, we assume that it is most challenging for the models to learn how *musically ambiguous* regions have to be labeled in order to predict the local key label as given by a specific annotator.

5. CONCLUSIONS

We approached the task of local key estimation in classical music recordings and systematically explored the efficacy of an HMM-based and a CNN-based approach. Using a cross-version dataset of Schubert's song cycle *Winterreise*, we trained, validated, and tested both systems on splits along songs or versions. Moreover, we explored a strict split where *neither* test songs *nor* test versions are shown during training. For the song and the "neither" split, we found that CNN and HMM models perform similarly well, reaching an accuracy of approximately 70%. Most of the observed errors can be explained through musical ambiguities where annotations are often subjective. Comparing results for different splits, we showed that generalization across versions ("album effect") does not pose major problems for the models. Knowing the specific songs (version split) leads to clearly higher results, especially for the CNN (96%). We call this the "cover song effect." While this is beneficial in our scenario, it means that the CNN (over)fits to the harmonic progressions of specific songs and learns how a single annotator labeled these songs. Song and "neither" split therefore show more realistic results as can be expected for unseen pieces. Yet, providing CNN models with more training data covering a wide variety of harmonic progressions is supposed to have high potential for improving local key estimation systems in general.

6. REFERENCES

[1] Taemin Cho and Juan Pablo Bello, "On the relative importance of individual components of chord recognition systems," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 477–492, 2014.

[2] Nanzhu Jiang, Peter Grosche, Verena Konz, and Meinard Müller, "Analyzing chroma feature types for automated chord recognition," in *Proc. AES Conf. on Semantic Audio*, 2011.

[3] Verena Konz, Meinard Müller, and Rainer Kleinertz, "A cross-version chord labelling approach for exploring harmonic structures—a case study on Beethoven's Appassionata," *Journal of New Music Research*, vol. 42, no. 1, pp. 61–77, 2013.

[4] Siddharth Sigtia, Nicolas Boulanger-Lewandowski, and Simon Dixon, "Audio chord recognition with a hybrid recurrent neural network," in *Proc. Int. Soc. for Music Inf. Retrieval Conf. (ISMIR)*, 2015, pp. 127–133.

[5] Brian McFee and Juan Pablo Bello, "Structured training for large-vocabulary chord recognition," in *Proc. Int. Soc. for Music Inf. Retrieval Conf. (ISMIR)*, 2017, pp. 188–194.

[6] Miguel A. Roig-Francolí, *Harmony in Context*, McGraw-Hill, New York, USA, 2011.

[7] Carl Dahlhaus, Julian Anderson, Charles Wilson, Richard L. Cohn, and Brian Hyer, "Harmony," in *Grove Music Online: Oxford Music Online*. Oxford University Press, 2001.

[8] Steven van de Par, Martin F. McKinney, and André Redert, "Musical key extraction from audio using profile training," in *Proc. Int. Soc. for Music Inf. Retrieval Conf. (ISMIR)*, 2006.

[9] Christof Weiß, "Global key extraction from classical music audio recordings based on the final chord," in *Proc. Sound and Music Computing Conf. (SMC)*, 2013, pp. 742–747.

[10] Yongwei Zhu and Mohan S. Kankanhalli, "Music scale modeling for melody matching," in *Proc. ACM Int. Conf. on Multimedia*, 2003, pp. 359–362.

[11] Christof Weiß and Julian Habryka, "Chroma-based scale matching for audio tonality analysis," in *Proc. Conf. on Interdisciplinary Musicology (CIM)*, 2014, pp. 168–173.

[12] Craig Stuart Sapp, "Visual hierarchical key analysis," *ACM Computers in Entertainment*, vol. 3, no. 4, pp. 1–19, 2005.

[13] Harald Grohganz, *Algorithmen zur strukturellen Analyse von Musikaufnahmen*, Ph.D. thesis, University of Bonn, 2015.

[14] Nanzhu Jiang and Meinard Müller, "Automated methods for analyzing music recordings in sonata form," in *Proc. Int. Conf. on Music Inf. Retrieval (ISMIR)*, 2013, pp. 595–600.

[15] Hendrik Purwins, Benjamin Blankertz, and Klaus Obermayer, "Constant Q profiles for tracking modulations in audio data," in *Proc. 2001 Int. Computer Music Conf. (ICMC)*, 2001.

[16] Christof Weiß, Frank Zalkow, Meinard Müller, Stephanie Klauk, and Rainer Kleinertz, "Computergestützte Visualisierung harmonischer Verläufe: Eine Fallstudie zu Wagners Ring," in *Proc. GI Jahrestagung*, 2017, pp. 205–217.

[17] Özgür İzmirli, "Localized key finding from audio using nonnegative matrix factorization for segmentation," in *Proc. Int. Soc. for Music Inf. Retr. Conf. (ISMIR)*, 2007, pp. 195–200.

[18] Hélène Papadopoulos and Geoffroy Peeters, "Local key estimation from an audio signal relying on harmonic and metrical structures," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1297–1312, 2012.

[19] Wei Chai and Barry Vercoe, "Detection of key change in classical piano music," in *Proc. Int. Soc. for Music Inf. Retrieval Conf. (ISMIR)*, 2005, pp. 468–474.

[20] Thomas Rocher, Matthias Robine, Pierre Hanna, and Laurent Oudre, "Concurrent estimation of chords and keys from audio," in *Proc. Int. Soc. for Music Inf. Retrieval Conf. (ISMIR)*, 2010, pp. 141–146.

[21] Matthias Mauch and Simon Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1280–1289, 2010.

[22] Filip Korzeniowski and Gerhard Widmer, "End-to-end musical key estimation using a convolutional neural network," in *Proc. European Signal Processing Conf. (EUSIPCO)*, 2017.

[23] Filip Korzeniowski and Gerhard Widmer, "Genre-agnostic key classification with convolutional neural networks," in *Proc. Int. Soc. for Music Inf. Retr. Conf. (ISMIR)*, 2018, pp. 264–270.

[24] Hendrik Schreiber and Meinard Müller, "Musical tempo and key estimation using convolutional neural networks with directional filters," in *Proc. Sound and Music Computing Conf. (SMC)*, 2019.

[25] Lesley Mearns, Emmanouil Benetos, and Simon Dixon, "Automatically detecting key modulations in J. S. Bach chorale recordings," in *Proc. Sound and Music Computing Conf. (SMC)*, 2011, pp. 25–32.

[26] Christian Dittmar, Bernhard Lehner, Thomas Prätzlich, Meinard Müller, and Gerhard Widmer, "Cross-version singing voice detection in classical opera recordings," in *Proc. Int. Soc. for Music Inf. Retrieval Conf. (ISMIR)*, 2015, pp. 618–624.

[27] Stylianos I. Mimilakis, Christof Weiß, Vlora Arifi-Müller, Jakob Abeßer, and Meinard Müller, "Cross-version singing voice detection in opera recordings: Challenges for supervised learning," in *Proc. 12th Int. Workshop on Machine Learning and Music (MML)*, 2019.

[28] Frans Absil, *Musical Analysis*, 6th edition, 2017.

[29] Frank Zalkow, Christof Weiß, Thomas Prätzlich, Vlora Arifi-Müller, and Meinard Müller, "A multi-version approach for transferring measure annotations between music recordings," in *Proc. AES Int. Conf. on Semantic Audio*, 2017, pp. 148–155.

[30] Meinard Müller, Frank Kurth, and Michael Clausen, "Chroma-based statistical audio features for audio matching," in *Proc. IEEE Workshop on Applications of Signal Processing (WASPAA)*, 2005, pp. 275–278.

[31] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[33] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. for Learning Representations (ICLR)*, 2015.

[34] Elias Pampalk, Arthur Flexer, and Gerhard Widmer, "Improvements of audio-based music similarity and genre classification," in *Proc. Int. Soc. for Music Inf. Retrieval Conf. (ISMIR)*, 2005, pp. 628–633.