In [9]:

```python
import pandas as pd
import numpy as np
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from sklearn.model_selection import train_test_split


train_csv = 'C:\\Users\\Hi\\OneDrive\\Desktop\\mnist_train.csv'
test_csv = 'C:\\Users\\Hi\\OneDrive\\Desktop\\mnist_test.csv'


train_data = pd.read_csv(train_csv)
test_data = pd.read_csv(test_csv)


X_train = train_data.iloc[:, 1:].values
y_train = train_data.iloc[:, 0].values

X_test = test_data.iloc[:, 1:].values
y_test = test_data.iloc[:, 0].values


X_train = X_train.reshape(-1, 28, 28, 1).astype('float32') / 255.0
X_test = X_test.reshape(-1, 28, 28, 1).astype('float32') / 255.0


y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)

print(f'Training data shape: {X_train.shape}')
print(f'Testing data shape: {X_test.shape}')
```

```
Training data shape: (60000, 28, 28, 1)
Testing data shape: (10000, 28, 28, 1)
```

In [11]:

```python
model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')  # 10 output units for digits 0-9
])


model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])


model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10, batch_size=128)


model.save('mnist_digit_recognition_model.h5')

print("Model trained and saved successfully.")
```

```
Epoch 1/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 10s 19ms/step - accuracy: 0.8538 - loss: 0.4963 - val_accuracy: 0.9815 - val_loss: 0.0601
Epoch 2/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 9s 19ms/step - accuracy: 0.9823 - loss: 0.0598 - val_accuracy: 0.9855 - val_loss: 0.0421
Epoch 3/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 10s 21ms/step - accuracy: 0.9876 - loss: 0.0414 - val_accuracy: 0.9884 - val_loss: 0.0378
Epoch 4/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 9s 19ms/step - accuracy: 0.9910 - loss: 0.0307 - val_accuracy: 0.9871 - val_loss: 0.0385
Epoch 5/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 9s 19ms/step - accuracy: 0.9919 - loss: 0.0249 - val_accuracy: 0.9896 - val_loss: 0.0302
Epoch 6/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 9s 19ms/step - accuracy: 0.9936 - loss: 0.0200 - val_accuracy: 0.9900 - val_loss: 0.0288
Epoch 7/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 9s 19ms/step - accuracy: 0.9954 - loss: 0.0154 - val_accuracy: 0.9899 - val_loss: 0.0311
Epoch 8/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 8s 17ms/step - accuracy: 0.9961 - loss: 0.0125 - val_accuracy: 0.9896 - val_loss: 0.0309
Epoch 9/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 8s 17ms/step - accuracy: 0.9967 - loss: 0.0109 - val_accuracy: 0.9910 - val_loss: 0.0281
Epoch 10/10
469/469 ━━━━━━━━━━━━━━━━━━━━ 8s 17ms/step - accuracy: 0.9976 - loss: 0.0073 - val_accuracy: 0.9921 - val_loss: 0.0278
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legac
y. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
Model trained and saved successfully.
```

In [12]:

```python
_, test_accuracy = model.evaluate(X_test, y_test, verbose=0)


print(f'Test accuracy: {test_accuracy * 100:.2f}%')
```

```
Test accuracy: 99.21%
```

In [16]:

```python
from tensorflow.keras.models import load_model
import numpy as np


model = load_model(r"C:\Users\Hi\mnist_digit_recognition_model.h5")


sample = X_test[0].reshape(1, 28, 28, 1)
prediction = model.predict(sample)


predicted_digit = np.argmax(prediction)
actual_digit = np.argmax(y_test[0])

print(f'Predicted digit: {predicted_digit}, Actual digit: {actual_digit}')
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or eva
luate the model.
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
Predicted digit: 7, Actual digit: 7
```

In [ ]:

```python
```