**ASSESMENT-4**

**PROBLEM STATEMENT-** Image classification on MNIST dataset (CNN model with Fully connected layer)

**SOURCE CODE:**

* Open the folder in the Spyder IDE
* The Command to install

**“pip install tensorflow”**

**File Name: program4.py**

import tensorflow as tf

from tensorflow.keras import layers, models, datasets

# Load and preprocess MNIST dataset

(train\_images, train\_labels), (test\_images, test\_labels) = datasets.mnist.load\_data()

train\_images = train\_images.reshape((60000, 28, 28, 1)).astype('float32') / 255

test\_images = test\_images.reshape((10000, 28, 28, 1)).astype('float32') / 255

# Define CNN architecture

model = models.Sequential([

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

    layers.MaxPooling2D((2, 2)),

    layers.Flatten(),

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

    layers.Dense(10, activation='softmax')

])

# Compile the model

model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

# Train the model

model.fit(train\_images, train\_labels, epochs=5, batch\_size=64, validation\_data=(test\_images, test\_labels))

# Evaluate the model

test\_loss, test\_acc = model.evaluate(test\_images, test\_labels)

print("\nTest accuracy:", test\_acc)

**Explanation:**

Here's a breakdown of the code:

1. **Load and Preprocess Data:**
   * It loads the MNIST dataset using TensorFlow's datasets module.
   * Preprocesses the images by reshaping them to the shape required by the CNN model and scaling pixel values to the range [0, 1].
2. **Define CNN Architecture:**
   * It defines a CNN architecture using the Sequential API in Keras.
   * The architecture consists of a convolutional layer with ReLU activation, followed by max-pooling layer, flattening layer, and two fully connected dense layers with ReLU and softmax activations, respectively.
3. **Compile the Model:**
   * It compiles the model using the Adam optimizer and sparse categorical cross-entropy loss function.
4. **Train the Model:**
   * It trains the model on the training data for 5 epochs with a batch size of 64.
5. **Evaluate the Model:**
   * It evaluates the trained model on the test data and prints the test accuracy.

**OUTPUT:**

****