import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.datasets import mnist

from tensorflow.keras.utils import to\_categorical

# Load and preprocess the data

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

# Reshape the data to add a channel dimension and normalize

x\_train = x\_train.reshape((x\_train.shape[0], 28, 28, 1)).astype('float32') / 255

x\_test = x\_test.reshape((x\_test.shape[0], 28, 28, 1)).astype('float32') / 255

# One-hot encode the labels

y\_train = to\_categorical(y\_train)

y\_test = to\_categorical(y\_test)

# Build the CNN model

model = models.Sequential([

layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.MaxPooling2D((2, 2)),

layers.Flatten(),

layers.Dense(64, activation='relu'),

layers.Dense(10, activation='softmax') # 10 classes for digits 0-9

])

# Compile the model

model.compile(optimizer='adam',

loss='categorical\_crossentropy',

metrics=['accuracy'])

# Train the model

model.fit(x\_train, y\_train, epochs=5, batch\_size=64, validation\_split=0.1)

# Evaluate the model

test\_loss, test\_acc = model.evaluate(x\_test, y\_test)

print(f"Test accuracy: {test\_acc:.4f}")