```python
import os
import cv2
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

print("Welcome to the hand sign recognition")

# Decide if to load an existing model or to train a new one
train_new_model = True

if train_new_model:
    # Loading the MNIST data set with samples and splitting it
    mnist = tf.keras.datasets.mnist
    (X_train, y_train), (X_test, y_test) = mnist.load_data()

    # Normalizing the data (making length = 1)
    X_train = tf.keras.utils.normalize(X_train, axis=1)
    X_test = tf.keras.utils.normalize(X_test, axis=1)

    # Create a neural network model
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Flatten(input_shape=(28, 28)))  # Specify input shape
    model.add(tf.keras.layers.Dense(units=128, activation=tf.nn.relu))
    model.add(tf.keras.layers.Dense(units=128, activation=tf.nn.relu))
    model.add(tf.keras.layers.Dense(units=10, activation=tf.nn.softmax))

    # Compiling and optimizing model
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

    # Training the model
    model.fit(X_train, y_train, epochs=3)

    # Evaluating the model
    val_loss, val_acc = model.evaluate(X_test, y_test)
    print(val_loss)
    print(val_acc)

    # Saving the model
    model.save('handwritten_digits_model.h5')  # Save as a file
else:
    # Load the model
    model = tf.keras.models.load_model('handwritten_digits_model.h5')
```

```
    # Saving the model
    model.save('handwritten_digits_model.h5')  # Save as a file
else:
    # Load the model
    model = tf.keras.models.load_model('handwritten_digits_model.h5')

# Load custom images and predict them
image_number = 1
while os.path.isfile('digits/digit{}.png'.format(image_number)):
    try:
        img = cv2.imread('digits/digit{}.png'.format(image_number))[:, :, 0]
        img = np.invert(np.array([img]))
        img = img / 255.0  # Normalize the image
        prediction = model.predict(img)
        print("The number is probably a {}".format(np.argmax(prediction)))
        plt.imshow(img[0], cmap=plt.cm.binary)
        plt.show()
        image_number += 1
    except Exception as e:
        print(f"Error reading image! Proceeding with next image... {e}")
        image_number += 1
```

handwritten_digits_recognition.py M

handwritten_digits_recognition.py > ...

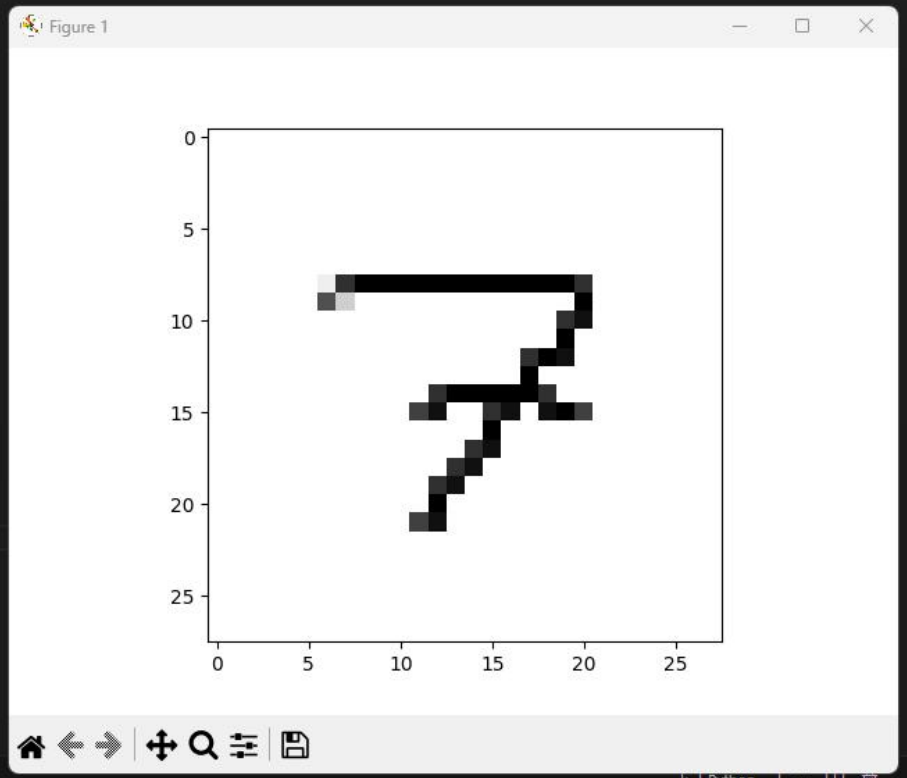
```python

    # Evaluating the model
    val_loss, val_acc = model.evaluate(X_test, y_test)
    print(val_loss)
    print(val_acc)

    # Saving the model
    model.save('handwritten_digits_model.h5')  # Save as a file
else:
    # Load the model
    model = tf.keras.models.load_model('handwritten_digits_model.h5')

# Load custom images and predict them
image_number = 1
while os.path.isfile('digits/digit{}.png'.format(image_number)):
    try:
        img = cv2.imread('digits/digit{}.png'.format(image_number))[:, :, 0]
        img = np.invert(np.array([img]))
        img = img / 255.0  # Normalize the image
        prediction = model.predict(img)
        print("The number is probably a {}".format(np.argmax(prediction)))
        plt.imshow(img[0], cmap=plt.cm.binary)
        plt.show()
        image_number += 1
    except Exception as e:
        print(f"Error reading image! Proceeding with next image... {e}")
        image_number += 1

```



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
The number is probably a 7
```