A NOVEL DATASET AND DEEP LEARNING BENCHMARK FOR CLASSICAL MUSIC FORM RECOGNITION AND ANALYSIS

Daniel Szelogowski   Lopamudra Mukherjee   Benjamin Whitcomb

University of Wisconsin - Whitewater

GitHub Repository & Supplement

University of Wisconsin Whitewater

Abstract

Automated computational analysis schemes for Western classical music analysis based on form and hierarchical structure have not received much attention in the literature so far. We provide a system for computational analysis of classical music, both for machine learning and music researchers. First, we introduce a labeled dataset containing 200 classical music pieces annotated by form and phrases. Then, by leveraging this dataset, we show that deep learning-based methods can be used to learn Form Classification as well as Phrase Analysis and Classification, for which few (if any) results have been reported yet. Taken together, we provide the community with a unique dataset as well as a toolkit needed to analyze classical music structure, which can be used or extended to drive applications in both commercial and educational settings.

Form Analyzer

- Classify classical piece as one of 12 possible forms:
  - Arch
  - Bar
  - Binary
  - Minuet & Trio
  - Ritornello
  - Rondo
  - Sonata
  - Ternary
  - Theme & Variation
  - Through Composed
  - Unary/Strophic
  - Unique

TreeGrad Model [33] – a hybrid Neural Network/Decision Tree with a very low sensitivity to noisy data; trains extremely quickly compared to standard Neural Network models

Peak-Picking Alg.

- Break down audio file using Onset Detection methods to discover the peak event audio frames
- Return this set of frames as a series of timestamps representing the musical phrases

Onset Detection Algorithm [35] – a fast, unsupervised peak-picking algorithm comparable to other Convolutional Neural Network models; could be used for self-supervised NN training

Phrase Analyzer

- Using the form classification and phrase timestamps, classify each timestamp sequentially
- Timestamps may include multiple labels (part and phrase) or individual (transition, phrase, etc.)

LSTM-Tree Model [12] – a hybrid Bidirectional-LSTM/Decision Tree that provides a more accurate output than either model individually; Bi-LSTM is used to fit D-Tree for final output

Prediction System

- Combine the outputs of all 3 major components
- Present final analysis formatted to match the training data (filename, form, and labeled timestamps)

Combined System – form classification and phrase timestamps are provided to both the end user and Phrase Analyzer, which outputs the combined **form**, **time**, and **phrase** predictions

Background

System 1 (1998)	Melody and harmony generator using **Feed-forward Neural Networks**; unable to learn higher-level musical structures occurring simultaneously and at multiple time scales or recognize melodic vs. harmonic context of notes and intervals
System 2 (2007)	Automatic musical style recognition through classification of harmonic, melodic, and rhythmic descriptors using **k-NN**, **Self-Organizing Maps**, and **Bayesian classification**; SOMs may be useful for formal analysis, system designed to recognize low-level features only
System 3 (2014, 2015, 2016)	Boundary recognition using **Convolutional Neural networks**, **Mel Spectrogram**, and **Self-Similarity Lag Matrices** to estimate fixed-depth segmentations based on SALAMI annotation levels; boundaries evaluated by time tolerance. Also used to generate audio thumbnails
System 4 (2020)	Automatic musical structure detection and segmentation using **multi-resolution community detection** and **graph theory** to perform boundary detection and structural grouping, yielding a structural hierarchy. Noted that CNNs will continue to lack improvement without recurrent layers

Methodology

Figure 1. Example of phrase labeling from analysis of Bourrée from J.S. Bach's BWV 996 on a human-annotated score, where the relation between phrases and their respective part can be seen hierarchically. On an analyzed score, it is standard that only the first instance of a structure is labeled, although it may continue far beyond the initial instance.

Develop a system of three components: Form Analyzer, Peak-Picking Algorithm, and Phrase Analyzer

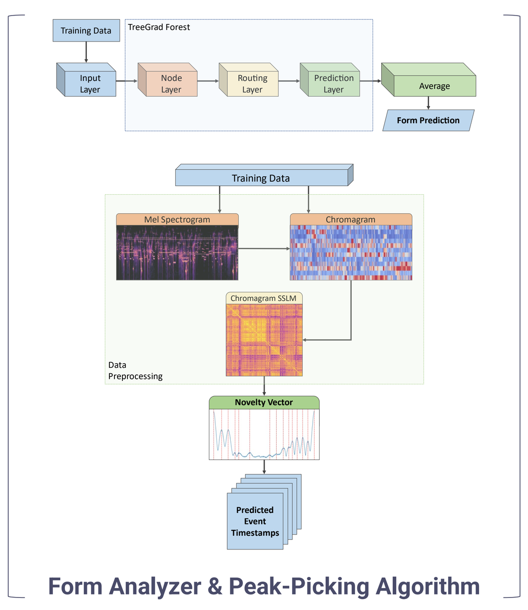
Use hybrid Neural-Decision Tree models to train quickly and reduce overfitting

Dataset built from 200 manually classified MIDIs, augmented with 5 different sets of permutations to expand dataset to 1,200

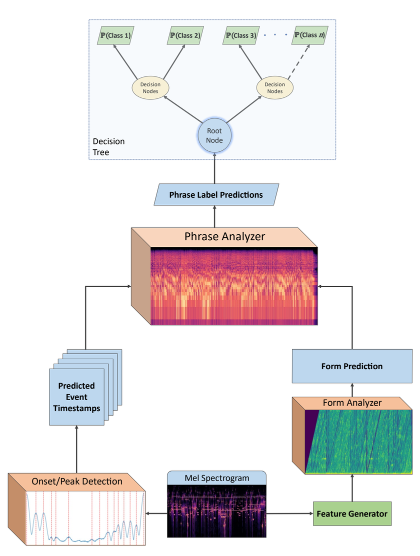
Contributions

- Using feature selection and elimination methods, we found that the two most important data features for Form Classification were the **music duration** and the **Self-Similarity Matrix (SSM) of the (Mel) Spectrogram**
- The dataset was **augmented** using **pitch, time, speed**, and **starting-point shifting** methods to expand the 200-piece dataset to 1200 – the data is publicly available on GitHub for extended contribution (https://github.com/danielathome19/Form-NN/tree/master/Data)
- Each piece of music was converted to its **Spectrogram SSM**, and the mean and variance were used to reduce the 2D array into 1D – a common approach for **feature scaling** and **dimensionality reduction** in signal processing
- This set of data (including the duration and numerous unused pre-calculated features) was stored as a **data table** for ease of extensibility and reduced computation time during training
- The data for the Form Analyzer was scaled using $X = \frac{(X - mean(X))}{std(x)}$, and using **Min-Max Scaling** for the Phrase Analyzer ($X = \frac{X - \min(X)}{\max(X) - \min(X)}$).

Data Extraction & Prep



Form Analyzer & Peak-Picking Algorithm



Phrase Analyzer (Final Output)

System Architecture

Experiments and Results

- A hybrid **Deep Neural-Decision Forest** architecture (known as **TreeGrad**) was used to fit the dataset as an ensemble network using **Stacking.** This model trained extremely quickly and fit to the dataset with **high accuracy and low error** – hence, overfitting was not an issue compared to **non-hybrid models** (CNN, DNN, etc.)
- To combat overfitting using the **pruning methods** employed by decision trees, TreeGrad models each tree in the ensemble as a three-layer neural network to create a **Neural Decision Tree**. Each Neural Decision Tree is comprised of a **Decision (Input) Layer**, **Node/Routing (Hidden) Layer** which controls the branching of each node in the tree, and a **Prediction (Output) Layer**
- Other machine learning algorithms such as **Random Forest** and **Extra Trees** were attempted, but provided unusable or highly-overfit output due to the Multilabel Classification; the LSTM-Tree also appears to **prioritize the large form labels**, and often tends to leave out the phrase label or generalizes it as a "section" without a unique letter
- The full (augmented) dataset was split into **85% training** and **15% testing** (or validation)
- The Form Analyzer was evaluated using both validation accuracy (or **Jaccard score** in this case) as well as **Precision/Recall/F1** scores. The final model solely uses the TreeGrad model to perform the prediction – the **Mel Spectrogram SSM** is calculated on the fly, then passed to the model along with the **duration**

Form Analyzer – TreeGrad

- The Peak-Picking (also called "**Onset Detection**") algorithm uses the **Mel Spectrogram** and **Self-Similarity Lag Matrix (SSLM) Chromagram** (a graph of pitch class distribution by time) to detect peaks in the audio
- The Chromagram SSLM is computed using **k-Nearest Neighbors** to cluster pitches in the Mel Spectrogram. The **computed vector of peaks**, represented by audio frames captured by the Short-Time Fourier Transform (STFT), is returned as **an array of timestamps**
- While the Peak-Picking Algorithm was not evaluated using a formal metric, the algorithm was tested against the training data and the output timestamps were often found to be **nearly identical** or **had a low enough difference** to be subjectively true (similar to human bias)
- Using the **timestamps** provided by the Peak-Picking Algorithm and the **labels** output from the Phrase Analyzer, the piece of music can be **score studied** (i.e., analyzed within the sheet music) much quicker for rehearsal and research use, for example
- The output of the novelty function was compared to **numerous hand-labeled pieces** from the dataset, and we found that the **difference was negligible**. Based on our comparisons, it was **more feasible** (and both faster and accurate) **to use the algorithm than to train a CNN** to perform the same task and greatly reduced system design time (given the lack of training necessary to perform the calculations)

Peak-Picking Algorithm – Onset Detection

- The features selected for the Phrase Analyzer include the **Form classification**, **timestamp**, **audio slice duration**, and the **Mel Spectrogram**. Hence, prediction requires both an accurate Form prediction and Peak-Picking results
- The model was implemented using a hybrid architecture – a **Bidirectional LSTM** (a form of Recurrent Neural Network) is fit to the data, then the output of the last hidden (Dense) layer is used to fit a **Decision Tree** to perform the final prediction (referred to as **LSTM-Tree**)
- This model is **much more difficult to score programmatically**, as numerous factors affect the final system. The labels are often **highly subjective**, and **some labels are implicit** (part A continues until timestamp *n* but is normally only labeled at the first occurrence).
- If the data was split into a test set, the results would likely be less truthful of the model's performance due to **poor generalizing**
- While there was currently little room for improving the model outside of **manually expanding the dataset** (after optimizing the hyper-parameters), we found that output of the final model was **objectively comparable** to our ground-truth analyses. As such, the model is practical enough to be used as an **assisting tool** for human analyses (such as for expanding the dataset), and was thus considered as good as currently possible

Phrase Analyzer – LSTM-Tree

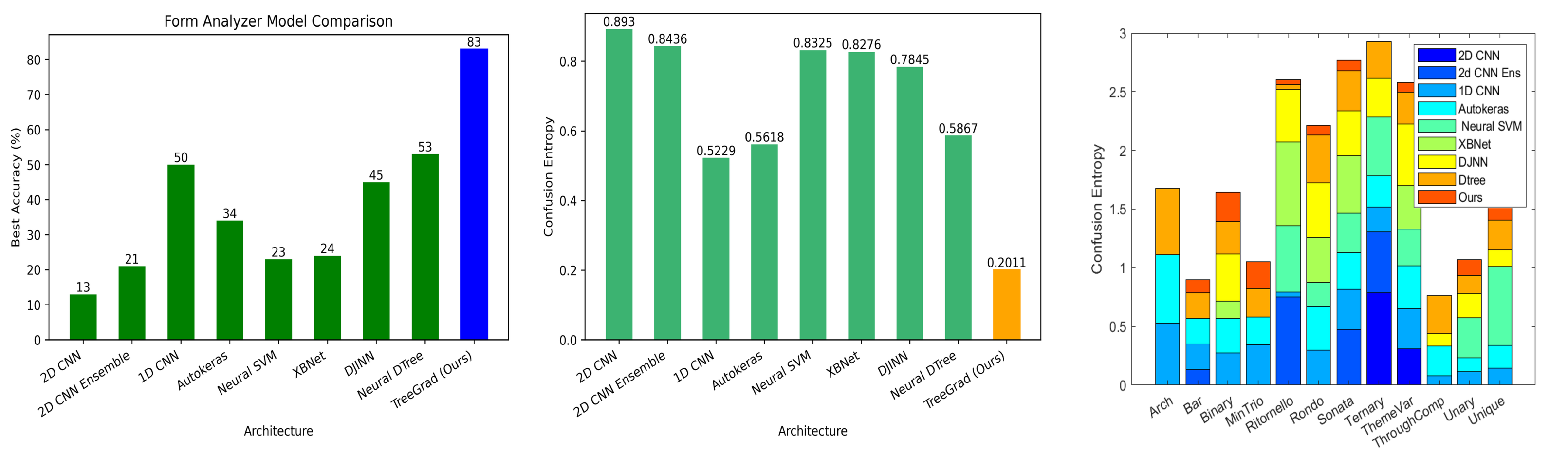


Figure 5. Form Analyzer Architecture Comparison with other methods (left), Confusion Entropy calculated for each method (center), and the class-wise Confusion Entropy for each method (right)

```
Performing predictions on anna-magdalena_book_14
        Predicted form: Unary
Performing predictions on brahms_opus117_1
        Predicted form: Ternary
Performing predictions on faure_nocturne_99_no10
        Predicted form: Sonata
Performing predictions on bthvn_pno_concerto_2_19_3
        Predicted form: Rondo
Performing predictions on schubert_D935_2
        Predicted form: Ternary
Performing predictions on schumann_evening_song
        Predicted form: Rondo
Performing predictions on schbrt_strquartet_13-mvt3
        Predicted form: MinTrio
Performing predictions on tchaik_nocturne_19_4
        Predicted form: Binary
```

Sample prediction output from Form Analyzer

Result #1 – Form Analyzer

The final Form Analyzer model achieved a **maximum accuracy of 83%** -- precision and recall were closely correlated to this score. May perform better as an **ensemble**

Result #2 – Peak-Picking Algorithm

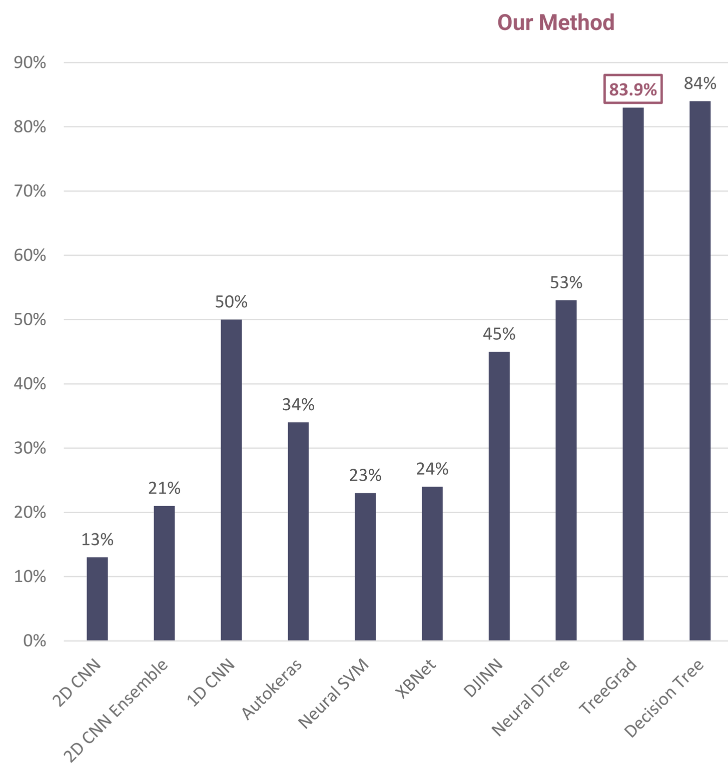
The Peak-Picking algorithm proved **comparable to other machine learning approaches** (CNN, Self-Organizing Maps), as even pre-labeled data points were nearly identical to those marked by a human analyst

Result #3 – Phrase Analyzer

In comparison to other models, **LSTM-Tree outperformed** both individual NNs (DNN, CNN), ML algorithms (decision tree, random forest), and TreeGrad

Result #4 – Approach

Using a **hybrid NN-Decision Tree approach** greatly reduced overfitting and thus increased accuracy for both Form and Phrase analyzers




```
brahms_opus117_1
        Ternary
      Guesser: LSTMTree   DcnTree TreeGrad
          0.0  Silence   Silence  Silence
          0.1   [A, a]       [A]        B
       21.177      [A]       [B]        B
        38.87 [A, sec] [CODA, f]        B
       57.121 [A, sec] [CODA, f]        B
       67.013      [A]  [A, sec]        B
        96.27      [B]  [A, sec]        B
      150.187      [b]       [A]        B
      167.741      [a]  [A, sec]        B
       186.41 [B, sec]       [B]        B
      200.899   [B, c]       [A]        B
      210.512   [CODA]       [A]        B
      223.608   [CODA]  [A, sec]        B
      237.958      [A]  [A, sec]        B
      252.029      [A]       [A]        B
      269.444      End       End      End
```

Final prediction system output includes Decision Tree and TreeGrad for comparison

Discussion

Curriculum Learning

The LSTM-Tree may benefit from using a **Curriculum Learning** approach, much like that of a traditional Form and Analysis class. An Autoencoder or Seq2Seq model may be useful in creating a more accurate/faster system

Music Education

The final Form-NN system is currently accurate enough to be implemented as the backend of a **higher-level system** such as an **assisted grading tool** for human-analyzed scores or a musical practice tool

Segmentation Model

The Peak-Picking algorithm could be used to train a more accurate **music segmentation network**, allowing the entire system to be treated as one large Deep Learning system

Anthology Compilation

The current dataset features class **imbalance**; anthologies of classical music classified by form are lacking, though this system could be used to assist in compiling such a work

Conclusion and Future Work

Methodology

We have devised a system for the task of automatic musical form recognition and analysis using hybrid Neural Network-Decision Tree models

Intuition

This system completely analyses a piece of classical music, including locating the points of musical events, labeling them by their structural classification, and classifying the piece by its large form structure

Contribution

We presented a new dataset that seeks to correct the errors presented by previous commonly used databases, including pre-computed spectral data (for training) and the form classification for each piece

Extension

The final system is in a usable state for individual use, anthology development, or implementation into a more complex piece of software

Analytical Extensions

While the current system is specific to classical music analysis, it could be extended to allow for the **classification of additional forms** including those found in popular music and more complex hybrid forms

Optical Music Recognition

Optical Music Recognition is another difficult task lacking substantial research – our methods could be potentially extended to perform visual music analysis and perform the segmentation/classification on the score

Forensic Musicology & Copyright

The system may be extendable for use in **Forensic Musicology**, using the system's output analysis in the comparison of multiple pieces of music for potentially similar or exact replications of musical phrases