**Name: Gaurang Vaghela**

**Rollno: TEAD-22561**

**ANN (SL-II)**

**Practical 12  
Problem Statement:** TensorFlow/Pytorch implementation of CNN

**Code:**

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

# Load MNIST dataset

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

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0 # Normalize

# Reshape for CNN (adding channel dimension)

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

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

# Define CNN model

model = keras.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.Conv2D(64, (3, 3), activation='relu'),

layers.Flatten(),

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

layers.Dense(10, activation='softmax') # Output layer for 10 classes

])

# Compile model

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

# Train model

model.fit(x\_train, y\_train, epochs=5, validation\_data=(x\_test, y\_test))

# Evaluate model

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

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

**Output:**

