

銘傳大學

資訊傳播工程學系

物聯網導論期末專案報告

居家照顧_跌倒偵測

指導教授：黃博俊 老師

學生：李旻珊 林采葳

中華民國一一〇年十二月二十四日

目錄

目錄.....	i
圖目錄	ii
一、 題目	1
二、 目的	1
三、 功能	1
四、 研究工具	2
五、 進行步驟	4
六、 程式碼講解	5
七、 結果呈現	9
八、 問題	10
九、 未來	11

圖目錄

圖 4-1 攝影機	2
圖 4-2 樹莓派	2
圖 4-3 麵包板	2
圖 4-4 語音卡	3
圖 4-5 喇叭	3
圖 4-6 擴充板	3
圖 4-7 LED 燈	4
圖 5-1 流程圖	5
圖 6-1 LED 程式碼 1	6
圖 6-2 LED 程式碼 2	6
圖 6-3 LED 程式碼 3	6
圖 6-4 語音程式碼 1	7
圖 6-5 語音程式碼 2	7
圖 6-6 語音程式碼 3	7
圖 6-7 鏡頭程式碼 1	8
圖 6-8 鏡頭程式碼 2	8
圖 6-9 鏡頭程式碼 3	8
圖 6-10 鏡頭程式碼 4	9
圖 6-11 鏡頭程式碼 5	9
圖 7-1 偵測到人	10
圖 7-2 跌倒狀態	10

一、 題目

居家照顧_跌倒偵測

二、 目的

台灣近年來人口老化速度越來越快，居家照顧相關的行業也越來越興盛，但是很多時候一個護理師通常會照顧 3 個以上的長者，因此有供不應求的情況。

我們希望利用攝影機，偵測是否有人，若有人的話則會發出警示聲音，並緩慢亮起紅燈；若沒人的話則不會有反應。更重要的是如果有人跌倒，則會發出警示聲，並快速閃爍紅燈。

三、 功能

利用攝影機進行偵測是否有人，若有人的話則會發出警示聲音，並緩慢亮起紅燈；若沒人的話則不會有反應。更重要的是如果有人跌倒，則會發出警示聲，並快速閃爍紅燈。

四、研究工具

1. 攝影機



圖 4-1 攝影機

2. 樹莓派



圖 4-2 樹莓派

3. 麵包板

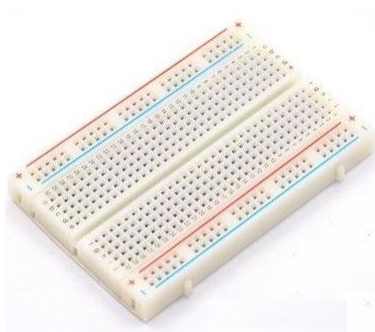


圖 4-3 麵包板

4. 語音卡

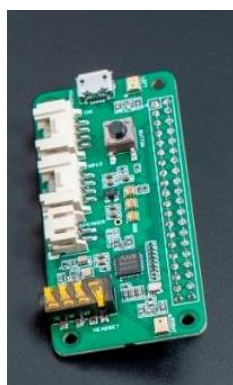


圖 4-4 語音卡

5. 喇叭



圖 4-5 喇叭

6. 擴充板

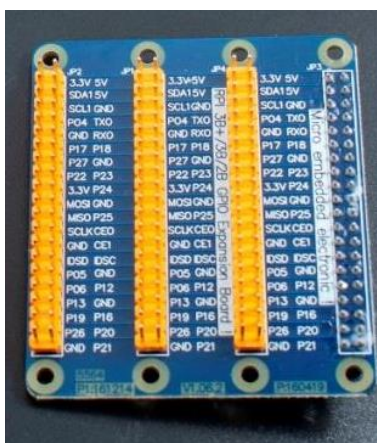


圖 4-6 擴充板

7. LED 燈



圖 4-7 LED 燈

五、進行步驟

- (1) 攝影機進行偵測，如果沒有人的話不會有任何的警示聲音與 LED 燈，只會印出 'No'。
- (2) 如果有偵測到人的話，則會檢查標籤是否為 'person'，如果是 person 的話則會印出 x、y 的 min、max 並計算 $rate = (ymax - ymin)/(xmax - xmin)$ ，並且開啟 LED 燈。
- (3) 接著進行第二個檢測，檢測 rate 值是否小於 0.6，若小於 0.6 代表這個人跌倒了，而喇叭會發出 '危險跌倒了' 的警示聲，並且 LED 呈現快速閃爍的狀態，最後則會輸出 'help' 進行警示。
- (4) 如果 rate 值大於 0.6，則會判斷這個人沒有跌倒，喇叭會發出 '注意有人喔' 的警示聲，LED 則緩慢的閃爍，最後再輸出 'YYYYYY'。

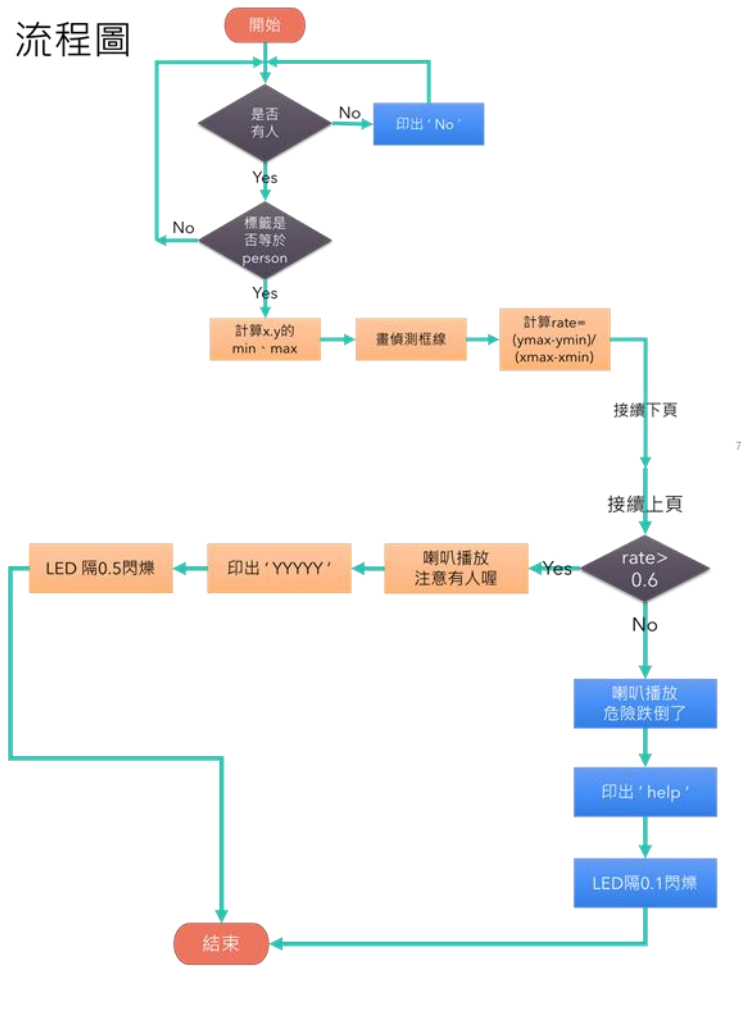


圖 5-1 流程圖

六、程式碼講解

1. LED 程式碼：

- (1) 接角設為 26，接地為 20（如圖 6-1 所示）。
- (2) 設定間隔閃爍秒數，並印出 HIGH、LOW 確認是否成功執行（如圖 6-2 所示）。
- (3) 確保程式中止時能夠執行並清掉 GPIO 的設定（如圖 6-3 所示）。


```

15
16 LED_PIN = 26
17 GPIO.setmode(GPIO.BOARD)
18 GPIO.setup(LED_PIN, GPIO.OUT)
19

```

圖 6-1 LED 程式碼 1

```

225 GPIO.output(LED_PIN, True)
226 print("HIGH")
227 time.sleep(0.5)
228 GPIO.output(LED_PIN, False)
229 print("LOW")
230 time.sleep(0.5)

```

圖 6-2 LED 程式碼 2

```

283 GPIO.cleanup()

```

圖 6-3 LED 程式碼 3

2. 語音 程式碼：

- (1) 放置 speak 程式碼（如圖 6-4 所示）。
- (2) 設定 sentence01、sentence02 分別不同的語句（如圖 6-5 所示）。
- (3) 設定 sentence 句子為 sentence01+警示語，並設置隔 2 秒播放一次（如圖 6-6 所示）。

```

21 def speak(sentence, lang, loops=1):
22     with tempfile.NamedTemporaryFile(delete=True) as fp:
23         tts=gTTS(text=sentence, lang=lang)
24         tts.save('{} .mp3'.format(fp.name))
25         mixer.init()
26         mixer.music.load('{} .mp3'.format(fp.name))
27         mixer.music.play(1)

```

圖 6-4 語音程式碼 1

```
191 sentence01='注意'  
192 sentence02='危險'
```

圖 6-5 語音程式碼 2

```
223 sentence=sentence01+'有人喔'  
224 print('YYYYY')  
225 GPIO.output(LED_PIN,True)  
226 print("HIGH")  
227 time.sleep(0.5)  
228 GPIO.output(LED_PIN,False)  
229 print("LOW")  
230 time.sleep(0.5)  
231  
232 else:  
233 sentence=sentence02+'跌倒了'  
234 print('help')  
235 GPIO.output(LED_PIN,True)  
236 print("HIGH")  
237 time.sleep(0.1)  
238 GPIO.output(LED_PIN,False)  
239 print("LOW")  
240 time.sleep(0.1)  
241  
242 speak(sentence, 'zh')  
243 time.sleep(2)
```

圖 6-6 語音程式碼 3

3. 鏡頭 程式碼：

- (1) 印出 classes，設定 target 為第 0 個（如圖 6-7 所示）。
- (2) 設定如果 classes 有第 0 個標籤(person)的話則計算 x.y 的 min 和 max、並開始畫框線（如圖 6-8 所示）。
- (3) 設定框線、label、rate 值，並印出 x.y 的 min 和 max，以及 rate 值（如圖 6-9 所示）。
- (4) 設定 rate 值>0.6 時，代表有偵測到人，而且是站立的，則喇

叭播放’ 注意有人喔’，印出’ YYYYYY’，LED 以 0.5 的速度閃爍；否則，代表有偵測到人，但是是跌倒的，則喇叭播放’ 危險跌倒了’，印出’ help’，LED 以 0.1 的速度閃爍（如圖 6-10 所示）；若沒有偵測到人，只會印出’ NO’，LED 則不會亮（如圖 6-11 所示）。

```

186 # Retrieve detection results
187 boxes = interpreter.get_tensor(output_details[0]['index'])[0] #
188 classes = interpreter.get_tensor(output_details[1]['index'])[0]
189 print('classes=', classes)
190 target=[0]
191 sentence01='注意'
192 sentence02='危險'

```

圖 6-7 鏡頭程式碼 1

```

196 # Loop over all detections and draw detection box if confidence is above minimum
197 for i in range(len(scores)):
198     if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):
199         if classes[i]==0:
200
201             # Get bounding box coordinates and draw box
202             # Interpreter can return coordinates that are outside of image dimension
203             ymin = int(max(1,(boxes[i][0] * imH)))
204             xmin = int(max(1,(boxes[i][1] * imW)))
205             ymax = int(min(imH,(boxes[i][2] * imH)))
206             xmax = int(min(imW,(boxes[i][3] * imW)))
207
208             cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (255, 151, 100), 2)

```

圖 6-8 鏡頭程式碼 2

```

207 cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (255, 151, 100), 2)
208
209 # Draw label
210 object_name = labels[int(classes[i])] # Look up object name from "labels" array using class index
211 label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: "person: 72%"
212 labelSize, baseline = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2) # Get font size
213 label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to top of window
214 cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseline-10), (255, 255, 255), cv2.FILLED)
215 cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2) # Draw label text
216
217 print('ymin=',ymin,'ymax=',ymax,'xmin=',xmin,'xmax=',xmax)
218 rate=(ymax-ymin)/(xmax-xmin)
219 print('rate=',rate)
220

```

圖 6-9 鏡頭程式碼 3

```

221 GPIO.output(LED_PIN,True)
222 if(rate>0.6):
223     sentence=sentence01+'有人喔'
224     print('YYYYY')
225     GPIO.output(LED_PIN,True)
226     print("HIGH")
227     time.sleep(0.5)
228     GPIO.output(LED_PIN,False)
229     print("LOW")
230     time.sleep(0.5)
231
232 else:
233     sentence=sentence02+'跌倒了'
234     print('help')
235     GPIO.output(LED_PIN,True)
236     print("HIGH")
237     time.sleep(0.1)
238     GPIO.output(LED_PIN,False)
239     print("LOW")
240     time.sleep(0.1)
241
242 speak(sentence, 'zh')
243 time.sleep(2)

```

圖 6-10 鏡頭程式碼 4

```

244 .....else:
245 .....GPIO.output(LED_PIN,False)
246 .....print("NO")

```

圖 6-11 鏡頭程式碼 5

七、結果呈現

使用樹莓派和攝影機皆可以偵測，標籤也可只偵測人，如我們設定的喇叭會發出注意的警示聲，也會印出‘YYYYY’，並且緩慢地閃爍 LED 燈（如圖 7-1 所示）。

假如 rate 值小於 0.6，則為跌倒狀態，系統也如我們設定的，喇叭會發出危險的警示聲，也會印出‘help’，並且快速地閃爍 LED 燈（如圖 7-2 所示）。

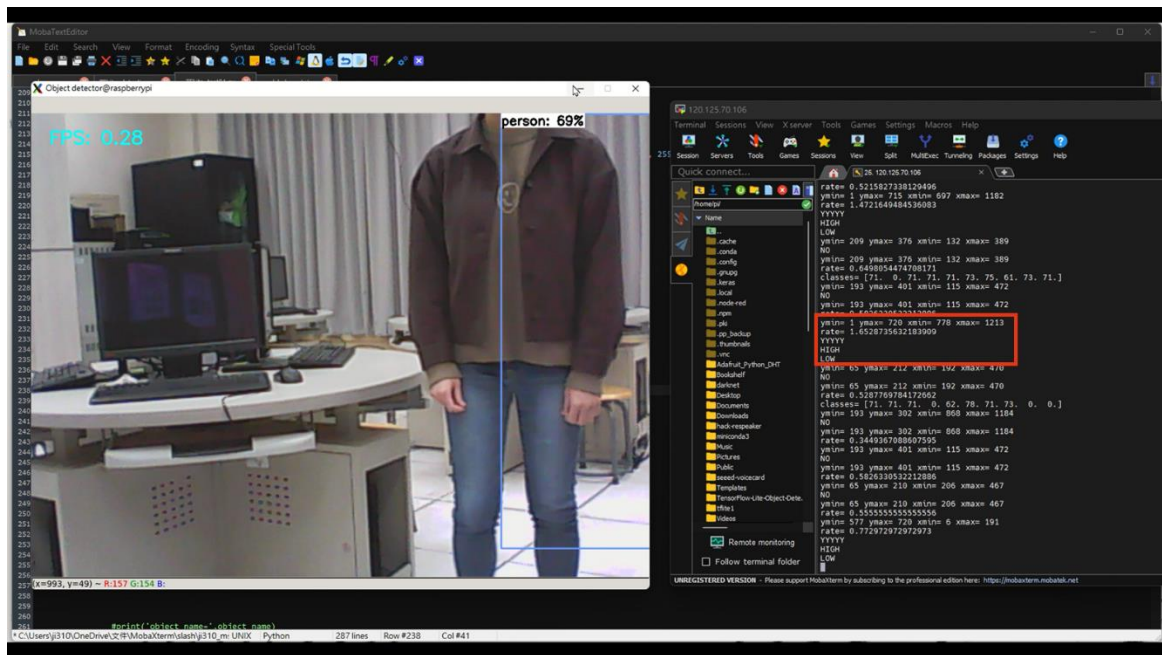


圖 7-1 偵測到人

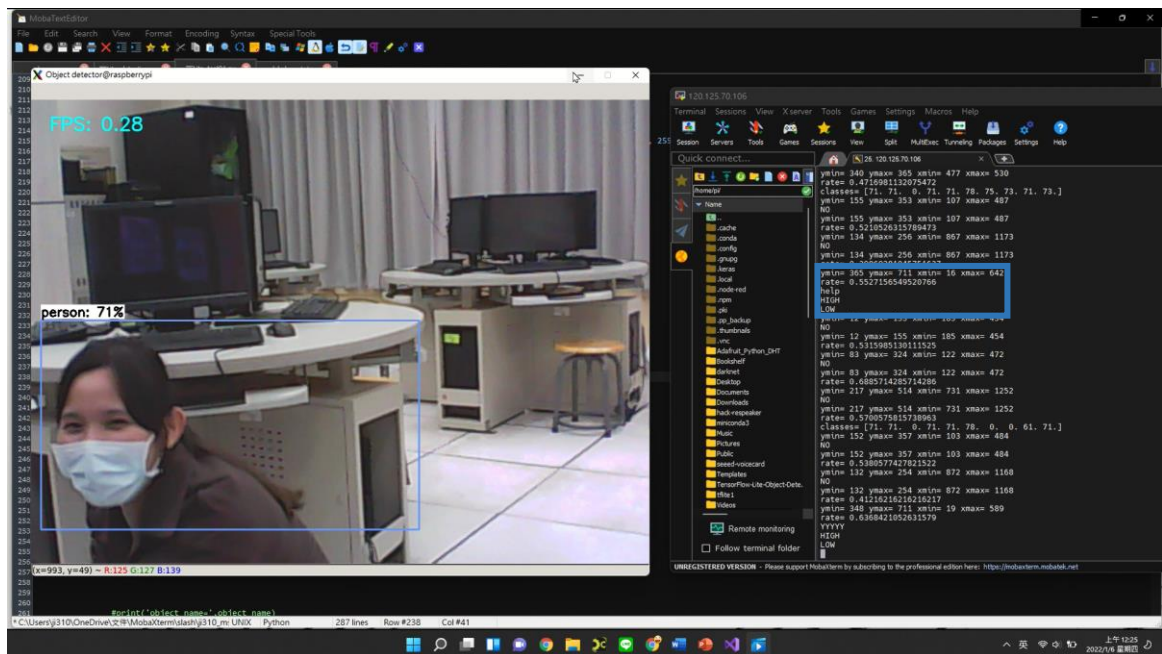


圖 7-2 跌倒狀態

八、問題

我們在做這個專題的時候，第一個遇到的困難是 rate 值的設定，rate

的值和攝影機擺放的位置也有間接的關係，需要完整的照到人，並且不斷的調整值，rate 值太大，可能人是坐著的狀態但卻顯示跌倒的回應；rate 值太小，也許人已經是平躺的狀態卻始終顯示站立的回應，因此這部分我們也測試滿久才找到一個平衡值。

再來是攝影機的拍攝速度，很多時候會是喇叭和 LED 燈已經有反應了，但攝影機的畫面卻來是卡頓、延遲的狀態，因此我們透過刪減一些不必要的程式碼，降低系統的作業時間，速度有比較快，但還是有卡頓和延遲的現象。

九、未來

我們希望未來可以更進一步利用 Line Bot 的方式，連接使用者的手機，可以更及時的透過訊息通知，避免錯過急救的黃金時間。