MIREX 2025 Submission: Degree-Based Automatic Chord Recognition with Enharmonic Distinction

Masayuki Doai
Keio University
masayukid@keio.jp

ABSTRACT

This extended abstract presents our submission on automatic chord estimation. We build upon ChordFormer, a model based on the Conformer architecture, by modifying it to estimate scale degrees rather than absolute pitch. This approach enables the system to recognize musical keys and distinguish enharmonic equivalents. Furthermore, we introduce the Octavewise Convolution Module, applied prior to the Conformer layers, which performs convolution on an octave basis. This design facilitates the capture of intervallic and scale-related features.

1. INTRODUCTION

Automatic chord estimation is the task of estimating the chords being played, along with their corresponding time intervals, from audio data. It is an important problem with applications in music analysis and music genre classification. In general, acoustic features are extracted—typically using techniques such as the Fourier transform—and used as inputs to a model. The model produces chord annotations containing the start time, end time, and chord label.

In previous studies, it has been common practice to standardize the notation of enharmonically equivalent pitches during preprocessing, thereby representing pitch classes with 12 categories. However, from the perspective of music analysis, distinguishing enharmonic equivalents is important, since the scale degree within a given key can reflect the harmonic function and role of a chord.

In this submission, we propose an approach that decomposes the chord recognition task into two subtasks: key estimation and scale-degree estimation. The final chord label is then determined based on the estimated key and scale degree.

2. METHOD

In this section, we describe an overview of the system, and our original contributions: Octavewise Convolution and a novel chord representation.

2.1 ChordFormer

Our model is based on ChordFormer [1]. ChordFormer adopts the Conformer architecture [2] that combines convolutional neural networks (CNNs) for capturing local patterns with the Transformer's self-attention mechanism for modeling long-term dependencies. The ChordFormer architecture consists of three primary modules: Preprocessing Module, Conformer Block, Decoding Model.

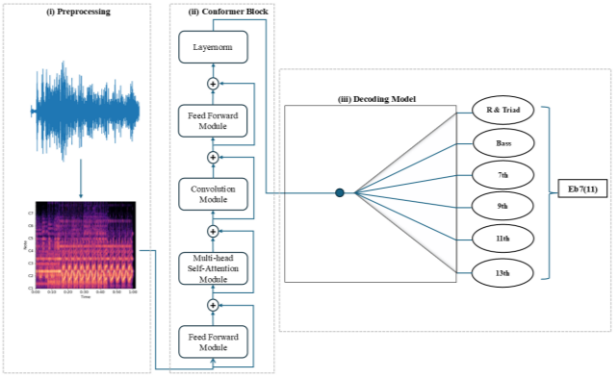

Figure 1. Overview of the ChordFormer architecture [1]

2.1.1 Feature Extraction

ChordFormer uses Constant-Q spectrograms as features, computed with the librosa library [3]. The audio signals were sampled at 22,050 Hz, and the spectrograms were computed with a hop length of 512. The frequency range was set from C1 (inclusive) to C8 (exclusive), with 36 bins per octave, resulting in a 252-dimensional feature representation. The spectrogram was then converted to a decibel scale with respect to the maximum amplitude.

2.1.2 Chord Representation

ChordFormer represents chords using a six-dimensional vector, with each dimension corresponding to an essential component of the chord. Specifically, the first dimension encodes the root note and triad type, the second dimension represents the bass note, and the third through sixth dimensions correspond to the seventh, ninth, eleventh, and thirteenth, respectively.

- **Root + Triad**: where root ∈ {N, C, C#/Db, …, B} and triad ∈ {N, maj, min, sus4, sus2, dim, aug}
- **Bass**: {N, C, C#/Db, …, B}
- **7th**: {N, 7, b7, bb7}
- **9th**: {N, 9, #9, b9}
- **11th**: {N, 11, #11}
- **13th**: {N, b13, 13}

2.2 Octavewise Convolution

In the proposed model, the acoustic features are first processed with a convolutional kernel of size one octave in the frequency direction (kernel_size = bins_per_octave), sliding by one semitone (stride = bins_per_octave // 12). The output is then passed through a linear layer to obtain a 256-dimensional representation, which is fed into the Conformer. This design aims to facilitate the model's ability to capture information related to pitch intervals and scale structure.

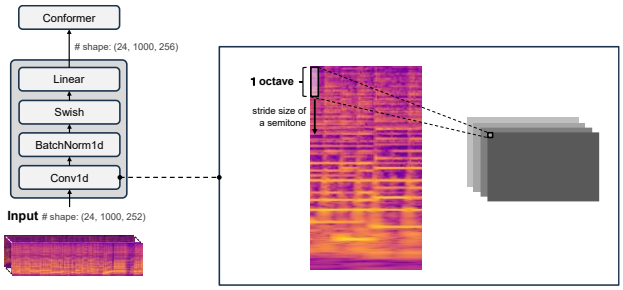

Figure 2. Visual illustration of Octavewise Convolution

2.3 Chord Representation

Conventional chord representations have two major limitations. First, since enharmonically equivalent notes are standardized, the system cannot output chord labels that are appropriate within a given key. For example, in a piece in B major, the 4-5-3-6 progression would be E–F#–D#–G#. However, if everything is written using flat notation, it would become E–Gb–Eb–Ab, which leads to awkward scale-degree labels such as bbVI or bIV when analyzing chord functions. This also raises the possibility that modulation may render contextual information ineffective. Second, chords that share similar pitch content and usage but are written differently (e.g., C6 and Am/C) are represented in very different ways. As a result, gradients may propagate in undesirable directions.

The proposed model is designed to distinguish enharmonically equivalent notes by decomposing the root into key and scale degree, thereby estimating three elements: key, root degree, and bass. In addition, 36 additional dimensions are introduced to represent the presence of each pitch from C to B, or intervals from perfect unison to major seventh relative to the root, as well as those relative to the bass. In total, chords are represented as 39-dimensional vectors.

- **Key**: {N, C, C#/Db, …, B}
- **Root Degree**: {N, I, #I, bII, …, VII}
- **Bass**: {N, C, C#/Db, …, B}
- **Absolute Pitches** (12 dim): {N, Contains}
- **Intervals from Root**: (12 dim): {N, Contains}
- **Intervals from Bass**: (12 dim): {N, Contains}

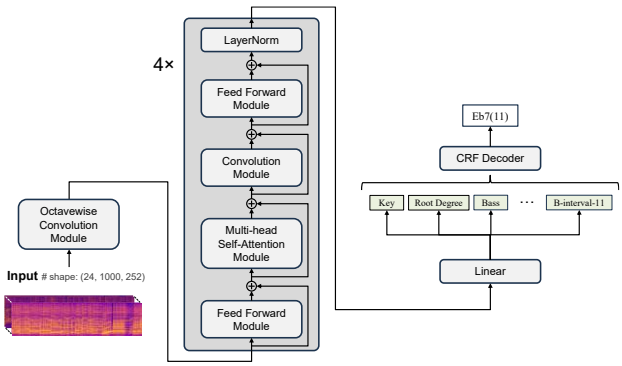

Figure 3. Overview of our system

3. TRAINING

In this section, we describe the data we used for training and related details of this submission.

3.1 Datasets

We used a total of 1,163 songs for training the model: 560 songs from the McGill-Billboard dataset [4] and 603 songs that we collected ourselves. From the McGill-Billboard dataset, we extracted musical key information from the SALAMI-format annotations, made some modifications, and then used it.

3.2 Data Augmentation

We employed two major strategies for data augmentation. The first, following previous studies, was to pitch-shift the input CQT spectrograms by -5 to +6 semitones and adjust the labels accordingly. The second was to apply SpecAugment from torchaudio, adding noise and performing time-stretching in the range of 0.8× to 1.2×, each with a probability of 50%.

3.3 Training Method

The model was trained following the same procedure as ChordFormer. Using the AdamW optimizer, the initial learning rate was set to 1.0×10^{-3}, and it was reduced by 90% if the validation loss did not improve for five consecutive epochs. To prevent overfitting, training was terminated once the learning rate dropped below 1.0×10^{-6}.

4. RESULTS

The evaluation metric (WCSR) computed using mir_eval [5] with 178 songs from the Isophonics dataset and 100 songs from the RWC-POP [6] dataset is presented below.

Metrics	Isophonics	RWC-POP
Root	0.8234	0.8454
MajMin	0.8139	0.8286
MajMin-Inv	0.7875	0.8082
Seventh	0.6645	0.6795
Seventh-Inv	0.6470	0.6616

Table 1. Weighted Chord Symbol Recall (WCSR) across datasets

5. REFERENCES

[1] M. W. Akram, S. Dettori, V. Colla, and G. Buttazzo, "Chord-Former: A Conformer-Based Architecture for Large-Vocabulary Audio Chord Recognition," *arXiv preprint* arXiv:2502.11840, 2025.

[2] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," Proc. *Interspeech* 2020, 5036-5040, doi: 10.21437/Interspeech.2020-3015

[3] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, vol. 8, 2015.

[4] 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*, Miami, FL, 2011, pp. 633–638.

[5] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in Proceedings of the 15th International Society for Music Information Retrieval Conference, 2014.

[6] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proceedings of the 3rd International Conference on Music Information Retrieval* (ISMIR), 2002, pp. 287–288.