#請準備好模型檔案

import tkinter as tk #Py圖形介面

import pyscreenshot as ImageGrab

import cv2

import matplotlib.pyplot as plt

import tensorflow as tf

import numpy as np

model=tf.keras.models.load\_model('CNNMnist.h5') #之前記得模型存檔的位置

fileName="test.jpg" #未來圖片存檔

#設定小畫家視窗

width = 280

height = 280

white = (255, 255, 255)

red = (255, 0, 0)

image1 = None

#滑鼠畫圖

def paint(event):

x1, y1 = (event.x + 1), (event.y + 1)

x2, y2 = (event.x - 1), (event.y - 1)

canvas1.create\_oval(x1, y1, x2, y2, fill="black", width=15) # On tkinter Canvas

#清除畫面

def clear ():

# Clear the SEEN canvas

canvas1.delete('all')

#存檔

def predict():

#存檔

x=root.winfo\_rootx()

y=root.winfo\_rooty()

x1=x+canvas1.winfo\_width()

y1=y+canvas1.winfo\_height()

ImageGrab.grab().crop((x,y,x1,y1)).save(fileName)

#辨識

img = cv2.resize(cv2.imread(fileName),(28,28))

pixel = (255-cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)) /255 # RGB->GRAY

fig = plt.gcf()

fig.set\_size\_inches(2,2)

plt.imshow(pixel, cmap='binary') # cmap='binary' 參數設定以黑白灰階顯示.

plt.show() #預覽寫字的結果

pixelarray=pixel.reshape(-1,28,28,1) #轉成輸入陣列<-CNN， -1-->自動分配

#pixelarray=np.asarray([pixel]) #轉成輸入陣列<-傳統類神經

label=model.predict(pixelarray) # 用MNIST模型來預測

maxindex = np.argmax(label)#找出0,1,2,...9，機率最大的輸出

#print(label) #顯示機率矩陣

if label.max()>0.5:

showtext="辨識結果=" + str(maxindex) + ", 機率=" + str(label[0][maxindex])

print(showtext)

textValue.set(showtext)

else:

print("無法辨識")

# 建立GUI視窗

root = tk.Tk()

# 建立畫布cv

canvas1 = tk.Canvas(root, width=width, height=height, bg='white')

canvas1.pack()

canvas1.bind("<B1-Motion>", paint) #設定滑鼠按下為繪圖

#建立辨識按鈕及清除按鈕

textValue = tk.StringVar()

textValue.set('')

label1=tk.Label(textvariable=textValue).pack()

button=tk.Button(text="辨識", command=predict).pack()

button=tk.Button(text="清除", command=clear).pack()

root.mainloop()