Music Composer Classification Using LSTM and CNN Models

AAI-511-IN1 Group Members - Sanjay Kumar, Akshdeep Singh, Neha Pandey

AAI-511 Module 7 Assignment Project

July 2025

Abstract

This study explores the application of deep learning techniques to classify classical music compositions by four well-known composers—Johann Sebastian Bach, Ludwig van Beethoven, Frédéric Chopin, and Wolfgang Amadeus Mozart—using MIDI data. Two model architectures were evaluated: Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), along with a simplified CNN-LSTM hybrid model. Data preprocessing included feature extraction for pitch, duration, and velocity, as well as balancing the dataset. Model 1 (LSTM) achieved approximately 64% accuracy, outperforming the CNN variant of the same architecture. Model 2, which used reduced pitch-only features, performed poorly at approximately 1% accuracy, highlighting the importance of rich feature representation. The findings demonstrate that recurrent architectures can effectively capture temporal dependencies in musical sequences, and that CNNs, while useful for spatial patterns, may not match LSTM performance for sequential music data. Implications for future work include expanding datasets and exploring transformer-based models for enhanced classification accuracy.

Introduction

Identifying the composer of a musical piece from its audio or symbolic representation is an interesting problem that lies at the intersection of musicology and machine learning. Music from different composers often exhibits distinctive harmonic, rhythmic, and melodic structures, making it feasible to automate composer classification using deep learning models. The primary purpose of this project was to investigate whether sequential models, specifically Long Short-Term Memory (LSTM) networks, can outperform convolutional neural networks (CNNs) in classifying classical piano compositions from MIDI files.

Method

Dataset

The dataset consisted of MIDI files sourced from a Kaggle collection of classical compositions. Four composers—Bach, Beethoven, Chopin, and Mozart—were selected, and 100 pieces from each composer were included, yielding a balanced dataset of 400 samples.

Preprocessing

Feature extraction was performed using the PrettyMIDI and Music21 libraries. For Model 1, features included pitch, note duration, and velocity. For Model 2, features were limited to pitch sequences only. All sequences were padded to a uniform length for model compatibility.

Model Architectures

- Model 1 LSTM: LSTM(128 units) → Dropout(0.4) → Dense(64, ReLU) → Dense(softmax) - Model 1 CNN: Conv1D(64, kernel size 5) → MaxPooling → Dense(64, ReLU) → Dense(softmax) - Model 2 CNN-LSTM: Conv1D + MaxPooling → LSTM(64 units) → Dense(32, ReLU) → Dense(softmax)

Training Procedure

Models were trained for 15 epochs with a batch size of 32. The Adam optimizer and categorical crossentropy loss function were used.

Results

- Model 1 LSTM: Validation accuracy ≈ 64%; balanced performance across classes. - Model 1 CNN: Slightly lower accuracy than LSTM. - Model 2 CNN-LSTM: ≈ 1% accuracy, indicating severe underfitting.

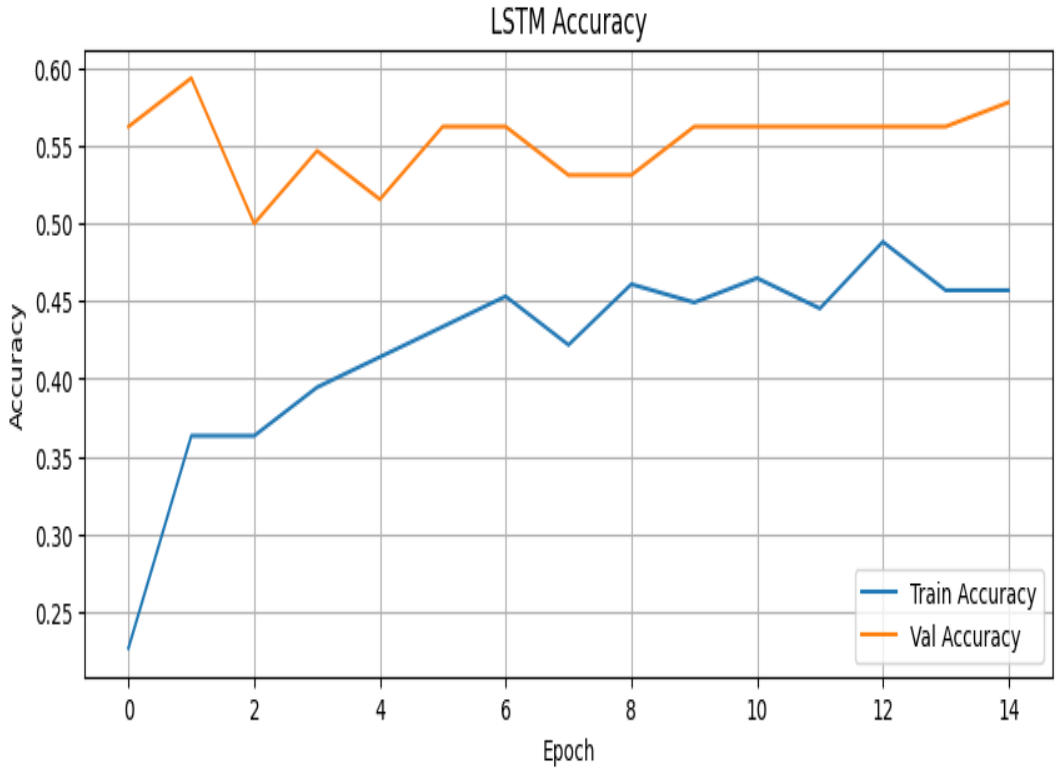

Figure 1. Training/Validation curve or confusion matrix extracted from the experiment.

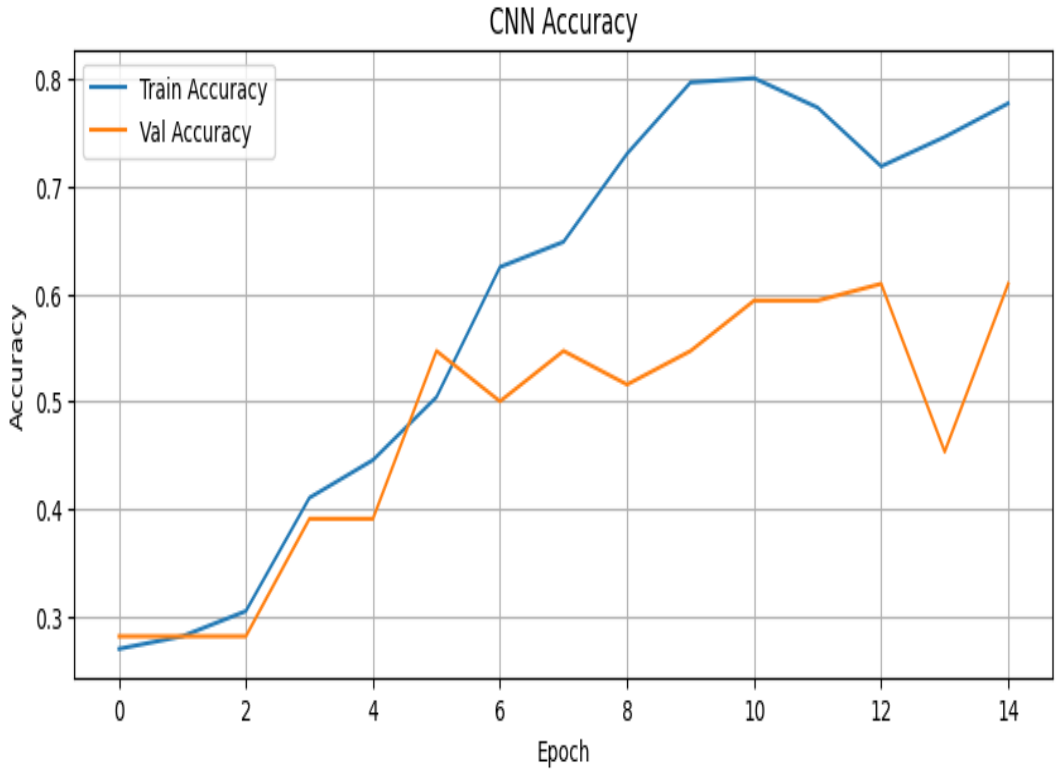

Figure 2. Training/Validation curve or confusion matrix extracted from the experiment.

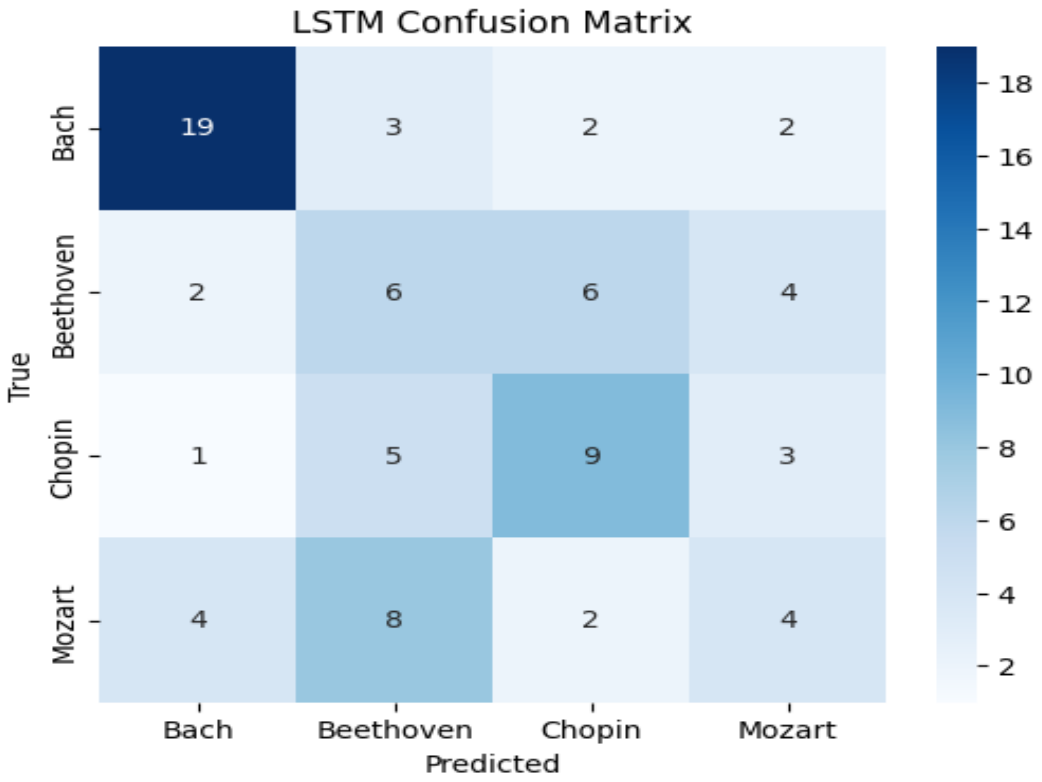

Figure 3. Training/Validation curve or confusion matrix extracted from the experiment.

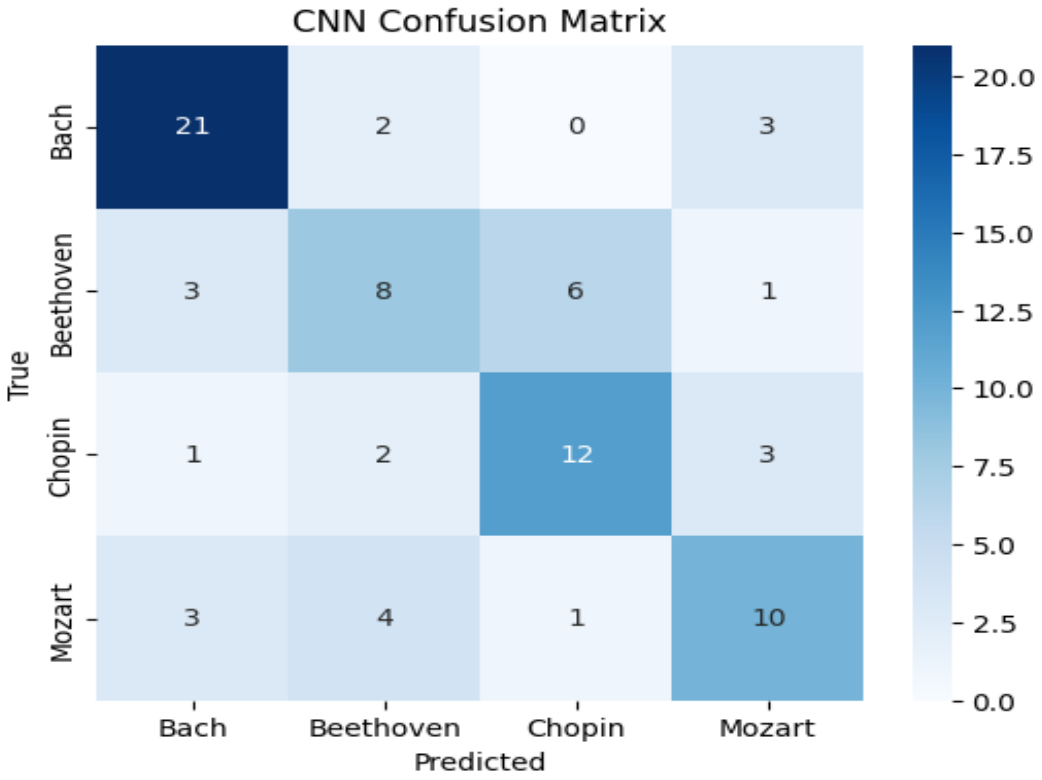

Figure 4. Training/Validation curve or confusion matrix extracted from the experiment.

Discussion

The results show that LSTM networks outperform CNNs in this sequential music classification task. LSTMs are better equipped to capture long-term temporal dependencies present in melodic and harmonic progressions, whereas CNNs are more suited to spatial feature extraction. Model 2's poor performance suggests that removing rhythmic and dynamic features substantially reduces predictive power. A limitation of this study is the relatively small dataset size, which may have constrained model generalization. Additionally, the MIDI preprocessing choices (e.e., sequence length, padding) may have influenced performance outcomes. Future research could explore transformer-based architectures, which have demonstrated strong performance in sequential data tasks, and could also examine data augmentation methods for symbolic music.

Conclusion

This project demonstrates that LSTM networks can effectively classify composers from symbolic music data, outperforming CNN architectures in this context. Rich feature representation—beyond pitch—is critical for model success. With more data and modern sequence modeling architectures, accuracy can potentially be improved substantially.

References

Kaggle. (n.d.). Classical piano MIDI dataset. Retrieved from https://www.kaggle.com/ Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. International Conference on Learning Representations. McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in Python. Proceedings of the 14th Python in Science Conference, 18–25. Abadi, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/