

第 7 章 Raspberry Pi 電腦視覺



章節目標：在這一章節，我們將為 Raspberry Pi 加上鏡頭，以 python CV2 進行影像擷取，並將影像利用 mediapipe 提取人體骨架，隨後用自己訓練的 ai 模型，辨識人物是否跌倒，並在網頁呈現即時圖像。



本章學習
內容摘要

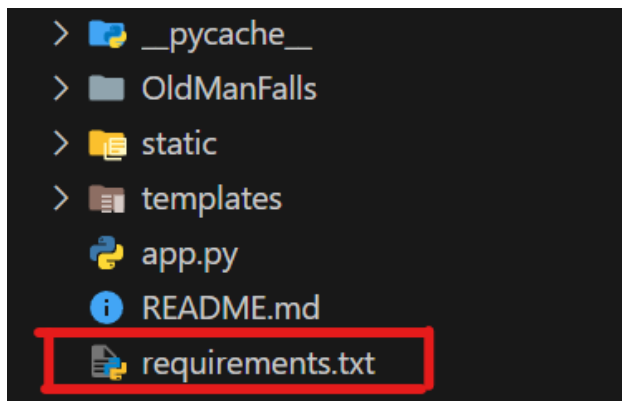
1. Raspberrypi 連接鏡頭。
2. 透過 FLASK 傳送即時影像。
3. 跌倒訓練資料採集。
4. 訓練自己的 AI 模型。
5. 透過 FLASK 傳送辨識結果。
6. 附錄-改為用影片作為輸入



環境安裝

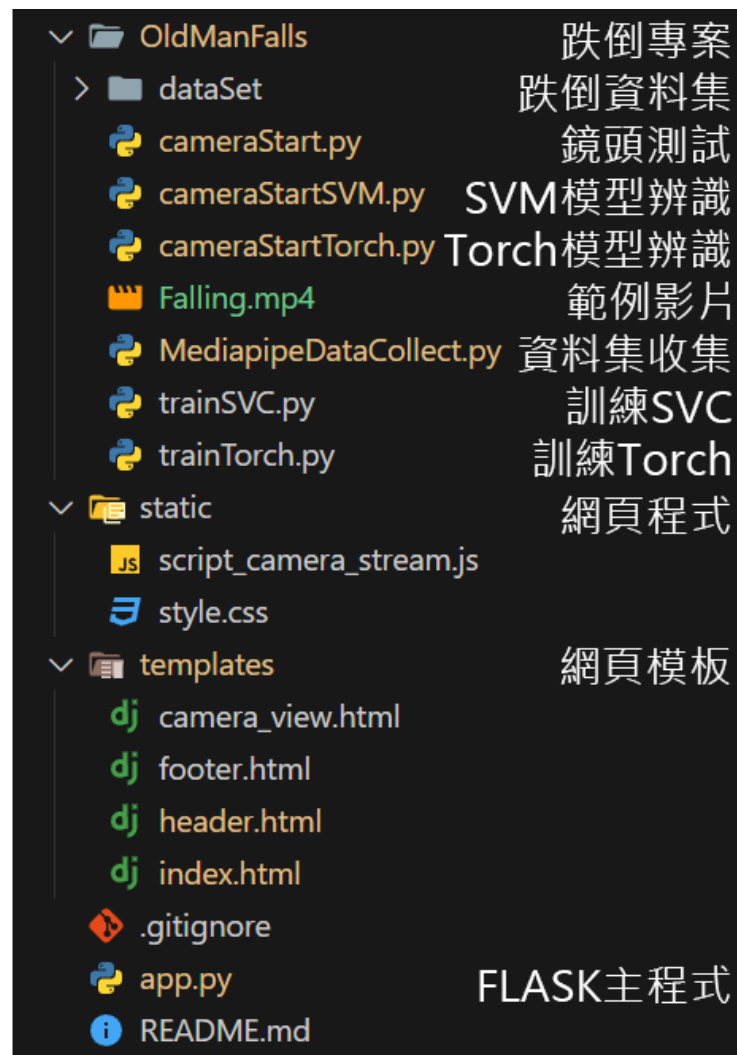
下載 requirements.txt，在同層級開啟 CMD，執行

```
$ pip install -r requirements.txt
```



1.1. RASPBERRYPI 連接鏡頭

- 1-1. 將發給你的羅技網路攝影機，或任意 USB 攝影機，插入樹梅派的任一 USB 口。
- 1-2. 開啟附件專案資料夾，結構如下



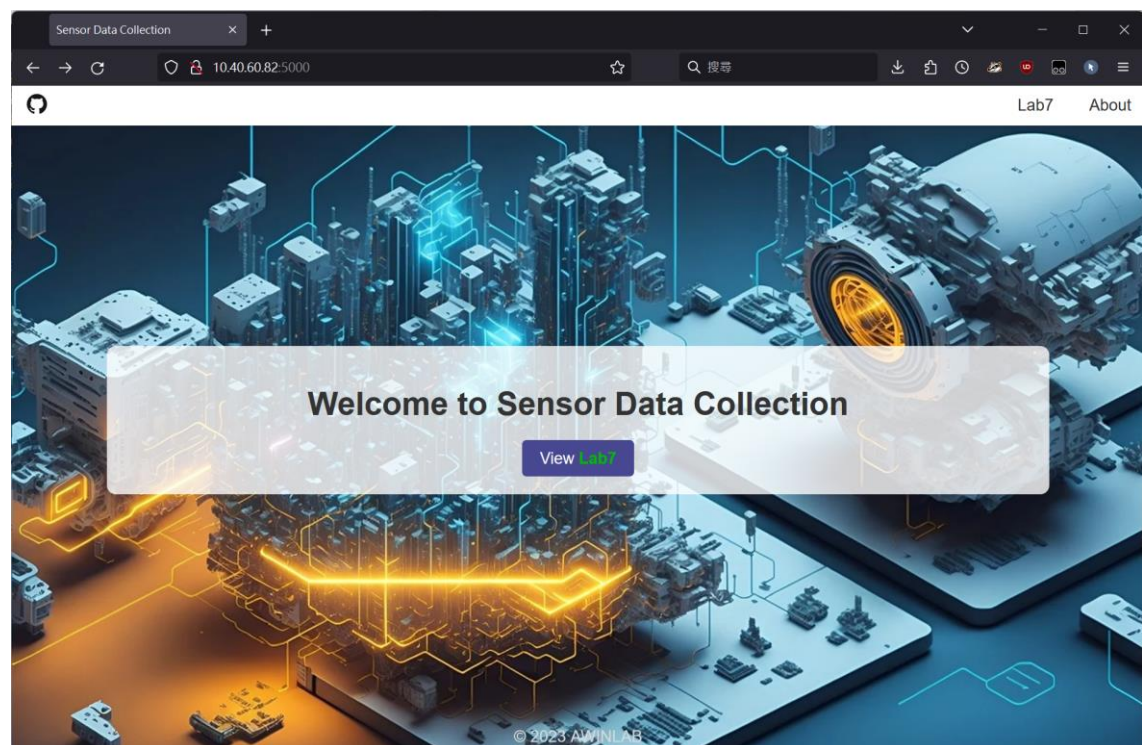
- 1-3. 執行 app.py

```
$ python app.py
```

- 1-4. 在終端輸出找到網頁的對外 ip，之後會用到。

```
PS D:\研究所\樹莓\3111_lab5 - 複製> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000 ← 本地ip
* Running on http://10.40.60.82:5000 ← 對外ip
Press CTRL+C to quit
```

1-5. 在連到同網路的另一台電腦瀏覽器網址欄，輸入對外 ip，可以看到 Lab 的網頁。



1-6. 按下 View Lab7，進入圖像傳輸頁面後，放旁邊備用。



1-7. 開啟跌倒專案內的 cameraStart.py，將檔案內的兩個 ip 位址都修改成剛剛得到的對外 ip，如：

```
http://xxx.xxx.xxx.xxx:5000/post_camera_frame
```

改為

```
http:// 你的對外 ip:5000/post_camera_frame
```

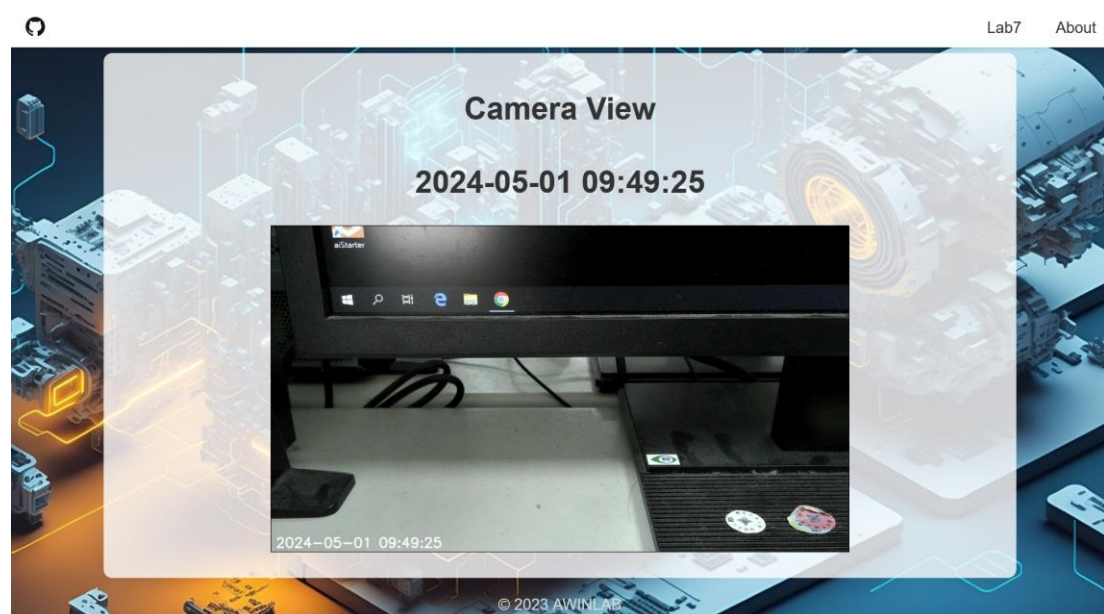
```
18
19     # 發送POST請求
20     try:
21         response = requests.post(
22             'http://10.40.60.82:5000/post_camera_frame', json=data)
23         if response.status_code == 200:
24             print("err sent successfully")
25
26
27
28
29
30
31
32
33
34     # 發送POST請求
35     try:
36         response = requests.post(
37             'http://10.40.60.82:5000/post_camera_frame', json=data)
38         if response.status_code == 200:
39             print("Image sent successfully")
40         else:
41             print(f"Failed to send image. Status code: {response.status_code}")
42     except requests.exceptions.RequestException as e:
43         print("Failed to send image:", e)
```

其他.py 檔裡面有這兩個 ip 的地方，之後也需要修改。

1-8. 執行 cameraStart.py

```
$ python cameraStart.py
```

1-9. 如果此時能在網頁上看到鏡頭畫面，說明這階段成功了



2.3. 跌倒訓練資料採集

2-1. 開啟跌倒資料夾內的 MediapipeDataCollect.py

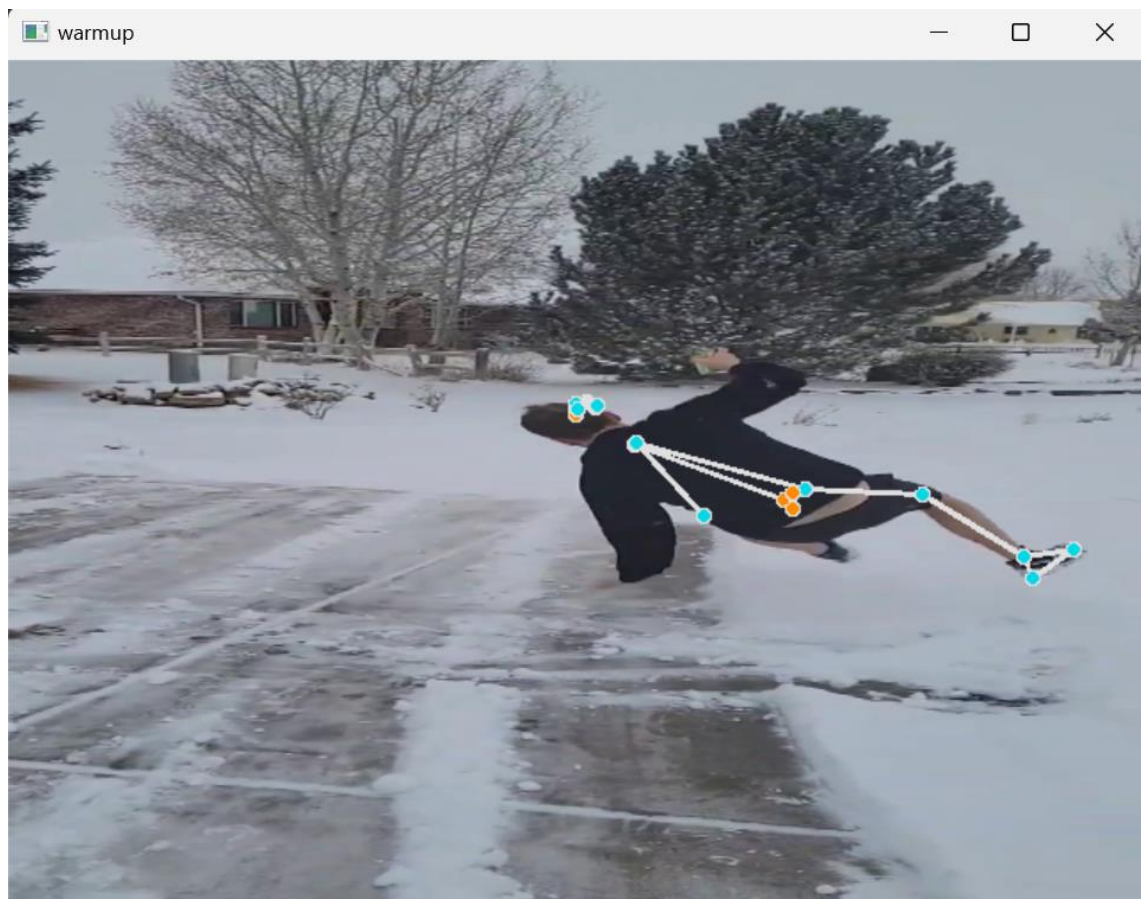
2-2. 確認 csv 的檔名，0 代表正常，1 代表跌倒，分別進行兩次採集。

```
31     time.sleep(5) # 給你5秒擺好POSE
32     # 設定CSV名稱
33     csv_filename = "./dataSet/0.csv"
34     # 將新的關鍵點座標陣列附加到CSV文件末尾
35     with open(csv_filename, 'a', newline='') as csvfile:
```

2-3. 執行 MediapipeDataCollect.py

```
$python MediapipeDataCollect.py
```

2-4. 成功執行可以看到骨架顯示，每 5 秒會擷取為一筆資料，WINDOWS 電腦會逼一聲提醒你。



2-5. 進行兩次資料採集，需全身入鏡，一次是正常站立及活動，一次是倒臥在地上，各需要大約 50 筆資料，姿勢越多樣越好。

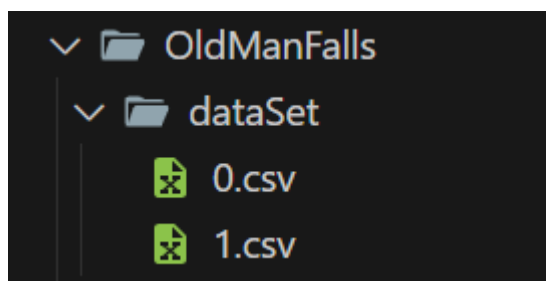


正常站立



跌倒

2-6. 兩次採集完後資料夾會有兩個 CSV 檔，裡面每列是一筆全身關節位置。



```
OldManFalls > dataSet > 0.csv > data
1 0.5166316628456116,0.20242682099342346,-0.25198429822921753,0.999254047870636,0.5210773348808289,0.1899314969778061,-
2 0.5506668090820312,0.1928868144750595,-0.26877158880233765,0.9998144507408142,0.551826536655426,0.18011532723903656,-
3 0.5712102651596069,0.20603051781654358,0.09869849681854248,0.9997596740722656,0.5665817260742188,0.19184726476669312,-
4 0.5367294549942017,0.2036317139863968,0.2627725899219513,0.9999274611473083,0.5323042869567871,0.1928594708442688,0.
5 0.5060552358627319,0.18532365560531616,-0.004841409157961607,0.9985745549201965,0.5126477479934692,0.1723505854606628
6 0.48642387986183167,0.1887362003326416,-0.1164478063583374,0.999160647392273,0.49316588044166565,0.17556272447109222,
7 0.5235576033592224,0.17433986067771912,-0.24818971753120422,0.9998651146888733,0.5269217491149902,0.1620677411556244,
```

3.4. 訓練自己的 AI 模型

3-1. 開啟跌倒專案內的 trainSVC.py

3-2. 執行 trainSVC.py

```
$python trainSVC.py
```

3-3. 訓練完成後會產生一個模型檔 svm_model

3-4. 如果想使用 pyTorch 訓練，執行 trainTorch.py，
會有額外加分

4.4. 透過 FLASK 傳送辨識結果

4-1. 依照你選擇的模型，開啟 cameraStartSVM.py 或
cameraStartTorch.py。

4-2. 修改檔案內的兩個 ip 位置，改為對外 ip

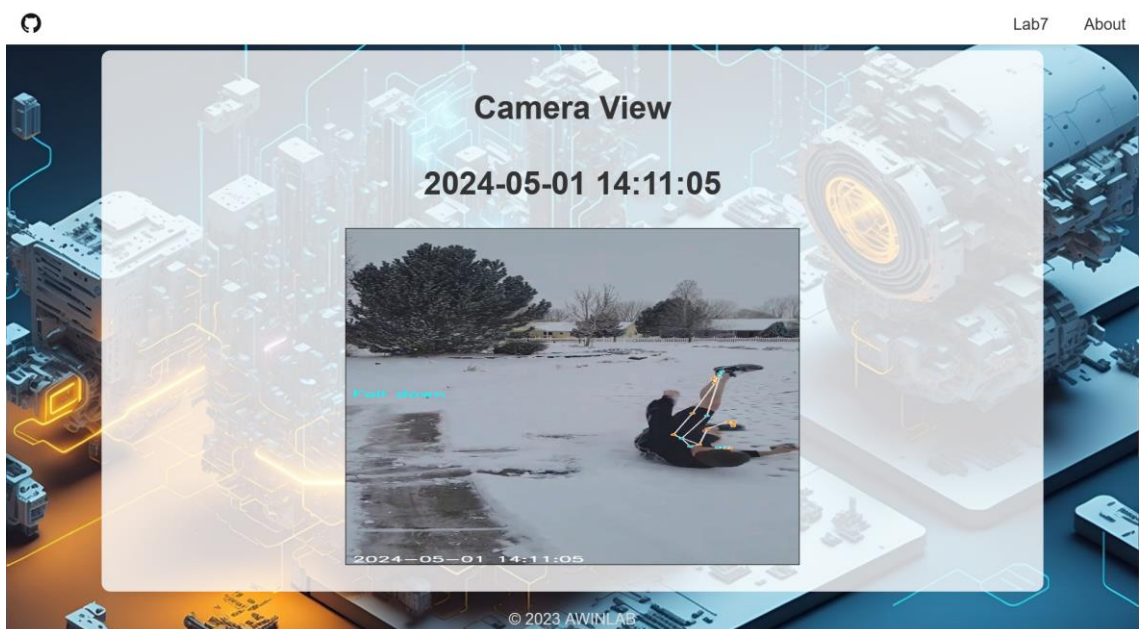
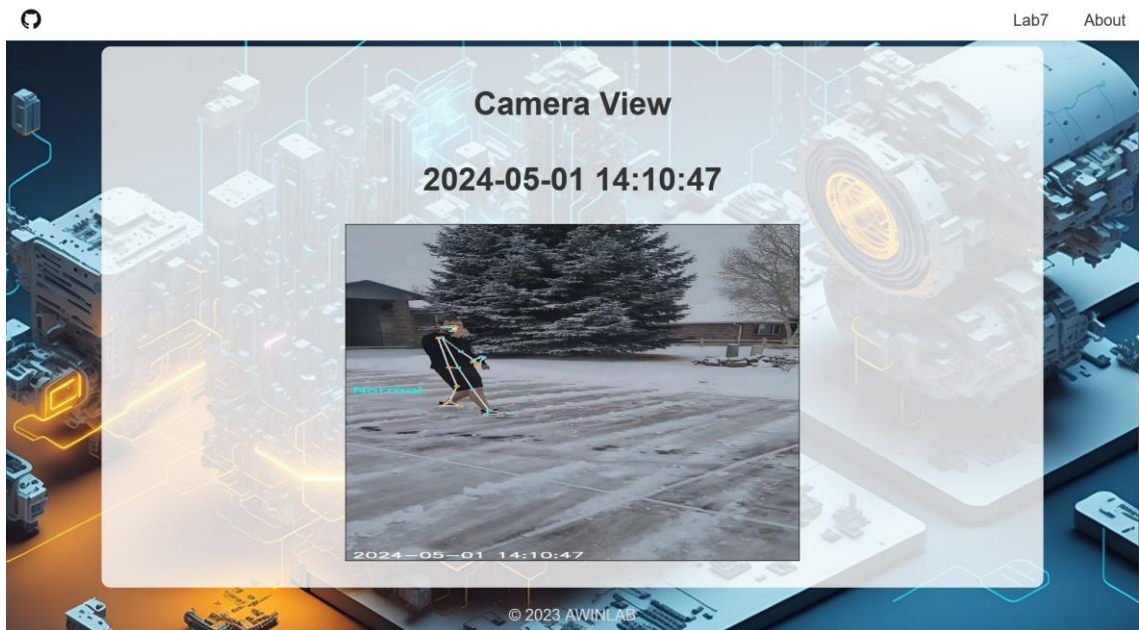
4-3. 確定接上鏡頭後執行，

```
$python cameraStartSVM.py
```

或

```
$ python cameraStartTorch.py
```

4-4. 辨識結果會傳到網頁上顯示，畫面左方顯示
'Normal',或'Fall down'。



5. 附錄-改為用影片作為輸入

開啟 cameraStartSVM.py 或 cameraStartTorch.py

```
18 # 開啟鏡頭  
19 cap = cv2.VideoCapture(0)
```

改為影片位址

```
18 # 開啟鏡頭  
19 cap = cv2.VideoCapture("./Falling.mp4")
```