Experiment Number : 6

Problem Statement: **Write Python code to apply Convolutional Neural Network for the given data set.**

NAME: Harshwardhan PatilROLLNO: 50

CLASS: IT-B BATCH: B1

**Code:**

**import tensorflow as tf**

**from tensorflow import keras**

**from tensorflow.keras import layers**

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

**import matplotlib.pyplot as plt**

**# Load dataset (MNIST as an example)**

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

**# Normalize and reshape data**

**x\_train = x\_train.reshape(-1, 28, 28, 1) / 255.0**

**x\_test = x\_test.reshape(-1, 28, 28, 1) / 255.0**

**# Define the 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')**

**])**

**# Compile the model**

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

**# Train the model**

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

**# Evaluate the model**

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

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

**# Plot training history**

**plt.plot(history.history['accuracy'], label='Training Accuracy')**

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

**plt.xlabel('Epochs')**

**plt.ylabel('Accuracy')**

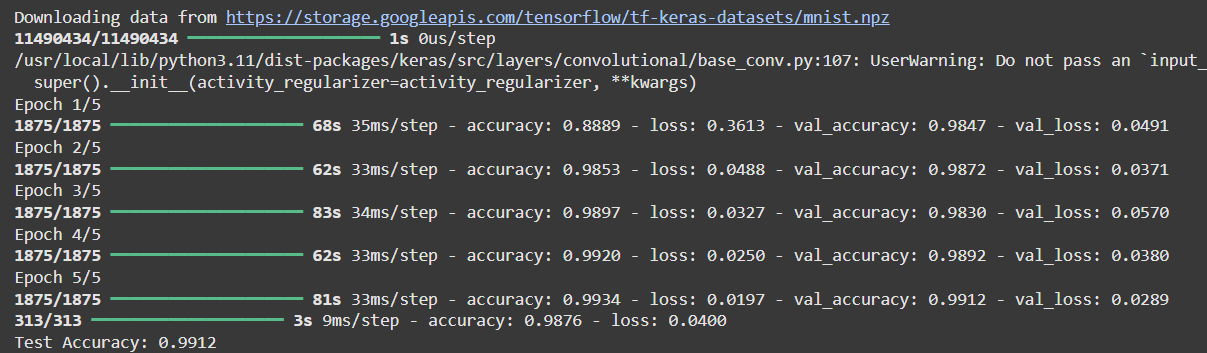
**plt.legend()**

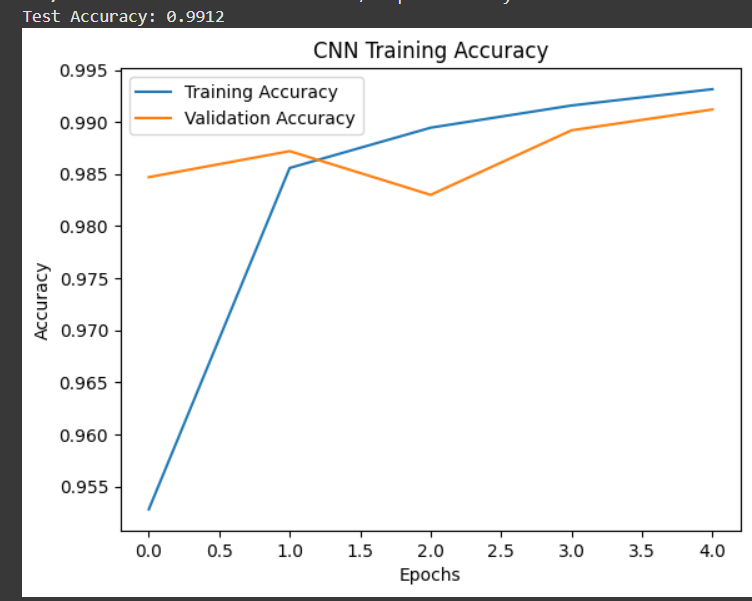
**plt.title('CNN Training Accuracy')**

**plt.show()**

**Output:**

**Accuracy=0.99%**

****

****