Deep Learning Lab Experiment - 5

Music Generation Using the MAESTRO Dataset

Develop and train a deep neural network to generate music sequences using the MAESTRO Dataset. The model should be capable of composing piano music that mimics the style of the training data. Load and process the MIDI files, analyze their structure, and convert them into piano roll or token-based sequences. Split the dataset into Training, Validation, and Test sets, apply scaling and normalization, and optionally use data augmentation techniques. Design a deep learning model using LSTM, GRU, or Transformer-based architectures, train and evaluate it, and analyze Loss, Accuracy, and the quality of generated music. Finally, discuss model performance, music generation quality, challenges, and potential improvements.

In [8]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [9]:

```
#Extract Notes and Piano_rolls from MIDI Files

import os
import numpy as np
import pandas as pd
import pretty_midi
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split

# Define the dataset path
dataset_path = "/content/drive/MyDrive/maestro-v3.0.0/"

# Load metadata CSV file
metadata_file = os.path.join(dataset_path, "maestro-v3.0.0.csv")
df = pd.read_csv('maestro-v3.0.0.csv')

selected_years = [2017]
df = df[df['year'].isin(selected_years)]

# Get list of MIDI files
midi_files = [os.path.join(dataset_path, f) for f in df['midi_filename'].tolist()]

# Function to extract note sequences from MIDI
def midi_to_notes(midi_file):
    pm = pretty_midi.PrettyMIDI(midi_file)
    notes = []

    for instrument in pm.instruments:
        if instrument.is_drum:
            continue  # Skip drum tracks

        for note in instrument.notes:
            notes.append([note.start, note.end, note.pitch, note.velocity])

    return np.array(notes)

# Process all MIDI files
all_notes = [midi_to_notes(f) for f in midi_files]
all_notes = np.concatenate(all_notes, axis=0)

# Exploratory Data Analysis (EDA) on MIDI Notes
plt.figure(figsize=(10, 4))
sns.histplot(all_notes[:, 2], bins=50, kde=True, color="blue")  # MIDI pitch distribution
plt.xlabel("MIDI Pitch")
plt.ylabel("Frequency")
plt.title("Pitch Distribution in MAESTRO Dataset")
plt.show()

# Function to convert MIDI to piano roll
def midi_to_piano_roll(midi_file, fs=100):
    pm = pretty_midi.PrettyMIDI(midi_file)
    piano_roll = pm.get_piano_roll(fs=fs)  # fs = time steps per second
    return piano_roll.T  # Transpose to have (time_steps, pitch_classes)

# Convert all MIDI files to piano rolls
piano_rolls = [midi_to_piano_roll(f) for f in midi_files]

# Define a fixed maximum sequence length
max_length = 1000
```

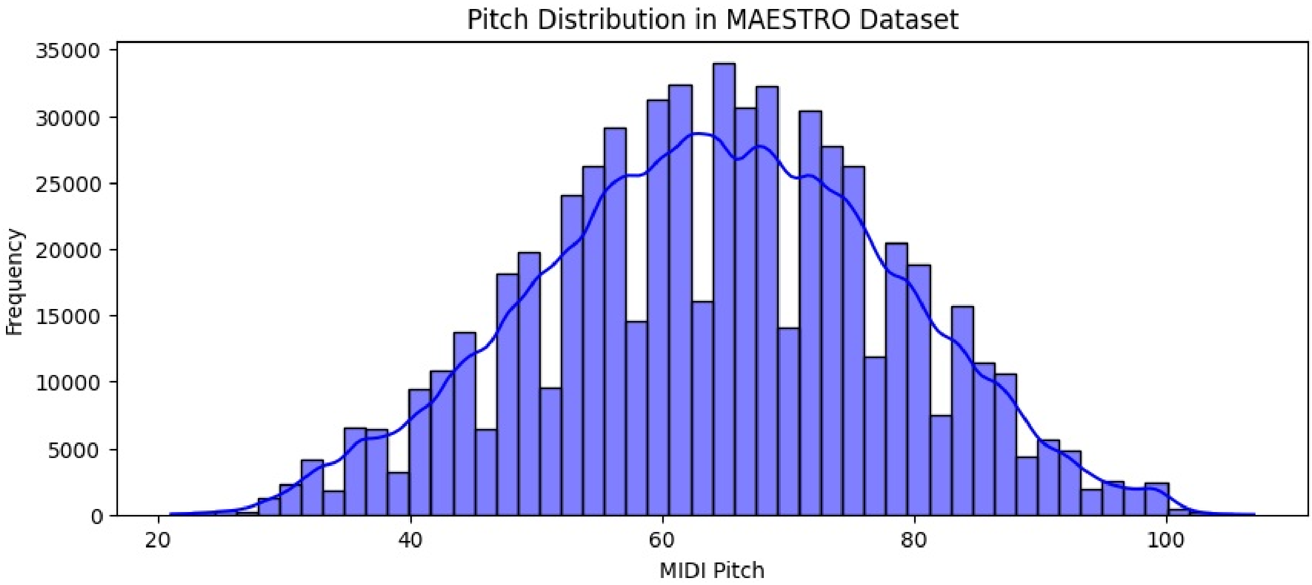
```python
# Pad or truncate sequences to the same length
piano_rolls = [x[:max_length] if x.shape[0] > max_length else np.pad(x, ((0, max_length - x.shape[0]), (0, 0)))

# Convert to NumPy array
X = np.array(piano_rolls)
max = np.max(X)
print(max)

# Train-Test-Validation Split
X_train, X_temp = train_test_split(X, test_size=0.2, random_state=42)
X_val, X_test = train_test_split(X_temp, test_size=0.5, random_state=42)

# Normalize input data
X_train = X_train / max  # Normalize MIDI velocities
X_val = X_val / max
X_test = X_test / max

# Print data shapes
print(f"Training Data Shape: {X_train.shape}")
print(f"Validation Data Shape: {X_val.shape}")
print(f"Test Data Shape: {X_test.shape}")
```




```
114.0
Training Data Shape: (112, 1000, 128)
Validation Data Shape: (14, 1000, 128)
Test Data Shape: (14, 1000, 128)
```

In [11]:
```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, LSTM, Dense, Dropout, TimeDistributed, BatchNormaliza

# Define Model
def build_cnn_lstm(input_shape=(1000, 128)):
    model = Sequential()

    # CNN Feature Extraction
    model.add(Conv1D(64, kernel_size=3, activation='relu', padding='same', input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(MaxPooling1D(pool_size=2))

    model.add(Conv1D(128, kernel_size=3, activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling1D(pool_size=2))

    # LSTM for Temporal Learning
    model.add(LSTM(128, return_sequences=True))
    model.add(Dropout(0.3))

    model.add(LSTM(64, return_sequences=True))
    model.add(Dropout(0.3))

    # Upsampling to match the original temporal dimension
    model.add(UpSampling1D(size=2))  # Upsample by a factor of 2
    model.add(UpSampling1D(size=2))  # Upsample by a factor of 2

    # Fully Connected Output Layer (Predict Note Activations)
    model.add(TimeDistributed(Dense(128, activation='sigmoid')))
```

```python
    # Compile Model
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

    return model


# Define input shape (time_steps, pitch_classes)
input_shape = (1000, 128)
model = build_cnn_lstm(input_shape)
model.summary()

# Train Model
history = model.fit(
    X_train, X_train,  # Autoencoder-style training
    epochs=200,
    batch_size=8,
    validation_data=(X_val, X_val)
)

# Evaluate Model
loss, accuracy = model.evaluate(X_test, X_test)
print(f"Test Loss: {loss:.4f}, Test Accuracy: {accuracy:.4f}")

# Plot Training History
plt.figure(figsize=(10, 4))

# Loss Curve
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training & Validation Loss')
plt.legend()

# Accuracy Curve
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training & Validation Accuracy')
plt.legend()

plt.show()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 1000, 64)	24,640
batch_normalization_2 (BatchNormalization)	(None, 1000, 64)	256
max_pooling1d_2 (MaxPooling1D)	(None, 500, 64)	0
conv1d_3 (Conv1D)	(None, 500, 128)	24,704
batch_normalization_3 (BatchNormalization)	(None, 500, 128)	512
max_pooling1d_3 (MaxPooling1D)	(None, 250, 128)	0
lstm_2 (LSTM)	(None, 250, 128)	131,584
dropout_2 (Dropout)	(None, 250, 128)	0
lstm_3 (LSTM)	(None, 250, 64)	49,408
dropout_3 (Dropout)	(None, 250, 64)	0
up_sampling1d_2 (UpSampling1D)	(None, 500, 64)	0
up_sampling1d_3 (UpSampling1D)	(None, 1000, 64)	0
time_distributed_1 (TimeDistributed)	(None, 1000, 128)	8,320

Total params: 239,424 (935.25 KB)

Trainable params: 239,040 (933.75 KB)

Non-trainable params: 384 (1.50 KB)

```
Epoch 1/200
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 62s 1s/step - accuracy: 0.0061 - loss: 0.6619 - val_accuracy: 0.0000e+00 - val_loss:
0.4925
Epoch 2/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 7s 158ms/step - accuracy: 0.0016 - loss: 0.3941 - val_accuracy: 0.0000e+00 - val_loss
: 0.1959
Epoch 3/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 127ms/step - accuracy: 0.0020 - loss: 0.1782 - val_accuracy: 0.0000e+00 - val_loss
: 0.1107
Epoch 4/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.0024 - loss: 0.1102 - val_accuracy: 0.0000e+00 - val_loss
: 0.0862
Epoch 5/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.0037 - loss: 0.0853 - val_accuracy: 0.0000e+00 - val_loss
: 0.0769
Epoch 6/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.0100 - loss: 0.0793 - val_accuracy: 0.0367 - val_loss: 0.
0723
Epoch 7/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.0190 - loss: 0.0770 - val_accuracy: 0.0367 - val_loss: 0.
0696
Epoch 8/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 154ms/step - accuracy: 0.0222 - loss: 0.0719 - val_accuracy: 0.0367 - val_loss: 0.
0678
Epoch 9/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 136ms/step - accuracy: 0.0223 - loss: 0.0690 - val_accuracy: 0.0367 - val_loss: 0.
0664
Epoch 10/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.0211 - loss: 0.0654 - val_accuracy: 0.0367 - val_loss: 0.
0654
Epoch 11/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.0254 - loss: 0.0681 - val_accuracy: 0.0367 - val_loss: 0.
0646
Epoch 12/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.0280 - loss: 0.0687 - val_accuracy: 0.0367 - val_loss: 0.
0641
Epoch 13/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.0228 - loss: 0.0672 - val_accuracy: 0.0367 - val_loss: 0.
0635
Epoch 14/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 146ms/step - accuracy: 0.0257 - loss: 0.0654 - val_accuracy: 0.0367 - val_loss: 0.
0629
Epoch 15/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 145ms/step - accuracy: 0.0246 - loss: 0.0646 - val_accuracy: 0.0367 - val_loss: 0.
0626
Epoch 16/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.0272 - loss: 0.0673 - val_accuracy: 0.0367 - val_loss: 0.
0623
Epoch 17/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.0222 - loss: 0.0676 - val_accuracy: 0.0367 - val_loss: 0.
0620
Epoch 18/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.0239 - loss: 0.0672 - val_accuracy: 0.0367 - val_loss: 0.
0618
Epoch 19/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 122ms/step - accuracy: 0.0265 - loss: 0.0633 - val_accuracy: 0.0367 - val_loss: 0.
0615
Epoch 20/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.0232 - loss: 0.0660 - val_accuracy: 0.0367 - val_loss: 0.
0611
Epoch 21/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.0244 - loss: 0.0666 - val_accuracy: 0.0366 - val_loss: 0.
0608
Epoch 22/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.0258 - loss: 0.0600 - val_accuracy: 0.0436 - val_loss: 0.
0606
Epoch 23/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.0285 - loss: 0.0599 - val_accuracy: 0.0367 - val_loss: 0.
0599
Epoch 24/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 122ms/step - accuracy: 0.0281 - loss: 0.0639 - val_accuracy: 0.0373 - val_loss: 0.
0594
Epoch 25/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.0280 - loss: 0.0639 - val_accuracy: 0.0367 - val_loss: 0.
0592
Epoch 26/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 147ms/step - accuracy: 0.0276 - loss: 0.0613 - val_accuracy: 0.0393 - val_loss: 0.
0590
Epoch 27/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 133ms/step - accuracy: 0.0270 - loss: 0.0601 - val_accuracy: 0.0367 - val_loss: 0.
0586
Epoch 28/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.0268 - loss: 0.0575 - val_accuracy: 0.0506 - val_loss: 0.
0582
```

```
Epoch 29/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 122ms/step - accuracy: 0.0314 - loss: 0.0605 - val_accuracy: 0.0367 - val_loss: 0.
0580
Epoch 30/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.0290 - loss: 0.0609 - val_accuracy: 0.0367 - val_loss: 0.
0577
Epoch 31/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.0288 - loss: 0.0595 - val_accuracy: 0.0367 - val_loss: 0.
0576
Epoch 32/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 147ms/step - accuracy: 0.0276 - loss: 0.0574 - val_accuracy: 0.0491 - val_loss: 0.
0574
Epoch 33/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 149ms/step - accuracy: 0.0317 - loss: 0.0609 - val_accuracy: 0.0367 - val_loss: 0.
0572
Epoch 34/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.0282 - loss: 0.0606 - val_accuracy: 0.0373 - val_loss: 0.
0570
Epoch 35/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.0302 - loss: 0.0594 - val_accuracy: 0.0367 - val_loss: 0.
0569
Epoch 36/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 118ms/step - accuracy: 0.0299 - loss: 0.0572 - val_accuracy: 0.0370 - val_loss: 0.
0568
Epoch 37/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 119ms/step - accuracy: 0.0366 - loss: 0.0589 - val_accuracy: 0.0717 - val_loss: 0.
0565
Epoch 38/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 155ms/step - accuracy: 0.0397 - loss: 0.0566 - val_accuracy: 0.0834 - val_loss: 0.
0562
Epoch 39/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 137ms/step - accuracy: 0.0376 - loss: 0.0577 - val_accuracy: 0.0594 - val_loss: 0.
0563
Epoch 40/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.0420 - loss: 0.0568 - val_accuracy: 0.0389 - val_loss: 0.
0559
Epoch 41/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.0369 - loss: 0.0558 - val_accuracy: 0.0729 - val_loss: 0.
0558
Epoch 42/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.0455 - loss: 0.0541 - val_accuracy: 0.0564 - val_loss: 0.
0555
Epoch 43/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 122ms/step - accuracy: 0.0480 - loss: 0.0567 - val_accuracy: 0.1059 - val_loss: 0.
0552
Epoch 44/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 142ms/step - accuracy: 0.0525 - loss: 0.0572 - val_accuracy: 0.0936 - val_loss: 0.
0551
Epoch 45/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 154ms/step - accuracy: 0.0499 - loss: 0.0599 - val_accuracy: 0.0976 - val_loss: 0.
0549
Epoch 46/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.0582 - loss: 0.0575 - val_accuracy: 0.1265 - val_loss: 0.
0546
Epoch 47/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.0566 - loss: 0.0566 - val_accuracy: 0.1061 - val_loss: 0.
0545
Epoch 48/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.0582 - loss: 0.0589 - val_accuracy: 0.1288 - val_loss: 0.
0540
Epoch 49/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.0642 - loss: 0.0549 - val_accuracy: 0.1245 - val_loss: 0.
0536
Epoch 50/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.0676 - loss: 0.0538 - val_accuracy: 0.1373 - val_loss: 0.
0535
Epoch 51/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 135ms/step - accuracy: 0.0684 - loss: 0.0519 - val_accuracy: 0.1670 - val_loss: 0.
0533
Epoch 52/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 142ms/step - accuracy: 0.0754 - loss: 0.0558 - val_accuracy: 0.1407 - val_loss: 0.
0531
Epoch 53/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.0865 - loss: 0.0565 - val_accuracy: 0.1359 - val_loss: 0.
0527
Epoch 54/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 125ms/step - accuracy: 0.0762 - loss: 0.0527 - val_accuracy: 0.1984 - val_loss: 0.
0524
Epoch 55/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 124ms/step - accuracy: 0.0840 - loss: 0.0546 - val_accuracy: 0.1861 - val_loss: 0.
0523
Epoch 56/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.0864 - loss: 0.0526 - val_accuracy: 0.1964 - val_loss: 0.
```

```
0519
Epoch 57/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 149ms/step - accuracy: 0.1052 - loss: 0.0515 - val_accuracy: 0.2019 - val_loss: 0.
0518
Epoch 58/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 149ms/step - accuracy: 0.0975 - loss: 0.0549 - val_accuracy: 0.1754 - val_loss: 0.
0515
Epoch 59/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.1120 - loss: 0.0531 - val_accuracy: 0.2036 - val_loss: 0.
0511
Epoch 60/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 124ms/step - accuracy: 0.1170 - loss: 0.0523 - val_accuracy: 0.2099 - val_loss: 0.
0508
Epoch 61/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 125ms/step - accuracy: 0.1233 - loss: 0.0530 - val_accuracy: 0.2101 - val_loss: 0.
0506
Epoch 62/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.1309 - loss: 0.0529 - val_accuracy: 0.2226 - val_loss: 0.
0503
Epoch 63/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 154ms/step - accuracy: 0.1191 - loss: 0.0495 - val_accuracy: 0.2342 - val_loss: 0.
0501
Epoch 64/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 141ms/step - accuracy: 0.1217 - loss: 0.0522 - val_accuracy: 0.2379 - val_loss: 0.
0498
Epoch 65/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.1478 - loss: 0.0484 - val_accuracy: 0.2399 - val_loss: 0.
0495
Epoch 66/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.1351 - loss: 0.0515 - val_accuracy: 0.2429 - val_loss: 0.
0491
Epoch 67/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.1521 - loss: 0.0483 - val_accuracy: 0.2679 - val_loss: 0.
0488
Epoch 68/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.1448 - loss: 0.0495 - val_accuracy: 0.2694 - val_loss: 0.
0485
Epoch 69/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 165ms/step - accuracy: 0.1540 - loss: 0.0503 - val_accuracy: 0.2804 - val_loss: 0.
0481
Epoch 70/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 132ms/step - accuracy: 0.1476 - loss: 0.0487 - val_accuracy: 0.2744 - val_loss: 0.
0478
Epoch 71/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.1766 - loss: 0.0509 - val_accuracy: 0.2701 - val_loss: 0.
0476
Epoch 72/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 119ms/step - accuracy: 0.1818 - loss: 0.0490 - val_accuracy: 0.2766 - val_loss: 0.
0471
Epoch 73/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.1826 - loss: 0.0483 - val_accuracy: 0.2743 - val_loss: 0.
0469
Epoch 74/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.1677 - loss: 0.0502 - val_accuracy: 0.3056 - val_loss: 0.
0466
Epoch 75/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 159ms/step - accuracy: 0.1842 - loss: 0.0503 - val_accuracy: 0.3166 - val_loss: 0.
0463
Epoch 76/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.1714 - loss: 0.0528 - val_accuracy: 0.3149 - val_loss: 0.
0461
Epoch 77/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 124ms/step - accuracy: 0.1898 - loss: 0.0515 - val_accuracy: 0.3399 - val_loss: 0.
0455
Epoch 78/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.1717 - loss: 0.0512 - val_accuracy: 0.3133 - val_loss: 0.
0451
Epoch 79/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.1892 - loss: 0.0464 - val_accuracy: 0.3287 - val_loss: 0.
0448
Epoch 80/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 130ms/step - accuracy: 0.1990 - loss: 0.0465 - val_accuracy: 0.3361 - val_loss: 0.
0444
Epoch 81/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 169ms/step - accuracy: 0.2092 - loss: 0.0481 - val_accuracy: 0.3094 - val_loss: 0.
0442
Epoch 82/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.2017 - loss: 0.0471 - val_accuracy: 0.3295 - val_loss: 0.
0439
Epoch 83/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 118ms/step - accuracy: 0.2268 - loss: 0.0448 - val_accuracy: 0.3214 - val_loss: 0.
0436
Epoch 84/200
```

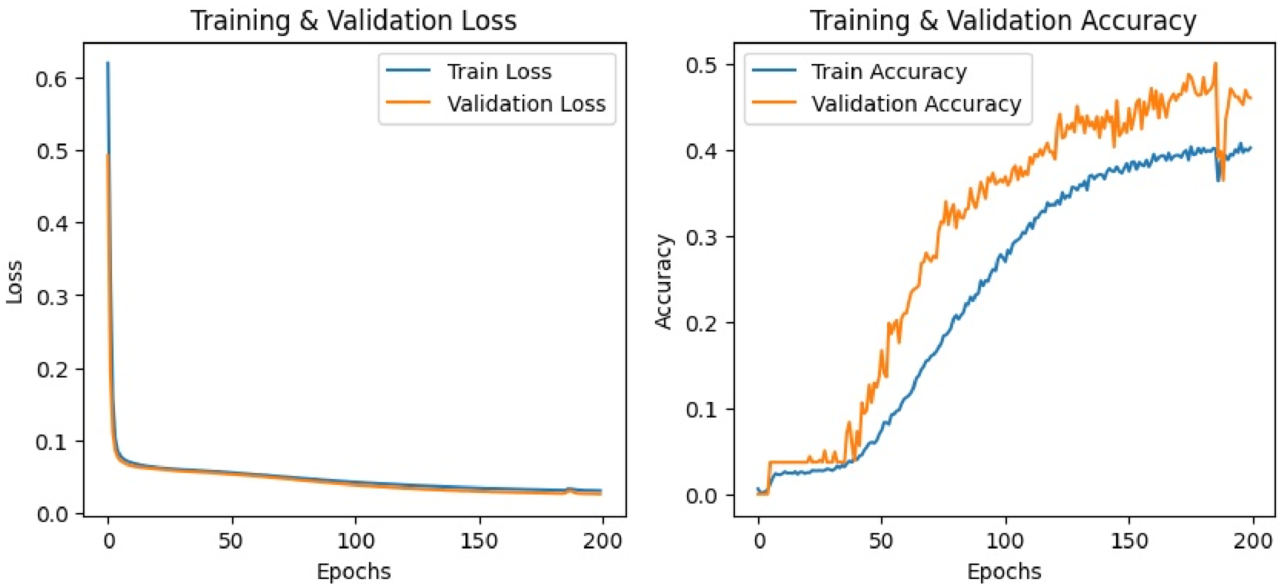
```
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.2188 - loss: 0.0465 - val_accuracy: 0.3204 - val_loss: 0.
0433
Epoch 85/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.2517 - loss: 0.0429 - val_accuracy: 0.3311 - val_loss: 0.
0429
Epoch 86/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.2031 - loss: 0.0452 - val_accuracy: 0.3319 - val_loss: 0.
0426
Epoch 87/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 128ms/step - accuracy: 0.2230 - loss: 0.0474 - val_accuracy: 0.3554 - val_loss: 0.
0424
Epoch 88/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 170ms/step - accuracy: 0.2253 - loss: 0.0446 - val_accuracy: 0.3399 - val_loss: 0.
0420
Epoch 89/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.2358 - loss: 0.0435 - val_accuracy: 0.3328 - val_loss: 0.
0416
Epoch 90/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.2270 - loss: 0.0439 - val_accuracy: 0.3455 - val_loss: 0.
0414
Epoch 91/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.2402 - loss: 0.0441 - val_accuracy: 0.3624 - val_loss: 0.
0412
Epoch 92/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.2612 - loss: 0.0429 - val_accuracy: 0.3544 - val_loss: 0.
0408
Epoch 93/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.2354 - loss: 0.0445 - val_accuracy: 0.3431 - val_loss: 0.
0405
Epoch 94/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 165ms/step - accuracy: 0.2555 - loss: 0.0462 - val_accuracy: 0.3676 - val_loss: 0.
0404
Epoch 95/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.2381 - loss: 0.0455 - val_accuracy: 0.3630 - val_loss: 0.
0400
Epoch 96/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 119ms/step - accuracy: 0.2765 - loss: 0.0422 - val_accuracy: 0.3731 - val_loss: 0.
0397
Epoch 97/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.2577 - loss: 0.0428 - val_accuracy: 0.3607 - val_loss: 0.
0394
Epoch 98/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.2724 - loss: 0.0418 - val_accuracy: 0.3622 - val_loss: 0.
0392
Epoch 99/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.2445 - loss: 0.0442 - val_accuracy: 0.3654 - val_loss: 0.
0391
Epoch 100/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 146ms/step - accuracy: 0.2769 - loss: 0.0438 - val_accuracy: 0.3621 - val_loss: 0.
0387
Epoch 101/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.2767 - loss: 0.0445 - val_accuracy: 0.3689 - val_loss: 0.
0387
Epoch 102/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 123ms/step - accuracy: 0.2700 - loss: 0.0438 - val_accuracy: 0.3591 - val_loss: 0.
0382
Epoch 103/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.2497 - loss: 0.0429 - val_accuracy: 0.3638 - val_loss: 0.
0379
Epoch 104/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.2877 - loss: 0.0416 - val_accuracy: 0.3774 - val_loss: 0.
0376
Epoch 105/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.2895 - loss: 0.0419 - val_accuracy: 0.3814 - val_loss: 0.
0375
Epoch 106/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 153ms/step - accuracy: 0.2747 - loss: 0.0431 - val_accuracy: 0.3651 - val_loss: 0.
0374
Epoch 107/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.3014 - loss: 0.0386 - val_accuracy: 0.3801 - val_loss: 0.
0370
Epoch 108/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 121ms/step - accuracy: 0.2848 - loss: 0.0432 - val_accuracy: 0.3693 - val_loss: 0.
0368
Epoch 109/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.2947 - loss: 0.0410 - val_accuracy: 0.3753 - val_loss: 0.
0366
Epoch 110/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.3058 - loss: 0.0420 - val_accuracy: 0.3709 - val_loss: 0.
0364
Epoch 111/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 152ms/step - accuracy: 0.3214 - loss: 0.0389 - val_accuracy: 0.3913 - val_loss: 0.
0361
```

```
Epoch 112/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 146ms/step - accuracy: 0.2946 - loss: 0.0412 - val_accuracy: 0.3841 - val_loss: 0.
0362
Epoch 113/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.3037 - loss: 0.0412 - val_accuracy: 0.3949 - val_loss: 0.
0358
Epoch 114/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 125ms/step - accuracy: 0.3112 - loss: 0.0391 - val_accuracy: 0.3921 - val_loss: 0.
0356
Epoch 115/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.3379 - loss: 0.0375 - val_accuracy: 0.4006 - val_loss: 0.
0352
Epoch 116/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.3119 - loss: 0.0377 - val_accuracy: 0.3935 - val_loss: 0.
0350
Epoch 117/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 142ms/step - accuracy: 0.3338 - loss: 0.0402 - val_accuracy: 0.4021 - val_loss: 0.
0350
Epoch 118/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 160ms/step - accuracy: 0.3443 - loss: 0.0406 - val_accuracy: 0.4078 - val_loss: 0.
0347
Epoch 119/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.3454 - loss: 0.0381 - val_accuracy: 0.3976 - val_loss: 0.
0344
Epoch 120/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.3392 - loss: 0.0387 - val_accuracy: 0.3976 - val_loss: 0.
0344
Epoch 121/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 124ms/step - accuracy: 0.3367 - loss: 0.0370 - val_accuracy: 0.3892 - val_loss: 0.
0340
Epoch 122/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.3583 - loss: 0.0382 - val_accuracy: 0.4284 - val_loss: 0.
0339
Epoch 123/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.3159 - loss: 0.0371 - val_accuracy: 0.4417 - val_loss: 0.
0338
Epoch 124/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 157ms/step - accuracy: 0.3381 - loss: 0.0382 - val_accuracy: 0.4136 - val_loss: 0.
0337
Epoch 125/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 133ms/step - accuracy: 0.3420 - loss: 0.0356 - val_accuracy: 0.4152 - val_loss: 0.
0333
Epoch 126/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.3449 - loss: 0.0379 - val_accuracy: 0.4286 - val_loss: 0.
0333
Epoch 127/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.3520 - loss: 0.0351 - val_accuracy: 0.4246 - val_loss: 0.
0329
Epoch 128/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.3591 - loss: 0.0351 - val_accuracy: 0.4309 - val_loss: 0.
0327
Epoch 129/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.3387 - loss: 0.0368 - val_accuracy: 0.4206 - val_loss: 0.
0326
Epoch 130/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 133ms/step - accuracy: 0.3631 - loss: 0.0364 - val_accuracy: 0.4508 - val_loss: 0.
0325
Epoch 131/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 144ms/step - accuracy: 0.3626 - loss: 0.0355 - val_accuracy: 0.4311 - val_loss: 0.
0323
Epoch 132/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.3770 - loss: 0.0347 - val_accuracy: 0.4384 - val_loss: 0.
0322
Epoch 133/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 122ms/step - accuracy: 0.3726 - loss: 0.0360 - val_accuracy: 0.4197 - val_loss: 0.
0320
Epoch 134/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.3486 - loss: 0.0347 - val_accuracy: 0.4396 - val_loss: 0.
0317
Epoch 135/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.3928 - loss: 0.0347 - val_accuracy: 0.4279 - val_loss: 0.
0317
Epoch 136/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 151ms/step - accuracy: 0.3623 - loss: 0.0380 - val_accuracy: 0.4322 - val_loss: 0.
0319
Epoch 137/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.3767 - loss: 0.0364 - val_accuracy: 0.4239 - val_loss: 0.
0313
Epoch 138/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.3523 - loss: 0.0366 - val_accuracy: 0.4385 - val_loss: 0.
0313
Epoch 139/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.3668 - loss: 0.0359 - val_accuracy: 0.4142 - val_loss: 0.
```

```
0311
Epoch 140/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 122ms/step - accuracy: 0.3789 - loss: 0.0348 - val_accuracy: 0.4375 - val_loss: 0.
0309
Epoch 141/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.3701 - loss: 0.0362 - val_accuracy: 0.4251 - val_loss: 0.
0310
Epoch 142/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 144ms/step - accuracy: 0.3733 - loss: 0.0337 - val_accuracy: 0.4456 - val_loss: 0.
0308
Epoch 143/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 125ms/step - accuracy: 0.3858 - loss: 0.0356 - val_accuracy: 0.4324 - val_loss: 0.
0307
Epoch 144/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.3520 - loss: 0.0355 - val_accuracy: 0.4386 - val_loss: 0.
0305
Epoch 145/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.3971 - loss: 0.0318 - val_accuracy: 0.4036 - val_loss: 0.
0303
Epoch 146/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.3569 - loss: 0.0369 - val_accuracy: 0.4569 - val_loss: 0.
0306
Epoch 147/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.3903 - loss: 0.0336 - val_accuracy: 0.4163 - val_loss: 0.
0301
Epoch 148/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 169ms/step - accuracy: 0.3887 - loss: 0.0349 - val_accuracy: 0.4207 - val_loss: 0.
0300
Epoch 149/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.3721 - loss: 0.0342 - val_accuracy: 0.4305 - val_loss: 0.
0300
Epoch 150/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.3828 - loss: 0.0342 - val_accuracy: 0.4200 - val_loss: 0.
0298
Epoch 151/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.3593 - loss: 0.0352 - val_accuracy: 0.4484 - val_loss: 0.
0298
Epoch 152/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 121ms/step - accuracy: 0.3817 - loss: 0.0356 - val_accuracy: 0.4239 - val_loss: 0.
0296
Epoch 153/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 145ms/step - accuracy: 0.3799 - loss: 0.0357 - val_accuracy: 0.4371 - val_loss: 0.
0294
Epoch 154/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 137ms/step - accuracy: 0.3926 - loss: 0.0332 - val_accuracy: 0.4538 - val_loss: 0.
0296
Epoch 155/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 119ms/step - accuracy: 0.3617 - loss: 0.0348 - val_accuracy: 0.4586 - val_loss: 0.
0293
Epoch 156/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.3950 - loss: 0.0331 - val_accuracy: 0.4243 - val_loss: 0.
0290
Epoch 157/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 120ms/step - accuracy: 0.3861 - loss: 0.0324 - val_accuracy: 0.4554 - val_loss: 0.
0290
Epoch 158/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.4024 - loss: 0.0332 - val_accuracy: 0.4402 - val_loss: 0.
0289
Epoch 159/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 170ms/step - accuracy: 0.4026 - loss: 0.0303 - val_accuracy: 0.4519 - val_loss: 0.
0288
Epoch 160/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.4146 - loss: 0.0319 - val_accuracy: 0.4718 - val_loss: 0.
0287
Epoch 161/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 120ms/step - accuracy: 0.3946 - loss: 0.0333 - val_accuracy: 0.4476 - val_loss: 0.
0287
Epoch 162/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 121ms/step - accuracy: 0.3886 - loss: 0.0335 - val_accuracy: 0.4686 - val_loss: 0.
0285
Epoch 163/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.3986 - loss: 0.0322 - val_accuracy: 0.4369 - val_loss: 0.
0284
Epoch 164/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 126ms/step - accuracy: 0.3963 - loss: 0.0341 - val_accuracy: 0.4644 - val_loss: 0.
0283
Epoch 165/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 157ms/step - accuracy: 0.3825 - loss: 0.0357 - val_accuracy: 0.4513 - val_loss: 0.
0284
Epoch 166/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 133ms/step - accuracy: 0.3905 - loss: 0.0328 - val_accuracy: 0.4376 - val_loss: 0.
0281
Epoch 167/200
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.3976 - loss: 0.0314 - val_accuracy: 0.4542 - val_loss: 0.
0281
Epoch 168/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 124ms/step - accuracy: 0.3887 - loss: 0.0323 - val_accuracy: 0.4603 - val_loss: 0.
0281
Epoch 169/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.3967 - loss: 0.0320 - val_accuracy: 0.4651 - val_loss: 0.
0279
Epoch 170/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 121ms/step - accuracy: 0.3983 - loss: 0.0332 - val_accuracy: 0.4572 - val_loss: 0.
0280
Epoch 171/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 157ms/step - accuracy: 0.4020 - loss: 0.0328 - val_accuracy: 0.4642 - val_loss: 0.
0276
Epoch 172/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 125ms/step - accuracy: 0.3789 - loss: 0.0339 - val_accuracy: 0.4644 - val_loss: 0.
0279
Epoch 173/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 126ms/step - accuracy: 0.3992 - loss: 0.0322 - val_accuracy: 0.4768 - val_loss: 0.
0276
Epoch 174/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.4125 - loss: 0.0325 - val_accuracy: 0.4683 - val_loss: 0.
0276
Epoch 175/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.3951 - loss: 0.0303 - val_accuracy: 0.4877 - val_loss: 0.
0275
Epoch 176/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 128ms/step - accuracy: 0.4088 - loss: 0.0326 - val_accuracy: 0.4850 - val_loss: 0.
0276
Epoch 177/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 151ms/step - accuracy: 0.3910 - loss: 0.0333 - val_accuracy: 0.4763 - val_loss: 0.
0273
Epoch 178/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 125ms/step - accuracy: 0.3945 - loss: 0.0301 - val_accuracy: 0.4662 - val_loss: 0.
0273
Epoch 179/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.4050 - loss: 0.0304 - val_accuracy: 0.4632 - val_loss: 0.
0271
Epoch 180/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 125ms/step - accuracy: 0.3971 - loss: 0.0319 - val_accuracy: 0.4642 - val_loss: 0.
0271
Epoch 181/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.3894 - loss: 0.0319 - val_accuracy: 0.4841 - val_loss: 0.
0271
Epoch 182/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 132ms/step - accuracy: 0.3976 - loss: 0.0321 - val_accuracy: 0.4649 - val_loss: 0.
0268
Epoch 183/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 147ms/step - accuracy: 0.3688 - loss: 0.0327 - val_accuracy: 0.4683 - val_loss: 0.
0269
Epoch 184/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.4187 - loss: 0.0312 - val_accuracy: 0.4711 - val_loss: 0.
0267
Epoch 185/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.3989 - loss: 0.0325 - val_accuracy: 0.4768 - val_loss: 0.
0268
Epoch 186/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.4044 - loss: 0.0302 - val_accuracy: 0.5006 - val_loss: 0.
0273
Epoch 187/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.3641 - loss: 0.0340 - val_accuracy: 0.3915 - val_loss: 0.
0302
Epoch 188/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 141ms/step - accuracy: 0.3799 - loss: 0.0328 - val_accuracy: 0.3981 - val_loss: 0.
0304
Epoch 189/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 173ms/step - accuracy: 0.3903 - loss: 0.0325 - val_accuracy: 0.3643 - val_loss: 0.
0289
Epoch 190/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 124ms/step - accuracy: 0.3959 - loss: 0.0324 - val_accuracy: 0.4357 - val_loss: 0.
0273
Epoch 191/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 126ms/step - accuracy: 0.3883 - loss: 0.0316 - val_accuracy: 0.4509 - val_loss: 0.
0269
Epoch 192/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 126ms/step - accuracy: 0.3793 - loss: 0.0323 - val_accuracy: 0.4712 - val_loss: 0.
0266
Epoch 193/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 122ms/step - accuracy: 0.3785 - loss: 0.0330 - val_accuracy: 0.4671 - val_loss: 0.
0266
Epoch 194/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 145ms/step - accuracy: 0.4141 - loss: 0.0299 - val_accuracy: 0.4615 - val_loss: 0.
0263
```

```
Epoch 195/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 155ms/step - accuracy: 0.3926 - loss: 0.0309 - val_accuracy: 0.4616 - val_loss: 0.
0264
Epoch 196/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 125ms/step - accuracy: 0.4039 - loss: 0.0307 - val_accuracy: 0.4558 - val_loss: 0.
0262
Epoch 197/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 122ms/step - accuracy: 0.4170 - loss: 0.0293 - val_accuracy: 0.4527 - val_loss: 0.
0261
Epoch 198/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 123ms/step - accuracy: 0.4246 - loss: 0.0302 - val_accuracy: 0.4696 - val_loss: 0.
0261
Epoch 199/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 2s 123ms/step - accuracy: 0.4002 - loss: 0.0309 - val_accuracy: 0.4619 - val_loss: 0.
0259
Epoch 200/200
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 162ms/step - accuracy: 0.4124 - loss: 0.0306 - val_accuracy: 0.4602 - val_loss: 0.
0260
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 143ms/step - accuracy: 0.4053 - loss: 0.0376
Test Loss: 0.0376, Test Accuracy: 0.4053
```



In [12]:

```
# Generate Music Sequence
generated_sequence = model.predict(X_test[:1])  # Generate for one test sample

print(generated_sequence)

# Convert Generated Sequence to MIDI
def piano_roll_to_midi(piano_roll, fs=100):
    pm = pretty_midi.PrettyMIDI()
    instrument = pretty_midi.Instrument(program=0)  # Acoustic Grand Piano

    for time, pitch_vector in enumerate(piano_roll):
        for pitch, velocity in enumerate(pitch_vector):
            if velocity > 0:
                note = pretty_midi.Note(
                    velocity=int(velocity * 127),
                    pitch=pitch,
                    start=time / fs,
                    end=(time + 1) / fs
                )
                instrument.notes.append(note)

    pm.instruments.append(instrument)
    return pm

# Convert and save generated MIDI file
generated_midi = piano_roll_to_midi(generated_sequence[0])
generated_midi.write("generated_music.mid")
print("Generated MIDI saved as 'generated_music.mid'")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 12s 12s/step
[[[5.52258243e-05 7.05885614e-05 1.06200554e-04 ... 1.47956089e-04
   1.15975396e-04 6.43615131e-05]
  [5.52258243e-05 7.05885614e-05 1.06200554e-04 ... 1.47956089e-04
   1.15975396e-04 6.43615131e-05]
  [5.52258243e-05 7.05885614e-05 1.06200554e-04 ... 1.47956089e-04
   1.15975396e-04 6.43615131e-05]
  ...
  [7.35896094e-07 1.02950116e-06 1.80438769e-06 ... 2.94909341e-06
   1.98770726e-06 9.09363905e-07]
  [7.35896094e-07 1.02950116e-06 1.80438769e-06 ... 2.94909341e-06
   1.98770726e-06 9.09363905e-07]
  [7.35896094e-07 1.02950116e-06 1.80438769e-06 ... 2.94909341e-06
   1.98770726e-06 9.09363905e-07]]]
Generated MIDI saved as 'generated_music.mid'
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js