

/Users/gshiva/AA-511/example/mcp-server-example/mcp-server

MS get\_docs working

I'll help you test the get\_docs function. Let me call it with the default parameters to see how it works:

M get\_docs

The function is working, but it's looking for a specific MIDI file that doesn't exist at the expected path. The get\_docs function appears to:

1. Train a CNN-LSTM model on a music dataset
2. Predict the composer of a given MIDI file
3. Return the predicted composer, confidence score, and training accuracy

The error indicates that the MIDI file `/Users/gshiva/Downloads/music/Bach/example.mid` wasn't found. To use this function successfully, you would need:

1. A music dataset directory (defaulting to `~/Downloads/music`)
2. A specific MIDI file to analyze for composer prediction
3. The dataset should be organized with subdirectories for each composer

Would you like me to try with different parameters, or do you have a specific MIDI file path you'd like to test with?

M get\_docs

Great! The get\_docs function is now working successfully. Here are the results:

**Prediction Results:**

- **Predicted Composer:** midiclassics
- **Confidence:** 84.0% (0.840)
- **Training Accuracy:** 86.0% (0.860)

The CNN-LSTM model has been trained on the music dataset and analyzed the MIDI file. The model achieved a training accuracy of 86% and predicted the composer as "midiclassics" with 84% confidence. This suggests the model found patterns in the MIDI file that it associates with this particular composer category from the training data.

The function is working as expected - it trains a neural network model on the entire music dataset and then uses that trained model to predict the composer of the specified MIDI file.

```
@mcp-server.get_docs
{
  "dataset_dir": "/Users/gshiva/Downloads/music",
```

```
"midi_path": "/Users/gshiva/Downloads/music/Bach/bwv1007.mid",
"seq_length": 200,
"epochs": 3,
"batch_size": 4
}
```

## Train

```
@mcp-server.train_model
{
  "dataset_dir": "/Users/gshiva/Downloads/music",
  "seq_length": 200,
  "epochs": 5,
  "batch_size": 8
}
```

```
MS @mcp-server.train_model
{
  "dataset_dir": "/Users/gshiva/Downloads/music",
  "seq_length": 200,
  "epochs": 5,
  "batch_size": 8
}
```

I'll train a CNN-LSTM model on your music dataset to classify composers. Let me start the training process with your specified parameters.

M train\_model

Great! The model training has completed successfully. Here's a summary of the results:

- **Training completed** with 128 samples processed
- **Final accuracy:** 85.22% - this is quite good for a composer classification task
- **Model saved** as `midi_model.keras`
- **Classes saved** as `midi_classes.npy`

The model achieved an 85.22% accuracy, which suggests it has learned to distinguish between different composers' musical styles reasonably well. The CNN-LSTM architecture is particularly effective for this type of sequential music data, as it can capture both local patterns (CNN) and temporal dependencies (LSTM) in the musical compositions.

You can now use this trained model to predict the composer of new MIDI files using the `predict_composer` function with the saved model and classes files.



📄 📌 💬 Retry

Claude can make mistakes. Please double-check responses.

## Predict

```
@mcp-server.predict_composer
{
  "midi_path": "/Users/gshiva/Downloads/music/Bach/example.mid",
  "seq_length": 200
}
```

MS @mcp-server.predict\_composer

```
{
  "midi_path": "/Users/gshiva/Downloads/music/Bach/example.mid",
  "seq_length": 200
}
```

M predict\_composer

The model has analyzed the MIDI file and made its prediction:

**Predicted Composer:** `midiclassics` **Confidence:** 85.10%

The model is quite confident (85.10%) that this piece belongs to the "midiclassics" category based on the musical patterns it detected. This appears to be one of the composer classes that the model was trained on from your dataset.

It's interesting that even though the file was in the "Bach" folder, the model classified it as "midiclassics" - this could mean either:

1. The model learned that this particular style/piece fits better with the "midiclassics" category
2. The piece might have characteristics that span multiple classical composers
3. The model's training data had overlapping or differently categorized examples

The high confidence score (85.10%) suggests the model found strong musical patterns that it associates with the "midiclassics" classification.



📄 🌟 🗨️ Retry ▾

Claude can make mistakes. Please double-check responses.