**Experiment-No.9**

**Objective:**to Implement pattern recognition problems of handwritten character/ digit recognition

|  |  |  |
| --- | --- | --- |
| **Scheduled Date:** | **Compiled Date:** | **Submitted Date:** |
| 20Nov 2024 | 20 Nov 2024 | 27 Nov2024 |
|  |  |  |

pip install tensorflow

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Flatten

from tensorflow.keras.utils import to\_categorical

# Load the MNIST dataset

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

# Preprocess the data

x\_train = x\_train.astype('float32') / 255.0  # Normalize the images to [0, 1]

x\_test = x\_test.astype('float32') / 255.0

y\_train = to\_categorical(y\_train, num\_classes=10)  # One-hot encoding

y\_test = to\_categorical(y\_test, num\_classes=10)

# Build the model

model = Sequential()

model.add(Flatten(input\_shape=(28, 28)))  # Flatten the input images

model.add(Dense(128, activation='relu'))   # Hidden layer

model.add(Dense(10, activation='softmax'))  # Output layer

# Compile the model

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

# Train the model

model.fit(x\_train, y\_train, epochs=5, batch\_size=32, validation\_split=0.1)

# Evaluate the model

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

print(f"Test accuracy: {test\_accuracy:.4f}")

# Predict on a sample from the test set

sample\_index = 0

sample\_image = x\_test[sample\_index].reshape(1, 28, 28)  # Reshape for prediction

predicted\_class = np.argmax(model.predict(sample\_image), axis=1)

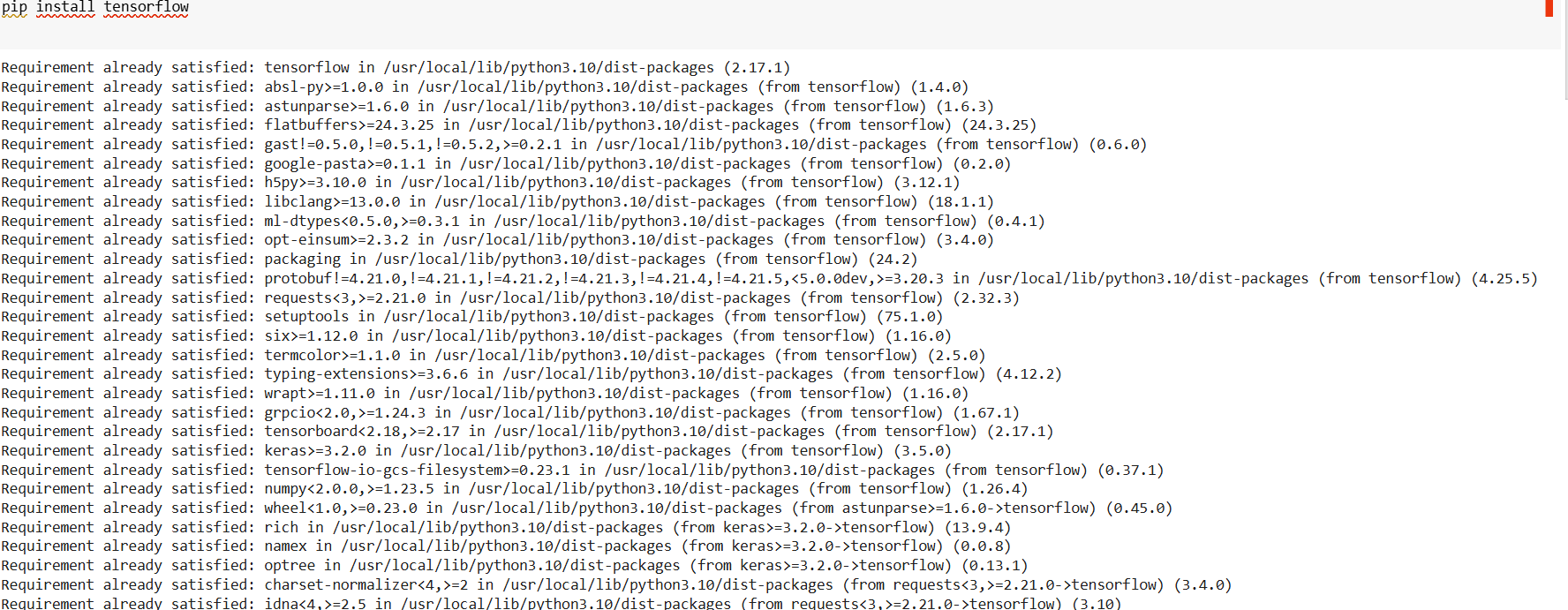
# Display the sample image and predicted class

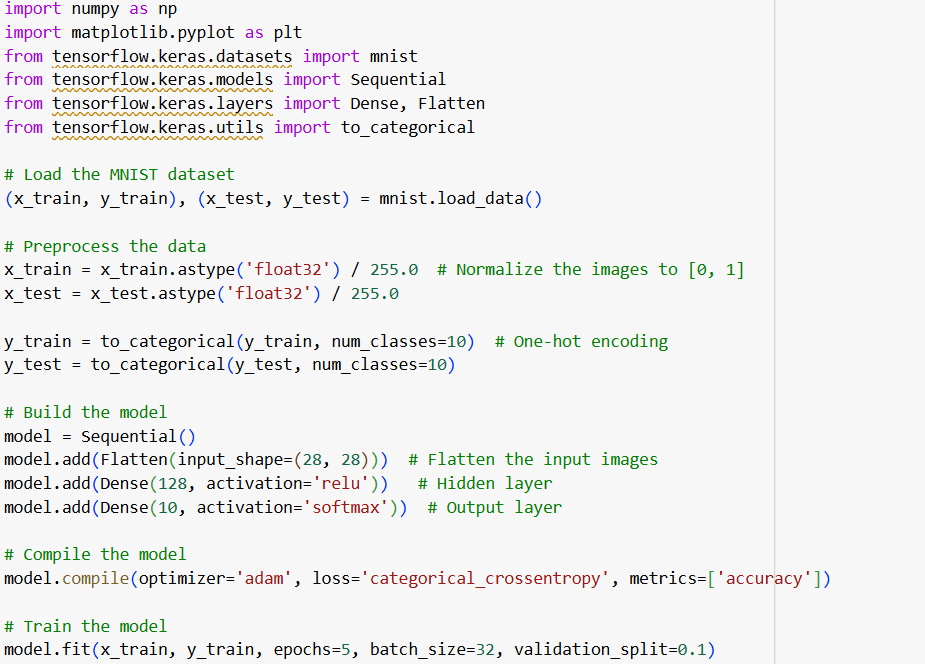
plt.imshow(x\_test[sample\_index], cmap='gray')

plt.title(f"Predicted Class: {predicted\_class[0]}")

plt.axis('off')

plt.show()

****

****

