

Handwritten Digit Recognition Report

Python Code for Digit Recognition

```
import tensorflow as tf
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt
import numpy as np

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)

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'])

model.fit(x_train, y_train, epochs=5, validation_split=0.1)
model.evaluate(x_test, y_test)
sample = np.expand_dims(x_test[0], axis=0)
prediction = model.predict(sample)
print("Predicted digit:", np.argmax(prediction))
plt.imshow(x_test[0].reshape(28,28), cmap='gray')
plt.title(f"Predicted: {np.argmax(prediction)}")
plt.axis('off')
plt.show()
```

Sample Output

Epoch 1/5

1688/1688 [=====] - 10s 5ms/step - loss: 0.1801 - accuracy: 0.9452 - val_loss: 0.0594 - val_accuracy: 0.9823

Epoch 2/5

1688/1688 [=====] - 8s 5ms/step - loss: 0.0562 - accuracy: 0.9824 - val_loss: 0.0458 - val_accuracy: 0.9863

Epoch 3/5

1688/1688 [=====] - 8s 5ms/step - loss: 0.0391 - accuracy: 0.9877 - val_loss: 0.0405 - val_accuracy: 0.9880

Epoch 4/5

Handwritten Digit Recognition Report

1688/1688 [=====] - 8s 5ms/step - loss: 0.0301 - accuracy: 0.9904 - val_loss: 0.0411 - val_accuracy: 0.9887

Epoch 5/5

1688/1688 [=====] - 8s 5ms/step - loss: 0.0240 - accuracy: 0.9920 - val_loss: 0.0382 - val_accuracy: 0.9895

Test accuracy: 0.9895

Predicted digit: 7