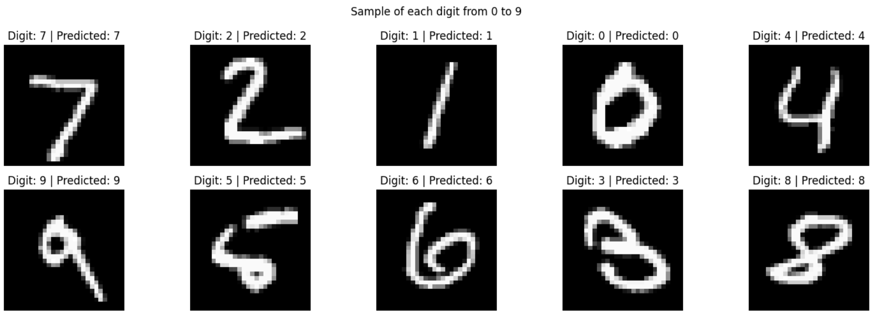
```python
# 📦 Importing required libraries
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist

# 📥 Load the MNIST dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# 🔄 Normalize the images
x_train = x_train / 255.0
x_test = x_test / 255.0

# 🧩 Reshape to fit CNN input
x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)

# 🧱 Build the CNN model
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    layers.MaxPooling2D(pool_size=(2, 2)),

    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),

    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')  # 10 classes for digits 0-9
])

# ⚙ Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# 🏋️ Train the model
model.fit(x_train, y_train, epochs=5, validation_data=(x_test, y_test))

# 📈 Evaluate the model
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print(f"\n🎯 Test Accuracy: {test_accuracy:.4f}")

# 💾 Save the model
model.save("digit_recognition_model.h5")

# 🔮 Predict on test data
predictions = model.predict(x_test)

# 🎨 Display one sample of each digit (0-9)
shown_digits = set()
plt.figure(figsize=(15, 5))

i = 0
count = 0
```

```
while len(shown_digits) < 10 and i < len(x_test):
    label = y_test[i]
    if label not in shown_digits:
        plt.subplot(2, 5, count + 1)
        plt.imshow(x_test[i].reshape(28, 28), cmap='gray')
        plt.title(f"Digit: {label} | Predicted: {np.argmax(predictions[i])}")
        plt.axis('off')
        shown_digits.add(label)
        count += 1
    i += 1

plt.suptitle("Sample of each digit from 0 to 9")
plt.tight_layout()
plt.show()
```

```
Epoch 1/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 52s 27ms/step - accuracy: 0.9034 - loss: 0.3117 - val_a
Epoch 2/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 83s 28ms/step - accuracy: 0.9867 - loss: 0.0451 - val_a
Epoch 3/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 82s 27ms/step - accuracy: 0.9911 - loss: 0.0277 - val_a
Epoch 4/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 49s 26ms/step - accuracy: 0.9929 - loss: 0.0226 - val_a
Epoch 5/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 49s 26ms/step - accuracy: 0.9956 - loss: 0.0143 - val_a
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 13ms/step - accuracy: 0.9889 - loss: 0.0359
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.sa

🎯 Test Accuracy: 0.9913
313/313 ━━━━━━━━━━━━━━━━━━━━ 3s 8ms/step
```


Sample of each digit from 0 to 9
Digit: 7 | Predicted: 7
Digit: 2 | Predicted: 2
Digit: 1 | Predicted: 1
Digit: 0 | Predicted: 0
Digit: 4 | Predicted: 4
Digit: 9 | Predicted: 9
Digit: 5 | Predicted: 5
Digit: 6 | Predicted: 6
Digit: 3 | Predicted: 3
Digit: 8 | Predicted: 8