# Created by J.TAKESHWAR

# Explanation and Instructions

## Code 1: Dataset Generation

### Overview

The first script generates a dataset of handwritten-style digit images (0-9) using the Python Imaging Library (PIL). It creates 500 samples for each digit and saves them in a structured directory.

### Steps in the Code

- Import Required Libraries: The script imports necessary libraries like numpy, matplotlib.pyplot, and PIL for image processing.

- Set Parameters: Defines image size, font size, and the number of images per digit.

- Create Directory Structure: A folder /mnt/data/generated\_mnist is created to store images.

- Generate Images:  
- A 28x28 grayscale image is created with a white background.  
- The digit (0-9) is drawn in black at a fixed position.  
- Each image is saved under /mnt/data/generated\_mnist/{digit}/.

- Visualize Some Samples: Displays one sample from each digit using matplotlib.

### Running the Code

Save the script as generate\_mnist.py and execute:

python generate\_mnist.py

This will generate and store images in /mnt/data/generated\_mnist.

## Code 2: Training a CNN Model

### Overview

The second script loads the generated dataset, preprocesses images, and trains a Convolutional Neural Network (CNN) for digit classification.

### Steps in the Code

- Import Required Libraries: TensorFlow and Keras are used for building and training the model.

- Set Parameters: Defines image size, batch size, number of classes, and training epochs.

- Load Dataset Using ImageDataGenerator:  
- Uses ImageDataGenerator to load images and apply preprocessing (rescaling and validation split).  
- Images are loaded in grayscale mode and categorized into 10 classes.

- Define CNN Model:  
- 3 convolutional layers with ReLU activation and max pooling.  
- Flatten layer followed by two dense layers (one hidden, one output).

- Compile and Train Model:  
- Optimizer: Adam  
- Loss Function: Categorical Crossentropy  
- Trains for 5 epochs.

- Save the Model: The trained model is saved as /mnt/data/cnn\_mnist\_model.h5.

### Running the Code

Save the script as train\_cnn.py and execute:

python train\_cnn.py

This will train the CNN model and save it for future use.

## Summary

- Code 1 generates a dataset of digits.

- Code 2 trains a CNN model using the generated dataset.

- The final model is saved and can be used for digit classification tasks.

This concludes the explanation of the scripts.