**Bach or Not, Classical Composer Classification**

University of San Diego

AAI-511-03 Neural Networks and Deep Learning

Carrie Little, Alexis Lim, Ahmad Milad

Dr. Mirsardar Esmaeili

08/11/2025

# Bach or Not, Classical Composer Classification

In this project, we explored how deep learning can be used to identify the composer of a piece of classical music. We focused on four well-known composers being Bach, Beethoven, Chopin, and Mozart, and trained two different models: a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network. Both models used note sequences extracted from MIDI files, and we compared how well each one performed in recognizing the unique musical patterns associated with each composer. In the end, the CNN model came out on top with higher accuracy and more balanced results across all four composers.

## Introduction

Every composer has a distinct musical style. Recognizing these styles programmatically is a tough challenge if not clearly defined. In this project, we wanted to see if deep learning could learn those differences directly from symbolic music data, specifically MIDI files, and use that knowledge to predict who wrote a given piece.

We built and compared two models: a CNN that looks for short repeating patterns (motifs), and an LSTM that tries to learn how those patterns unfold over time. Both models were trained on the same dataset and tested on how well they could classify pieces by composer.

## Dataset Description

The dataset came from Kaggle’s “midi-classic-music” collection, which includes hundreds of classical pieces. We narrowed it down to just the four composers required for the assignment: Bach, Beethoven, Chopin, and Mozart. Each file was parsed using pretty\_midi to extract the sequence of pitches, which served as the core input for our models.

## Methodology

### Data Preprocessing

To make the music readable by a deep learning model:

* We converted each piece into a list of note pitches.
* These sequences were mapped to integers and padded to the same length (we used a fixed max length of 500 notes to avoid memory issues).
* Composer labels were encoded into numbers.
* We split the data into training and test sets using stratified sampling to make sure each composer was fairly represented.
* We didn’t include tempo, duration, or other stats in the model input, but we did use them to explore the dataset during feature exploration.

**CNN Architecture**

The CNN model was designed to detect small-scale note patterns that might appear repeatedly in different pieces. It included:

* Two convolutional layers (with 64 and 128 filters)
* Max pooling and global max pooling layers to reduce dimensionality
* A dense (fully connected) layer
* Dropout for regularization
* A softmax output layer to classify the composer

### LSTM Architecture

### The LSTM model was built to focus on the sequence and flow of notes over time:

### An embedding layer to learn a dense vector for each pitch

### A 128-unit LSTM layer to learn long-range dependencies

### A dense layer followed by dropout

### A softmax layer for the final prediction

### We used the Adam optimizer, applied early stopping during training, and reduced the learning rate if validation loss stopped improving.

### Results & Evaluation

#### CNN Performance

#### Accuracy: 80%

#### Macro F1-score: 0.69

#### Bach F1: 0.9

#### Beethoven F1: 0.64

#### Chopin F1: 0.67

#### Mozart F1: 0.55

#### LSTM Performance

#### Accuracy: 66%

#### Macro F1-score: 0.42

#### Bach F1: 0.82

#### Beethoven F1: 0.34

#### Chopin F1: 0.35

#### Mozart F1: 0.16

## Discussion

The CNN model ended up being the better all-around performer. It not only had the highest accuracy but also did a solid job across all four composers. This makes sense since CNNs are great at spotting local musical phrases or repeated motifs that are common in classical music.

The LSTM model did well with Bach, which isn’t surprising. His music tends to be more structured and layered, which fits well with how LSTMs work. But it struggled with the other composers, especially Mozart, where the F1 score dropped to just 0.16. This suggests the LSTM may have overfit to the kinds of patterns found in Bach’s music and couldn’t generalize as well.

We didn’t end up including a hybrid CNN-LSTM model in the final submission, mainly because of complexity and runtime issues. Surprisingly, the simpler models actually performed better for this dataset.

**Conclusion & Future Work**

Overall, this project showed that deep learning, especially CNNs, can do a pretty good job of identifying classical composers from MIDI note data. While LSTMs can capture more structure and long-term flow, they seemed to require more data and tuning to really perform well across all composers.

Going forward, we’d like to:

* Try Transformer models, which could potentially capture both local and global patterns more effectively
* Add data augmentation, like transposing pieces or stretching sequences
* Include more musical elements beyond just pitch, such as rhythm, key, dynamics, and harmony
* Balance the dataset better to avoid skew toward Bach

Even with the challenges, it was exciting to see how much musical style can be learned by machines just from raw note sequences.

# References

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Kaggle (n.d.). MIDI Classical Music Dataset. https://www.kaggle.com/datasets/blanderbuss/midi-classic-music

Pretty MIDI Library: https://github.com/craffel/pretty-midi