|  |  |
| --- | --- |
| Semester | B.E. Semester VII |
| Subject | Deep Learning |
| Subject Professor In-charge | Dr. Nayana Mahajan |
| Laboratory | M201B |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student Name | Harsh Jian | Division | | B |
| Roll Number | 22108B0054 | Batch | | 4 |
| Grade and Subject Teacher’s Signature |  | |  | |

|  |  |
| --- | --- |
| Experiment Number | 4 |
| Experiment Title |  |
| Resources / Apparatus Required | Software: |
| Algorithm |  |
| Program code | import tensorflow as tf  from tensorflow.keras import datasets, layers, models  import matplotlib.pyplot as plt  (x\_train, y\_train), (x\_test, y\_test) = datasets.mnist.load\_data()  x\_train, x\_test = x\_train / 255.0, x\_test / 255.0  x\_train, x\_test = x\_train[..., tf.newaxis], x\_test[..., tf.newaxis]  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\_data=(x\_test, y\_test), verbose=1)  test\_loss, test\_acc = model.evaluate(x\_test, y\_test, verbose=0)  print(f"Test Accuracy: {test\_acc:.4f}")  plt.figure(figsize=(12,5))  plt.subplot(1,2,1)  plt.plot(history.history['accuracy'], label='train acc')  plt.plot(history.history['val\_accuracy'], label='val acc')  plt.xlabel('Epochs'); plt.ylabel('Accuracy'); plt.legend(); plt.title('Accuracy')  plt.subplot(1,2,2)  plt.plot(history.history['loss'], label='train loss')  plt.plot(history.history['val\_loss'], label='val loss')  plt.xlabel('Epochs'); plt.ylabel('Loss'); plt.legend(); plt.title('Loss')  plt.show() |
| Output |  |
|  |  |
| Conclusion |  |