Optical Music Recognition for LilyPond File Generation

Evan Matthews

Executive Summary

This report explores the feasibility of Lilypond as a digital music format in the process of optical music recognition and translation for full scores. A simple Convolutional Recurrent Neural Network (CRNN) implementation was trained on Inventions by Johann Sebastian Bach, with intention for test outputs of other Bach works to be analyzed for generalization in the Lilypond format. Overall, an implementation of the Shi, et al. CRNN model was unsuccessful in providing Lilypond predictions for analysis. However, current state-of-the-art CRNN with other digital music formats provide an extensive survey of what is can be accomplished in full-score optical music recognition.

Department of Computer Science
University of Illinois Urbana-Champaign
United States
May 10th, 2024

Optical Music Recognition for LilyPond File Generation

Evan Matthews

1 INTRODUCTION

1.1 Initial Proposal

In the world of music or audio transcription/arrangement, the complexity of sound and score data has allowed human performance to remain the current state-of-the-art. Several factors contribute to this circumstance: the number of potential audio and score file types, ambiguity on how performance qualities are notated, and overall inconsistencies between recordings and their respective scores. Machine learning models, in turn, have been a crucial step towards reducing these inconsistencies. Their ability to learn the nonlinearities and artistic qualities that otherwise plague audio computation have allowed for noteworthy advancements such as WaveNet [10]. The remaining issue in the process of sound generation is the amount of data available to train with. Worthwhile audio data remains difficult to collect due to its large size and potential copyright issues. In particular, trying to condition off another medium is incredibly difficult as current datasets lack the correlations necessary to streamline the audio transcription process. Datasets such as MAESTRO [4] are close to ideal results, but some intermediate steps are needed to learn correlations between audio files and musical scores.

With the following conditions in mind, I propose an Image-to-MIDI model to serve a few purposes. First, this model serves to convert image data to relative musical data to a high degree of accuracy. The process of converting MIDI to an image is trivial, but the opposite is a known Optical Music Recognition (OMR) problem in that the correlation is nonexistent. Second, a highly reliable model of this type is capable of serving as an intermediate step in future audio computation endeavors, as the nontrivial nature of OMR is a barrier for reliable model training between scores and other types of audio data. Finally, a conversion of this type allows for more flexible datasets because MIDI can be converted into a number of other audio file types.

1.2 Known Limitations

Given the vast complexity of classical western music notation and intricacies in notation/time alignment, the proposed model maintins a rather pessimistic approach. That is, given a small subset of musical data (*among all notated music*) and some model, results can be expected to only match expected outputs through one or more of: pitch, rhythm, duration, and overall formatting. Existing research in OMR supports these expectations. First, several digital notation systems exist for music, including:

- Standards like *MusicXML*, *Lilypond* and *MIDI*,
- Software-specific formats from *Musescore*, *Sibelius* and *Finale*,
- *KERN* from the Humdrum tool-set [8],
- Mayer, et al.'s *Linearized MusicXML* [6], and

- Contreras, et al.'s untitled "end-to-end OMR" encoding language [2].

Each format provides benefits and drawbacks towards generalized OMR, but combined research efforts have yet to hone in on a particular format. Second, current research continues to limit its effective musical scope: constant genre, time period composed, single vs. multi-line pieces, and single vs. multiple measures, to name a few. These limitations are to be expected as a means of balancing experiment accuracy with the subset of musical notation to be recognized. However, no paper to date has intended to capture a high accuracy while completely generalizing the space of recognizable music.

Finally, the state-of-the-art for machine-learning-based OMR lies in the implementation of Convolutional Recurrent Neural Networks (CRNN). This model type is preferred over CNN for its ability to learn data as order-dependent sequences- a crucial philosophy in parsing and understanding musical scores in general. However, it should be noted that CNNs are still viable for their non-order-dependent musical problems, such as Nugroho and Zahra's work on individual note and duration recognition [7].

1.3 Research Questions

Despite low expectations for training and results, I believe that a few questions can be posed and answered for the purpose of bounding requirements on a larger, all-encompassing Image-to-MIDI model:

RQ1 What forms of recognition can be expected from a generalized CRNN implementation?
RQ2 How does Lilypond recognition compare against other standard and custom music formats?
RQ3 What conflicts currently prevent state-of-the-art models from further generalization?

For **RQ1**, I aim to recover noticeable results regarding the CRNN model's ability to generalize sequences of data from scores. Hence, whether or not order-dependent training can pick up on note, rhythm or duration sequences will be crucial to my final analysis. Next, promising results from **RQ1** will be compared against existing research to answer **RQ2**. In particular, the generalized CRNN model's ability to render score data as Lilypond (.ly) files is compared against other notation systems to determine if a more complex Lilypond-based model would be successful. Finally, for **RQ3**, the results of CRNN model training are criticized on what should be generalized or further implemented in order to reliably translate score images into Lilypond data. This research question refers to all-encompassing limitations such as in **1.2** along with experiment-specific limitations.

2 BACKGROUND

2.1 Optical Music Recognition

Current research in Optical Music Recognition (OMR) can be placed in one of two categories:

- **Symbol Recognition:** the ability for a model to recognize individual symbols on a score.
- **Music Transcription:** the ability for a model to recognize and transcribe multiple symbols or long passages into a new musical format.

The problem of Symbol Recognition is nearly identical to that of text recognition with the exception of a separate set of possible symbols. Additionally, unlike situations where text is perfectly aligned for recognition, musical symbols need to be recognized with respect to their shape and position in order to achieve a high accuracy. All notes, for example, have unique pitches dependent on their placement in a score, (on a "line" or "space" in a staff, above or below a staff). The playback of notes is also determined by surrounding symbols such as accidentals (natural, sharp, flat), rhythmic symbols (the type of duration, a "dot" for an additional half of that note's duration). Finally, the highest level of context is staff-related information, which includes clefs (treble, bass, alto, tenor; determining the relation between note placement and audible pitches), time signatures and key signatures (for adding context for an entire piece of music).

On the other hand, Music Transcription encompasses the entire Symbol Recognition problem, (iteratively for some amount of sheet music) as well as problems with correlating recognized symbols in sequence. On its own, Symbol Recognition could recognize every component of a piece of music, but these results lack the ordered context necessary for transcription. State-of-the-art transcription through machine learning, as a result, has been dominated by the Convolutional Recurrent Neural Network (CRNN). Unlike CNNs, CRNNs make use of recurrent layers, (such as bidirectional LSTM layers in Shi, et al. [9]) in order to contextualize sequences of data forwards and backwards. These layers, along with attention cells focused on deriving pitch, rhythm and durational context, allow for CRNNs to recognize and output ordered sequences of logical musical data.

Additionally, it is important to recognize that musical data sequences are ordered along multiple dimensions. Contreras, et al. provides an excellent diagram of this ordering in their work with respect to how notes have a separate "ordering" from other symbols. State-of-the-art CRNNs, in turn, are able to recognize this multi-dimensional ordering to a point, (see 5.3.1) so long as specific elements of the music are kept constant. In the case of most OMR research, the rhythms and durations are kept constant across multiple voices, (unless only a single voice is used).

2.2 Lilypond

Lilypond was chosen for dataset file representation as it maintains accurate translation between digital (MIDI) and visual (image) notation systems [5]. In particular, translation from

Figure 1: multi-dimensional ordering in sheet music [2].

Lilypond to MIDI is trivial, and Lilypond provides a more intuitive representation of musical information compared to raw bytes of MIDI data. This notation language consists of multiple layers to separate score components:

- **Document level:** components related to page layout and high-level musical details, (number of instruments/tracks, score engraving information).
- **Music level:** components related to low-level musical details, (notes, rhythms, durations, key/time signatures, tempo).

At the document level, aspects of a score that stay mostly or completely constant throughout a piece are indicated by specific, indent-sensitive keywords. For example, the number of voices/instruments and respective number of staves are initialized with the $\backslash Voice$ and $\backslash context$ commands, while high-level musical information is initialized by commands such as $\backslash time$ for time signature, $\backslash key$ for key signature, and $\backslash tempo$ for performance speed in beats per minute.

At the music level, rhythmic musical symbols are notated according to their pitches and relative time duration. pitches are all characters a, b, c, d, e, f, g, r, where r is a "rest" meaning no pitch occurs. To ascend or descend in pitch beyond a single musical "octave," the $'$ and $,$ characters are appended to indicate one or more octave ascendings or descendings, respectively. Pitches are also proceeded by a rhythmic value representing its duration in time. These values are typically powers of two, (but can technically be any positive integer), and they dictate how long a pitch is played with respect to the piece's time signature. For example, a score with time signature $= 2/4$ indicates that each measure has two beats, and each beat is the length of a quarter note (hard-coded in the denominator). In this case, the line "$c4 \quad d4$" would represent two quarter (4) notes or a single measure in the provided time signature. Additionally, notes without defined durational values take on the previous note's duration in a line, so a series of equal-duration notes is represented by one durational value. Finally, separations between measures are made with "$|\%n$", marking the end of measure n. Figure 2 compares the rendering of a single staff of music with its relative Lilypond notation. [1]

---

[1] The time signature C or "common-time" is another common way of writing 4/4.

```
trackBchannelB = \relative c {
        r16 c' d e f d e c g'8 c b32 a b16 c8
        | % 2
        d16 g, a b c a b g d'8 g f32 e f16 g8
        | % 3
}
```
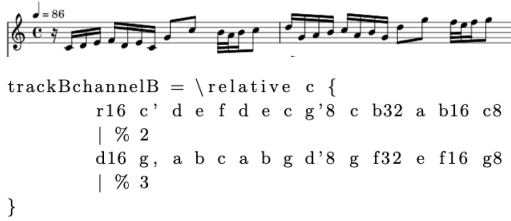
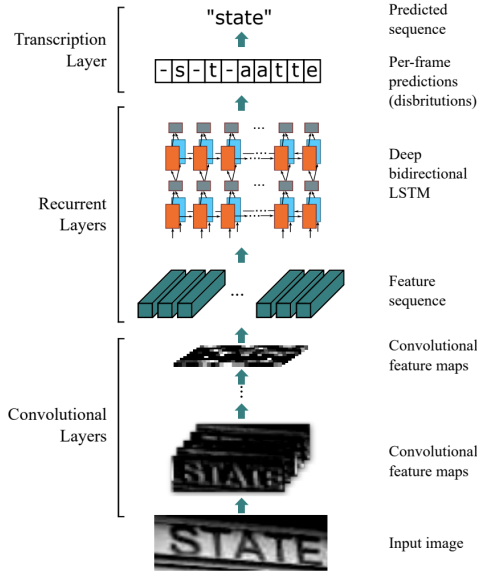**Figure 2: Lilypond (.ly) code and rendering.**

Figure 3: General CRNN architecture, as modeled by Shi, et al. [9].

3 EXPERIMENT

3.1 CRNN Model

The proposed Image-to-MIDI model is a smaller model referencing Shi, et al.'s pioneering work in the creation of CRNNs [9]. The model itself consists of multiple two-dimensional Convolutional/MaxPooling pairs for the purpose of extracting features from the intial score images. Batch normalization layers are also applied to refine data recovered from the feature extraction. Lastly, features are pushed through two bidirectional LSTM layers- the recurrent layers of the model- for data prediction with respect to "sequences" of the images. In this case, sequences of the image data are the pixel columns of $(W \times H)$ images, where W is the pixel width of standard "letter" paper commonly used for scores, and H is the height of the image dependent on the overall "length" of the music.

3.2 Data Specfications

MIDI data used for this experiment has been narrowed down to strictly fifteen two-stave "Inventions" by Johann Sebastian Bach [1] to maintain rhythmic and compositional uniformity.

These works maintain relatively similar quantization schemes (rhythms and durations of notes), are short, and have several clear transcriptions for research, education, and entertainment purposes.

Testing is performed against a handful of arbitrary "preludes" and "fugues," also written by Bach. Limiting the training and testing sets, (especially in a way that limits the ability to calculate an overall accuracy), is purposefully done in the context of this experiment. First, there are no expectations for relatively "accurate" recreations of score images in Lilypond. A proper training model would require extensive tokenization for all symbolic details that help to render a Lilypond file, which is outside of the scope of this research endeavor. Rather, output is manually analyzed against its correlated image to determine what musical or notational qualities were generalized. Second, a broader dataset would drastically complicate the CRNN model's ability to generalize key features of score images. Focus on a single composer, musical time period, and music "type" (prelude, waltz, rondo, etc.) allows the model to more quickly generalize aspects of the subset of music, at the cost of potentially overfitting to the subset.

3.3 Quantization Rendering

Similar to simple transformations performed on images, (rotations, coloring, scaling), output for sheet music is determined by both technical and musical constraints that don't affect the "performance" of the composition. In particular, when score images are rendered from the Bach lilypond files, (trivially translated from the Bach MIDI Index [1]), there is a required "quantization" value which affects the rhythms and durations visually. This quantization, designated $Q \in \mathbb{Z}^+$, represents what specific note duration counts as a "beat" within a given score. Figure 5 lists the most common duration mappings that occur within this experiment.

Drastic change of Q results in score images that appear muddled or illegible despite conveying the same musical information, as shown in Figure 6. In order to prevent drastic shifts due to quantization, an optimal quantization $\hat{Q}$ is calculated with

$$\hat{Q} = \max_{t \in \text{MIDI events}} \frac{\tau * 10^{-6}}{t} * n * \text{floor}(d/4) \qquad (1)$$

where t is the number of seconds for a MIDI event to occur, τ is the provided MIDI tempo in beats per microsecond, and (n, d) are the numerator and denominator, respectively, of the piece's time signature. Additionally, possible $\hat{Q}$ are limited to powers of 2 greater than 0 in order to reduce complication from compound time signatures, (where beats are not represented by durations in Figure 5).

4 RESULTS

Due to technical conflicts with my intended dataset, (variable-height images against Lilypong string data), the CRNN model proved unsuccessful in training and producing any results

Figure 4: Excerpt of Invention No.4; cropped/prepared rendering of Invention No.8.

Figure 5: Table of common duration-integer mappings, from Francis Hamzagic [3].

Figure 6: Levels of quantization $Q = (1, 32)$ in rendering, respectively.

in the Lilypond formatting language. While this result complicates my analysis of Lilypond for machine learning, a meta-analysis of state-of-the-art OMR research with CRNN models still provides significant findings towards the previously mentioned research questions.

5 DISCUSSION

The following section details my meta-analysis of reference work with respect to the previously mentioned research questions (**RQ1-3**).

5.1 Model Generalization (RQ1)

Given the visual encoding aspects of Lilypond, a simple CRNN model would struggle to accomplish several layers of recognition at once. However, multiple simple CRNNs would be successful in learning the Lilypond file schematic and musical information provided that both levels are tokenized and separated. The work of Nugroho and Zahra, for instance, validates the ability of a simple CNN architecture to solely recognize musical symbols in context [7].

With additional features and data preparation, CRNN models are shown to accurately parse and transcribe full scores of monophonic[2] and homophonic [3] music [2]. In turn, the additional preparation, which forcibly standardizes the data to assist in training, sterilizes the diverse nuances that come with all possible sheet music, including physical aging, printing error, different notation fonts, sizes and shapes of scores, and the immense dictionary of musical symbols, (standardized or unique to specific pieces of music).

5.2 Format Comparison (RQ2)

Lilypond is not, by all means, a perfect music formatting language. Being a format focused on the visual clarity of musical notation, I find that the aforementioned "document level" greatly hinders Lilypond from achieving basic recognition. In particular, additional keywords for the representation and position of solely visual elements of a score, along with their necessary language features like brackets and spacers, strongly inhibit basic CRNN models from making noticeable music generalizations.

More complex CRNN models and state-of-the-art work currently provides towards support for Lilypond in this context. Mayer, et al.'s *Linearized MusicXML* [6], a specially-designed superset of MusicXML, demonstrates how complex CRNNs can accurately generalize document-level and music-level data.

5.3 Model Criticism (RQ3)

My CRNN implementation, along with the CRNN models for which my research references, are not without active barriers preventing consistent musical generalization. In staying consistent with my description of the Lilypond formatting language, my critiques against CRNNs are categorized by what further work should be done in order for music-level and document-level generalizations to occur.

5.3.1 Music-level Generalization. The primary issue holding current CRNN models back is their inability to generalize to polyphonic [4] music. That is, state-of-the-art models fail to solve the problem of recognizing and transcribing several different "voices" of music at the same time. Solutions related to the separation of these voices, (i.e. removing sequences as they are recognized from the original score), approach

the homophonic algorithm, but they also introduce problems. For instance, when a voice is removed, determining whether intersecting voices were affected is rather difficult.

Additionally, a majority of OMR research relies on datasets consisting of digital or digitized musical scores. This choice for data is fair when considering that OMR models require upwards of hundreds of scores for proper training [6], but it also limits the diversity of scores and contexts that ML transcription can be used for. In particular, standards for scores have changed several times over centuries of music history, yet the formatting of historical scores is often left behind in modern transcriptions. The solution to this issue, while partly related to current models, lies in the amount of trainable data across all eras of classical music and the ability to notate historical scores. Modern notation software, while addressing current standards, thankfully includes typing for historical symbols, but these symbols have yet to be properly implemented in popular formatting languages. Historical datasets are also quite possible, albeit as scanned images with no manually-transcribed digital reference as of the current day.

5.3.2 Document-level Generalization. At the document level, the greatest issue plaguing OMR research is the lack of a formatting language standard, or lack of similarity between all formatting languages. Every formatting language has separate means of conveying equivalent musical information based on the user's needs, (i.e. MIDI for real-time data, MusicXML for web-related music, Lilypond for polished layouts). However, as the needs of users vary in the numerous niches of music production, technology and composition, digital music encoding and datasets slowly diminish the possibility of standardization. While MIDI and MusicXML are currently the most popular for datasets, both formats lack consistency in rendering sheet music across all digital notation platforms. I believe consistency on this front is crucial for reducing variability in musical machine learning datasets. In particular, sheet music should "look the same" regardless of the formatting language it is imported from.

6 CONCLUSIONS

Overall, despite unsuccessful training of a simple CRNN model on Lilypond data, the following meta-analysis proved to be quite beneficial in analyzing the Lilypond formatting language. Existing research from several CRNN papers confirms the viability of Lilypond as a format for machine learning, and I believe a combination of this, tokenization for layout-related encoding, and active development of the language would allow it to thrive in current research. I'm especially interested in seeing Lilypond extend into OMR research as its backend contains useful tools for conversion to MIDI and MusicXML, formats with plenty of existing datasets but higher complexity in generalization.

In the future, I would like to focus on a tokenization system for future Lilypond ML models, as this sudden obstacle was the reason for the experiment being unsuccessful. Additionally, given my interest in music notation and conversion to

---

[2] *monophonic* refers to having one voice of musical information.
[3] *homophonic* refers to having multiple voices of music sharing the same note rhythms and durations.
[4] *polyphonic* refers to having multiple voices of music with different rhythms and durations.

related formats, I am hoping to generate non-text-related outputs from digital music data, including audio for performances, video for artistic interpretation, and more scientific metrics to bridge the gap between "musical emotion" and psychology.

7 CONFLICT OF INTEREST

My ongoing plans for thesis work, advised by Paris Smaragdis, revolve around the generation of audio conditioned on score images. While this report actively assists in my understanding of the correlations between digital music formats and computer vision, I do not believe that it trivializes any future work in this topic.

REFERENCES

[1] [n. d.]. Complete Bach Midi Index. https://www.bachcentral.com/midiindexcomplete.html

[2] María Alfaro-Contreras, JoséM. Iñesta, and Jorge Calvo-Zaragoza. 2023. Optical music recognition for homophonic scores with neural networks and synthetic music generation. *International Journal of Multimedia Information Retrieval* 12, 1 (2023), 12. https://doi.org/10.1007/s13735-023-00278-5

[3] Francis Hamzagic. 2018. Music Theory for Producers - Time Signature Part 2. https://creatingtracks.com/music-theory-producers-time-signature-part-2/

[4] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1lYRjC9F7

[5] Lilypond 2024. Lilypond music notation for everyone. https://lilypond.org/doc/v2.25/Documentation/web/index

[6] Jiří Mayer, Milan Straka, Jan Hajič jr. au2, and Pavel Pecina. 2024. Practical End-to-End Optical Music Recognition for Pianoform Music. arXiv:2403.13763 [cs.CV]

[7] Douglas Rakasiwi Nugroho and Amalia Zahra. 2024. Musical Note Position and Duration Recognition Model in Optical Music Recognition Using Convolutional Neural Network. *Journal of Image and Graphics* 12, 1 (Jan 2024), 32–39. https://doi.org/10.18178/joig.12.1.32-39

[8] Antonio Ríos-Vila, David Rizo, José M. Iñesta, and Jorge Calvo-Zaragoza. 2023. End-to-end optical music recognition for pianoform sheet music. *International Journal on Document Analysis and Recognition (IJDAR)* 26, 3 (2023), 347–362. https://doi.org/10.1007/s10032-023-00432-z

[9] Baoguang Shi, Xiang Bai, and Cong Yao. 2015. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. arXiv:1507.05717 [cs.CV]

[10] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. arXiv:1609.03499 [cs.SD]