**PROJECT TITLE : RECOGNIZING HANDWRITTEN DIGITS USING MACHINE LEARNING**

**PROGRAM:**

**# Import required libraries**

**import numpy as np**

**import tensorflow as tf**

**from tensorflow.keras.datasets import mnist**

**from tensorflow.keras.models import Sequential**

**from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout**

**from tensorflow.keras.utils import to\_categorical**

**# Enable memory growth for GPU (optional)**

**physical\_devices = tf.config.list\_physical\_devices('GPU')**

**if physical\_devices:**

**try:**

**tf.config.set\_memory\_growth(physical\_devices[0], True) # Updated API**

**except RuntimeError as e:**

**print(f"GPU memory growth setting failed: {e}")**

**# Load the MNIST dataset**

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

**# Reshape and normalize the input data**

**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**

**# Convert class labels to one-hot encoding**

**y\_train = to\_categorical(y\_train, 10)**

**y\_test = to\_categorical(y\_test, 10)**

**# Build the CNN model**

**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'),**

**Dropout(0.5),**

**Dense(10, activation='softmax') # 10 classes for digits 0–9**

**])**

**# Compile the model**

**model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])**

**# Print model summary**

**model.summary()**

**# Train the model**

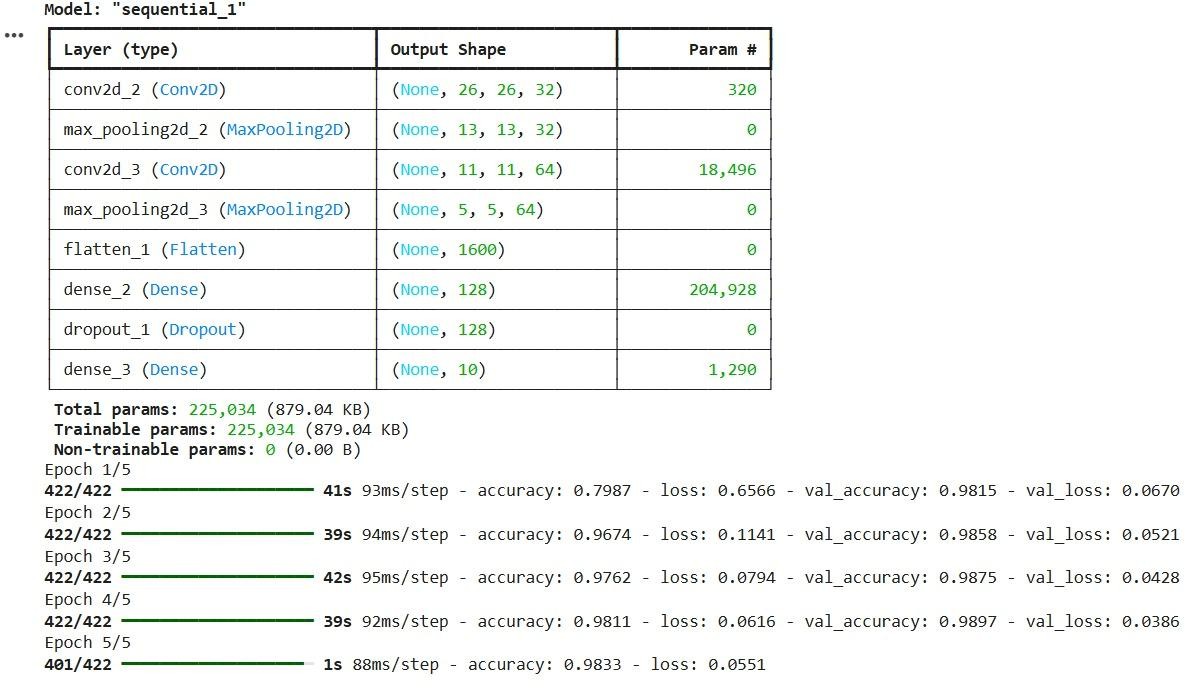
**model.fit(X\_train, y\_train, batch\_size=128, epochs=5, validation\_split=0.1)**

**# Evaluate the model**

**loss, accuracy = model.evaluate(X\_test, y\_test)**

**print("\nTest Accuracy: {:.4f}".format(accuracy))**

**output:**

****