DEEP LEARNING FOR MUSICAL FORM: RECOGNITION AND ANALYSIS

Daniel Szelogowski

Master Thesis

MS Computer Science

April 2022

University of Wisconsin - Whitewater

Outline of Presentation

Committee

Dr. Lopamudra Mukherjee, *supervisor*

Dr. Hien Nguyen

Dr. Zachary Oster

Dr. Benjamin Whitcomb

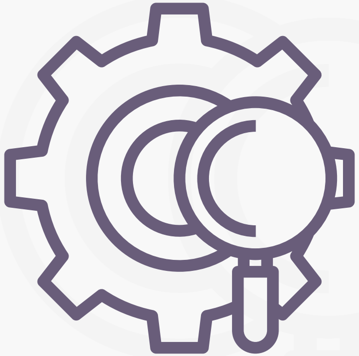
Introduction

- [Classical] **Musical form analysis** is a rigorous task that frequently challenges the expertise of human analysts and signal processing algorithms alike

- While numerous systems have been proposed to perform the tasks of **musical segmentation**, **genre classification**, and **single-label segment classification** in **popular music**, none have specifically focused on the analytical process used by classical musicians

- As well, **current datasets** used for related research tasks **lack standardized analytical conventions**, including **form classification**, and suffer from **erroneous annotations** and **extensibility** due to the data sources used for the music

- We propose a new system to perform the task of **automatic musical form analysis** using deep learning models, as well as a new **standardized dataset**



Motivation

X

Problem Definition

Several attempts to **autonomously analyze and segment musical form** using artificial intelligence algorithms have been made (including novelty methods, community detection algorithms, and neural networks), but **none have proven to be sufficient**

1

Limitation #1

Current systems focus on **popular music tasks**, such as **Verse-Chorus segmentation/classification** and **genre classification** – although attempts have been made to segment classical music by phrases without classifying

2

Limitation #2

The only existing datasets of phrase-analyzed classical music (SALAMI) feature numerous errors, **lack standardized analytical conventions** (to allow for genre flexibility), and uses live recordings rather than basing timestamps on the score

3

Limitation #3

No such tool exists for **automatic** (or **computer assisted**) classical form analysis – this time-consuming task must be done entirely by hand, and translating this to an AI-usable format requires a double analysis of **both the sheet music and a reliable audio file** for timestamping

Motivation – Potential Applications

1

Music Rehearsal and Pedagogy

Music practice and analysis tools, such as **dividing a piece by themes** for rehearsal, **assignment generation** using peak-picking, or a **grading system** for human-analyzed scores

2

Audio Thumbnails and Fingerprints

Audio thumbnail/fingerprint generation, as applicable for a **streaming service or web store** (iTunes, Facebook Music Sharing, Amazon music)

3

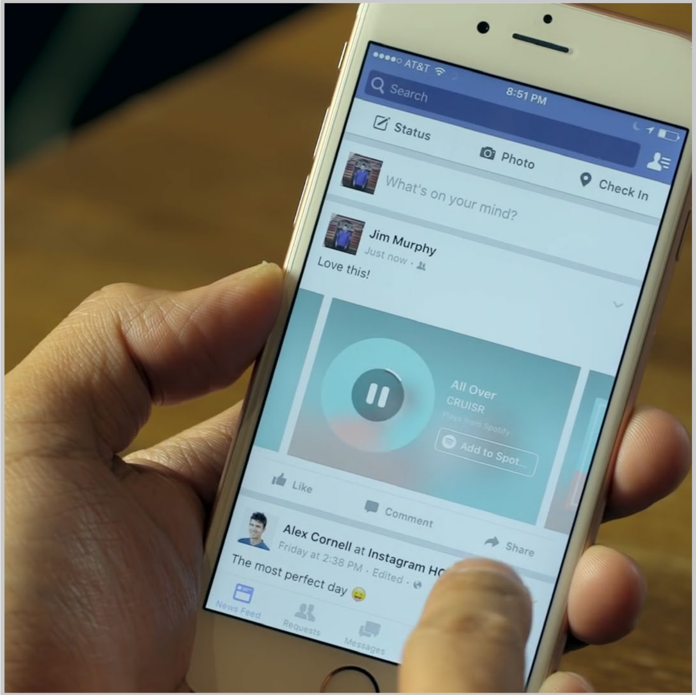
Forensics and Copyright Detection

Forensic Musicology, where the analysis may be used to **compare numerous pieces of music** for similar or exact replications of musical phrases, motives, or other structures

4

Audio-Video Analysis

Extension to video analysis to apply both **visual and audio cues** to the **media's structure**, whether formal in design (such as a music video, musical, etc.) or not (a movie/TV show)



Facebook's audio sharing feature for Spotify /iTunes[1] (Plays a 30 second song preview)

[1] Image Source: https://digiday.com/media/facebook-now-lets-users-share-music-listen-30-second-song-snippets/

Motivation – Potential Applications

5

Generalized Audio Structure Analysis

Performing structural analysis on any given audio or waveform, including **spoken audio** where patterns of repeated language may be used (i.e., poetry, forensic investigation, lecture, neuro-linguistic programming, **Natural Language Processing (NLP)** problems)

6

Optical Music Recognition and Analysis

Optical Music Recognition (OMR) – analyzing a piece using the sheet music, much like a human analyst, or correcting optical sheet music transcriptions using formal structures

7

Audio and Spoken Language Classification

Audio classification by content, with or without music (e.g., for **sorting a web database**, hearing-impaired **accessibility tools**, **language classification** from audio recordings [NLP])

8

Lacking Musicology Research

Production of **musical form/analysis-based anthologies**, alongside other fields of musicology that lack significant research and technological advancement

Anthology books are widely used in all fields of music[2] – though Form Analysis has very few

[2] Image Source: https://www.amazon.com/Faber-Music-Contemporary-Piano-Anthology/dp/0571541585

Goals and Objectives

Goal #1

Provide a new model to **perform full form analysis** (form classification, segmentation, part/phrase labeling), rather than simply peak-picking and segmentation, and **expand upon existing model architecture designs** using recurrent memory cells to better recognize repeated audio patterns

Goal #2

Develop **a new, musically accurate dataset** by common analytical conventions, including categorical divisions by large musical form, and provide appropriate **evaluation metrics** and **accurate model performance** results

Goal #3

Present a more accurate deep learning model to perform both **form-level** and **part/phrase-level analysis** using suitable algorithms for the network architecture and signal processing by extensive and exhaustive research

Goal #4

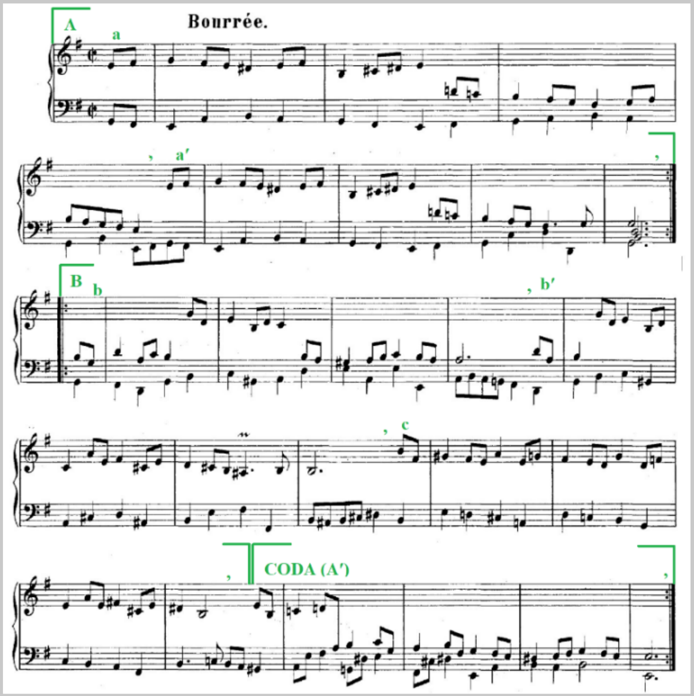
Examine the previous work done in the field through **extensive background research** and **contribute to the literature** by obtaining improved results from previous studies in the formal analysis of music using machine learning and peak picking methods

Background

- Classical music form analysis facilitates a combination of classification and segmentation tasks, including **form classification**, **structural segmentation**, and **multilabel large-** and **small-segment classification** – tasks that lack feasible algorithms, machine learning models, and extensive research

- Classical pieces are broken down hierarchically by their **large form** (two-part/binary, three-part/ternary, ...), **Part** divisions (part A, B, A', Development, CODA, …), and **Phrases** (a, a', b, c, theme, variation, transition, section, …) – though many additional labels may be employed, including for other sub-structures

- Form analysis has many applications in the world of music (especially for directors), and a viable analytical system would greatly benefit **performing musicians** and **academic researchers**, both in musicology and signal processing

- One of the greatest difficulties in musical analysis is the **lack of complete ground truth** – pieces of music are frequently interpreted differently by different analysts, and **may be classified or even analyzed differently** at the phrase or part levels

Analysis of Bourrée from J.S. Bach's BWV 996 (Rounded Binary form)

Literature Review

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 – Main Contributions

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

Methodology – System Components

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

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

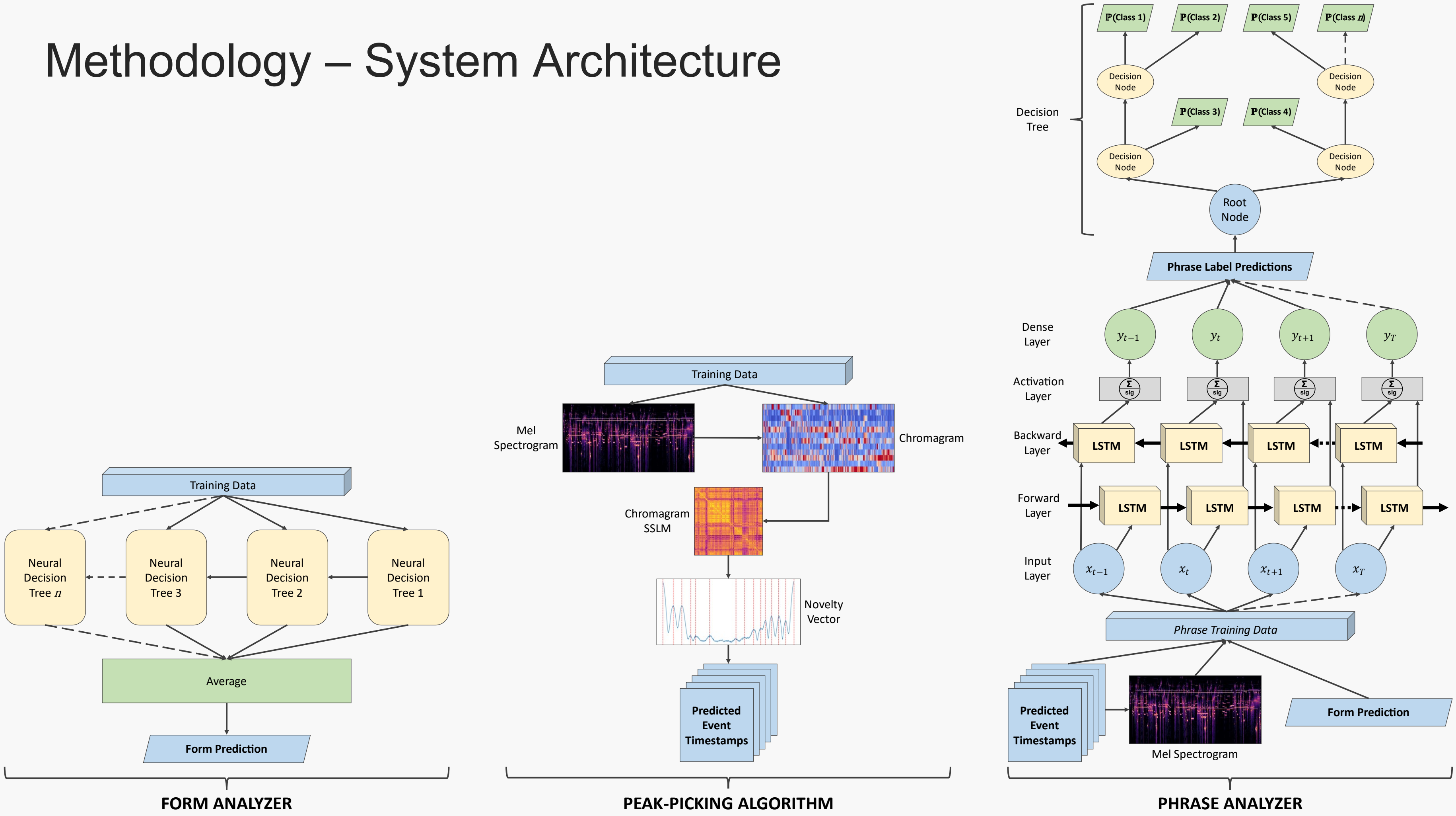
Phrase Analyzer

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

Prediction System

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

Methodology – System Architecture



Data Extraction and Preparation

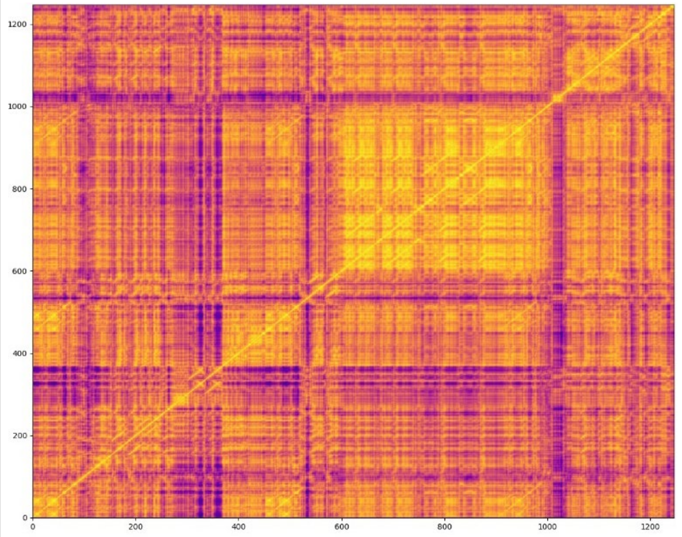
- Using feature selection and elimination methods, we found that the two most important data features for Form Classification were the **music duration** and the **Self-Similarity Matrix (SSM) of the (Mel) Spectrogram**

- 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

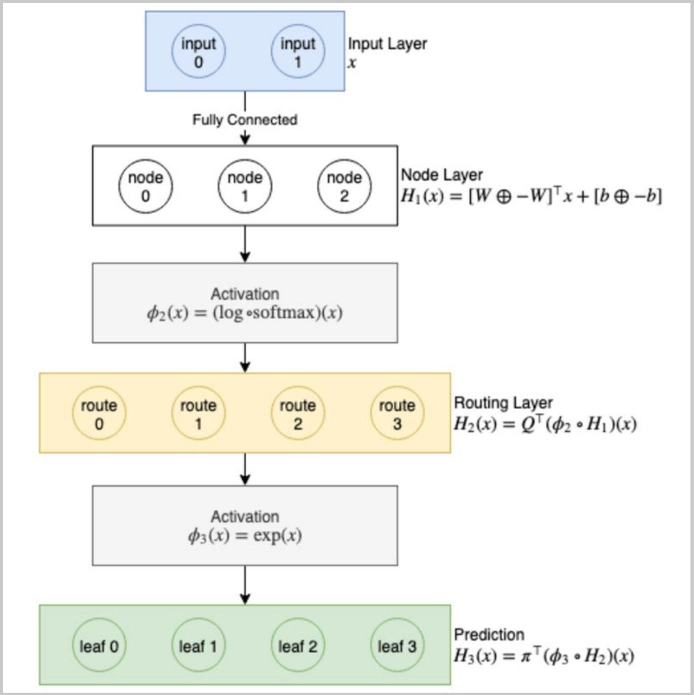
- The data for the Form Analyzer was scaled using $X = \frac{(X - mean(X))}{std(x)}$, and using **Min-Max Scaling** for the Phrase Analyzer ($X = \frac{X - \min(X)}{\max(X) - \min(X)}$).



Mel Spectrogram SSM

Form Analyzer Architecture – TreeGrad

- 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.)
- **Duration** was found to be the most important feature in **classifying the form**, a hint well-used by human analysts
- Labels were encoded using **Integer encoding**
- 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**



TreeGrad Architecture

Form Analyzer Architecture – TreeGrad

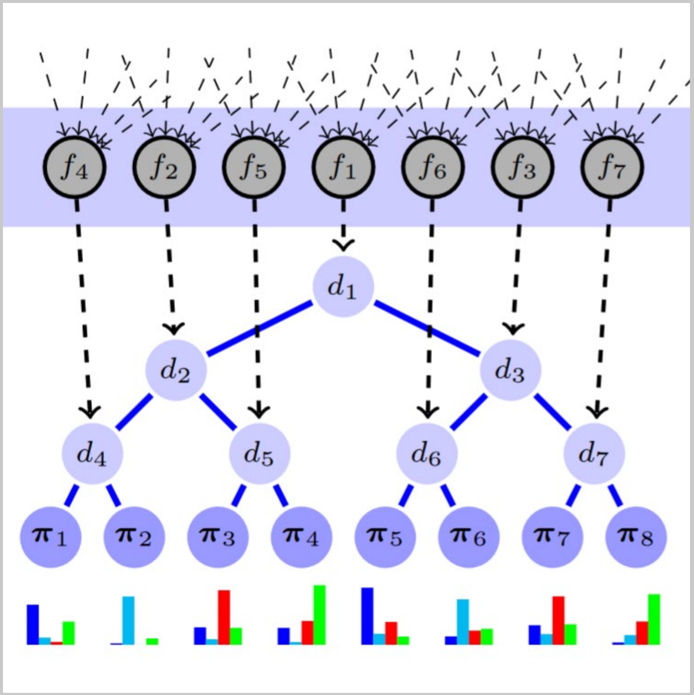
- Each tree in the TreeGrad forest uses probabilistic routing computed by the **sigmoid function**, replicating the probabilistic output of a **Dense Neural Network**

- Both the **Input** and **Dense** layers are trainable, allowing the weights of each layer to be bounded by the $\ell_p - norm$ with $p \geq 1$, conforming the model to the **AdaNet Generalization Bounds** (creating better comparisons between the complexities of models in an ensemble and the overall training loss)

- All trees in the forest are combined into a **Stacked** ensemble, where multiple well-performing models are combined, and the **average** of their output is returned as the final prediction

Form Analyzer Parameters

	# Of Leaves (Per Tree)	Max Depth	Learning Rate	# Of Estimators (Trees)	Batch Size	Refit Splits?
Value	31	-1 (∞)	0.1	100	32	True



Probabilistic routing and output of a Neural Decision Tree modeled as a CNN [92]

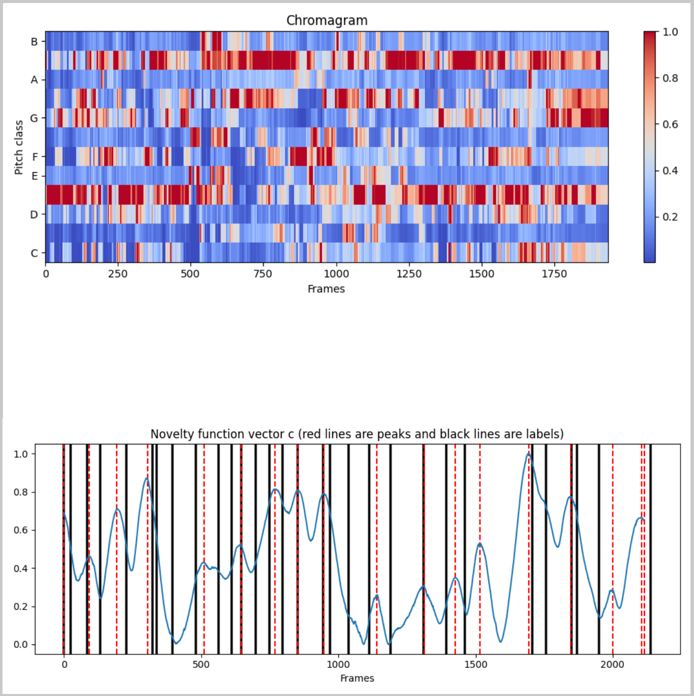
Peak-Picking Algorithm – Onset Detection

- 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 algorithm **does not employ machine learning techniques** to perform the peak-detection directly, it was **found to be comparable to other CNN architectures** discovered in the literature review and greatly reduced system design time (given the lack of training necessary to perform the calculations)



Example Chromagram and Novelty Vector

Phrase Analyzer Architecture – LSTM-Tree

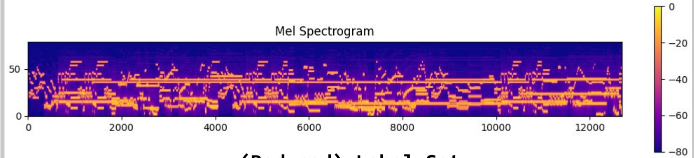
- Data for the Phrase Analyzer is divided into **a series of timestamps** which are used to slice the audio into segments between these points

- 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. Labels were encoded using **One-Hot Encoding**

- 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**)

Phrase Analyzer Parameters

	LSTM Units	LSTM Dropout	Merge Mode	Loss	Activation	Optimizer	Batch Size	Epochs
Value	4	0.2	Concat	Binary Cross-entropy	Sigmoid	Adam	1	5

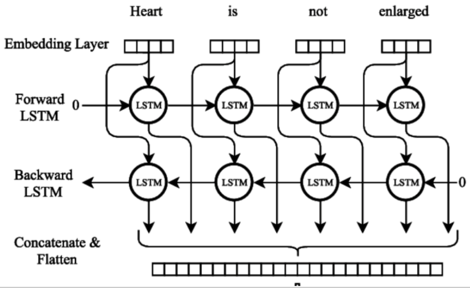


(Reduced) Label Set

Large Parts = { A, B, C, D, Theme, Variation, CODA, End, Silence }

Phrases = { a, b, c, d, e, f, transition, codetta, section (abbr. “sec”) }

Theme Variations = { characteristic, figural, melodic, ornamental, simplifying }



Example Mel Spectrogram, Label Set Universe, and Bi-LSTM Architecture

Evaluation – Experimental Results

- For the Form Analyzer, the TreeGrad model simply takes in the SSM and duration of a piece of music and outputs the **predicted classification**, which is compared against the **ground truth label**

- For a musicologist, this system alone could be used to discover **when and why** a composer may choose a particular form over another, for example

- 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

A person performing music analysis using the sheet music for the piece[3]

[3] Image Source: https://makingmusicmag.com/music-analysis-essay/

Evaluation – Form Analyzer

- The full (augmented) dataset was split into **83.1% training** and **16.9% 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**
- The output provided by the prediction is **the large form classification** expected by the model, which was found to be 83% accurate – a surprisingly good performance given the **subjective nature** of the form classification

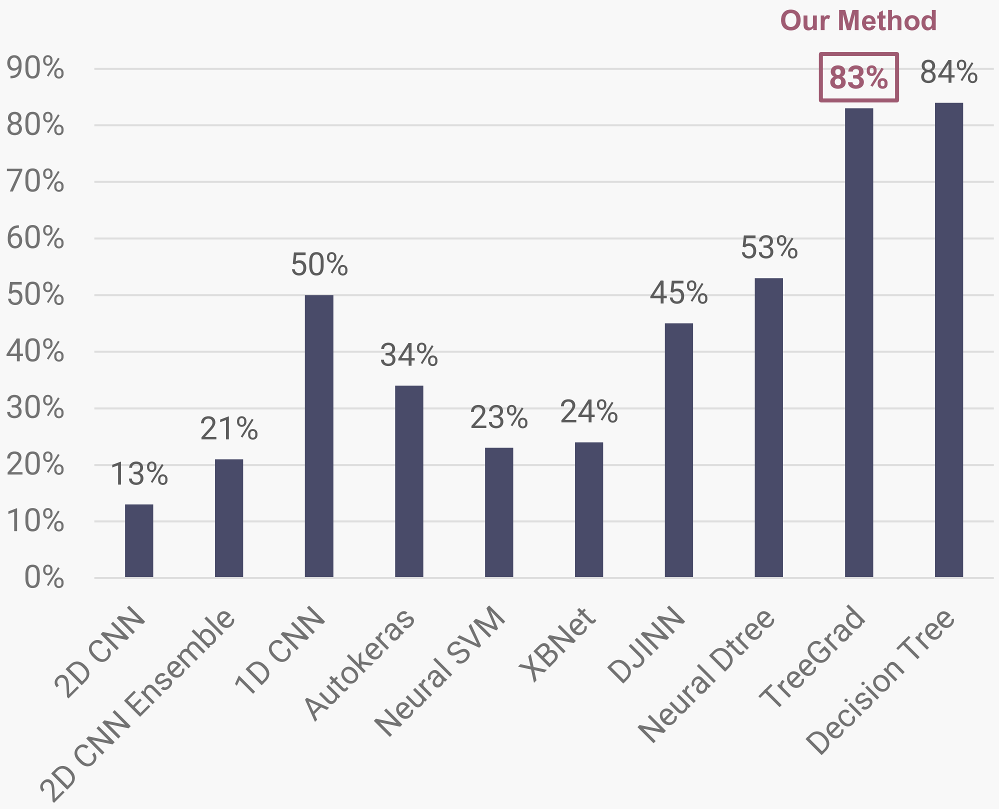
Result #1

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**

```
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

Results – Form Analyzer Compared to Other Methods



Form Analyzer

83%

Validation Accuracy
(Jaccard Score)

Form Analyzer

84%

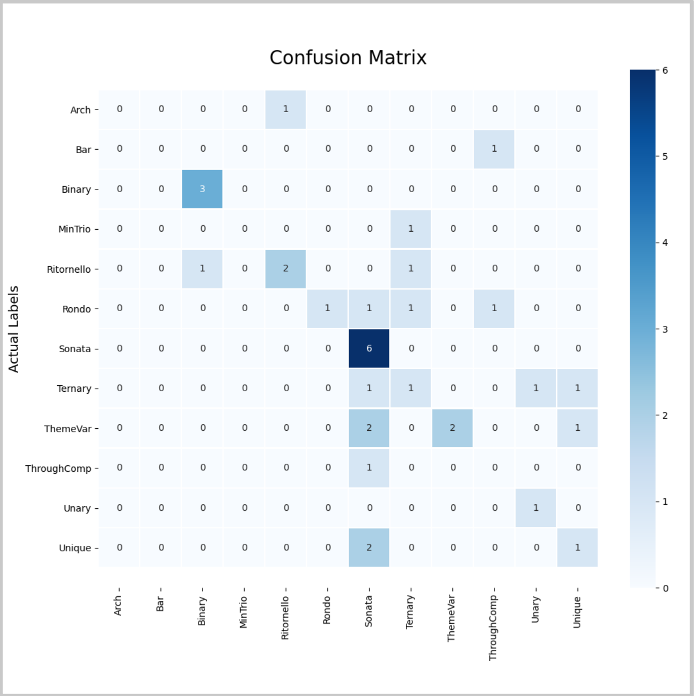
Precision

Form Analyzer

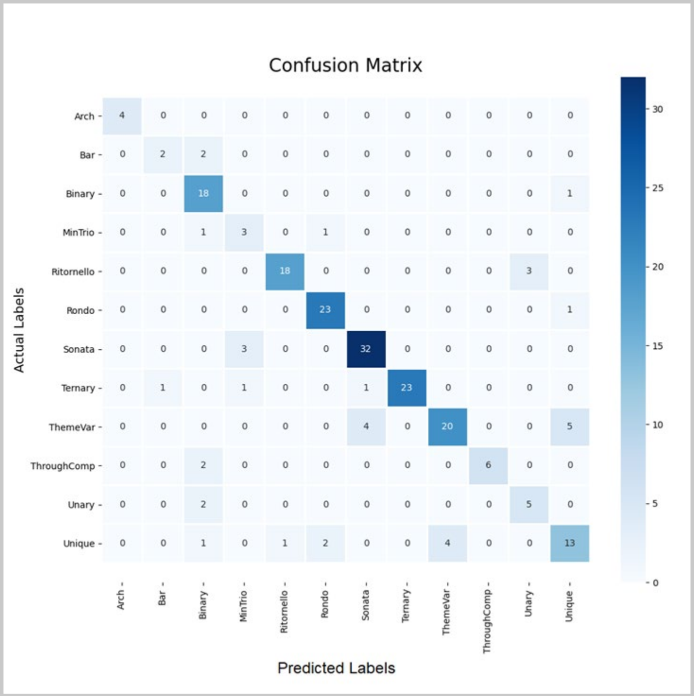
82%

Recall & F1

Results – Form Analyzer Compared to Other Methods



Best CNN Confusion Matrix
(50% Accuracy, 56% Precision, 44% Recall, 41% F1)



TreeGrad Confusion matrix
(83% Accuracy, 84% Precision, 82% Recall & F1)

Evaluation – Peak-Picking Algorithm

- The authors of the full algorithm propose further modifications for segment-segment comparisons [80], but these **greatly increase the number of matrix computations** and add substantial overhead

- Results were found to be **identical** for almost all pieces of music regardless of the proposed additions — since the timestamps are independent of the classification analysis, a segment-segment comparison is unnecessary for onset detection

- The output of the novelty function was compared to **numerous hand-labeled pieces** from the dataset, and we found that the **difference was negligible**

```
Event: 0:00:00          Ground Truth: 0:00:00          Difference:  0.000000
Event: 0:00:13.235374   Ground Truth: 0:00:01.150000   Difference: -12.085374
Event: 0:00:28.560544   Ground Truth: 0:00:23.995000   Difference: -4.565544
Event: 0:00:40.124082   Ground Truth: 0:00:29.945000   Difference: -10.179082
Event: 0:01:00.186122   Ground Truth: 0:00:37.545000   Difference: -22.641122
Event: 0:01:18.715646   Ground Truth: 0:00:48.645000   Difference: -30.070646
Event: 0:01:42.678639   Ground Truth: 0:00:56.495000   Difference: -46.183639
Event: 0:01:54.102857   Ground Truth: 0:01:40.195000   Difference: -13.907857
Event: 0:02:14.443537   Ground Truth: 0:01:59.161000   Difference: -15.282537
Event: 0:02:27.678912   Ground Truth: 0:02:22.011000   Difference: -5.667912
Event: 0:02:44.397279   Ground Truth: 0:02:52.811000   Difference:  8.413721
Event: 0:02:56.100136   Ground Truth: 0:03:19.611000   Difference:  23.510864
Event: 0:03:09.196190   Ground Truth: 0:03:32.411000   Difference:  23.214810
Event: 0:03:22.292245   Ground Truth: 0:03:51.761000   Difference:  29.468755
Event: 0:03:36.642177   Ground Truth: 0:04:10.761000   Difference:  34.118823
Event: 0:03:59.072653   Ground Truth: 0:04:24.211000   Difference:  25.138347
Event: 0:04:15.233741   Ground Truth: 0:04:30.761000   Difference:  15.527259
Event: 0:04:31.255510   Ground Truth: 0:04:38.761000   Difference:  7.505490
Event: 0:04:41.983129   Ground Truth: 0:04:47.761000   Difference:  5.777871
Event: 0:05:01.209252   Ground Truth: 0:04:59.911000   Difference: -1.298252
Event: 0:05:19.878095   Ground Truth: 0:05:06.511000   Difference: -13.367095
Event: 0:05:34.228027   Ground Truth: 0:05:31.761000   Difference: -2.467027
Event: 0:05:48.438639   Ground Truth: 0:05:43.861000   Difference: -4.577639
Event: 0:06:08.918639   Ground Truth: 0:06:08.001000   Difference: -0.917639
Average (absolute) time difference: ±14.828637755102045
```

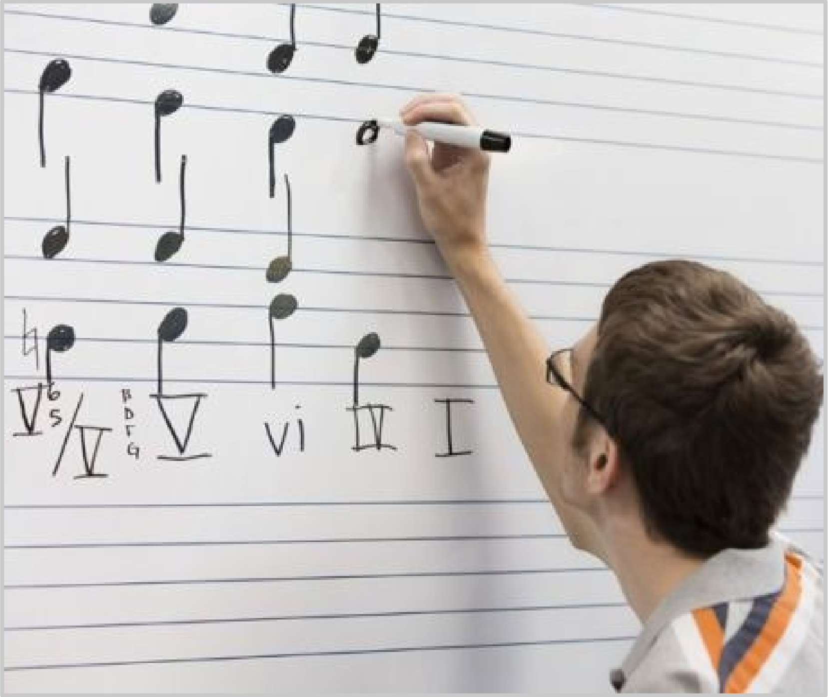
Demonstration of peak-picking algorithm compared to ground truth annotation

Results – Peak-Picking Algorithm

- 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

- One issue the algorithm faces is that some pieces of music that are exceptionally short in duration **only return the onset of the start and end** of the piece – for such examples, the halfway point (*duration* / 2) is added to the novelty vector

Result #2

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

A demonstration of harmonic analysis, part of the intuition behind analyzing musical phrase segmentation[4]

[4] Image Source: https://www.pycosom.com/music-theory.html

Evaluation – Phrase Analyzer

Associated Challenges

- This model is **much more difficult to score programmatically**, as numerous factors affect the final system

- The model receives the timestamps from the Peak-Picking algorithm, which is also difficult to compare to a human annotated ground truth due to **integrative disagreement**

- 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)

Phrase Analyzer

100%

Accuracy
(Hamming Score)

Formal Plan of Vivaldi's "Winter," Op. 8, No. 4, Second Movement

Sections	[A]				[B]				
Measure #'s	1-2	3-5	5-7	7-8	9-10	11	12-13	14-16	17-18
Phrases	a	b	ext.	---------	a′	ext.	b′	ext.	---------
Harmonic Motion (Phrase Level)	I-V-I	V^6 ________	$^5_3$		=V	V^7 I	IV V	V	I
		=	I	V-I	I-V-I				
Keys	E♭		B♭			E♭			
Large-Scale Harmonic Plan	I		V			I		V	I

A full analysis example, displaying the carryover of the Part labels and the repetition of Phrase labels[5]

[5] Figure Source: https://apcentral.collegeboard.org/courses/ap-music-theory/classroom-resources/teaching-score-analysis-ap-music-theory

Evaluation – Phrase Analyzer

Associated Challenges

- Form analysis, especially onset detection, operates on **fuzzy logic rules**, so the peak-picking algorithm and phrase analyzer may or may not be considered accurate based on the bias of a human analyst and/or their conventions
- 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**
- The dataset is too small to split well into training and testing proportions – however, expanding the dataset would require many analysts all trained on the same methodology **analyzing each piece individually** before reaching an agreement for the ground truth labels (two analysts may even label varying numbers of events)

```
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
```

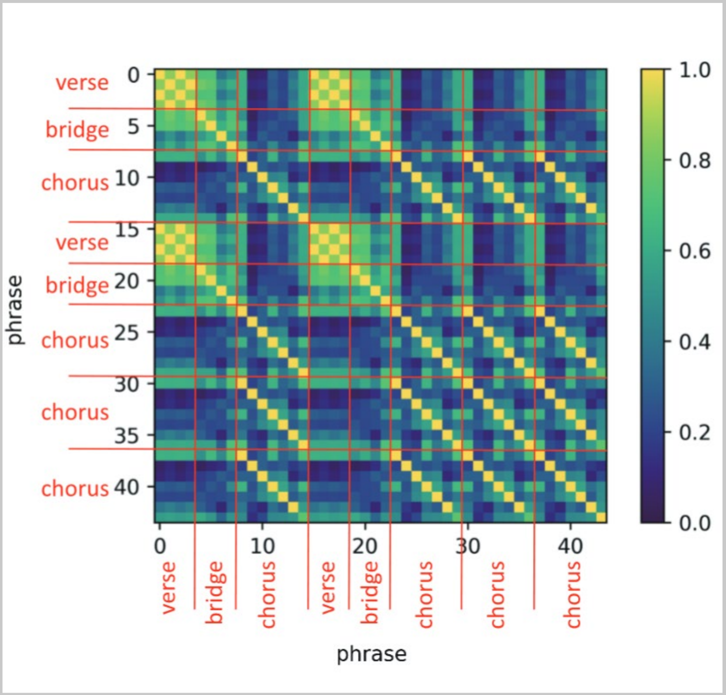
Final prediction system output includes Decision Tree and TreeGrad for comparison

Results – Phrase Analyzer

- 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

- Given the evaluation constraints for the phrase analysis, we found that the **LSTM-Tree** was sufficient to avoid harsh overfitting compared to other attempted models (DNN, RNN, TreeGrad), likely the result of the **boosting** from the decision tree



A Self-Similarity Matrix of a popular-form piece of music analyzed using an extended Word2Vec model[6]

[6] Image Source: http://dx.doi.org/10.2197/ipsjjip.27.278

Results – Phrase Analyzer

- 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

Result #3

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

Result #4

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

Discussion

Discussion #1

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

Discussion #2

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

Discussion #3

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

Discussion #4

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

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

Conclusion

In this thesis, we proposed a new system to perform the task of automatic musical form analysis using deep learning models, as well as a new standardized dataset

Future Work

Suggestion #1

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

Suggestion #2

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

Suggestion #3

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

References

1. Jacopo de Berardinis, Michalis Vamvakaris, Angelo Cangelosi, and Eduardo Coutinho. 2020. Unveiling the hierarchical structure of music by multi-resolution community detection. *Transactions of the International Society for Music Information Retrieval* 3, 1: 82–97. https://doi.org/10.5334/tismir.41
2. Douglass Marshall Green. 1979. *Form in tonal music: An introduction to analysis*. Cengage Learning.
3. Thomas Grill and Jan Schlüter. 2015. *Structural segmentation with convolutional neural networks MIREX submission*.
4. Dominik Hörnel and Wolfram Menzel. 1998. Learning musical structure and style with neural networks. *Computer Music Journal* 22, 4: 44–62. https://doi.org/10.2307/3680893
5. Tim O'Brien. 2016. Musical Structure Segmentation with Convolutional Neural Networks. *17th International Society for Music Information Retrieval Conference*.
6. Pedro Ponce de León and José Iñesta. 2007. Pattern recognition approach for music style identification using shallow statistical descriptors. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37, 2: 248–257. https://doi.org/10.1109/tsmcc.2006.876045
7. Karen Ullrich, Jan Schlüter, and Thomas Grill. 2014. Boundary Detection in Music Structure Analysis using Convolutional Neural Networks. *15th International Society for Music Information Retrieval Conference*.
8. Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. 2019. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access* 7: 19143–19165. http://dx.doi.org/10.1109/ACCESS.2019.2896880
9. Jordan Smith. 2019. "SALAMI Annotator's Guide." *GitHub*. Retrieved December 21, 2021 from https://github.com/DDMAL/salami-data-public/blob/master/SALAMI%20Annotator%20Guide.pdf.
10. McGill DDMAL. 2009. A Structural Analysis of Large Amounts of Music Information (SALAMI). *DDMAL*. Retrieved from https://ddmal.music.mcgill.ca/research/SALAMI/.
11. Leon Stein. 1979. *Structure & Style: The Study and Analysis of Musical Forms, Expanded Edition*. Summy-Birchard Music.

References

12. Vansh Sethi. 2019. Types of Neural Networks (and what each one does!) Explained. *Towards Data Science*. Retrieved December 22, 2021 from https://towardsdatascience.com/types-of-neural-network-and-what-each-one-does-explained-d9b4c0ed63a1

13. DeepAI. 2019. Feed Forward Neural Network. *DeepAI*. Retrieved December 22, 2021 from https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network

14. Naveen Joshi. 2019. 3 types of neural networks that AI uses. *Allerin*. Retrieved from https://www.allerin.com/blog/3-types-of-neural-networks-that-ai-uses

15. Bossy Mostafa, Noha El-Attar, Samy Abd-Elhafeez, and Wael Awad. 2020. Machine and Deep Learning Approaches in Genome: Review Article. Alfarama Journal of Basic & Applied Sciences 0, 0 (August 2020), 0–0. DOI:https://doi.org/10.21608/ajbas.2020.34160.1023

16. Quantum Neural Network — PennyLane. *PennyLane*. Retrieved December 23, 2021 from https://pennylane.ai/qml/glossary/quantum_neural_network.html

17. Mike Wang. 2021. Deep Q-Learning Tutorial: minDQN. *Towards Data Science*. Retrieved December 23, 2021 from https://towardsdatascience.com/deep-q-learning-tutorial-mindqn-2a4c855abffc

18. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. arXiv:312.5602v1

19. Daniel Szelogowski. 2020. Generative Deep Learning for Virtuosic Classical Music: Generative Adversarial Networks as Renowned Composers. arXiv:2101.00169

20. DeepAI. 2019. Recurrent Neural Network. *DeepAI*. Retrieved December 23, 2021 from https://deepai.org/machine-learning-glossary-and-terms/recurrent-neural-network

21. Christopher Olah. 2015. Understanding LSTM Networks. *Colah's Blog*. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

22. Daniel Szelogowski. 2021. Emotion Recognition of the Singing Voice: Toward a Real-Time Analysis Tool for Singers. arXiv:2105.00173

23. Aravindpai Pai. 2020. ANN vs CNN vs RNN. *Analytics Vidhya*. Retrieved from https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

24. DeepAI. 2019. Machine Learning. *DeepAI*. Retrieved December 23, 2021 from https://deepai.org/machine-learning-glossary-and-terms/machine-learning

25. IBM Cloud Education. 2020. What is Machine Learning? *IBM*. Retrieved from https://www.ibm.com/cloud/learn/machine-learning

References

26. Kristin Burnham. 2020. Artificial Intelligence vs. Machine Learning: What's the Difference? *Northeastern University Graduate Programs*. Retrieved December 23, 2021 from https://www.northeastern.edu/graduate/blog/artificial-intelligence-vs-machine-learning-whats-the-difference/

27. Sunil Ray. 2017. Commonly Used Machine Learning Algorithms. *Analytics Vidhya*. Retrieved December 23, 2021 from https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

28. DeepAI. 2019. Deep Learning. *DeepAI*. Retrieved December 23, 2021 from https://deepai.org/machine-learning-glossary-and-terms/deep-learning

29. DeepAI. 2019. Neural Network. *DeepAI*. Retrieved December 23, 2021 from https://deepai.org/machine-learning-glossary-and-terms/neural-network

30. Baeldung. 2020. Epoch in Neural Networks. Baeldung on Computer Science. Retrieved from https://www.baeldung.com/cs/epoch-neural-networks

31. MIR. Why_mir. Music Information Retrieval. Retrieved from https://musicinformationretrieval.com/why_mir.html

32. M. Sai Chaitanya and Soubhik Chakraborty. 2021. *Musical information retrieval. Signal Analysis and Feature Extraction using Python*. GRIN Verlag.

33. Timothy C. Justus and Jamshed J. Bharucha. 2002. Music Perception and Cognition. In *Stevens' Handbook of Experimental Psychology, Third Edition, Volume 1: Sensation and Perception*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 453-492. DOI:http://dx.doi.org/10.1002/0471214426.pas0111

34. Sumit Saha. 2018. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. *Towards Data Science*. Retrieved December 24, 2021 from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

35. The University of Kansas. 2014. Form Analysis. *KU School of Music*. Retrieved from https://music.ku.edu/form-analysis

36. Stony Brook. MUS Degrees & Requirements. *Stony Brook Undergraduate Bulletin*. Retrieved from https://www.stonybrook.edu/sb/bulletin/current/academicprograms/mus/degreesandrequirements.php

37. Sayantini. 2019. What is Fuzzy Logic in AI and What are its Applications? *Edureka*. Retrieved December 25, 2021 from https://www.edureka.co/blog/fuzzy-logic-ai/

38. Benjamin Whitcomb. 2021. *Form Terms*.

References

39. Jeremy Burns. 2018. 53-Form and Analysis Pt.2: Small Forms. *Music Student 101*. Retrieved December 26, 2021 from https://musicstudent101.com/53-form-and-analysis-pt.2-small-forms.html

40. Bryan Townsend. 2013. Phrase, Motif and Theme. *The Music Salon*. Retrieved December 26, 2021 from https://themusicsalon.blogspot.com/2013/01/phrase-motif-and-theme.html

41. Stefan Kostka, Dorothy Payne, and Byron Almén. 2017. Tonal Harmony (8th ed.). McGraw-Hill Education.

42. Ritesh Singh. 2020. Understanding neural networks through visualization. *Druva*. Retrieved December 26, 2021 from https://www.druva.com/blog/understanding-neural-networks-through-visualization/

43. Olalekan Ogunmolu, Xuejun Gu, Steve Jiang, and Nicholas Gans. 2016. Nonlinear Systems Identification Using Deep Dynamic Neural Networks. (October 2016).

44. Ashutosh Tripathi. 2021. What is the main difference between RNN and LSTM. *Data Science Duniya*. Retrieved December 27, 2021 from https://ashutoshtripathi.com/2021/07/02/what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/

45. Michael Tilmouth. 2001. *Strophic*. Oxford University Press. Retrieved December 27, 2021 from http://dx.doi.org/10.1093/gmo/9781561592630.article.26981

46. Samuel Chase. 2021. What Is Through Composed Form in Music? *Hello Music Theory: Learn Music Theory Online*. Retrieved December 28, 2021 from https://hellomusictheory.com/learn/through-composed-form/

47. Robert Hutchinson. 2021. Sonata Form. *Music Theory for the 21st-Century Classroom*. Retrieved December 29, 2021 from https://musictheory.pugetsound.edu/mt21c/SonataIntroduction.html

48. Steven Ashby. Sonata-Allegro Form. *Music Appreciation (MHIS 243)*. Retrieved December 29, 2021 from https://rampages.us/mhis243/lectures/lesson-8/sonata-allegro-form/

49. Anuj Sable. 2021. Introduction to Audio Analysis and Processing. *Paperspace Blog*. Retrieved from https://blog.paperspace.com/introduction-to-audio-analysis-and-synthesis/

50. Meinard Müller. 2021. *Fundamentals of Music Processing: Using Python and Jupyter Notebooks*. Springer Nature.

51. Meinard Müller. 2015. C2_STFT-Basic. *International Audio Laboratories Erlangen*. Retrieved December 29, 2021 from https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html

52. Slim Essid. 2018. Audio Data Analysis. *Telecom ParisTech*. Retrieved from https://perso.telecom-paristech.fr/essid/ces_ds/audio-analysis-lecture_2018.pdf

References

53. Desh Raj. 2019. A note on MFCCs and delta features. *Desh2608 Github*. Retrieved December 30, 2021 from https://desh2608.github.io/2019-07-26-delta-feats/

54. Dishashree Gupta. 2020. Activation Functions. *Analytics Vidhya*. Retrieved January 22, 2022 from https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/

55. 2021. What Is a Decision Tree? *Master's in Data Science*. Retrieved April 1, 2022 from https://www.mastersindatascience.org/learning/introduction-to-machine-learning-algorithms/decision-tree/

56. Sruthi E R. 2021. Random Forest. *Analytics Vidhya*. Retrieved April 1, 2022 from https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

57. 2018. What is TensorFlow? *Databricks*. Retrieved April 2, 2022 from https://databricks.com/glossary/what-is-tensorflow

58. Codecademy. What is Scikit-Learn? *Codecademy*. Retrieved April 2, 2022 from https://www.codecademy.com/article/scikit-learn

59. Introduction to NumPy. *W3Schools*. Retrieved April 2, 2022 from https://www.w3schools.com/python/numpy/numpy_intro.asp

60. Mode Resources. 2016. Pandas. *Mode Resources*. Retrieved April 2, 2022 from https://mode.com/python-tutorial/libraries/pandas/

61. Librosa. *Librosa — Librosa 0.9.1 Documentation*. Librosa. Retrieved April 2, 2022 from https://librosa.org/doc/latest/index.html

62. James Robert. jiaaro/pydub @ GitHub. *Pydub*. Retrieved April 2, 2022 from http://pydub.com/

63. 2017. Understanding Python Pickling with example. *GeeksforGeeks*. Retrieved April 2, 2022 from https://www.geeksforgeeks.org/understanding-python-pickling-example/

64. Scikit-Learn. 1.10. Multiclass and multilabel algorithms — scikit-learn 0.15-git documentation. *Scikit-Learn*. Retrieved April 2, 2022 from https://scikit-learn.org/0.15/modules/multiclass.html

65. Saugata Paul. 2019. A detailed case study on Multi-Label Classification with Machine Learning algorithms and predicting movie tags based on plot summaries! *Medium*. Retrieved April 2, 2022 from https://medium.com/@saugata.paul1010/a-detailed-case-study-on-multi-label-classification-with-machine-learning-algorithms-and-72031742c9aa

66. Daniel Szelogowski. 2021. GitHub - DanielAtHome19/Form-NN: In progress master thesis project. *GitHub*. Retrieved April 2, 2022 from https://github.com/danielathome19/Form-NN

67. Lorraine Li. 2019. Principal Component Analysis for Dimensionality Reduction. *Towards Data Science*. Retrieved April 2, 2022 from https://towardsdatascience.com/principal-component-analysis-for-dimensionality-reduction-115a3d157bad

68. Stefan Hrouda-Rasmussen. 2021. The Curse of Dimensionality. *Towards Data Science*. Retrieved April 2, 2022 from https://towardsdatascience.com/the-curse-of-dimensionality-5673118fe6d2

69. Jason Brownlee. 2019. A Gentle Introduction to Imbalanced Classification. *Machine Learning Mastery*. Retrieved April 2, 2022 from https://machinelearningmastery.com/what-is-imbalanced-classification/

References

70. Resampy. 2016. Core functionality — resampy 0.1.5 documentation. *Resampy Documentation*. Retrieved April 2, 2022 from https://resampy.readthedocs.io/en/0.1.5/api.html

71. Brian McFee. 2018. resampy/interpn.py at master · bmcfee/resampy. *GitHub*. Retrieved April 2, 2022 from https://github.com/bmcfee/resampy/blob/master/resampy/interpn.py

72. Weisstein. Gamma Function -- from Wolfram MathWorld. *Wolfram MathWorld*. Retrieved April 2, 2022 from https://mathworld.wolfram.com/GammaFunction.html

73. Théo Royer. 2019. Pitch-shifting algorithm design and applications in music. Thesis. KTH Royal Institute of Technology - School of Electrical Engineering and Computer Science. Retrieved from http://kth.diva-portal.org/smash/get/diva2:1381398/FULLTEXT01.pdf

74. Meinard Müller. 2015. C2_STFT-Inverse. *International Audio Laboratories Erlangen*. Retrieved April 2, 2022 from https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Inverse.html

75. MATLAB. Inverse short-time Fourier transform. MATLAB istft. Retrieved April 2, 2022 from https://www.mathworks.com/help/signal/ref/istft.html

76. William Sethares. 2007. phase vocoder in Matlab. *University of Wisconsin*. Retrieved April 2, 2022 from https://sethares.engr.wisc.edu/vocoders/phasevocoder.html

77. Chris Tralie. 2014. Musical Pitches and Chroma Features. *ECE 381 Data Expeditions Lab 1*. Retrieved April 3, 2022 from https://www.ctralie.com/Teaching/ECE381_DataExpeditions_Lab1/

78. Carlos Hernandez-Olivan, Jose R. Beltran, and David Diaz-Guerra. 2020. Music Boundary Detection using Convolutional Neural Networks: A comparative analysis of combined input features. arXiv:2008.07527

79. Meinard Müller. 2015. C2_STFT-Inverse. *International Audio Laboratories Erlangen*. Retrieved April 2, 2022 from https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Inverse.html

80. Joan Serrà, Meinard Müller, Peter Grosche, and Josep Arcos. 2014. Unsupervised Music Structure Annotation by Time Series Structure Features and Segment Similarity. IEEE Transactions on Multimedia 16, 5 (August 2014), 1229–1240. DOI:https://doi.org/10.1109/TMM.2014.2310701

81. Librosa. 2021. librosa.segment — librosa 0.8.1 documentation. *Librosa Documentation*. Retrieved April 3, 2022 from https://librosa.org/doc/0.8.1/_modules/librosa/segment.html

82. Christopher De Sa. 2018. Lecture 2: k-nearest neighbors / Curse of Dimensionality. *CS 4/5780 Introduction to Machine Learning*. Retrieved April 3, 2022 from https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote02_kNN.html

83. Scikit-Learn. sklearn.feature_selection.SelectKBest. *scikit-learn*. Retrieved April 3, 2022 from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest

84. Scikit-Learn. 1.13. Feature selection. scikit-learn. Retrieved April 3, 2022 from https://scikit-learn.org/stable/modules/feature_selection.html

85. Jason Brownlee. 2017. What is the Difference Between a Parameter and a Hyperparameter? Machine Learning Mastery. Retrieved April 4, 2022 from https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/

86. Tushar Sarkar. 2021. XBNet : An Extremely Boosted Neural Network. arXiv.org. arXiv:2106.05239

87. Cheshta Dhingra. 2020. A Visual Guide to Gradient Boosted Trees (XGBoost). Towards Data Science. Retrieved April 4, 2022 from https://towardsdatascience.com/a-visual-guide-to-gradient-boosted-trees-8d9ed578b33

References

88. K. D. Humbird, J. L. Peterson, and R. G. McClarren. 2017. Deep neural network initialization with decision trees. arXiv.org. arXiv:1707.00784

89. Prashant Gupta. 2017. Regularization in Machine Learning. *Towards Data Science*. Retrieved April 4, 2022 from https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a

90. Amar Budhiraja. 2018. Dropout in (Deep) Machine learning - Amar Budhiraja. *Medium*. Retrieved April 4, 2022 from https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

91. Chapman Siu. 2019. Transferring Tree Ensembles to Neural Networks. In *Neural Information Processing*. Springer International Publishing, Cham, 471–480. Retrieved April 5, 2022 from http://dx.doi.org/10.1007/978-3-030-36711-4_39

92. Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. 2015. Deep Neural Decision Forests. In *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE. Retrieved April 5, 2022 from http://dx.doi.org/10.1109/iccv.2015.172

93. Scikit-Learn. 1.12. Multiclass and multioutput algorithms. *scikit-learn*. Retrieved April 5, 2022 from https://scikit-learn.org/stable/modules/multiclass.html

94. Papers With Code. Papers with Code - BiLSTM Explained. *Papers With Code*. Retrieved April 5, 2022 from https://paperswithcode.com/method/bilstm

95. Raghav Aggarwal. 2019. Bi-LSTM - Raghav Aggarwal. *Medium*. Retrieved April 5, 2022 from https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0

96. Michael Phi. 2020. Illustrated Guide to LSTM’s and GRU’s: A step by step explanation. *Towards Data Science*. Retrieved April 5, 2022 from https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

97. Keras Team. Keras documentation: TimeDistributed layer. *Keras*. Retrieved April 5, 2022 from https://keras.io/api/layers/recurrent_layers/time_distributed/

98. Jason Brownlee. 2017. Why One-Hot Encode Data in Machine Learning? *Machine Learning Mastery*. Retrieved April 5, 2022 from https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

99. Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-Validation. SpringerLink. Retrieved April 5, 2022 from https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_565

100. Koo Ping Shung. 2020. Accuracy, Precision, Recall or F1? *Towards Data Science*. Retrieved April 5, 2022 from https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

101. MMA. 2020. Metrics for Multilabel Classification. *Mustafa Murat ARAT*. Retrieved April 5, 2022 from https://mmuratarat.github.io/2020-01-25/multilabel_classification_metrics

102. Johnson and Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data* 6, 1 (March 2019), 1–54. DOI:https://doi.org/10.1186/s40537-019-0192-5

103. Daniel Godoy. 2019. Understanding binary cross-entropy / log loss: a visual explanation. *Towards Data Science*. Retrieved April 5, from https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

104. GeeksforGeeks. 2020. Intuition of Adam Optimizer. GeeksforGeeks. Retrieved April 6, 2022 from https://www.geeksforgeeks.org/intuition-of-adam-optimizer/

105. Xin Wang, Yudong Chen, and Wenwu Zhu. 2020. A Survey on Curriculum Learning. arXiv:2010.13166

QUESTIONS

Thank you!

DEEP LEARNING FOR MUSICAL FORM: RECOGNITION AND ANALYSIS

Daniel Szelogowski

Master Thesis
MS Computer Science

April 2022
University of Wisconsin - Whitewater