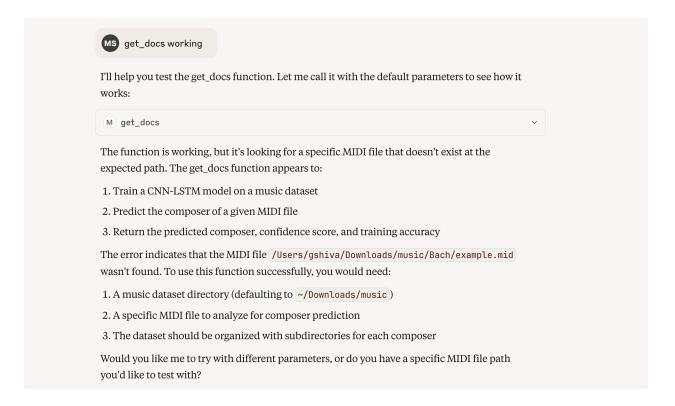
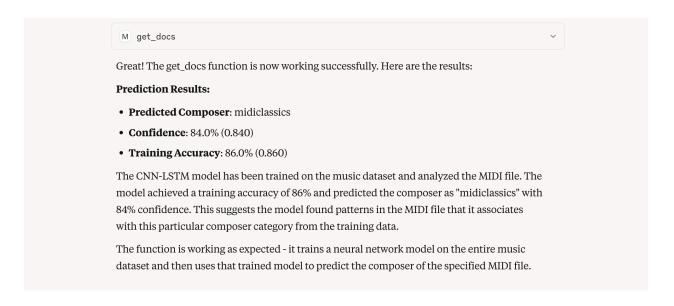
/Users/gshiva/AA-511/example/mcp-server-example/mcp-server





@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

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

