@mcp-server.train\_model { "dataset\_dir": "/Users/gshiva/Downloads/music", "seq\_length": 200, "epochs": 5, "batch\_size": 8 }

Edit

I'll train the model using your specified parameters. Let me start the training process with your dataset.

M

train\_model

Request

{

`epochs`: 5,

`batch\_size`: 8,

`seq\_length`: 200,

`dataset\_dir`: `/Users/gshiva/Downloads/music`

}

Response

{"classes": ["Albe\u0301niz", "Alkan", "Ambroise", "Arensky", "Arndt", "Bach", "midiclassics"], "lstm": {"path": "lstm\_model.keras", "accuracy": 0.852173924446106}, "cnn": {"path": "cnn\_model.keras", "accuracy": 0.852173924446106}, "classes\_path": "midi\_classes.npy"}

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

**Training Results:**

* **LSTM Model Accuracy**: 85.22%
* **CNN Model Accuracy**: 85.22%
* **Training Epochs**: 5
* **Batch Size**: 8
* **Sequence Length**: 200

**Saved Files:**

* LSTM model: lstm\_model.keras
* CNN model: cnn\_model.keras
* Classes file: midi\_classes.npy

**Detected Composers:** The model was trained to recognize 7 different composers/sources:

1. Albéniz
2. Alkan
3. Ambroise
4. Arensky
5. Arndt
6. Bach
7. midiclassics

Both models achieved the same accuracy of about 85.2%, which is quite good for a 5-epoch training run. The models are now ready to predict composers from MIDI files. You can use them to classify new MIDI files by calling the prediction function with the path to a MIDI file.

Retry

MS

@mcp-server.predict\_composer { "midi\_path": "/Users/gshiva/Downloads/music/Bach/example.mid", "seq\_length": 200 }

Edit

I'll predict the composer for the Bach MIDI file you specified.

M

predict\_composer

Request

{

`midi\_path`: `/Users/gshiva/Downloads/music/Bach/example.mid`,

`seq\_length`: 200

}

Response

{"model\_type": "lstm", "composer": "midiclassics", "confidence": 0.8252313733100891}

The prediction results are in:

**Model Used**: LSTM (selected based on sequence length) **Predicted Composer**: midiclassics **Confidence**: 82.52%

Interestingly, the model predicted "midiclassics" rather than "Bach" for this particular MIDI file, despite it being in the Bach folder. This could happen for several reasons:

1. The specific MIDI file might have characteristics that are more similar to the "midiclassics" training data
2. The file might be a Bach piece that was processed or arranged in a way that makes it stylistically closer to the midiclassics category
3. The model might need more training data or epochs to better distinguish between Bach and midiclassics

The confidence level of 82.52% is reasonably high, suggesting the model is fairly certain about its prediction, even though it doesn't match the folder name.

Retry

MS

@mcp-server.predict\_composer { "midi\_path": "/Users/gshiva/Downloads/music/Bach/bwv1007.mid", "seq\_length": 200, "model\_type": cnn }

Edit

I'll predict the composer for the Bach BWV1007 MIDI file using the CNN model specifically.

M

predict\_composer

Request

{

`midi\_path`: `/Users/gshiva/Downloads/music/Bach/bwv1007.mid`,

`model\_type`: `cnn`,

`seq\_length`: 200

}

Response

Error executing tool predict\_composer: [Errno 2] No such file or directory: '/Users/gshiva/Downloads/music/Bach/bwv1007.mid'

The prediction failed because the file /Users/gshiva/Downloads/music/Bach/bwv1007.mid doesn't exist at that path.

It looks like the specific MIDI file bwv1007.mid isn't present in your Bach folder. You might want to:

1. Check what MIDI files are actually available in your Bach folder
2. Use a different filename that exists in the dataset
3. Verify the exact filename and path

Would you like to try with a different MIDI file, or would you prefer to check what files are available in your Bach directory first?

Retry

MS

@mcp-server.predict\_composer { "midi\_path": "/Users/gshiva/Downloads/music/Bach/example.mid", "seq\_length": 200, "model\_type": cnn }

Edit

I'll predict the composer for the same Bach example file, but this time using the CNN model specifically.

M

predict\_composer

Request

{

`midi\_path`: `/Users/gshiva/Downloads/music/Bach/example.mid`,

`model\_type`: `cnn`,

`seq\_length`: 200

}

Response

{"model\_type": "cnn", "composer": "midiclassics", "confidence": 0.8653297424316406}

Here are the CNN model prediction results:

**Model Used**: CNN (as requested) **Predicted Composer**: midiclassics **Confidence**: 86.53%

Interesting comparison with the previous LSTM prediction:

* **LSTM**: Predicted "midiclassics" with 82.52% confidence
* **CNN**: Predicted "midiclassics" with 86.53% confidence

Both models agree on the prediction but the CNN model shows higher confidence (86.53% vs 82.52%). This suggests that for this particular MIDI file, both the LSTM and CNN models consistently identify stylistic features that align more closely with the "midiclassics" category rather than the "Bach" category, despite the file being in the Bach folder.

The CNN model's slightly higher confidence might indicate that it's better at capturing the spatial/structural patterns in this particular piece that distinguish it from typical Bach compositions in the training set.

Retry

MS

@mcp-server.predict\_composer { "midi\_path": "/Users/gshiva/Downloads/music/Bach/example.mid", "seq\_length": 200, "model\_type": lstm}

Edit

I'll predict the composer for the Bach example file using the LSTM model specifically.

M

predict\_composer

Request

{

`midi\_path`: `/Users/gshiva/Downloads/music/Bach/example.mid`,

`model\_type`: `lstm`,

`seq\_length`: 200

}

Response

{"model\_type": "lstm", "composer": "midiclassics", "confidence": 0.8252313733100891}

Here are the LSTM model prediction results:

**Model Used**: LSTM (as requested) **Predicted Composer**: midiclassics **Confidence**: 82.52%

Now we have a complete comparison across all three predictions for the same file:

| **Model Selection** | **Model Used** | **Predicted Composer** | **Confidence** |
| --- | --- | --- | --- |
| Auto (seq\_length=200) | LSTM | midiclassics | 82.52% |
| Explicit CNN | CNN | midiclassics | 86.53% |
| Explicit LSTM | LSTM | midiclassics | 82.52% |

**Key Observations:**

1. **Consistent Prediction**: All models consistently predict "midiclassics" rather than "Bach"
2. **CNN vs LSTM**: The CNN model shows higher confidence (86.53%) compared to LSTM (82.52%)
3. **Auto-selection**: When using seq\_length=200 without specifying model type, the system automatically chose LSTM
4. **Model Agreement**: Both models agree on the classification, suggesting the file genuinely has characteristics more similar to the "midiclassics" training data

This consistency across models suggests the prediction is likely accurate based on the musical features the models learned during training.