Chord Detection Using Deep Learning

{ Xinquan Zhou and Alexander Lerch }
Georgia Tech Center for Music Technology

Introduction

Chord detection is to automatically recognize the chord progression in a music recording. Several studies indicate that deep learning methods can be very successful when applied to Music Information Retrieval (MIR) tasks, especially when used for feature learning [1]. Deep learning, with its potential to untangle complicated patterns in large amounts of data, should be well suited for the task of chord detection.

Data set

Our data set is a 317-piece collection:

- 180 songs from the Beatles dataset
- 100 songs from the RWC Pop dataset
- 18 songs from the Zweieck dataset and
- 19 songs from Queen dataset

The target chords are major and minor triads for every root note plus one none class resulting in 25 (12+12+1) chord labels. Ground truth time-aligned chord symbols are mapped into the major/minor dictionary:

$$Chord_{majmin} \subset \{N\} \cup \{S \times maj, min\}$$

Triad major/minor and seventh major/minor are mapped into the corresponding major/minor label. Other chord types are treated as unknown chords.

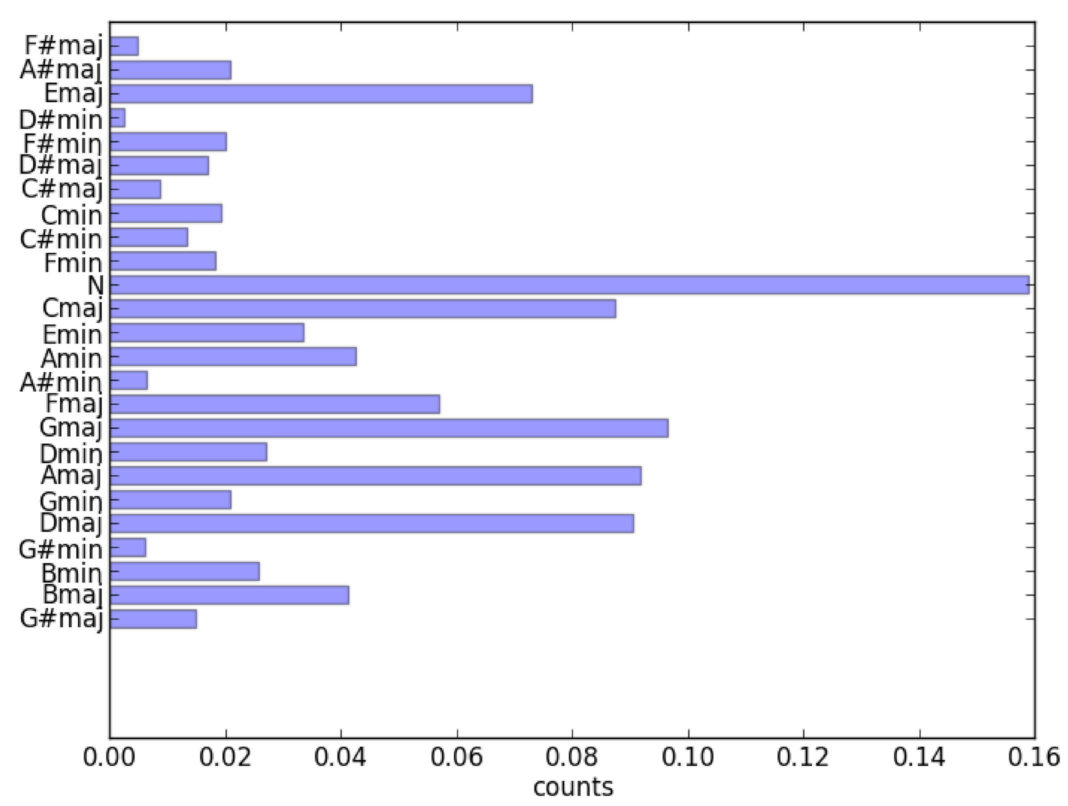

Figure 1: Chord distribution in our dataset

Proposed Methods

In this work, we investigate Deep Networks (DNs) for learning high-level and more representative features in the context of chord detection, effectively replacing the widely used pitch chroma intermediate representation. And feed the features into post classifiers to get the final results. The whole system is shown as below.

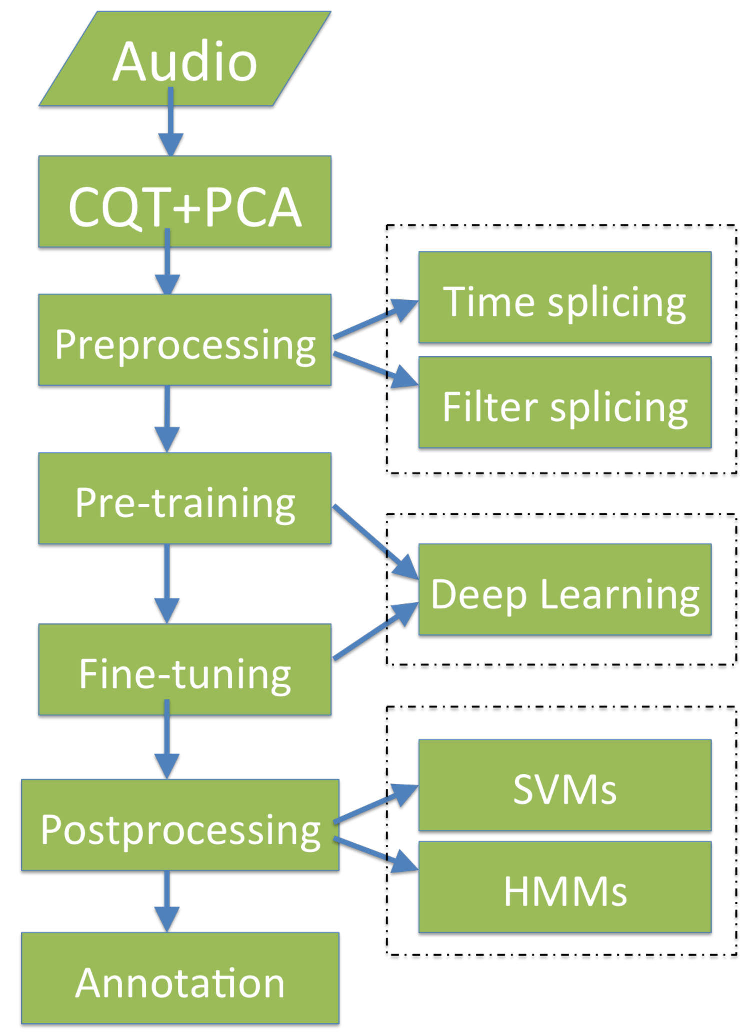

Figure 2: The flow chart of the system

For pre-processing, we employ both time splicing and "convolution" (spliced filters) methods.

- Time splicing is to simply join several adjacent frames into a "super" frame
- The "convolution" is to use a set of heuristic templates to convolve with the input frames and splice the results together, followed by an optional pooling operation.

An illustration of the convolution method is shown in the following figures:

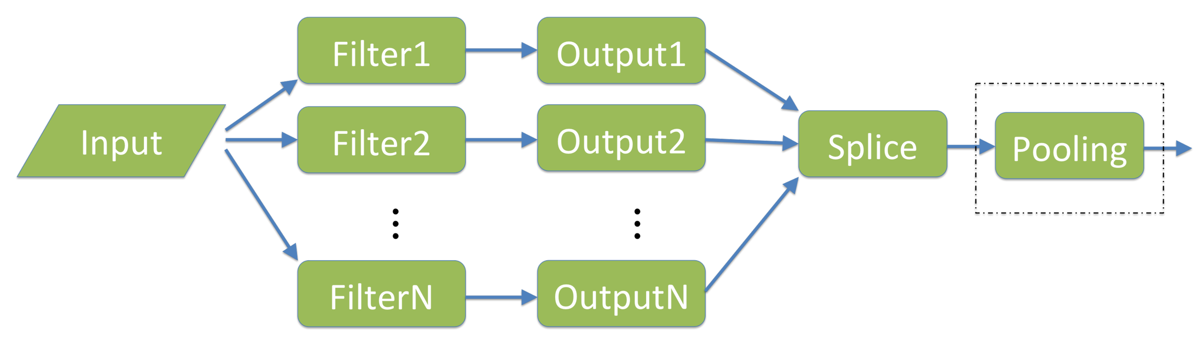

Figure 3: Spliced filters illustration

For networks architectures we compare bottleneck and the common ones;

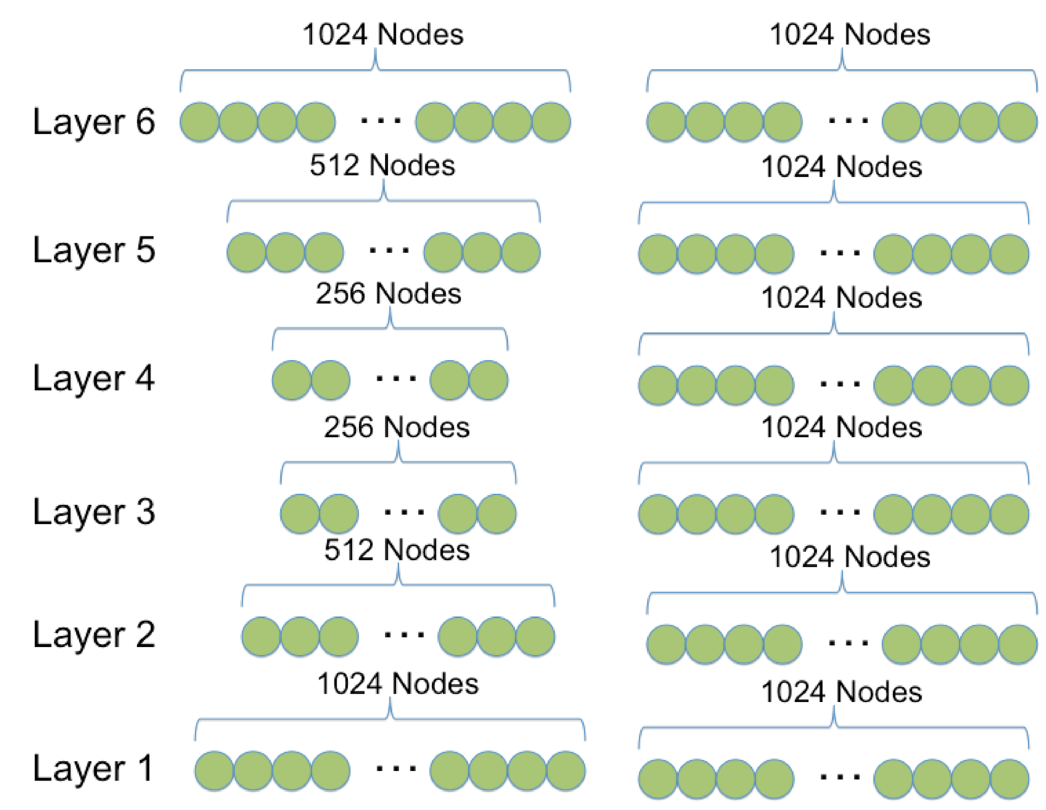

Figure 4: Architectures of bottleneck and common networks

Results

- Three different post classifiers are compared. The Weighted Chord Symbol Recall (WCSR) are used to evaluate the system.
- Both architectures is evaluated in comparison. WCSR computed on the training set is reported as well. Different pre-processing methods are applied respectively.

Classifier	Argmax	SVMs	HMMs
WCSR	0.648	0.645	0.755

Figure 5: Results using different post-classifiers

Method	WCSR
Chordino	0.625
Proposed (w/o max pooling)	0.919
Proposed (with max pooling)	0.916

Figure 6: Final results

- Finally, we compare our system with Chordino[2] under the default settings based on our dataset.

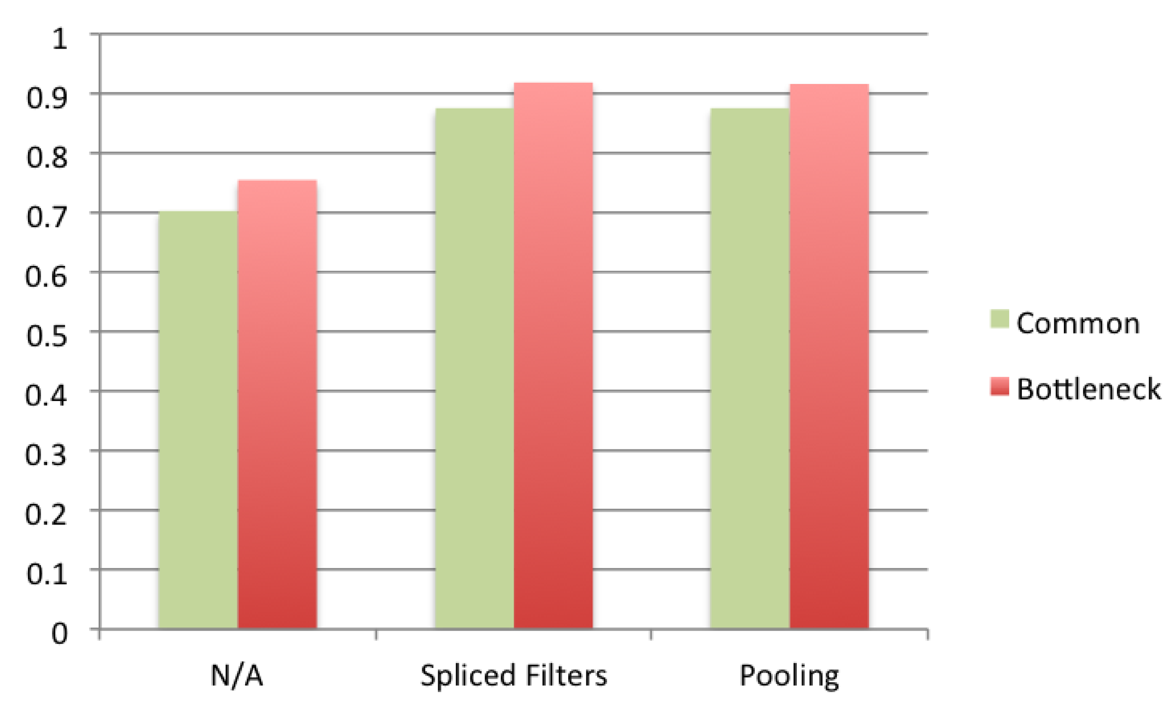

Figure 7: Results using different architectures

Conclusion

- Our model is able to learn high-level probabilistic representations for chords.
- The use of a bottleneck architecture is advantageous as it reduces overfitting and significantly increases classifier performance (p = 0.023).
- HMMs fits the task better than the static classifiers.
- The choice of appropriate filtering and splicing can significantly influence the performance.
- The pooling operation is preferable because it reduce the system complexity without sacrificing the performance.

References

[1] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, pages 339–344. Utrecht, The Netherlands, 2010.

[2] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 135–140, 2010.

Contact Information

Email xzhou89, alexander.lerch@gatech.edu
Phone +1 (404) 398 7794