

# 手寫數字辨識 Digit-Recognizer

敘述，基於上次使用 CNN 架構訓練 Mnist 資料集結果，並實作一個畫板實時的將使用者畫出來的手寫數字圖進行辨識。

## #環境

```
-pip install keras opencv-python tensorflow
```

透過上禮拜的訓練，儲存模型訓練結果(儲存格式.h5)，並保留至 local。

```
#model.save()
```

因 colab 並不支援彈出視窗(畫板)，請同學在 local 端使用 jupyter notebook 或者其他平台進行操作，並安裝 keras、opencv-python 以及 tensorflow 套件。

## 1. 環境設定及函示庫

- 首先，導入必要的 Python 函示庫：`cv2` 用於計算機視覺操作，`numpy` 用於數據處理。此外，從 `keras.models` 導入 `load_model` 來加載預訓練的模型。

## 2. 加載模型

- 使用 `load_model` 函數加載一個預訓練的神經網絡模型，這裡以 `'mnist_model.h5'` 為例，這是一個針對 MNIST 數字識別訓練的模型。

## 3. 初始化圖像和變數

- 創建一個 256x256 像素的全黑圖像 `img` 作為畫板，其中每個像素值初始化為 0 (代表黑色)。
- 定義 `drawing` 變數為 `False`，用來表示是否開始繪圖。
- `(ix, iy)` 初始化為 `(-1, -1)`，用來存儲滑鼠在圖像上的當前位置。

## 4. 定義滑鼠事件函數 [事件連結](#)

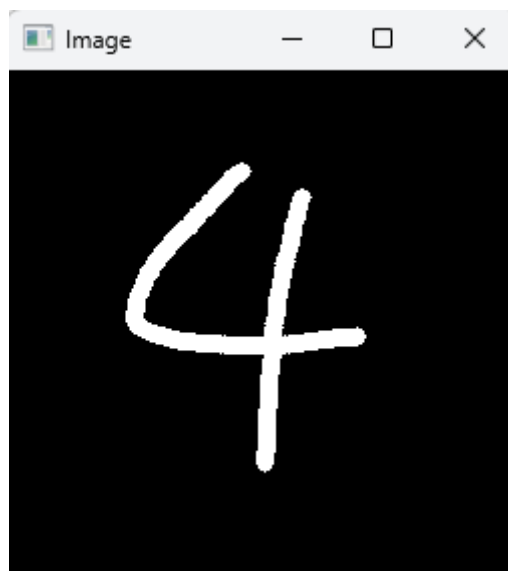
- 定義一個名為 `mouse_down_event` 的函數來處理滑鼠的不同事件 (按下、移動、釋放)。
- 當 **滑鼠左鍵按下**，設置 `drawing` 為 `True` 並記錄滑鼠位置到 `(ix, iy)`。
- 當 **滑鼠移動** 且 `drawing` 為 `True` 時，在圖像上從上一個點 `(ix, iy)` 到當前點 `(x, y)` 畫一條白線，並更新 `(ix, iy)`。
- 當 **滑鼠左鍵釋放**，停止繪畫。如果此時使用者按下 "c" 鍵，則對當前繪製的圖像進行預測。將圖像縮放到 28x28 像素，轉換數據類型並正規化後，使用加載的模型進行預測，並打印預測結果。最後重置圖像以供再次繪圖。

## 5. 顯示圖像和綁定事件

- 使用 `cv2.imshow("Image", img)` 顯示初始化的圖像。
- 使用 `cv2.setMouseCallback("Image", mouse_down_event)` 將滑鼠事件回調函數綁定到窗口。
- 使用 `cv2.waitKey(0)` 讓視窗持續顯示直到有鍵盤事件發生。
- 使用 `cv2.destroyAllWindows()` 在程序結束時關閉所有窗口。

## 6. 開始和結束

- 程序在執行完上述設定後進入等待狀態，直到使用者進行滑鼠操作或按鍵結束程序。



按下 "c" 鍵

輸出: Predict Number: 4