Name: Suchit Jundare Roll no.:15 BE: AI & DS Experiment No.8 DL

Code:

import os

import numpy as np

import matplotlib.pyplot as plt import tensorflow as tf

from tensorflow.keras import layers, models from tensorflow.keras.preprocessing import image

(x\_train, y\_train), (x\_test, y\_test) = tf.keras.datasets.mnist.load\_data() x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

x\_train = x\_train.reshape(-1, 28, 28, 1)

x\_test = x\_test.reshape(-1, 28, 28, 1)

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')

])

model.compile(optimizer='adam',

loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

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

test\_loss, test\_acc = model.evaluate(x\_test, y\_test) print(f"\nTest accuracy: {test\_acc:.4f}")

plt.figure(figsize=(12,4)) plt.subplot(1,2,1)

plt.plot(history.history['accuracy'], label='Train') plt.plot(history.history['val\_accuracy'], label='Validation') plt.title('Accuracy')

plt.xlabel('Epoch') plt.ylabel('Accuracy') plt.legend()

plt.subplot(1,2,2) plt.plot(history.history['loss'], label='Train')

plt.plot(history.history['val\_loss'], label='Validation') plt.title('Loss')

plt.xlabel('Epoch') plt.ylabel('Loss') plt.legend() plt.show()

def load\_custom\_image(img\_path):

img = image.load\_img(img\_path, color\_mode='grayscale', target\_size=(28, 28))

img\_array = image.img\_to\_array(img) img\_array = 255 - img\_array img\_array = img\_array / 255.0

img\_array = img\_array.reshape(1, 28, 28, 1) return img\_array

custom\_folder = 'custom\_images'

print(f"\nPredicting digits in folder: '{custom\_folder}'\n")

for filename in os.listdir(custom\_folder):

if filename.lower().endswith(('.png', '.jpg', '.jpeg')): path = os.path.join(custom\_folder, filename)

img = load\_custom\_image(path) pred = model.predict(img) label = np.argmax(pred)

plt.imshow(img.reshape(28,28), cmap='gray') plt.title(f"Prediction: {label} ({filename})") plt.axis('off')

plt.show()

Epoch 1/5

**1688/1688** ━━━━━━━━━━━━━━━━━━━━ **55s** 32ms/step - accuracy: 0.8951 - loss: 0.3409 - val\_accuracy: 0.9832 - val\_loss: 0.0583

Epoch 2/5

**1688/1688** ━━━━━━━━━━━━━━━━━━━━ **80s** 31ms/step - accuracy: 0.9830 - loss: 0.0549 - val\_accuracy: 0.9895 - val\_loss: 0.0361

Epoch 3/5

**1688/1688** ━━━━━━━━━━━━━━━━━━━━ **57s** 34ms/step - accuracy: 0.9890 - loss: 0.0358 - val\_accuracy: 0.9880 - val\_loss: 0.0436

Epoch 4/5

**1688/1688** ━━━━━━━━━━━━━━━━━━━━ **78s** 31ms/step - accuracy: 0.9921 - loss: 0.0266 - val\_accuracy: 0.9898 - val\_loss: 0.0348

Epoch 5/5

**1688/1688** ━━━━━━━━━━━━━━━━━━━━ **52s** 31ms/step - accuracy: 0.9940 - loss: 0.0184 - val\_accuracy: 0.9903 - val\_loss: 0.0346

**313/313** ━━━━━━━━━━━━━━━━━━━━ **4s** 13ms/step - accuracy: 0.9863

- loss: 0.0430

Test accuracy: 0.9885

