CODE

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import os
# from IPython.display import Image
import easyocr


def minist_traning():
"""
Does the traning from the MNIST dataset
:return: None
"""
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print(x_train.shape)
x_train = tf.keras.utils.normalize(x_train, axis=1)
x_test = tf.keras.utils.normalize(x_test, axis=1)
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten(input_shape=(28, 28)))
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))
model.compile(optimizer='adam',              loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
model.fit(x_train, y_train, epochs=3)
accuracy, loss = model.evaluate(x_test, y_test)
print(accuracy)
print(loss)
model.save('digits.model')
return model
```

```
def predict_using_trained():
"""
Takes the images that are present in the trained folder and uses the mnist traning to predict the numbers
:return: None
"""
model = minist_traning()
cur_wd = os.getcwd()
path = cur_wd + "\\" + "Trained"
print(path)
imgs = os.listdir(path)
for cur_img in range(1,len(imgs)+1):
os.chdir(path)
img = cv.imread(f'{cur_img}.png')[:,:,0]
img = np.invert(np.array([img]))
prediction = model.predict(img)
print(f'The result is probably:{np.argmax(prediction)}')
plt.imshow(img[0], cmap=plt.cm.binary)
plt.show()
os.chdir('..')


def easy_ocr():
"""
This module uses the easy ocr library to predict the output
:return: None
"""
reader = easyocr.Reader(['en','hi'])
cur_wd = os.getcwd()
path = cur_wd + "\\" + "EasyOCR"
imgs = os.listdir(path)
for cur_img in range(1,len(imgs)+1):
os.chdir(path)
pre_img = cur_img
img = cv.imread(f'{cur_img}.png')[:, :, 0]
img = np.invert(np.array([img]))
plt.imshow(img[0], cmap=plt.cm.binary)
```

```
plt.show()
output = reader.readtext(f'{cur_img}.png')
tup1 = output[0]
print(f" Predicted Words / characters are : {tup1[1]} ", end=" ")
print(f" Accuracy: {tup1[2]*100} ",end= "\n")
print(output)
os.chdir('..')


if __name__=="__main__":
print("\n Executing the pre trained function \n")
predict_using_trained()
print("\n Executing the easy ocr function \n")
easy_ocr()
```