LEARNING PITCH-CLASS REPRESENTATIONS FROM SCORE–AUDIO PAIRS OF CLASSICAL MUSIC

Christof Weiß[1,2], **Johannes Zeitler**[1], **Tim Zunner**[1], **Florian Schuberth**[1], **Meinard Müller**[1]
[1] International Audio Laboratories Erlangen, [2] LTCI, Télécom Paris, Institut Polytechnique de Paris
{christof.weiss,meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Chroma or pitch-class representations of audio recordings are an essential tool in music information retrieval. Traditional chroma features relying on signal processing are often influenced by timbral properties such as overtones or vibrato and, thus, only roughly correspond to the pitch classes indicated by a score. Deep learning provides a promising possibility to overcome such problems but requires large annotated datasets. Previous approaches therefore use either synthetic audio, MIDI-piano recordings, or chord annotations for training. Since these strategies have different limitations, we propose to learn transcription-like pitch-class representations using pre-synchronized score–audio pairs of classical music. We train several CNNs with musically inspired architectures and evaluate their pitch-class estimates for various instrumentations including orchestra, piano, chamber music, and singing. Moreover, we illustrate the learned features' behavior when used as input to a chord recognition system. In all our experiments, we compare cross-validation with cross-dataset evaluation. Obtaining promising results, our strategy shows how to leverage the power of deep learning for constructing robust but interpretable tonal representations.

1. INTRODUCTION AND RELATED WORK

In the field of music information retrieval (MIR), many algorithms rely on pitch-class or chroma representations for analyzing audio recordings. Such representations capture the signal's energy distribution over the twelve chromatic pitch classes (ignoring octave information) and, thus, allow for a direct musical interpretation. Chroma features have been successfully used for different MIR tasks such as chord recognition [1–4], key estimation [5], structure analysis [6], or audio retrieval [7, 8] especially for Western music. While traditional chroma features were designed in a handcrafted fashion based on signal processing techniques [9–12], such features exhibit several drawbacks caused by audio-related artifacts such as timbral characteristics, overtones, vibrato, or transients. Moreover, the relative loudness of a note directly influences the feature representation.

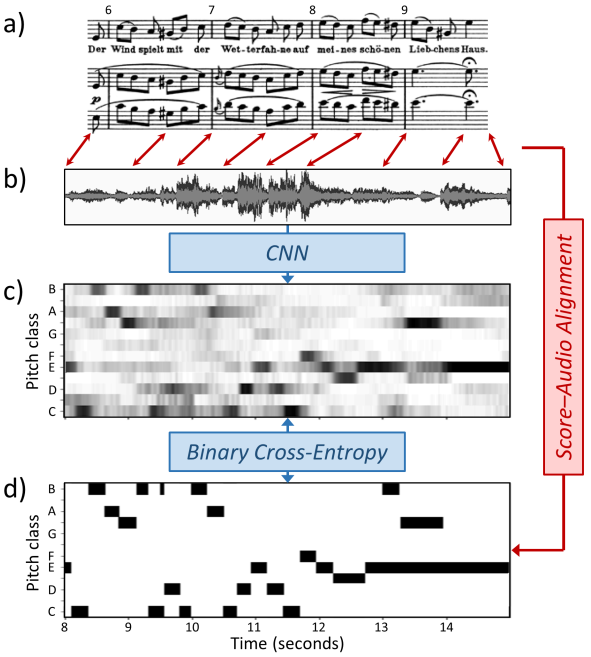

Figure 1. Illustration of the pitch-class training strategy with an example from Schubert's *Winterreise* [13]. (a) Score. (b) Audio recording. (c) Pitch-class estimates of the CNN. (d) Pitch-class labels derived from aligned score.

Over the years, a number of solutions for these problems were proposed involving spectral whitening [14], peak picking [12], overtone removal [3, 15], or timbre homogenization [16]. Most of these techniques led to improved results for tasks such as chord recognition [1–4] or audio retrieval [7]. However, the problem remains challenging since improvements for one task may deteriorate another—a good chroma for music synchronization [16] might be worse for chord recognition [2], or removal of harmonics might introduce sub-harmonic artifacts [3]. Finally, chroma features are often noisy compared to the pitch classes in the score, limiting their interpretability by musicologists as well as their potential for visualization and cross-modal retrieval and analysis applications.

To overcome such problems, more recent strategies make use of deep neural networks for learning chroma representations from data [17–21]. For the successful training of high-capacity networks, large amounts of annotated recordings are necessary. Since manual creation of pitch-class annotations is tedious and requires expert knowledge, there are several alternative strategies, all of which have their benefits and limitations. Early approaches to a "deep chroma" make use of chord labels to derive pitch-

Attribution: C. Weiß, J. Zeitler, T. Zunner, F. Schuberth, and M. Müller, "Learning Pitch-Class Representations from Score–Audio Pairs of Classical Music", in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, Online, 2021.

Table 1. Datasets and annotations used in this work ("Perf.": number of performances per piece).

ID	Dataset Name	Instrumentation	Pitch annotation type	Chords	Tracks	Pieces	Perf.[1]	hh:mm
SWD	Schubert Winterreise [13]	Piano, voice	Aligned scores	yes	216	24	9	10:50
BSD	Beethoven Piano Sonatas [25][2]	Piano	Aligned scores	yes	192	32	6	62:30
WaR	Wagner Ring [26][2]	Orchestra, voice	Aligned scores	no	33	11	3	43:13
MuN	MusicNet [27]	Piano, strings, winds	Aligned scores	no	330	330	1	34:08
SMD	Saarland Music Data [22]	Piano	MIDI piano / Disklavier	no	50	50	1	4:43

class annotations [17, 18]. As shown in [17], this leads to a chroma extractor that has a strong bias towards the chords' pitch classes (in [17], these are triads) and does not actually detect the pitch classes notated in the score, thus limiting interpretability, generalization to other chord vocabularies and genres, and applicability to other tasks. As an alternative strategy for obtaining training data, symbolic music representations were used to render synthetic audio recordings together with the corresponding annotations [19]. While this is pragmatic, systems trained on synthetic data often show limited generalization to recorded audio. Another strategy makes use of MIDI-fied instruments (e. g., Disklaviers) for capturing pitch information, which led to a number of comprehensive piano transcription datasets [22–24]. However, these approaches are limited to piano or MIDI-fied instruments, and the use of the sustain pedal constitutes a problem for determining the perceptually relevant duration of a note.

In this paper, we target a pitch-class representation that relates to the task of multi-pitch estimation (MPE) or framewise transcription [28]. Concretely spoken, we aim for *detecting the framewise activity of all pitch classes indicated by the score* (multi-pitch-class estimation, see Figure 1c). In an ideal scenario, such a representation helps to close the gap between audio- and symbolic-based MIR and, as a consequence, is well-interpretable and capable of generalizing to different music genres, instrumentations, and MIR tasks. For this purpose, we propose an alternative training strategy using score–audio pairs of classical music that are pre-aligned using music synchronization techniques [29]. As an alternative to this, weakly-annotated score–audio pairs were recently used for training using an attention mechanism [30], the CTC loss [20], or a multi-label CTC variant that can deal with polyphonic pitch-class representations [21]. A (strong) aligment strategy similar to ours was successfully applied for multi-instrument music transcription with the MusicNet dataset [27], which we include in this paper. We prepare three further classical music datasets comprising several performances of Schubert's song cycle *Winterreise* [13], the first movements of Beethoven's piano sonatas [25], and Wagner's four-opera cycle *Der Ring des Nibelungen* [26]. We generate pitch-class annotations for these datasets using symbolic scores, manual measure annotations [26], and music synchronization techniques [29]. The data comprises various styles and instrumentations including piano, orchestra, chamber music, as well as singing voice.

As our first contribution, we use this data for supervised learning of a transcription-like pitch-class representation with a medium-sized, musically motivated convolutional neural network (CNN) inspired by [20, 31, 32]. We test the network's pitch-class estimates using evaluation measures from music transcription. Second, we compare this CNN with other architectures such as wider and deeper networks, inception blocks, and residual connections. Third, we test the benefit of the learned features for harmony analysis, specifically chord recognition for classical music. To systematically assess the role of the input features, we employ a controlled and well-understood chord recognition approach based on hidden Markov models (HMMs) [2, 4]. We compare the novel features with traditional chroma features and idealized pitch-class representations derived from the score. In all stages, we compare cross-validation results on individual datasets with cross-dataset results to systematically test generalization [33].

The remainder of paper is organized as follows. In Section 2, we introduce the datasets used for our experiments. Section 3 describes our CNN-based feature learning. In Section 4, we evaluate the learned pitch-class features. Section 5 discusses chord recognition results using the learned features. Section 6 concludes the paper.

2. DATASETS

As mentioned in Section 1, the limited availability of annotated data is a major issue for multi-pitch and pitch-class estimation—a "key challenge" of music transcription [34]. Since manual annotation is tedious and requires expert annotators, several workarounds were proposed [35]. A common approach involves the use of MIDI-fied pianos (Disklaviers) for simultaneously generating audio and annotations, leading to piano transcription datasets such as SMD [22], MAPS [23], or MAESTRO [24].

Beyond the solo piano scenario, there are only few and small datasets with pitch annotations such as Bach10 [36], TRIOS [37], or PHENICX-Anechoic [38] (all ≤10 pieces), which often involve multi-track recordings to simplify the manual annotation process [36–38] or to automatically generate annotations using a monophonic F0-tracker as done for MedleyDB [39]. Since this leads to F0 annotations following the performed frequencies rather than the pitches in the score, we do not use MedleyDB here.

As a further strategy, score–audio pairs of classical music can be exploited to generate pitch (class) annotations. This requires score–audio synchronization methods [29]. A dataset created with this strategy is MusicNet (MuN) [27], which comprises pitch annotations for 330 audio recordings of piano and chamber music. For our experiments, we reduce the pitch annotations to the pitch-class level. As another score–audio dataset, we make use of

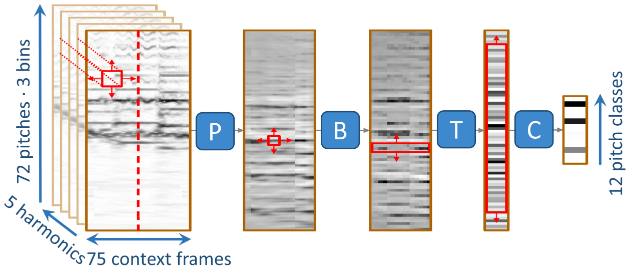

Figure 2. Illustration of the CNN architecture (`Basic`).

the Schubert Winterreise Dataset (`SWD`) [13], which comprises recorded performances (two of nine freely available), scores, measure positions, and chord annotations. We use the scores (MIDI files) and measure annotations together with a synchronization algorithm [29] based on dynamic time warping (DTW) to generate pitch-class annotations.[1] Using the same strategy, we create two further private[2] datasets: The Beethoven Sonatas Dataset (`BSD`) comprises the 32 first movements of Beethoven's piano sonatas in six versions. Using DTW-based alignment [29], we generate pitch-class annotations from corresponding scores and chord labels based on the annotations by Chen and Su [25]. In a similar fashion, we create pitch-class annotations for Wagner's four-opera cycle *Der Ring des Nibelungen* (`WaR`) based on manual measure annotations [26] and a full score (first act of *Die Walküre*) or a piano-reduced score (remaining acts), respectively. Table 1 gives an overview of the datasets used in this paper.

3. DEEP-LEARNING METHODS

In this section, we describe our CNN-based approach for extracting pitch-class representations and discuss our design choices, motivated by related work. Previous deep-learning approaches for pitch-class representations use a variety of architectures including fully-connected [17, 40] and convolutional neural networks (CNNs) [17, 19, 20], where the latter often exhibit large kernels in the last layers to aggregate harmonic information. Due to the lower number of parameters, we pursue a CNN-based approach inspired by [20, 32], summarized in Figure 2 and Table 2.

Input representation. As network input, spectral representations are used most frequently, either generated by a short-time Fourier transform [17] or a constant-Q transform (CQT) [40]. The CQT can be extended to a harmonic CQT (HCQT) with CQTs in harmonic frequency ratios stacked on top of each other, thus allowing for convolutions across harmonics (overtones) along the channel axis [32]. As our input representation, we use such a HCQT with five harmonics (no sub-harmonic). Based on audio sampled at 22050 Hz, we use a CQT hopsize of 384 samples resulting in a feature rate of roughly 57.4 Hz.[3] Our HCQT spans 72 semitones (6 octaves) starting at C1 and a resolution of three bins per semitone. We choose a centering strategy with bins corresponding to integer MIDI pitches placed between the two surrounding bins.

Table 2. Musically informed CNN architecture (`Basic`).

Function	Kernel size, #	Stride	Output Shape	Activ.
		Prefiltering (P):		
LayerNorm			$216\times75\times5$	
Conv2D	$15\times15,\ N_0$	$(1,1)$	$216\times75\times N_0$	LReLU
MaxPool		$(1,2)$	$216\times37\times N_0$	
Dropout				
		Binning to MIDI pitches (B):		
Conv2D	$3\times3,\ N_1$	$(3,3)$	$72\times12\times N_1$	LReLU
MaxPool		$(1,2)$	$72\times6\times N_1$	
Dropout				
		Time reduction (T):		
Conv2D	$1\times6,\ N_2$	$(1,1)$	$72\times1\times N_2$	LReLU
Dropout				
		Chroma reduction (C):		
Conv2D	$1\times1,\ N_3$	$(1,1)$	$72\times1\times N_3$	LReLU
Dropout				
Conv2D	$61\times1,\ 1$	$(1,1)$	$12\times1\times1$	Sigmoid

Context frames. To accurately predict a frame, a network needs information about the context surrounding the target frame. When using a single-stage system, this can be done by feeding multiple time frames of the spectral representation to the network [17, 19, 32]. For our network, we feed the network with 75 context frames (37 to each side of the target frame), corresponding to 1.3 sec at a frame rate of 57.4 Hz. Thus, we feed the network with an input tensor of shape $216\times75\times5$ to predict a pitch-class activation vector of size 12 (see Table 2).

Basic CNN architecture. Our proposed CNN filters the input data in a musically meaningful way. Table 2 gives detailed information about the proposed model; Figure 2 provides a schematic illustration. First, we perform layer normalization to ensure zero mean and unit variance for each input sample followed by a (trainable) linear transformation of the normalized input tensor. Next, N_0 (default 20) feature maps are extracted in the **Prefiltering** layer (P). Using a kernel size of 15×15 allows the network to detect, e. g., vibrato for singing. The second convolutional layer performs a **Binning to MIDI pitches** (B) by moving a 3×3 kernel with stride 3 and no padding along the pitch axis, so that each output bin corresponds to an integer MIDI pitch. We learn N_1 (default 20) feature maps. Third, a convolution across time performs a **Time reduction** (T), resulting in N_2 (default 10) feature maps with 72 bins each. Fourth, we perform pitch-class or **Chroma reduction** (C): After reducing the representation to N_3 (default 1) channels with a 1×1 convolution, we move a kernel with length $72-11=61$ along the pitch axis. In all convolutional layers, we use LeakyReLU activation (negative slope 0.3) to prevent vanishing gradients. MaxPooling along time reduces the number of parameters and forces generalization. Dropout (rate 0.2) hampers overfitting while retaining a large amount of information. We use sigmoid activation in the final layer and train with binary cross-entropy loss between predicted pitch-class vectors $\mathbf{p}\in[0,1]^{12}$ and bi-

[1] With this paper, we publish pitch and pitch-class annotations for the `SWD`, to be found at `https://zenodo.org/record/5139893/`.

[2] These datasets cannot be published due to copyright issues.

[3] As the only parameter, the CQT is determined by the hopsize, which must be an integer multiple of powers of two.

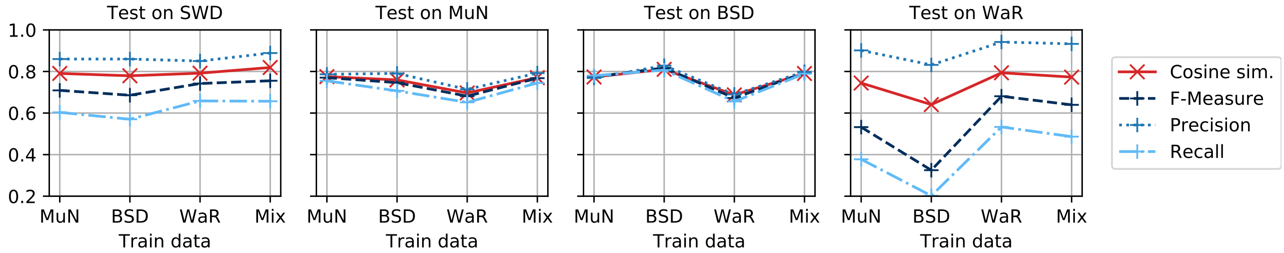

Figure 3. Pitch-class estimation results (`Basic` model) for different datasets (train/test subsets of each dataset are disjoint).

nary, multi-hot target vectors $\mathbf{t} \in \{0, 1\}^{12}$, obtained from the score's note occurrences without weighting (Figure 1).

Larger CNN architectures. In addition to this basic CNN architecture (denoted as `Basic` in the following) with roughly 27k convolutional parameters (plus the parameters of layer normalization), we test a number of extended network architectures. A simple strategy is to increase the number of learned feature maps ($N_0 \dots N_3$). For the architecture `BasicLast10`, we increase the last layer to $N_3 = 10$ (28k conv. params.). For the architecture `Wide`, we increase the number of channels in all layers by a factor of five so that $N_0 = N_1 = 100$, $N_2 = 50$, $N_3 = 5$ (233k conv. params.). Since the choice of the prefiltering kernel size is difficult, we adopt the concept of inception blocks [41], where the input is filtered by kernels with different sizes in parallel. For this `WideInception` architecture, we use kernel sizes 3×3, 9×9, 15×15, and 27×27, leaving the total number of kernels as in `Wide`. As an alternative, we test a `Deep` architecture with more hidden layers, replicating the first layer (P) five times. All remaining layers and parameters are identical to `Basic`. Since training deep architectures is difficult due to vanishing gradients, we test the `DeepResNet` architecture with residual connections [42]. We add shortcut connections to the five P layers, leaving the remainder identical to `Deep`.[4]

4. EVALUATING PITCH-CLASS ESTIMATION

In the following, we evaluate the pitch-class estimates of different networks, trained and tested on various datasets. We measure the frame-wise precision, recall, and F-measure (F) using a threshold of 0.5 (motivated by the sigmoid activation) as well as the cosine similarity (CS) between targets and non-thresholded predictions.

Evaluating general settings. We start with several experiments in a cross-validation on `SWD` (train on seven, test on two performances i. e., a version split [33]), which serves as our development set to decide on general settings. We train all networks with Adam [43] on mini-batches of size 25 using learning rate scheduling and early stopping. For the `Basic` architecture (as described in Section 3), we obtain F=0.832 and CS=0.836 on the test versions of `SWD`, which is already a promising result. Precision (0.850) is slightly higher than recall (0.814). Since the choice of the input HCQT's frame rate is important, we compare this result (with a frame rate of 57.4 Hz) to the use of a smaller rate (10.1 Hz) while holding the (physical) amount of context constant by adjusting CNN kernel shapes accordingly. With this smaller frame rate, we obtain slightly worse results of F=0.820 and CS=0.833. Thus, we use the finer resolution of 57.4 Hz in the following. Next, we test the influence of context frames: Reducing the original context of 75 frames (roughly 1.3 sec) to 51 frames (0.9 sec) leads to decreased results of F=0.827, with 25 frames to a further decrease of F=0.823. We thus opt for the larger context of 75 frames. Finally, we test different kernel sizes in the first layer (prefiltering P). Compared to `Basic` with 15×15 kernels, 9×9 kernels lead to F=0.835, and 5×5 kernels to F=0.824. Though the 9×9 kernels perform slightly better than the 15×15 kernels, we choose the larger kernel size since it spans a larger pitch range and may be better capable of, e. g., detecting vibrato.

Evaluating different datasets. With these parameter choices, we now perform a cross-dataset experiment. In addition to the `SWD` dataset (two versions for test), we use `MuN` (50 pieces test, 280 pieces train/val), `BSD` (two versions test, four versions train/val), and `WaR` (test on *Die Walküre*, 1st act, train/val on all other acts). We further compile a `Mix` train set, which encompasses the train subsets from `MuN`, `BSD`, and `WaR` to equal parts (using subsampling). Note that train subsets of a dataset are never used for testing (and vice versa) even for cross-dataset splits. Figure 3 shows the pitch-class estimation results for all train/test combinations. Using the same source for train and test set (`MuN`–`MuN`, `BSD`–`BSD`, `WaR`–`WaR`), the respective combination yields best results. In those cases, the `Mix` train set always achieves second-best results. In the case of a completely unknown test set (`SWD`), the diverse train data in `Mix` yields best results, slightly worse than the cross-validation results in the previous paragraph.

Evaluating CNN architectures. We now compare the `Basic` model to the larger architectures introduced in Section 3. Figure 4 illustrates the respective results using the same test datasets as for the previous experiment and the `Mix` training set. Compared to the `Basic` architecture, `BasicLast10` has an increased number of channels in the final layer (N_3=10), which yields slightly better results than `Basic` with only about 600 more parameters. In comparison, the `Wide`, `WideInception`, `Deep`, and `DeepResNet` architectures increase the number of parameters by a factor of roughly ten. All of them yield better results than `Basic`, except for the `WaR` test set. Although we can achieve minor improvements by the use of inception blocks and skip connections, the performance metrics

[4] Our source code (Keras) and pre-trained models are available under `https://github.com/christofw/pitchclass_cnn/`.

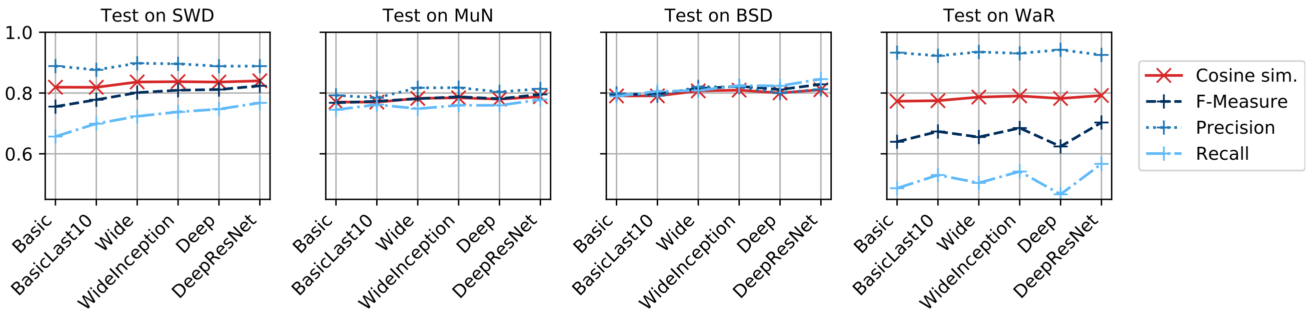

Figure 4. Pitch-class estimation results for different architectures trained on `Mix` dataset (subsets of `BSD`, `MuN`, `WaR`).

of the four most complex architectures are quite similar. Comparing these networks' results on `SWD` (cross-dataset) with our first experiment—a cross-validation on `SWD` with F=0.832 and CS=0.836—, we notice almost identical results. From this, we draw the important conclusion that a larger, diverse training set (e. g., `Mix`) together with a high-capacity network (e. g., `Wide`) can compensate for not "knowing" the particular dataset (here the style of *Winterreise* and the combination of piano and singing). We therefore use the `Wide` network trained on mixed datasets for the following chord recognition experiment. [5]

5. APPLICATION FOR CHORD RECOGNITION

Besides visualization purposes, pitch-class representations serve as front-end features for various MIR applications. To examine the effectiveness of our learned features for the important task of chord recognition, we present systematic experiments using the chord annotations of `SWD` and `BSD`. Rather than optimizing the chord recognition performance, we want to analyze the features' influence and test the hypothesis that our learned features behave similar to features derived from the score (the training targets for our CNNs). To gain these insights, we do not use an end-to-end chord-recognition approach but opt for a traditional yet effective method based on HMMs and Gaussian chord models [4]. For train/test of the HMM, we again compare cross-dataset results with cross-validation results on each dataset, making sure that neither a specific song nor a specific performance are seen during training (neither split) to avoid the kind of "musical overfitting" observed in [33].

Chord recognition method. On the training set, we learn multivariate Gaussian chord models in the pitch-class space $\mathbb{R}^{12}$. We cyclically shift and average the models of each chord type in order to obtain transposition-invariant models, which we use for generating the HMM's emission probabilities. Inspired by [4], we apply a uniform transition matrix with a high self-transition probability, which we optimize on the validation set together with other hyperparameters (log compression strength and pre-filtering length for the input features). We simplify the chord annotations of `SWD` and `BSD` to three common chord vocabularies: `MajMin` comprises the 24 major and minor triads, `Triads` adds the 12 diminished and 4 augmented triads resulting in 40 chords, and `Sevenths` further adds five types of seventh chords (dom7, maj7, min7, half-dim7, dim7) amounting to 91 chords. [6]

Evaluating feature variants. First, we assess the effectiveness of our pitch-class features (denoted as P_{CNN}) and compare those with other feature variants. To obtain P_{CNN}, we train the `Wide` model in a cross-dataset split (train data similar to `Mix` but leaving out the target dataset): For `SWD`, we train on `BSD`, `MuN`, and `WaR`; for `BSD`, we train on `SMD`, `MuN`, and `WaR` (we replace `BSD` with `SMD` to include a piano dataset). The resulting features are re-sampled to 10 Hz. For comparison, we consider three traditional chroma variants based on a CQT (P_{CQT}), an STFT (P_{STFT}), and an IIR filterbank (P_{IIR}), respectively. [7] As baseline, we use an idealized binary feature (P_{Score}) derived from the aligned score (the CNN's training targets). We train and validate the HMM in a cross-validation setting, making sure that neither test performances nor pieces are seen during training [33].

Figure 5 shows the results, reporting chord-symbol recall (which equals the F-measure and accuracy when ignoring no-chord frames). Looking at the traditional feature variants (P_{CQT}, P_{STFT}, P_{IIR}), we observe varying performance, with best results for P_{CQT} on `SWD` and for P_{STFT} on `BSD`. Over all datasets, our CNN-based feature P_{CNN} systematically outperforms the traditional variants with substantial improvements for the more complex vocabularies `Triads` and `Sevenths`. Most remarkably, P_{CNN} almost reaches the performance of the idealized chroma P_{Score}. This is a promising result, indicating that for the task of chord recognition, the *signal-processing* challenge of extracting pitch-class information from audio recordings can be approached in a suitable way using deep learning, while the remaining challenge mainly lies in the mapping of pitch-class information to chord labels.

Evaluating train/test splits. Next, we test the generalization behavior of the chord recognition system (Figure 6). To this end, we compare the cross-validation of the previous experiment with cross-dataset evaluation where we train and test on the other dataset, respectively (note that we speak of training chord recognition—the fea-

[5] For training large networks, a sufficient amount of data is necessary to prevent overfitting. Our largest model has roughly 550k parameters (including layer normalization). As a comparison, 550k frames of training data sampled at 50 Hz give a dataset of about three hours. The amount of data in `MuN` (34 hours), for example, is large compared to the number of parameters, which is even more the case for the larger `Mix` dataset.

[6] We do not discriminate between chords that are identical on the pitch-class level, e. g., C aug and E aug or C dim7 and E♭ dim7.

[7] For implementation details, please see `https://librosa.org/`.

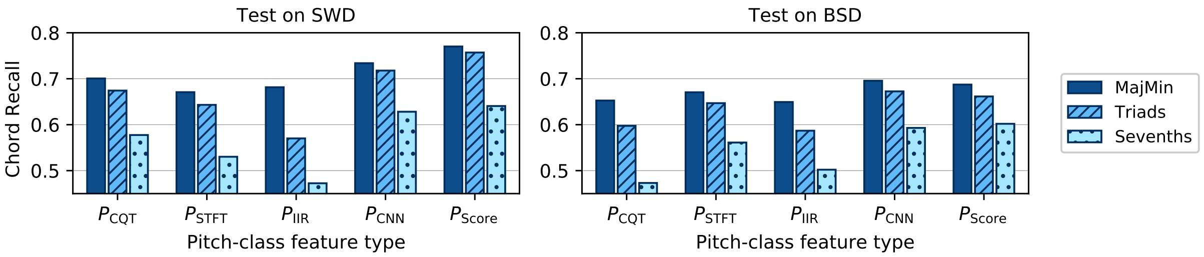

Figure 5. Chord recognition results based on different pitch-class features for `SWD` (left) and `BSD` (right), trained/tested with cross-validation using different chord vocabularies.

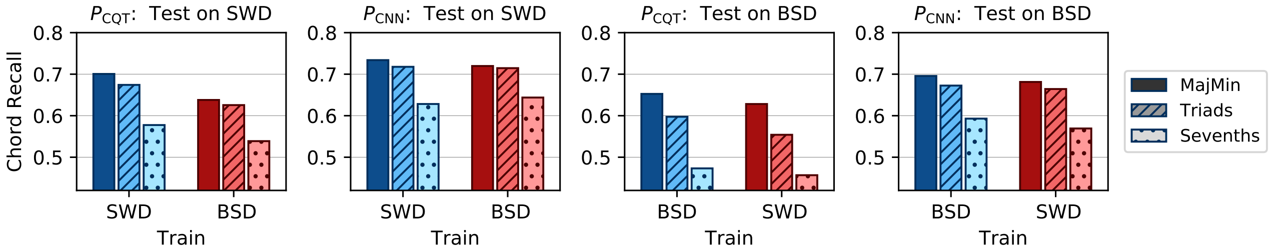

Figure 6. Chord recognition results using features P_{CQT} and P_{CNN} for `SWD` (left) and `BSD` (right), trained/tested with cross-validation (blue) and cross-dataset evaluation (red).

tures P_{CNN} are always trained in a cross-dataset split). Using the traditional feature P_{CQT} (left), we observe a clearly worse performance for the cross-dataset experiment (red), which is musically more challenging. When using P_{CNN} (second plot), this drop does not occur—we observe almost identical results for cross-validation and cross-dataset. Testing on `BSD` (right plots), this tendency is weaker. Still, we conclude that our score-based feature P_{CNN} not only leads to better but also to more robust chord recognition systems, which are widely capable of generalizing to unseen music.

Comparing score- and chord-based pitch classes. Overall, our chord recognition results are not as high compared to, e. g., recent results for pop music [17–19]—even for our baseline feature P_{Score}. On the one hand, this might be due to the simpler system (HMM) we use compared to recent approaches. As a main difference, however, the methods of [17, 18] directly use the chord labels to train a *chord-related* pitch-class representation (instead of a *score-oriented* one). To test the potential of this strategy, we use another baseline feature (P_{Chord}) derived from the chord annotations (without any reduction to a smaller vocabulary), thus capturing idealized, binary activities of the chords' pitch classes. As Figure 7 indicates, we observe a large increase for both datasets. This is of course expected (confirming a similar baseline experiment for pop music in [17]). The comparison of P_{Chord} with P_{Score} and P_{CNN} tells us that the main challenge of chord recognition (at least for our datasets) is a *musical* one: Even when knowing the pitches from the score, it is difficult to decide on which pitches are relevant for the annotated chords. This is of course not trivial and touches questions of musical style, music theory concepts, and annotator subjectivity [33,44,45]. Therefore, deep-learning approaches for mapping score information to chord labels such as [17, 18] are promising. We think that such methods could benefit from using our score-based features as input, thus helping to improve generalization. Beyond that, we want to again emphasize that our aim is not to improve chord recognition itself but to obtain a pitch-class representation that helps to close the gap between audio- and symbolic-based approaches to, e. g., harmony analysis and generalizes to unseen recordings. As our experiments indicate, deep learning allows us to take a crucial step towards this goal.

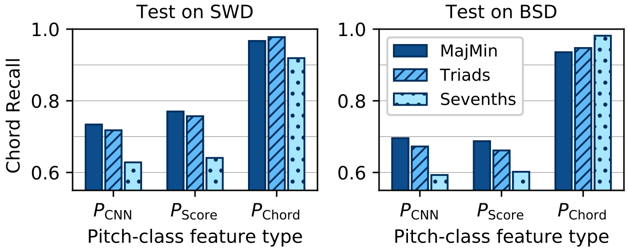

Figure 7. Chord recognition based on P_{CNN} compared with score (P_{Score}) and chord-label (P_{Chord}) baselines.

6. CONCLUSIONS

We presented a CNN-based approach for extracting transcription-like pitch-class representations from music audio recordings. As our main contribution, we proposed a novel strategy for training CNNs with pre-aligned score–audio pairs of classical music. We tested the effectiveness of this approach for pitch-class estimation by comparing different CNN architectures and dataset splits. Using the features as input to a traditional chord recognition system led to improved results and generalization compared to traditional features and is almost on par with symbolic pitch-class features. We conclude that the signal processing challenge of extracting pitch-class information from audio recordings can be successfully approached with deep learning, thus serving as an excellent basis to approach the musical challenge of finding the relevant pitch classes for chords and other harmonic structures—an interesting observation that should be verified for genres beyond Western classical music in future work.

7. ACKNOWLEDGEMENTS

This work is supported by the German Research Foundation (DFG WE 6611/1-1, 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. We thank Leo Brütting and Michael Kohl for help with dataset preparation and Geoffroy Peeters for fruitful discussions.

8. REFERENCES

[1] M. Stein, B. M. Schubert, M. Gruhne, G. Gatzsche, and M. Mehnert, "Evaluation and comparison of audio chroma feature extraction methods," in *Proceedings of the AES Convention*, Ilmenau, Germany, 2009.

[2] N. Jiang, P. Grosche, V. Konz, and M. Müller, "Analyzing chroma feature types for automated chord recognition," in *Proceedings of the AES Conference on Semantic Audio*, Ilmenau, Germany, 2011.

[3] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 135–140.

[4] T. Cho and J. P. Bello, "On the relative importance of individual components of chord recognition systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 477–492, 2014.

[5] E. Gómez and P. Herrera, "Estimating the tonality of polyphonic audio files: Cognitive versus machine learning modelling strategies," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Barcelona, Spain, 2004, pp. 92–95.

[6] R. B. Dannenberg and M. Goto, "Music structure analysis from acoustic signals," in *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. New York, NY, USA: Springer, 2008, vol. 1, pp. 305–331.

[7] M. Müller and S. Ewert, "Towards timbre-invariant audio features for harmony-based music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 649–662, 2010.

[8] J. Serrà, E. Gómez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 1138–1151, 2008.

[9] T. Fujishima, "Realtime chord recognition of musical sound: A system using common lisp music," in *Proceedings of the International Computer Music Conference (ICMC)*, Beijing, China, 1999, pp. 464–467.

[10] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2001, pp. 15–18.

[11] M. Müller, F. Kurth, and M. Clausen, "Chroma-based statistical audio features for audio matching," in *Proceedings of the IEEE Workshop on Applications of Signal Processing (WASPAA)*, New Paltz, USA, 2005, pp. 275–278.

[12] E. Gómez, "Tonal description of music audio signals," PhD Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[13] C. Weiß, F. Zalkow, V. Arifi-Müller, H. Grohganz, H. V. Koops, A. Volk, and M. Müller, "Schubert Winterreise dataset: A multimodal scenario for music analysis," *ACM Journal on Computing and Cultural Heritage (JOCCH)*, vol. 14, no. 2, pp. 25:1–18, 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5139893

[14] A. P. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 255–266, 2008.

[15] K. Lee, "Automatic chord recognition from audio using enhanced pitch class profile," in *Proceedings of the International Computer Music Conference (ICMC)*, New Orleans, USA, 2006, pp. 306–311.

[16] M. Müller, S. Ewert, and S. Kreuzer, "Making chroma features more robust to timbre changes," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[17] F. Korzeniowski and G. Widmer, "Feature learning for chord recognition: The deep chroma extractor," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016, pp. 37–43.

[18] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 188–194.

[19] Y. Wu and W. Li, "Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model," *IEEE/ACM Transactions on Audio, Speech & Language Processing*, vol. 27, no. 2, pp. 355–366, 2019.

[20] F. Zalkow and M. Müller, "Using weakly aligned score–audio pairs to train deep chroma models for cross-modal music retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, Canada, 2020, pp. 184–191.

[21] C. Weiß and G. Peeters, "Training deep pitch-class representations with a multi-label CTC loss," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021.

[22] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland music data (SMD)," in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, 2011.

[23] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[24] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, USA, 2019.

[25] T. Chen and L. Su, "Functional harmony recognition of symbolic music data with multi-task recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 90–97.

[26] C. Weiß, V. Arifi-Müller, T. Prätzlich, R. Kleinertz, and M. Müller, "Analyzing measure annotations for Western classical music recordings," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, 2016, pp. 517–523.

[27] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[28] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[29] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[30] R. Nishikimi, E. Nakamura, M. Goto, and K. Yoshii, "End-to-end melody note transcription based on a beat-synchronous attention mechanism," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2019, pp. 26–30.

[31] A. Elowsson and A. Friberg, "Modeling music modality with a key-class invariant pitch chroma CNN," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 541–548.

[32] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 tracking in polyphonic music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 63–70.

[33] C. Weiß, H. Schreiber, and M. Müller, "Local key estimation in music recordings: A case study across songs, versions, and annotators," *IEEE/ACM Transactions on Audio, Speech & Language Processing*, vol. 28, pp. 2919–2932, 2020.

[34] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[35] L. Su and Y. Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription," in *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Plymouth, UK, 2015, pp. 309–321.

[36] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 18, no. 8, pp. 2121–2133, 2010.

[37] J. Fritsch and M. D. Plumbley, "Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 888–891.

[38] M. Miron, J. Carabias-Orti, J. Bosch, E. Gómez, and J. Janer, "Score-informed source separation for multichannel orchestral recordings," *Journal of Electrical and Computer Engineering*, 2016.

[39] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 155–160.

[40] Y. Wu and W. Li, "Music chord recognition based on MIDI-trained deep feature and BLSTM-CRF hybrid decoding," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 376–380.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, 2015.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations (ICLR)*, San Diego, USA, 2015.

[44] Y. Ni, M. McVicar, R. Santos-Rodríguez, and T. D. Bie, "Understanding effects of subjectivity in measuring chord estimation accuracy," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2607–2615, 2013.

[45] H. V. Koops, W. B. de Haas, J. A. Burgoyne, J. Bransen, A. Kent-Muller, and A. Volk, "Annotator subjectivity in harmony annotations of popular music," *Journal of New Music Research*, vol. 48, no. 3, pp. 232–252, 2019.