TITLE: Recognizing handwritten digits with deep learning for smarter AI applications

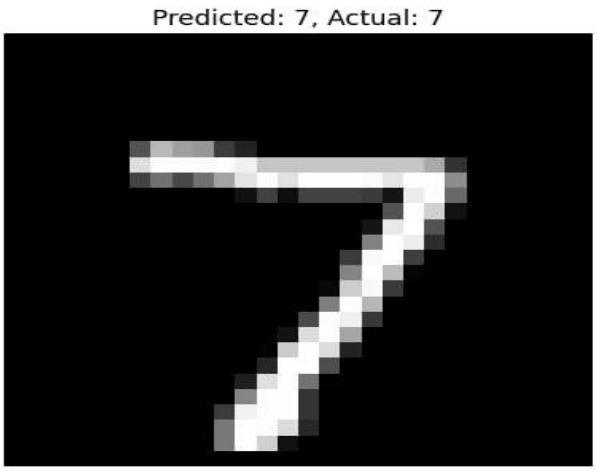
PROGRAM:

```
import tensorflow as tf

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D

import matplotlib.pyplot as plt

# Load MNIST dataset

(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Normalize pixel values (0-255 to 0-1)

x_train = x_train / 255.0

x_test = x_test / 255.0

# Reshape data to fit CNN input (batch, height, width, channels)

x_train = x_train.reshape(-1, 28, 28, 1)

x_test = x_test.reshape(-1, 28, 28, 1)

# Build CNN model

model = Sequential([

    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28, 28, 1)),

    MaxPooling2D(pool_size=(2, 2)),

    Conv2D(64, kernel_size=(3, 3), activation='relu'),

    MaxPooling2D(pool_size=(2, 2)),

    Flatten(),

    Dense(128, activation='relu'),

    Dense(10, activation='softmax')
```

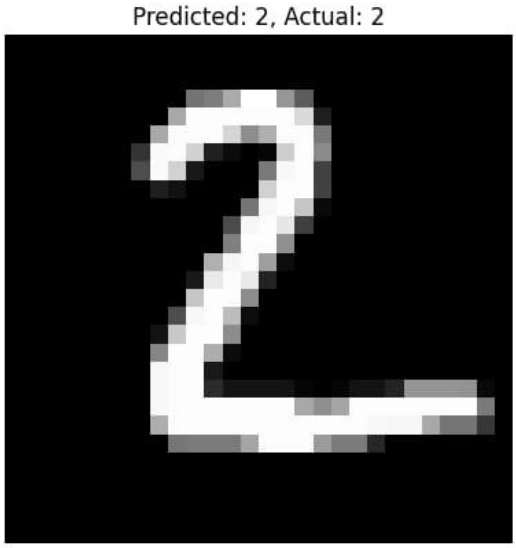
```
])

# 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 on test data

test_loss, test_acc = model.evaluate(x_test, y_test)

print(f'Test accuracy: {test_acc:.4f}')

# Predicting on test images

predictions = model.predict(x_test)

# Plot some predictions

for i in range(5):

    plt.imshow(x_test[i].reshape(28,28), cmap='gray')

    plt.title(f"Predicted: {tf.argmax(predictions[i])}, Actual: {y_test[i]}")

    plt.axis('off')

    plt.show()
```

OUTPUT:

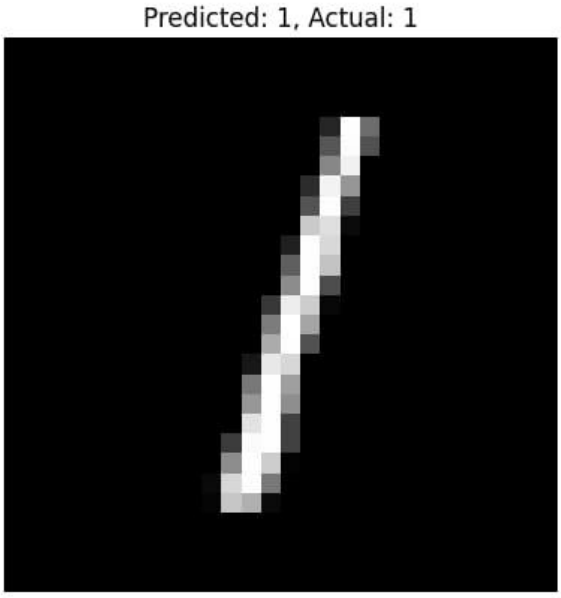
Prediction 1:



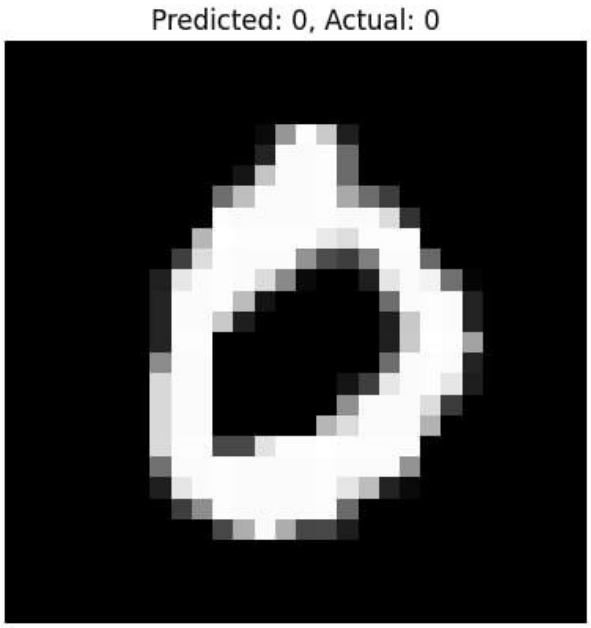
Prediction 2:



Prediction 3:



Prediction 4:



Prediction 5: