# 國立高雄科技大學

# 嵌入式系統設計實習

智慧社交距離監控

期末報告第3組

#### 組員:

周登豐 F110112108 沈易賢 F110112109 黃子衡 F110112110

指導教授: 陳建宏

中華民國一百一十一年五月

# 目 錄

1	`	研究目的	.3
2	•	設計與方式	-4
	1.	系統架構	-4
	2.	硬體架構	-5
	3.	軟體環境	-6
3	`	實驗結果	8
4	`	名言論1	2
矫	付件	:一 參考資料1	2

## 1、 研究目的

2019年因 Covid19病毒肆虐,除了戴口罩保護自己之外,最重要的就是與他人維持社交距離,而許多人在無意識的情況下會忘記保持距離,若能有一個設備能提醒人們,將會有很大的幫助。

### 2、 設計與方式

## A. 系統架構

利用 python 隨時讀取攝影機,再使用程式從影像抓取行人,並以各人輪廓中心點取出間距,間距小於 1.5m 時框出該行人,並辨識該行人衣服顏色,再依顏色發出語音提醒該行人。

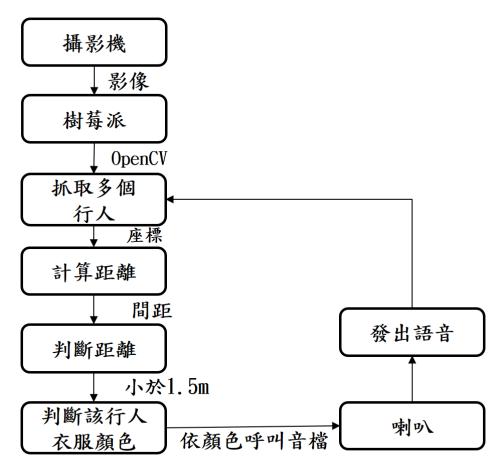


圖1系統架構圖

## B. 硬體架構

使用 Raspberry pi 作為主控接收資料,攝影機透過 USB 傳輸影像到 Raspberry pi,透過 USB 輸出給喇叭發出聲音警告、提醒。

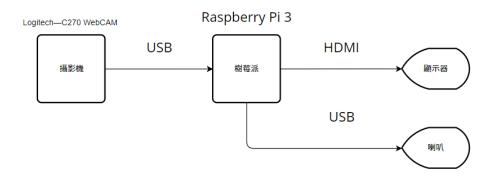


圖 2.硬體架構圖

### C. 軟體架構

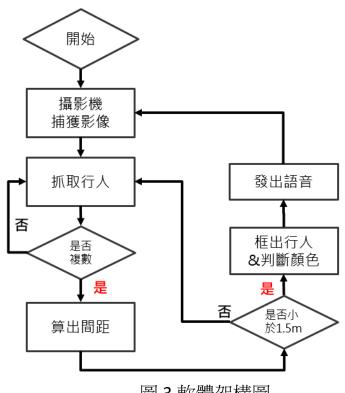


圖 3.軟體架構圖

## 行人辨識:

使用 OPEN CV 內建的行人辨識模型 (HOGDescriptor\_getDefaultPeopleDetector),是使用方 向梯度直方圖(Histogram of oriented gradient)處理影 像,透過支持向量機(Support Vector Machine)算法來 訓練出行人辨識模型。

## 距離計算:

辨識為行人後,以行人的中心點,使用直線距離 公式來計算兩人的距離,如果小於 1.5 公尺,則違反 社交距離。

## 顏色辨識:

透過 HSV 來判斷衣物顏色,設定飽和度(S)為 100%,明度(V)為 100%,根據色相(H)來辨識顏色。

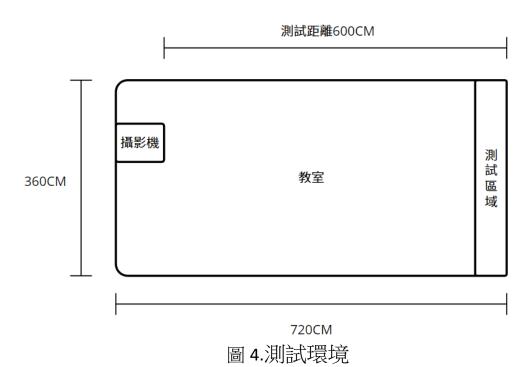
H value	0~6 163~255	6~21	23~32	33~77	78~130	131~163
color						

表 1.色相值對應的顏色

## 3、實驗結果

## 實驗環境:

教室長 720 CM 寬 360 CM,攝影機到測試區 域為 600 CM。



## PC 測試

1.當人小於社交距離時抓取

마다 많은 .	132
距離:	The second second second second
25 致 -	133
TT IAT -	
距離:	129
OF BA	132
此上 丙世:	
25 至 -	132
THE PAR -	
距離:	133
무는 당한 .	134
此上 丙世:	
25 班 -	134
THE PAR -	
距離:	132
OF BA	132
此上 丙世:	The second secon
品號-	131
TT 144 -	
距離:	128
무는 많은 .	126
距離:	
<b>电阻阻阻阻阻阻阻阻阻阻阻阻阻阻阻</b>	122
THE TAR	
距離:	124
	121
距離:	121

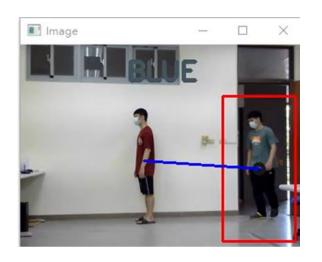


圖 5.小於社交距離

### 2.當人大於社交距離時不做反應

| 128 | 128 | 128 | 128 | 128 | 132 | 132 | 133 | 134 | 133 | 134 | 133 | 134 | 133 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134 | 134

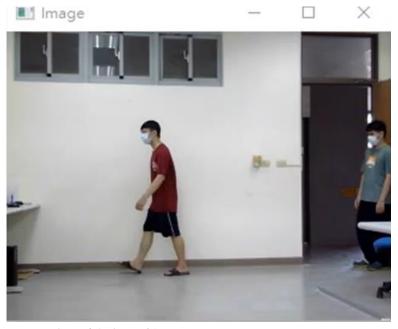


圖 6.大於社交距離

#### 3.當人又小於社交距離時警告

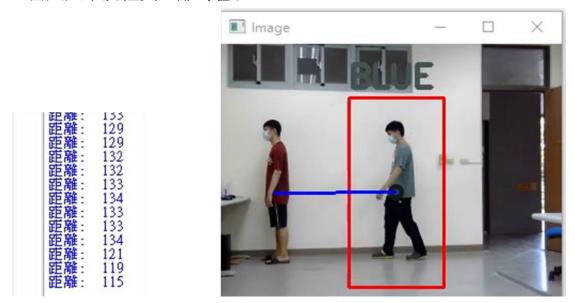


圖 7.移動後小於社交距離

#### 4.當人還是小於社交距離時持續警告

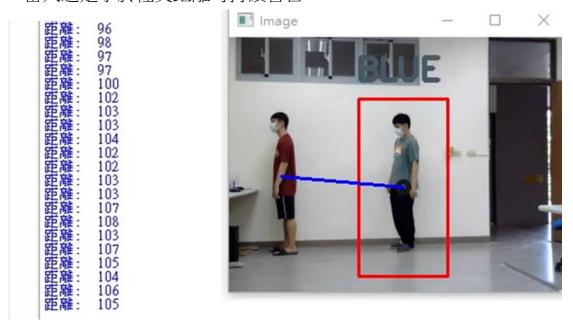


圖 8.持續小於社交距離

#### 5.直到保持社交距離時取消警告



圖 9.大於社交距離

## 樹莓派上執行

```
win width=240
win high=280
cap.set(3,win width) # 德克斯斯度 E
cap.set(4,win high) # 德克斯斯度 E
while cap.isOpened():

# Reading the video stream
ret, image = cap.read()
hsv frame = cv2.cvtColor(image, cv2.ColoR_BGR2HSV)
if ret:
image = inutils.resize(image,width=nin(800, image.shape[1]))

# Betection all the regions in the

BEE: 116
BEE: 127
BEE: 128
BEE: 129
BEE: 134
BEE: 193
BEE: 193
BEE: 195
```

圖 10.樹莓派上執行的結果

### 4、結論

這個專題主要的功能是透過影像辨識來提醒人們維持社交距離。 透過發出聲音來警告排隊的人,但有時警告會被忽視,因此提出透過衣物顏色辨識來指出違反社交距離的人。 最後在樹梅派上實現,能達成大部分辨識的功能。

## 附件一 參考資料

- Simple Color recognition with Opencv and Python —
   https://pysource.com/2021/10/19/simple-color-recognition-with-opencv-and-python/
- 2. Pedestrian Detection using OpeCV-Pytjon <a href="https://www.geeksforgeeks.org/pedestrian-detection-using-opencv-python/">https://www.geeksforgeeks.org/pedestrian-detection-using-opencv-python/</a>