

室內空氣監測及分析之物聯網應用



學生：謝承憲
指導教授：朱健銘

中國文化大學
地理學系
畢業專題

中華民國108年5月

摘要

目前市面上的空氣監測設備相當昂貴，每台儀器可以監測的項目也有限，亦須定期更換耗材，維護成本高。本研究設計、開發體積小、價格相對便宜的空氣監測系統，用以監測校園中室內空氣，俾利往後整合空調系統。本研究設計之裝置適合布置於各個具有無線網路(Wi-Fi)環境的校園室內空間中，監測裝置能將空氣感測器所收集到的空氣資訊彙整和上傳雲端空間，方便往後學校網頁依照需求呼叫第三方WebService服務，且依身分別開放權限讓學生、教職員和訪客都可以藉由網頁瀏覽到校園空間空氣品質之相關資訊。本研究僅設計物聯網中的監測部分，而數據分析與預測、空調系統回饋等部分並非本研究討論範疇。

為了測試本研究所設計之空氣監測設備是否能正常運作，以及瞭解人數是否對於原本具有空調設備的教室之空氣品質有所影響，本研究將針對二氧化碳、總揮發性有機物、溫度、濕度和人數進行實際的監測實驗。除了空氣成分有相對應的感測器負責監測外，人數監測的部分將利用影像物件偵測的技術完成。本研究增加了「影像物件偵測」的部分係與之前文獻比較不同的地方，不僅增加了監測項目的豐富性，也能夠提供以後的研究者參考，協助校園室內空氣監測與空調設備整合的智慧空調系統。根據實驗結果發現以上四項監測目標與人數均相關性不高，本研究也因此找到往後可以做進一步調整的方向。

目錄

第一章 前言	1
1.1 研究背景和動機	1
1.2 研究目的	1

第二章 文獻回顧	2
2.1 物聯網(IoT, Internet of Things)	2
2.2 室內空氣品質 (IAQ, Indoor Air Quality)	2
2.3 室內空氣污染物分類及影響因子	3
2.4 二氧化碳(CO ₂ , Carbon Dioxide)	4
2.5 總揮發性有機物(TVOC, Total Volatile Organic Compound)	4
2.6 濕度(Humidity)	5
2.7 溫度(Temperature)	5
第三章 研究方法	6
3.1 研究流程	6
3.2 監測方式	7
3.3 研究區域	8
3.4 監測設備	9
3.5 程式架構與說明	14
3.6 統計方法	15
第四章 研究成果與分析	16
4.1 監測結果	16
4.2 監測資料分析	19
第五章 結論	26
5.1 結論	26
5.2 建議	26
參考資料	27
學術論文	27
網路資料	28
附錄	29
本研究所撰寫之程式碼	29

圖目錄

圖2.1 室內空氣品質污染物質成份分類圖 (李彥頤 , 2004) 3

圖3.1 本研究流程架構圖 6

圖3.2 監測設備和研究區域示意圖 8

圖3.4 監測設備實際位置 8

圖3.3 研究區域實際狀況 8

圖3.5 樹莓派	9
圖3.6 CJMCU-8128電路板和感測器	10
圖3.7 樹梅派相機模組	11
圖3.8 YOLO影像物件偵測示意圖(林家平, 2018)	12
圖3.9 YOLOV3與其他訓練模型的精度和執行速度比較(Joseph Redmon & Ali Farhadi, 2018)	12
圖3.10 影像物件偵測實際應用情形	12
圖3.11 ThingSpeak網頁 (https://thingspeak.com/channels/660161/private_show)	13
圖3.12 程式邏輯和架構圖	14
圖4.1 各時段CO2濃度	16
.....	16
圖4.2 各時段TVOC濃度	16
圖4.3 各時段氣溫	17
圖4.4 各時段濕度	17
圖4.5 各時段人數	18
圖4.6 3月12日各項監測資料	20
圖4.7 3月12日 CO2-人數迴歸分析	21
圖4.8 3月12日 TVOC-人數迴歸分析	21
圖4.9 3月12日 溫度-人數迴歸模型	22
圖4.10 3月12日 濕度-人數迴歸模型	22
圖4.11 3月15日各項監測資料	23
圖4.12 3月15日 CO2-人數迴歸模型	24
圖4.13 3月15日 TVOC-人數迴歸模型	24
圖4.14 3月15日 溫度-人數迴歸模型	25
圖4.15 3月15日 濕度-人數迴歸模型	25

第一章 前言

最近幾年物聯網(IOT, Internet of Things)逐漸興起, 人們開始藉由這資訊載體提供的便利, 來識別、偵測和控制環境, 將彼此獨立物件之間的資訊做交流, 以達到現實世界的數據化。與此同時, 對於環境監控的議題也逐漸被大量討論, 許多人開始關注所身處的環境好壞, 大眾對於生活的品質要求漸漸提升。

1.1 研究背景和動機

現代人處於室內之時間相當長, 室內空氣的品質影響人類的健康時有所聞。當室內人數密度過高或通風換氣效率不佳時, 常會造成二氧化碳濃度累積, 濃度一旦高於一定比例, 人會感到呼吸費力或困難, 亦會產生頭痛、嗜睡、反射減退或倦怠等症狀, 並且容易感到疲勞而影響生活作息。

校園內的室內空間大多屬於教室, 不論是使用次數, 亦或是人數上都較為多, 因此對於空調設備有一定需求。但由於教室內空調多有可由人手控的裝置, 因此控制上可能不夠精確而造成電力浪費、資源損耗的情形發生, 本研究將對此問題設計監測設備。

1.2 研究目的

- 1.瞭解室內人數與空氣因子之間的關係。
- 2.提升室內空氣監測之效率, 彌補手動測量的不便性。
- 3.藉由安裝簡易的室內空氣感測器監測教室內部空氣品質。
- 4.自動監測室內環境以期做為日後整合空調設備的依據。

第二章 文獻回顧

2.1 物聯網(IoT, Internet of Things)

物聯網技術是能使獨立工作的物體之間能相互連通的技術, 是網際網路、傳統電信網等資訊承載體。通過物聯網技術可以利用中心電腦對機器、裝置和人員進行集中管理與遠端控制, 類似自動化操控系統。同時透過對小事件資料的收

集，最後可以聚整合大量資料，對於減少車禍、都市更新、災害預測與犯罪防治和流行病控制等等社會的重大改變，實現物與物相聯。物聯網技術將現實世界數位化，拉近分散的資訊，統整物與物的數位資訊。而本研究將開發用於監測空氣品質的後端系統，簡單來說，就是利用感測器自動監測室內空氣，並將監測所得的資料彙整、並上傳雲端，提供學校建立室內智慧空調的基礎。

2.2 室內空氣品質 (IAQ, Indoor Air Quality)

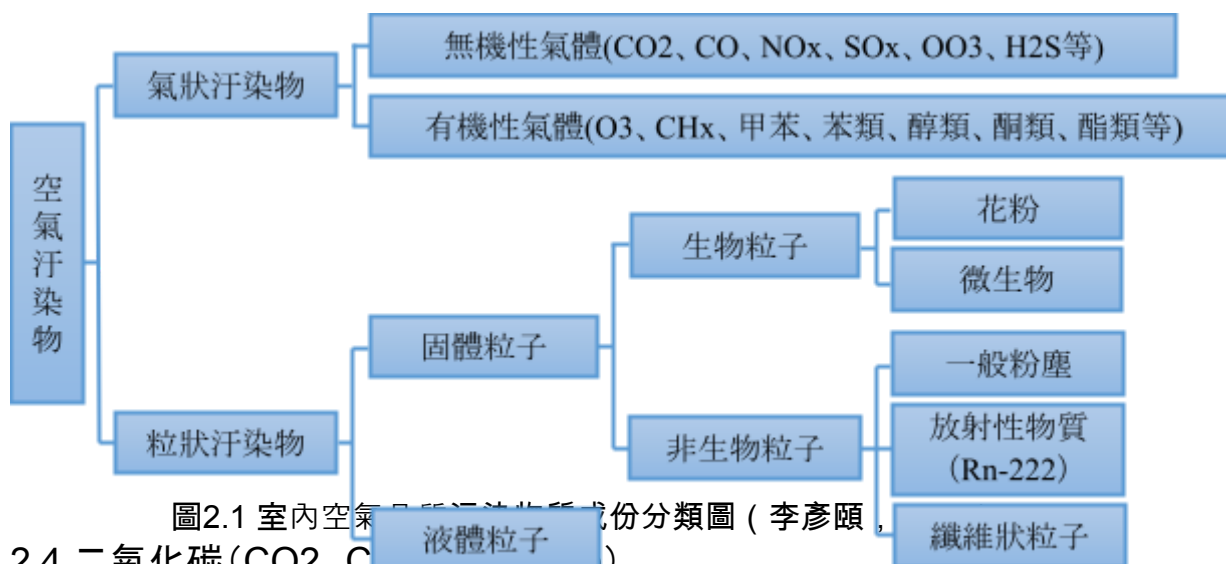
根據世界衛生組織資料，每年將近有380萬人的死亡、以及5歲以下兒童肺炎死亡中的50%，是與家庭空氣污染(HAP, Household Air Pollution)有關。而自從20世紀末，美國的環保署(US EPA, United States Environmental Protection Agency)及國家職業安全衛生研究所(NIOSH, National Institute for Occupational Safety and Health)等組織開始注意建築物內密閉和半密閉空間的空氣品質後，室內空氣品質的研究工作才逐漸被重視進而管制。隨著建築物興建技術和設計不斷變革，一般民眾在辦公室中活動機會增加，關於「病態建築物症候群(SBS, Sick Building Syndrome)」的報導更是引起更多人的注意與關心。

室內空氣品質 (IAQ) 是建築物及建築物內以及附近的空氣品質，影響在建築物內生活人們的健康及舒適。室內空氣品質會受氣體（如一氧化碳、氬、揮發性有機物、懸浮粒子）、生物(如：微生物、黴菌、細菌)或是其他會影響人們身體健康的物質所影響。主要提昇室內空氣品質的主要方式是生成源的控制、過濾、再配合通風來稀釋污染物質。而近幾十年來，建築物普遍採用空調，導致建築物使用者與室外環境愈趨隔絕，從外進入室內的空氣有限，不足以稀釋在室內積聚的空氣污染物及二氧化碳。

2.3 室內空氣污染物分類及影響因子

根據美國國家職業安全健康研究所(NIOSH)所做的室內空氣品質問題調查，室內汙染物源共包括：滲入外氣、室內人員、空調系統、建築材料、室內有機物質、事務(燃燒)器具與用品等六大類。而世界衛生組織(WHO, World Health Organization)根據污染物特性分類為：粒狀污染物、氣狀污染物及生物性氣膠三大類。而關注室內空氣品質的美國冷凍空調協會 (ASHRAE, American Society of Heating, Refrigerating and Air-Conditioning Engineers)則依據室內外汙染源

做區分。根據上述三個組織的分類，經常被探討的主要污染物有：一氧化碳、二氧化碳、甲醛、揮發性有機物質、臭氧、粒狀污染物、生物性氣膠(真菌、細菌等)。根據上述結論，本研究針對二氧化碳、揮發性有機物質兩種氣狀污染物作為監測目標。



2.4 二氧化碳(CO₂, Carbon Dioxide)

二氧化碳(CO₂)是人類排放的室內污染物中，相對比較容易測量的物質，並且與人類代謝活動相關。室內過高的二氧化碳濃度可能導致居住者昏昏欲睡、頭痛或活動能力變低。室外CO₂濃度通常為350-450ppm，而認為可接受的最大室內CO₂濃度為1000ppm。在大多數建築物中，人類是室內二氧化碳的主要供應來源。室內CO₂濃度是相對於室內人員密度和代謝活性的室外空氣的通風的充足性的指標。室內的CO₂濃度與上述室外濃度的差應降低至小於600ppm。NIOSH認為，超過1000二氧化碳的室內空氣中的濃度ppm的是提示通氣不足的指標。英國的學校標準規定，所有教學和學習空間的二氧化碳，當坐在頭部高度並在一整天內平均測量時，不應超過1,500 ppm。一整天是指正常上課時間，即上午9:00至下午3:30，包括午休時間等空閒時段。在香港，環保署為辦公樓及公眾地方制定了室內空氣質素指標，其中二氧化碳含量低於1,000 ppm被視為良好。歐洲標準將二氧化碳限制在3,500 ppm。這些限制為避免意識喪失(昏厥)，並且避免在較低二氧化碳濃度下開始出現的認知能力和能量受損。CO₂在封閉的辦公房濃度可在45分鐘內增加至超過1000 ppm。封閉或密閉空間內的二氧化碳濃度可在封閉後45分鐘內增加至1,000 ppm。例如，在一個3.5×4米(11英尺×13英尺)大小的辦公室中，大氣二氧化碳在通風停止和門窗關閉的45分鐘內從500 ppm增加

到超過1,000 ppm。

2.5 總揮發性有機物(TVOC, Total Volatile Organic Compound)

按照世界衛生組織的定義，如果在平均氣壓101.32kPa下，該化合物的沸點在50°C-250°C，即為揮發性有機物，在常溫下以氣體形式存在。可以進一步分為八類：烷類、芳烴類、烯類、鹵代烴類、酯類、醛類、酮類和其他。而行政院環境保護署環署所訂定總揮發性有機化合物 (TVOC)的含量標準，為少於0.56ppm (等同1287微克/立方米,1.287毫克/立方米或560ppbv0.2)，揮發性有機物危害相當明顯，當居室中的揮發性有機物濃度超過一定濃度時，在短時間內人們感到頭痛、噁心、嘔吐、四肢乏力；嚴重時會抽搐、昏迷、記憶力減退。揮發性有機物傷害人的肝臟、腎臟、大腦和神經系統，其中還包含了許多致癌物質。除了室內裝潢建材會散發揮發性有機物以外，人類本身也會藉由呼吸，代謝出揮發性有機物（刘畅，鄭云昊，劉兆榮，要茂盛，2016），因此室內空氣中揮發性有機物污染已引起各國重視，也是本次研究監測項目之一。

2.6 濕度(Humidity)

濕度(Humidity)一般在氣象學中指的是空氣溼度，它是空氣中水蒸氣的含量，不含水蒸氣的空氣被稱為乾空氣。由於大氣中的水蒸氣可以占空氣體積的0%到4%，一般在列出空氣中各種氣體的成分的時候是指這些成分在乾空氣中所占的比例。空氣溼度與呼吸之間的關係非常緊密。在一定的溼度下氧氣比較容易通過肺泡進入血液。濕度45%~55%是讓人感覺舒適的，在50%的溼度下感覺最舒適。過熱而不通風的房間裡的相對溼度一般比較低，這可能對皮膚不良和對黏膜有刺激作用。溼度過高影響人調節體溫的排汗功能，人會感到悶熱。而根據文獻指出濕度對於真菌和黴菌的增加有關聯（趙壽川，2018），該研究將濕度設定為研究範圍，因此本研究也將濕度納入監測項目之一。

2.7 溫度(Temperature)

溫度(Temperature)表示物體冷熱程度的物理量，微觀上來講是物體分子熱運動的劇烈程度，並且溫度只能通過物體隨溫度變化的某些特性來間接測量。溫度也會影響生物體內許多的反應，恆溫動物會調節自身體溫，若體溫升高即為發

熱，是一種醫學症狀。生物體也會感覺溫度的冷熱，但感受到的溫度受風寒效應影響，因此也會和周圍風速有關。溫度係為本研究重要監測目標，根據監測資料以個別調整教室內的空調系統以達到節能減碳的目的。

第三章 研究方法

本研究採用自動連續監測和數據分析的方法，針對本校教室內空氣因子：二氧化碳、總揮發性有機物、溫度、濕度進行統計分析和討論。

3.1 研究流程

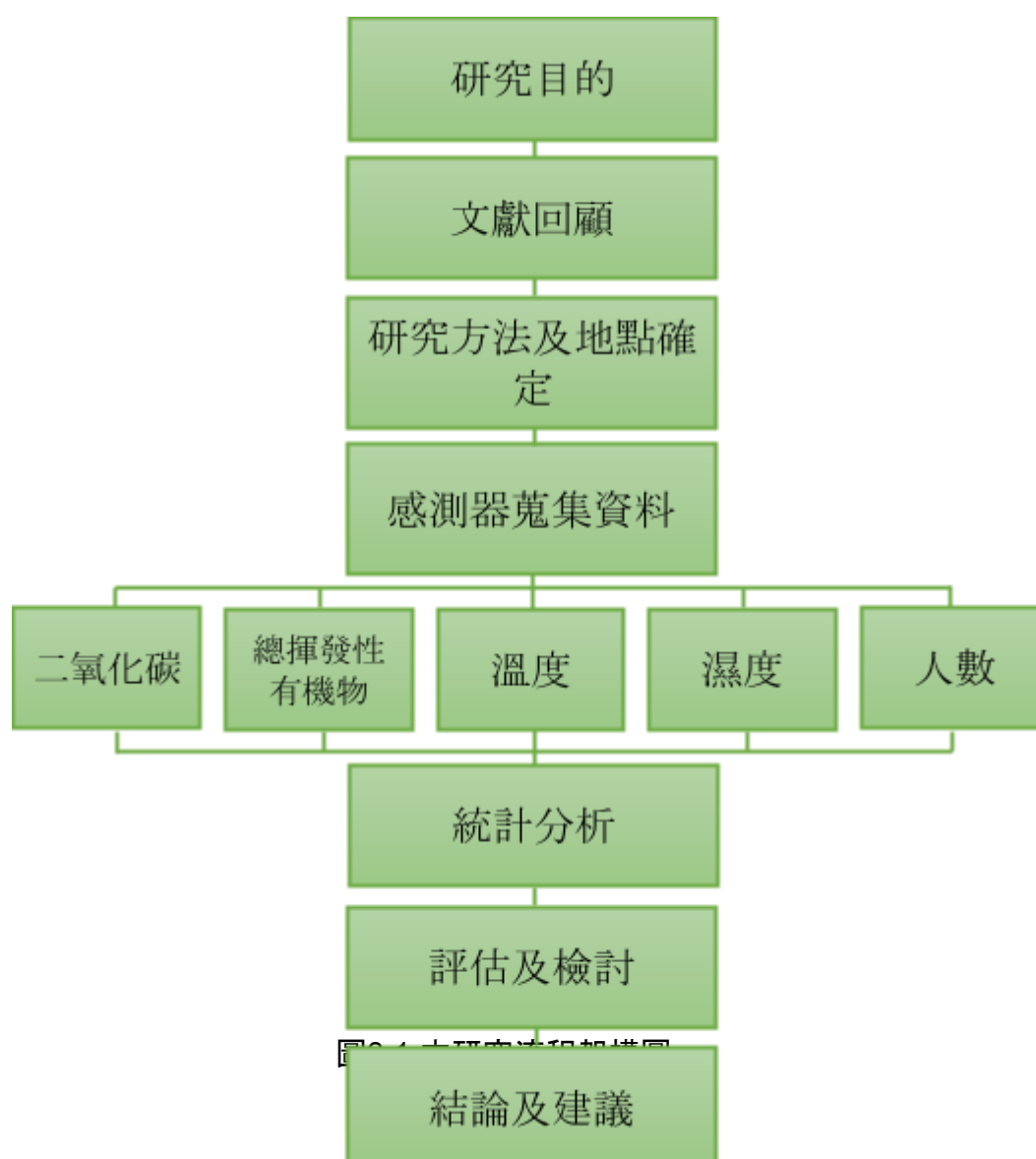


圖 3-1 本研究流程架構圖

3.2 監測方式

本研究於本校大義館某教室建置獨立的偵測器，進行連續 24 小時偵測，設

定每一分鐘讀取一次感測器、記錄一筆數據和上傳監測資料，所監測的項目包括：二氧化碳(CO₂)、總揮發性有機物(TVOC)、溫度(Temperature)、溼度(Humidity)以及人數(Human)共5個項目，透過本校網路，將數據傳至資料儲存平台中記錄，本實驗監測設備放置離地面高3公尺處的天花板。

依據「室內空氣品質檢驗測定管理辦法」第13、14和15條之規定：

- (1) 公告場所設置自動監測設施
 - (2) 之數目，除中央主管機關另有規定者外，依其公告管制室內空間樓地板面積每二千平方公尺(含未滿)，應設置一台自動監測設施。但其樓地板面積有超過四千平方公尺以上之單一無隔間室內空間，得減半計算應設置自動監測設施數目，且減半計算後數目不得少於二台。前項設置自動監測設施之監測採樣位置，應具代表性且分布於各樓層，於同樓層者應平均分布於樓層空間。
 - (3) 公告場所設置自動監測設施應量測之室內空氣污染物項目如下：
 - 一、二氧化碳。
 - 二、其他經中央主管機關指定者。
- (三)前二條規定之自動監測設施，應符合下列規定：
- 一、有效測定範圍應大於該項室內空氣污染物之室內空氣品質標準值上限。
 - 二、配有連續自動記錄輸出訊號之設備，其紀錄值應註明監測數值及監測時間。
 - 三、室內空氣經由監測設施之採樣口進入管線到達分析儀之時間，不得超過二十秒。
 - 四、取樣及分析應在六分鐘之內完成一次循環，並應以一小時平均值作為數據紀錄值。其一小时平均值為至少十個等時距數據之算術平均值。
 - 五、每月之監測數據小時紀錄值，其完整性應有百分之八十有效數據。
 - 六、採樣管線及氣體輸送管線材質具不易與室內空氣污染物產生反應之特性。

3.3 研究區域

本研究選定本校大義館五樓526教室為實驗區域，而主要選擇的原因為：

- (1) 位於室內空間；
- (2) 使用頻率高；
- (3) 具有室內空調；
- (4) 能接收到學校的Wi-Fi。

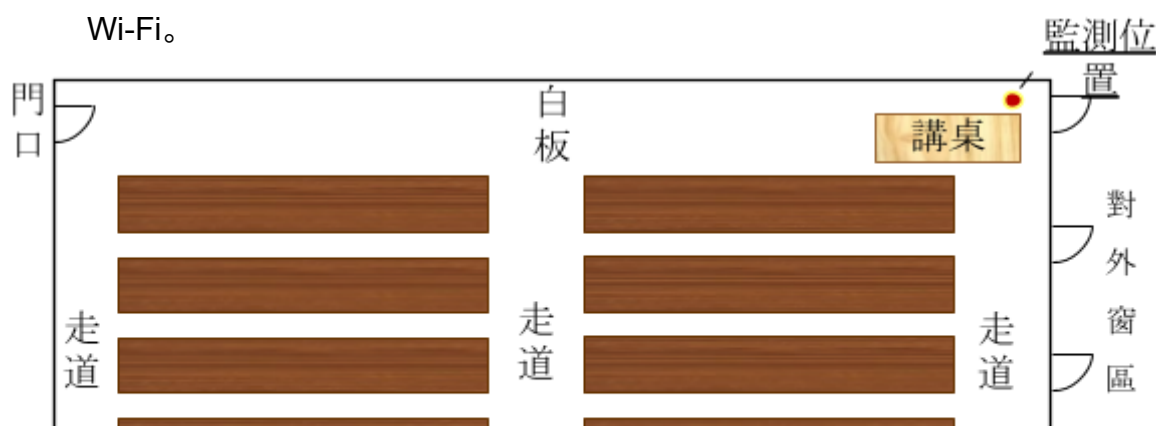


圖3.2 監測設備和研究區域示意圖



3.4 監測設備

3.4-1 樹莓派

樹莓派(Raspberry Pi)，是一款基於Linux的單晶片電腦。它由英國的樹莓派基金會所開發，目的是以低價硬體及自由軟體促進學校的基本電腦科學教育。樹莓派配備一枚博通(Broadcom)出產的ARM架構700MHz BCM2835處理器，256MB記憶體(B型已更新到512MB記憶體，本研究採用樹莓派3 B型)，使用SD卡當作儲存媒體，且擁有一個Ethernet、兩個USB埠、以及HDMI(支援聲音輸出)和RCA端子輸出支援。樹莓派面積只有一張信用卡大小，體積大概是兩個火柴盒大小，可以執行線上遊戲和進行1080p影片的播放。操作系統採用開源的Linux系

統: Debian、ArchLinux, 自帶的Iceweasel、KOffice等軟體, 能夠滿足基本的網路瀏覽、文字處理以及電腦學習的需要。樹莓派基金會提供了基於ARM架構的Debian、Arch Linux和Fedora等的發行版供大眾下載, 還計劃提供支援Python作為主要程式語言, 支援BBC BASIC、C語言和Perl等程式語言。樹莓派3的處理器更新為了64位元的博通BCM2837, 而根據 RPi FAQ的資料, BCM2835 的工作溫度範圍在 -40°C 到 85°C。

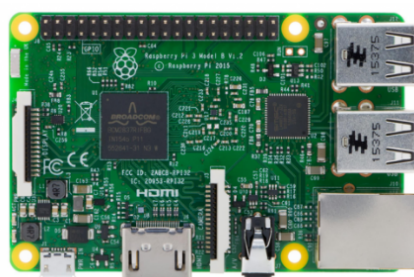


圖3.5 樹莓派
3.4-2 Debian作業系統

Debian是完全由自由軟體組成的類UNIX作業系統, 因其給予使用者的眾多選擇而聞名。現時Debian提供了超過25,000個軟體, 超過50,000個軟體包, 並正式支援10個電腦系統結構。其包含的多數軟體使用GNU通用公共許可協定授權, 並由Debian計劃的參與者組成團隊對其進行打包、開發與維護。

3.4-3 CJMCU-8128感測器

CJMCU-8128是集CCS811、HDC1080和BME280為一體的感測模組, 作為CO₂、VOCS、溫度、濕度和氣壓的氣體感測器, 大小為15mm * 21mm, CJMCU-8128氣體芯片系統的芯片(SSoC)和溫濕度感測器, 採用低功耗感測技術檢測室內揮發性有機化合物(VOCs)。

- (1) CCS811: 是由一家名為Adafruit Industries的開源硬件公司設計。能感測總揮發性有機物(TVOC), 包括等效二氧化碳(eCO₂)和MOX等級。此感測器主要用於室內空氣品質監測, 提供CO₂的百萬分率(PPM)讀數和TVOC的十億分率(PPB)讀數, 以及溫度值。
- (2) HDC1080: 數位濕度感測器, 是由一家名為Texas Instruments的半導體製造商所設計, 是一款具有整合式溫度感測器的數字濕度感測器, 其能以超低功耗提供出色的測量精度。支援較寬的工作電源電壓範圍, 並且該器件可為各類常見應用提供低成本和低功耗優勢。濕度和溫度感測器均經過

出廠校準。

- (3) BME280: 由Bosch Sensortec公司專為移動設備而開發設計之環境感測器，監測濕度和氣壓，具有極快的反應時間，小尺寸和低功耗是一大特點。該感測器可以用於室內氣候控制應用的智能手機，亦可整合到體育運動中進行高度分析，以實現更好的培訓績效監控等。

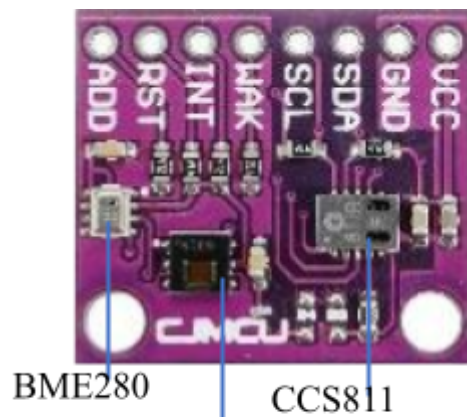
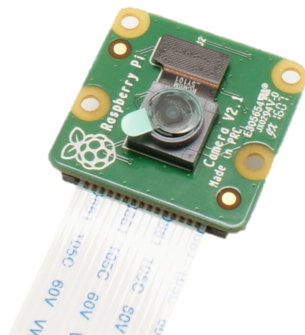


圖3.6 UMCU-8128電路板和感測器

3.4-4 樹梅派相機模組(Raspberry Pi Camera Module)

樹梅派相機模組由一個小的(25mm×20mm×9mm)電路板組成，通過柔性印刷電路板(FPC, Flexible Printed Circuit)連接到Raspberry Pi的相機串行接口(CSI, Camera Serial Interface)匯流排連接器。相機的感光元件具有800 萬像素的解析度，並具有固定焦距鏡頭。該相機的軟體支援靜止圖像的解析度最高可達2592x1944，影像解析度為1080p30、720p60和



640x480p60/90。

圖3.7 樹梅派相機模組

3.4-5 YOLOV3

YOLOV3 為YOLO 系列 (YOLO, You Only Look Once) 目前的最新版本, 是關於物件偵測 (object detection) 的類神經網路演算法, 延續前兩版的成果, YOLOV3 並沒有做革命性的創新, 而是參考其他的論文對本身的模型做優化。顧名思義, YOLO只需要在影像上執行卷積神經網路 (CNN, Convolutional Neural Network) 一次即可, 除此之外, YOLO使用單一神經網路, 以至於它可以直接優化端到端(end-to-end)的訓練, 如同圖3.7所示:YOLO系統將輸入的影像分為 $S * S$ 個網格大小, 而 S 的大小是可以做調整的, 每個網格可以預測出定界框 (Bounding Box) 數量與它們的機率值, 機率值可以反映出定界框中所標出的物件之可能性, 以及YOLO對該物件所預測的位置之準確性。YOLO以小眾架構 Darknet 實作, 實作該架構的作者 Joseph Redmon 沒有用到任何著名深度學習框架, 輕量、依賴少、演算法高效率, 在工業應用領域很有價值, 例如行人偵測、工業影像偵測等等, 圖3.8為YOLOV3與其他訓練模型的精度和執行速度比較。

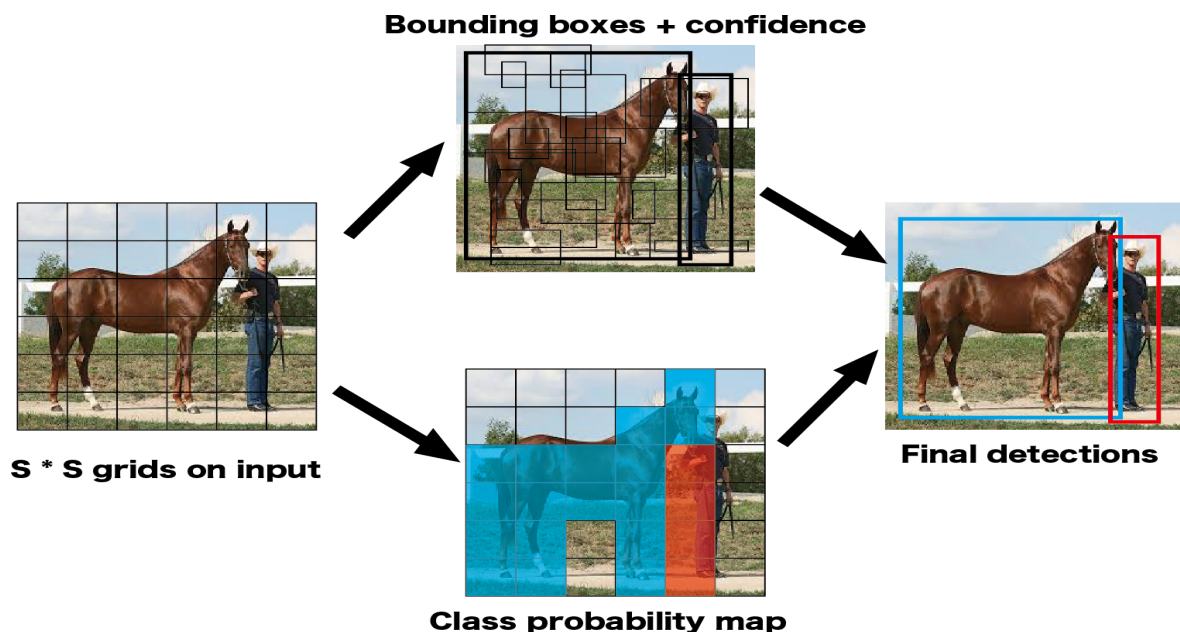
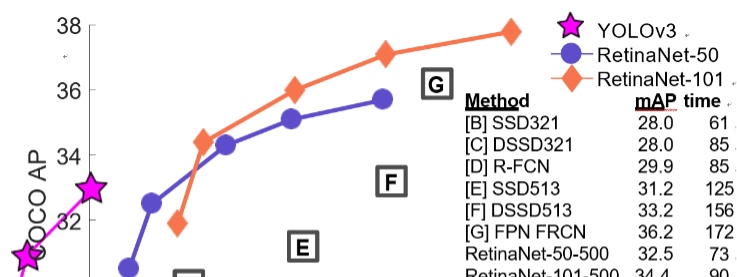


圖3.8 YOLO影像物件偵測示意圖(林家平, 2018)



The figure consists of two side-by-side photographs of a classroom. The left photograph is the original image, showing students seated at long wooden tables. The right photograph is the same scene with computer vision annotations. Numerous bounding boxes in various colors (blue, green, yellow, red, magenta) are drawn around objects in the scene. Each box is accompanied by a label and a percentage, indicating the model's confidence in its classification. Labels include 'person', 'chair', 'dining table', and 'table'. For example, a person in the foreground is labeled 'person 25%' and a chair next to them is labeled 'chair 49%'. The annotations are dense, covering most of the visible objects in the room.

3.4-6 ThingSpeak

ThingSpeak™

Channels ▾ Apps ▾ Community Support ▾ Commercial Use How to Buy Account ▾ Sign Out

airtest4

Channel ID: 660161
Author: chdajn123
Access: Private

Private ViewPublic ViewChannel SettingsSharingAPI KeysData Import / Export

+ Add Visualizations

+ Add Widgets

Export recent data

MATLAB Analysis

MATLAB Visualization

Channel Stats

Created: 4 months ago
Last entry: about a month ago
Entries: 922

Field 1 Chart

airtest4

680

Field 2 Chart

airtest4

42.5

圖3.11 ThingSpeak網頁 (https://thingspeak.com/channels/660161/private_show)

3.5 程式架構與說明

本研究所設計之程式主要分為三大部分，首先，為CJMCU-8128電路板上的三顆感測器所偵測到的氣體資訊，接著是PiCamera所拍攝影像的人數影像辨識，最後彙整監測資料、紀錄並上傳雲端。

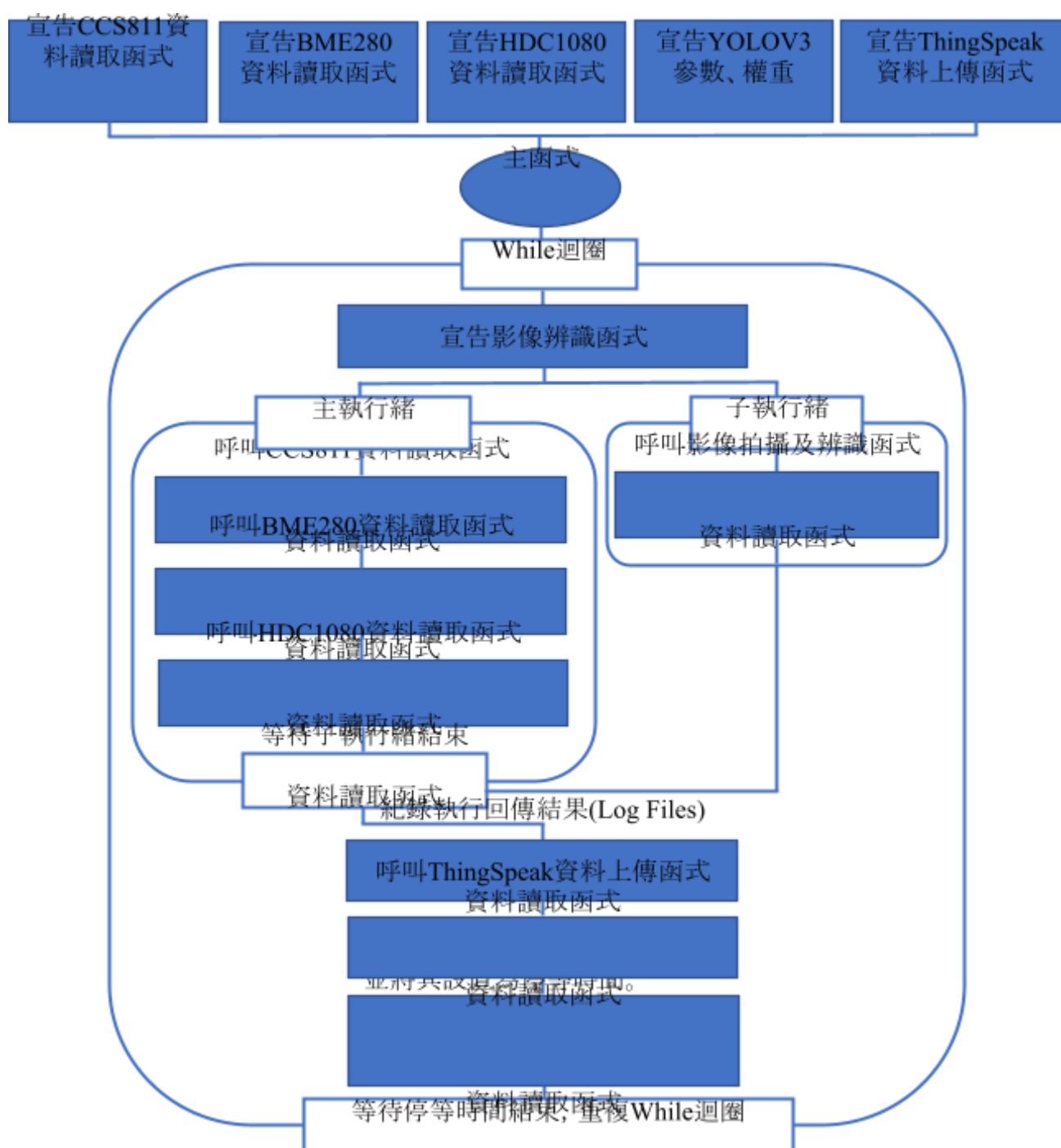


圖3.12 程式邏輯和架構圖

3.6統計方法

3.6-1 簡單線性迴歸分析

簡單線性迴歸分析用於探討單一自變數及依變數(連續變數)之間的關係，其估計式為： $y = a * x + b$ ，而誤差項需滿足三大假設：

- (1) 常態性(Normality)：本研究相信人數的變化，會對空氣造成直接影響，因此假設母體資料服從常態分配(Normal Distribution)。
- (2) 獨立性(Independency)：本研究相信造成人數與空氣變數對於預估值之間的誤差值，皆由獨立因素造成，兩兩之間互相不影響，所以假設誤差項之間相互獨立。
- (3) 變異數同質性(Constant Variance)：各組樣本必須取自變異數相等的母體。

由於本研究相信「人」是造成室內空氣汙染物濃度增加的主要關鍵，而是室內空調設備也需要根據空氣品質和人數進行調整，因此將人數設定為自變數，而空氣監測項目則設定為依變數。

3.6-2 決定係數 (Coefficient of determination)

決定係數在統計學中是用於衡量依變量的變異中可由自變量來解釋的所占比例，進而以此來判斷統計模型的解釋能力。 SS_{Reg} 為迴歸模式的變異量，是各個空氣變數的預估值到實際監測到的平均值之間的變異量； SS_{Total} 則稱為實際監測值的變異量。當兩者相除後的商數介於0到1之間，愈接近1者表示該模型愈符合期待、愈有解釋能力。因此決定係數寫成數學式即為：

$$R^2 = SS_{Reg} / SS_{Total}$$

本研究將藉由「決定係數」來進行監測資料之分析，以及評估系統運作和擺放位置的監測品質。

第四章 研究成果與分析

4.1 監測結果

根據本研究設計，藉由監測裝置所得到下列數據：1.CO2、2.TVOC、3.氣溫、4.濕度、5.人數等5項資料。

CO₂:

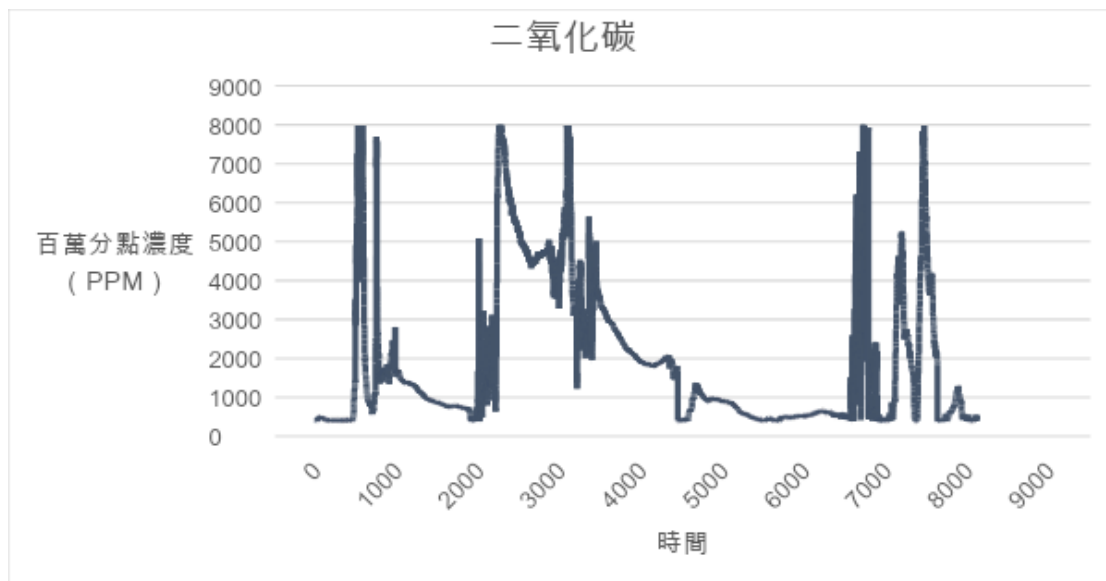


圖4.1 各時段CO₂濃度

TVOC:

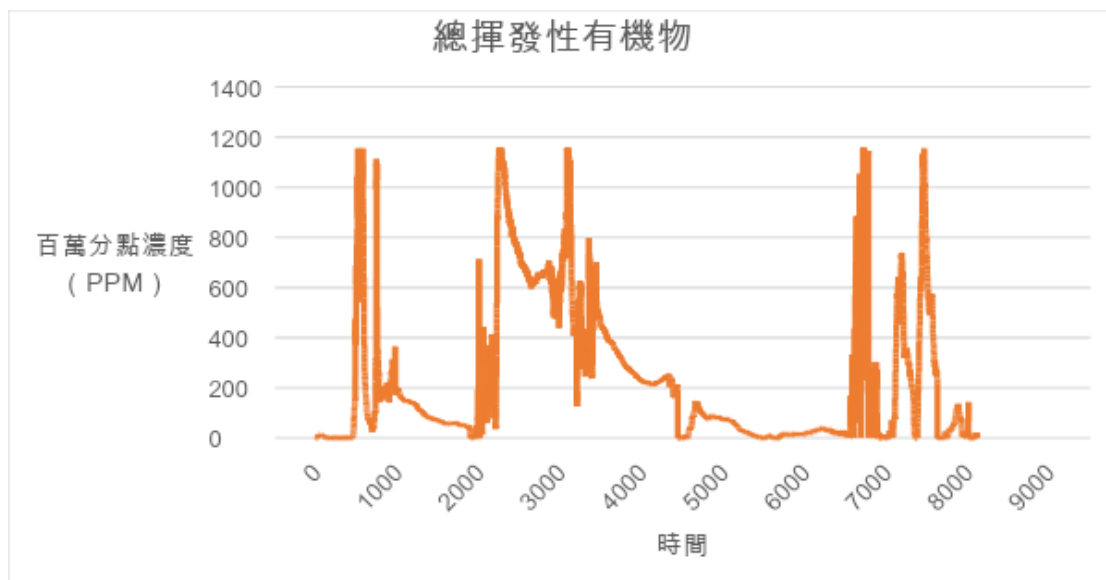


圖4.2 各時段TVOC濃度

氣溫:



圖4.3 各時段氣溫

濕度：

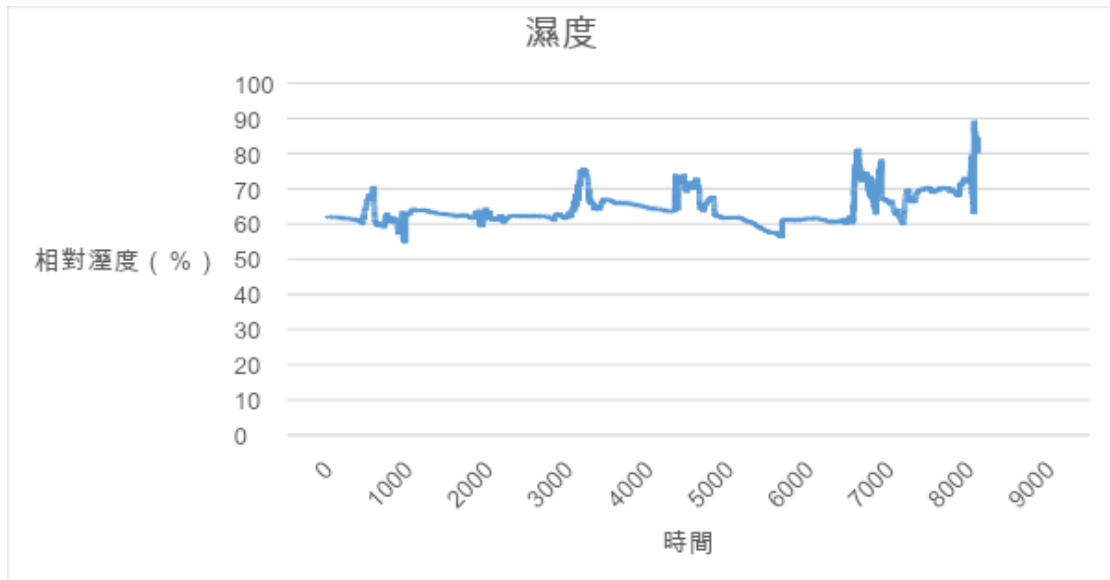


圖4.4 各時段濕度

人數：

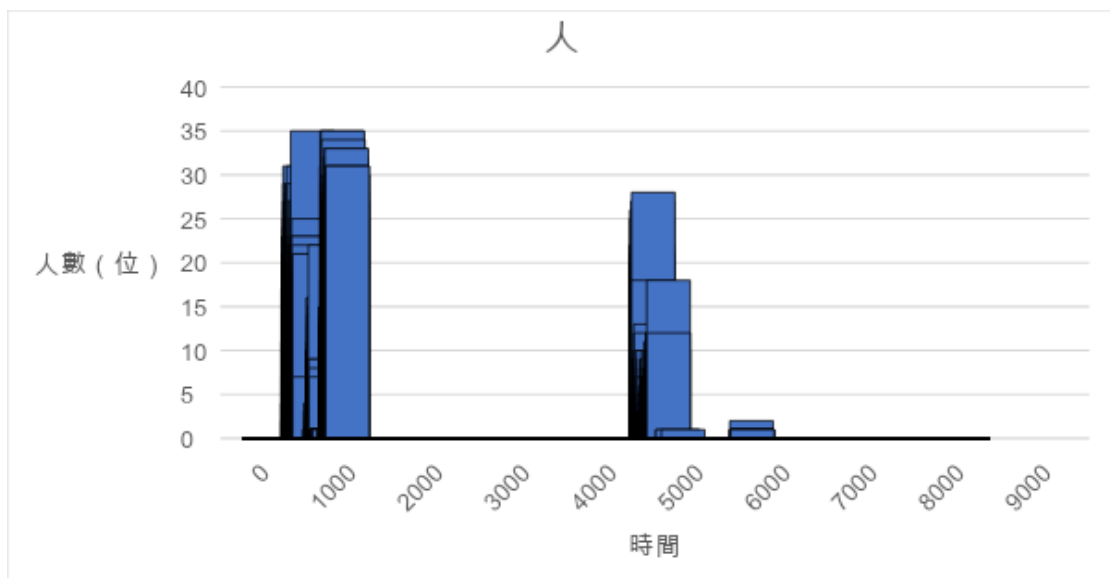


圖4.5 各時段人數

4.2 監測資料分析

由於監測時間為一個禮拜，部分天數沒有上課，以及在人數的辨識上有些錯誤，因此選擇人數較多、資料品質相對穩定的天數進行分析，分別為：2019年3月12日和2019年3月15日。將以上兩天的人數與四項影響空氣品質因素的資料在去掉極端值之後，做簡單線性迴歸分析。

4.2-1 簡單線性迴歸分析

以上兩天各有四張表格分別為：二氧化碳、總揮發性有機物、溫度、濕度等四項與人數所做的迴歸分析，如圖4.6至4.15。

根據3月12日的結果可以發現四項皆與人數有正相關，但相關性不高，可以歸納幾個影響相關性的原因，相關性高的：1.由於受限於教室人數的影像判釋需要良好的拍攝視野所以選擇在教室的邊緣架設儀器、2.儀器架設位置靠近窗邊，容易因為人為的行為而造成室內空氣成分改變，如：開啟窗戶、3.影像辨識有

一定的精度，而非百分之百準確，加上人類行為不可預測性，如：上課趴在桌上睡覺、轉頭和同學聊天等因素，增加了影像辨識的難度。以上幾點皆是影響3月12日相關性的可能原因。

根據3月15日的結果卻發現二氧化碳、總揮發性有機物以及溫度三者對於人數的迴歸直線斜率皆為負。當天早上10點為研究區域的第一堂課，而當天和前一天的室外各小時平均氣溫約在攝氏20度以下，我們可以假設研究區的對外窗那幾天並沒有開啟。因此我們可以推論：剛開始上課時，教室二氧化碳、總揮發性有機物維持一定濃度，而室內溫度也保持一定水準。但是隨著人數越來越多，室內空氣逐漸讓人感到悶熱，室內人員開啟空調設備或是開啟對外窗戶，而使得室內的二氧化碳和揮發性有機物的濃度逐漸下降，回到平常具有空調設備作用的空氣品質。對照12日和15日的數據，我們可已得出結論：當教室密閉且未開空調設備時，空氣汙染物質和人數多寡並無絕對關係，反而是受到空調關閉後，汙染物長時間累積所影響，而溫度、濕度則不盡然如此。

(1) 3月12日監測資料和簡單線性迴歸模型

The diagram illustrates a layout with two main rectangular regions, one on the left and one on the right, separated by a vertical line. Each region is divided into two horizontal sections by a horizontal line. The bottom section of the right region is further divided into two smaller horizontal sections by a horizontal line. The top section of the right region is labeled '1'.

圖4.6 3月12日各項監測資料

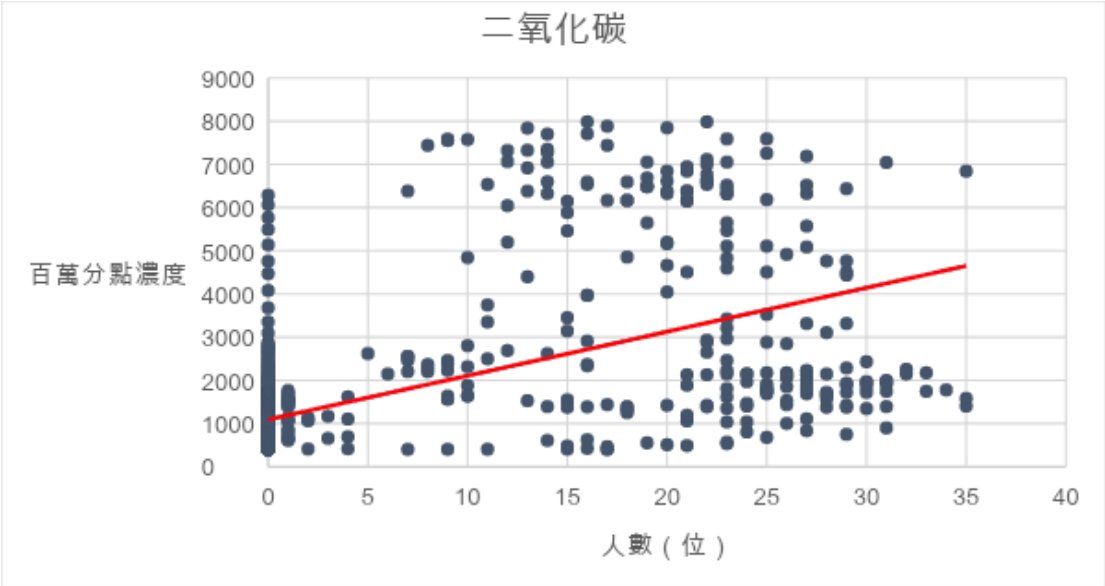


圖4.7 3月12日 CO2-人數迴歸分析

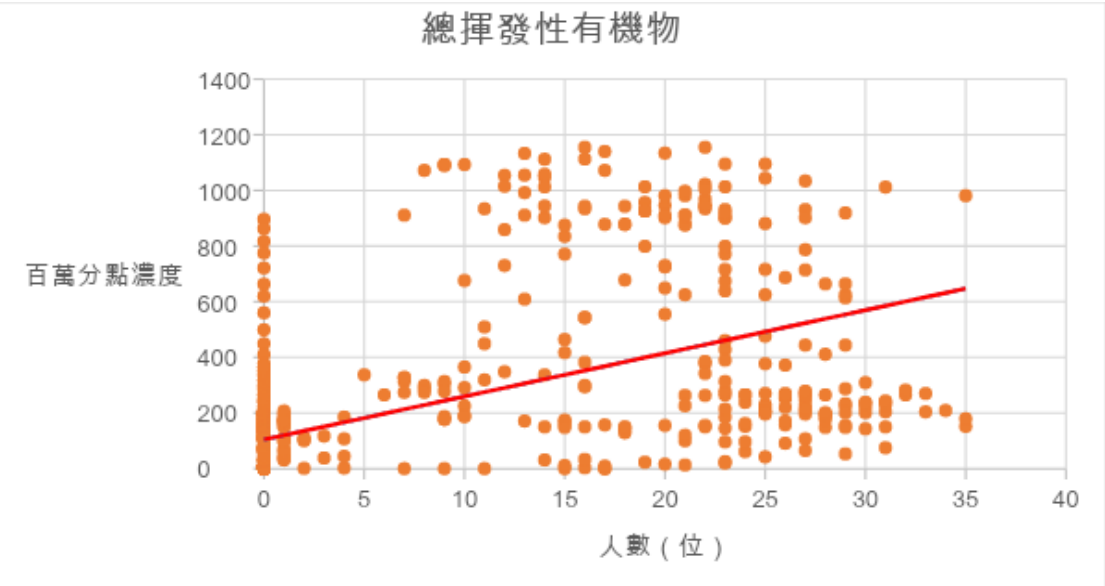


圖4.8 3月12日 TVOC-人數迴歸分析

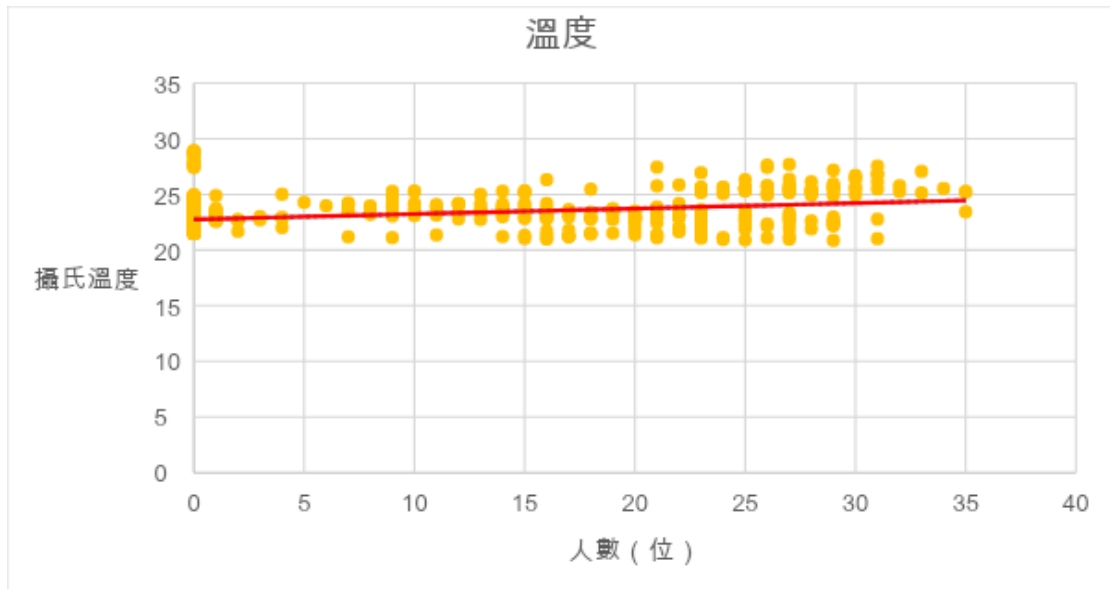


圖4.9 3月12日 溫度-人數迴歸模型

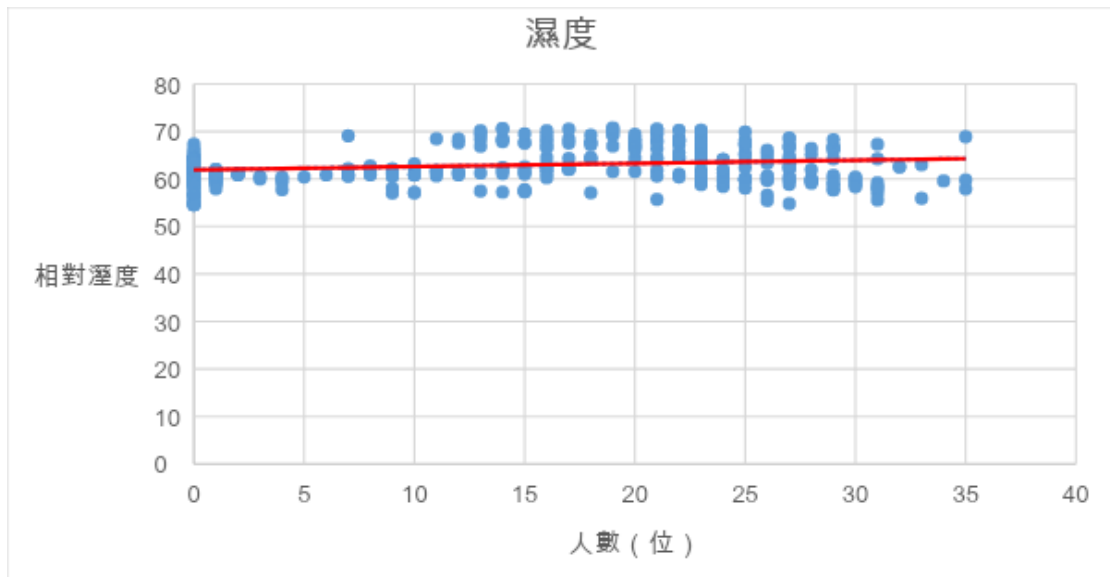


圖4.10 3月12日 濕度-人數迴歸模型

(2) 3月15日監測資料和簡單線性迴歸模型

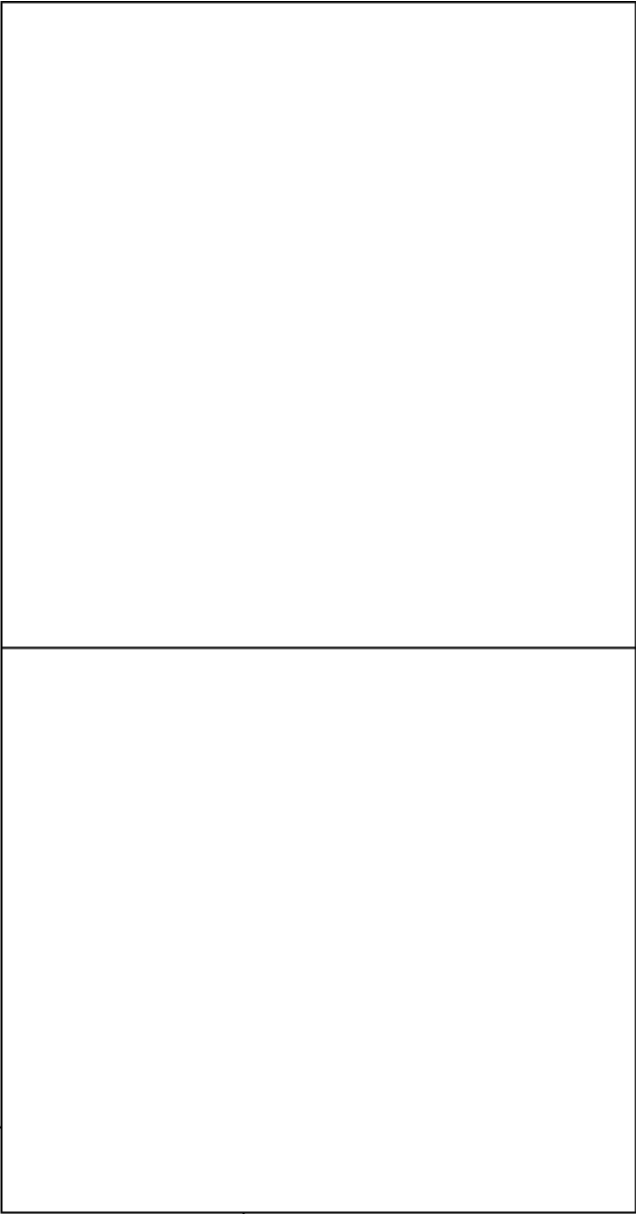
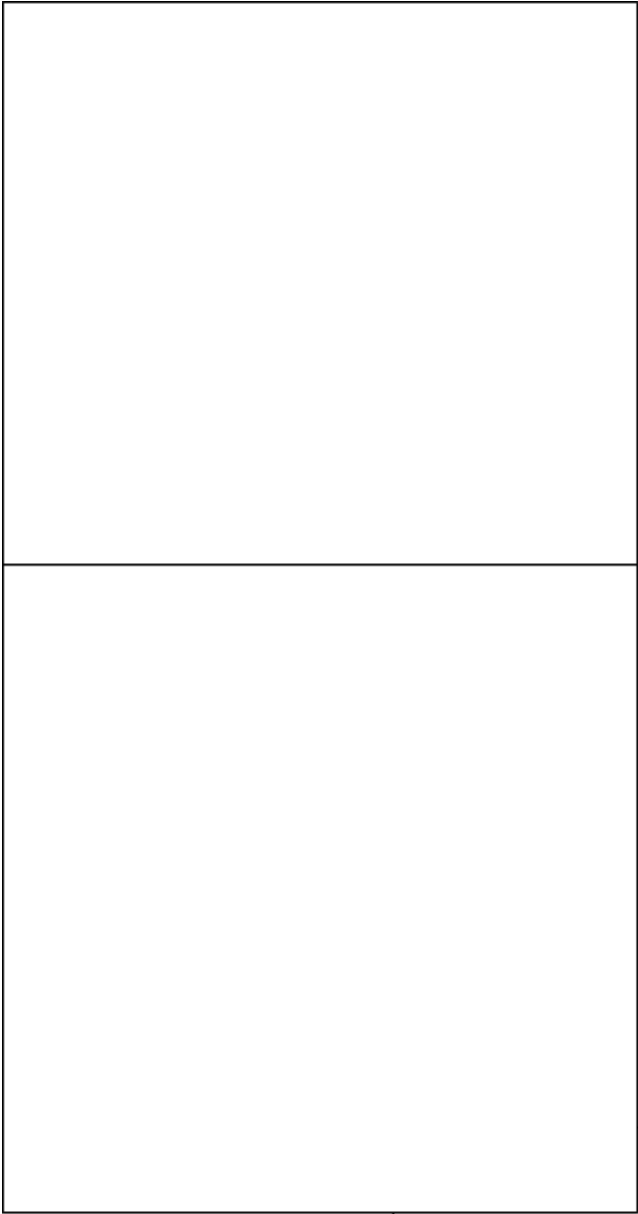


圖4.11 3月15日各項監測資料

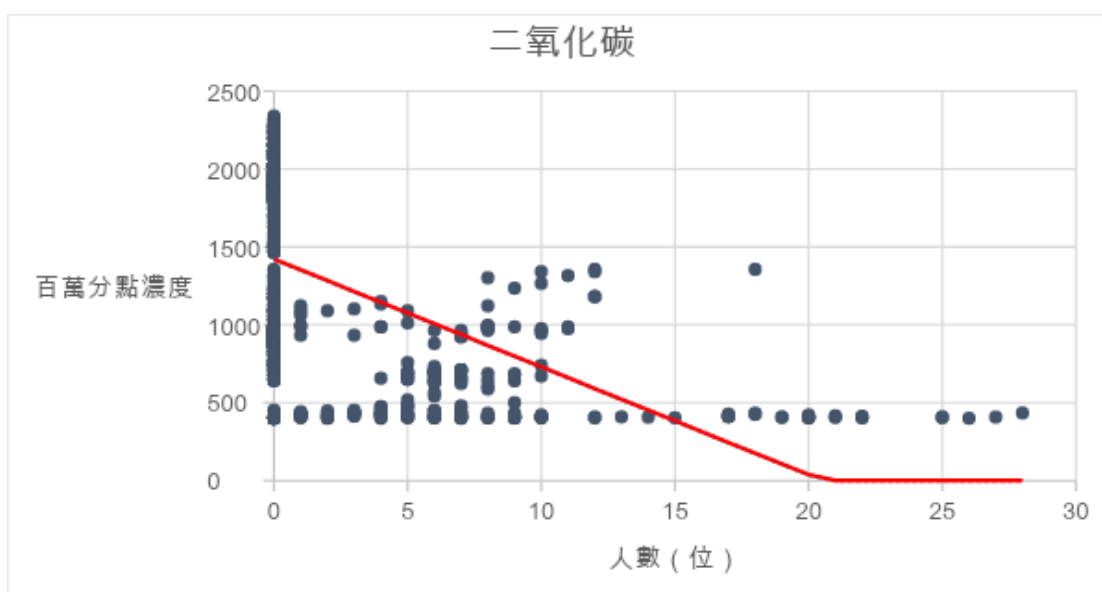


圖4.12 3月15日 CO2-人數迴歸模型

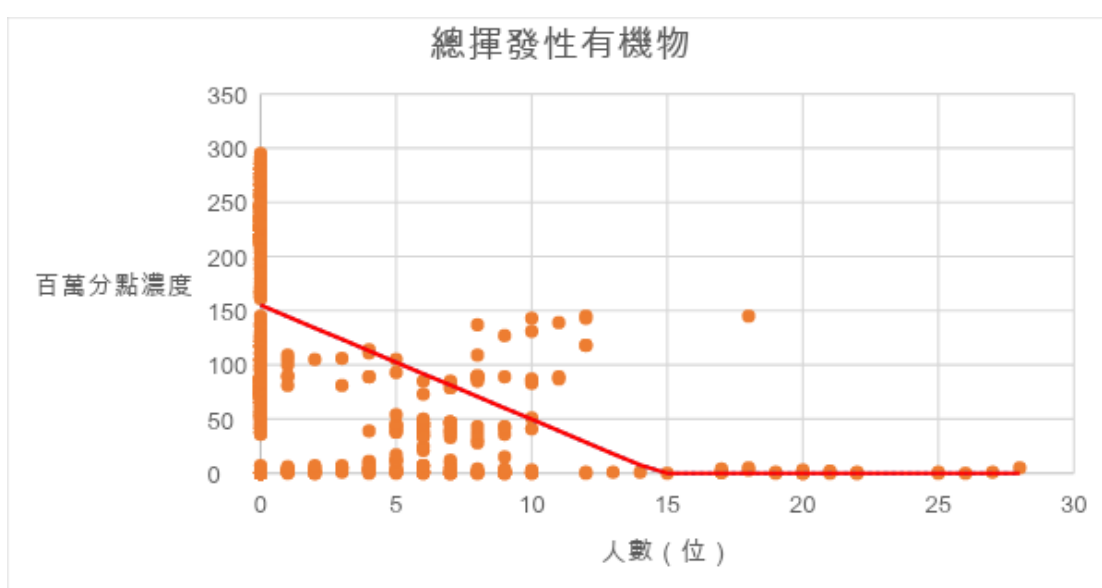


圖4.13 3月15日 TVOC-人數迴歸模型

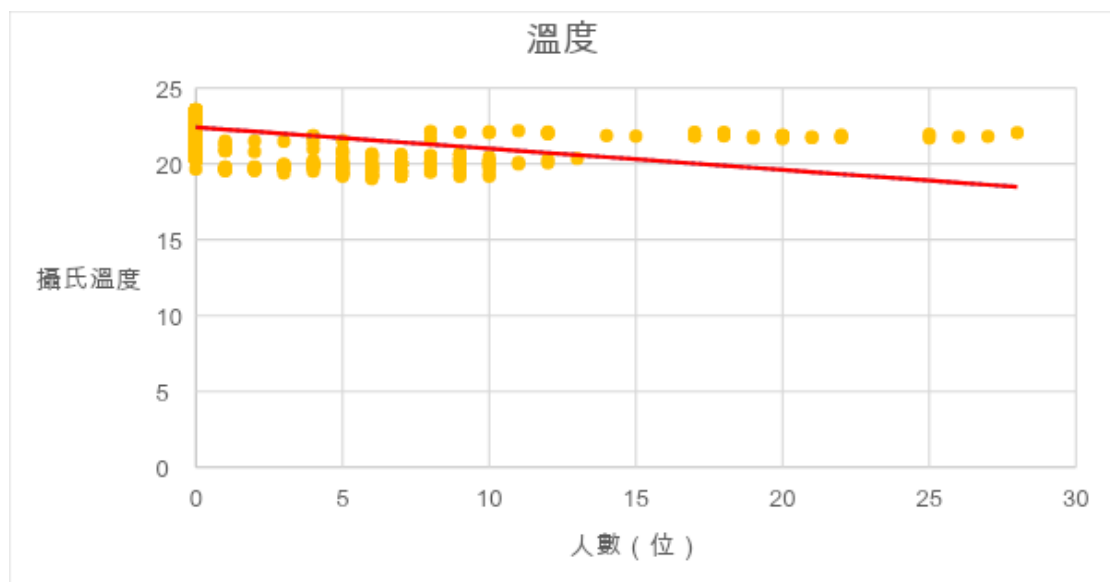
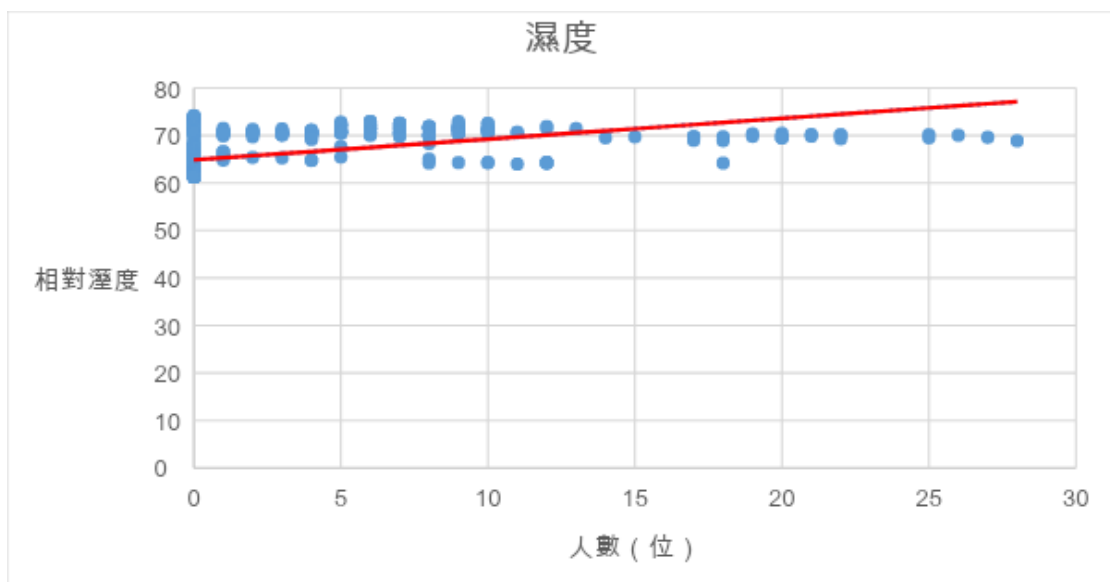


圖4.14 3月15日 溫度-人數迴歸模型

圖4.15 3月15日 濕度-人數迴歸模型



第五章 結論

5.1 結論

本研究所設計開發、用於監測校園的空氣監測系統，不僅可以與市面上的手持式檢測儀器一樣，具有相同的便利性，也具有將監測資料上傳雲端的功能，除此之外，本設備可以隨時擴充新的監測項目，不受限一機一功能的限制，更重要的是，所上傳到雲端的空氣資料，可以提供第三方WebService服務，學校可以根據需求，讓學生和教職員隨時隨地都可以從學校的網站上得到最新的室內空氣品質資訊。

本系統有著自動化運行的機制，不需要人為操作開關，因此即使突然遭遇斷電、恢復供電後，依然可以繼續執行任務，為24小時持續運作的裝置。雲端資料庫不僅可以顯示即時監測資訊，也同樣可以瀏覽或下載過去的歷史資料，提供之後研究者建立數據分析的基礎。除了主要的空氣監測項目之外，人數計算是本研究與過去文獻不同的部分，本研究利用影像物件偵測技術，辨識出當時室內概略人數，儘管準確率有可能受到外在因素的干擾，但卻增加了監測項目的豐富性，也是日後研究者可以用來作為數據分析和改善的重要依據。

由於本校尚未有室內空氣監測與空調設備整合的智慧空調系統，希望藉由本研究所設計出的監測系統，提供未來研究者參考方向。

5.2 建議

1. 本研究發現監測項目與人數的正相關性不高，可能與本設備擺放於窗戶旁邊的位置有關，未來可以對擺放位置進行調整。
2. 本研究只針對二氧化碳和總揮發性有機物、溫度、相對濕度進行監測，未來能夠增加其他空氣監測項目，讓校園室內空氣品質監測更完善。
3. 未來可以繼續將調整空調設備的系統完成，幫助校園完成智慧空調的目標和有效利用能源的方式。

參考資料

學術論文：

1. 王揚舜, 2010, 國民小學室內空氣品質影響因子及關聯性研究, 碩士論文,

樹德科技大學，建築與環境設計研究所，高雄。

- 2.李彥頤，2004，辦公空間室內空氣品質管制策略之研究，博士論文，國立成功大學，建築研究所，台南。
- 3.林聖家，2017，模組化感測器在環境偵測系統之物聯網應用，碩士論文，國立虎尾科技大學，光電工程系光電與材料科技碩士班，雲林。
- 4.林家平，2018，A YOLO-based Traffic Counting System，碩士論文，國立中央大學，資訊工程研究所，桃園。
- 5.洪維仁，2016，大學室內場所二氧化碳與甲醛濃度分佈探討，碩士學位論文，國立屏東科技大學，環境工程與科學系，屏東。
- 6.趙壽川，2018，新建醫院地下空間室內空氣品質之調查以美食商場與病歷室為例，碩士論文，天主教輔仁大學，公共衛生研究所，新北市。
- 7.蔡佳哲，2016，改善與監控室內空氣品質物聯網系統，碩士論文，國立高雄應用科技大學，電子工程系，高雄。
- 8.賴威文，2016，運用二氧化碳感測器實現不具隱私侵犯之室內人體偵測研究，碩士論文，大同大學，通訊工程研究所，台北。
- 9.刘畅，郑云昊，刘兆荣，要茂盛，2016，人体呼出气中内源性挥发性有机物筛选研究，研究報告，北京大学环境科学与工程学院，环境模拟与污染控制国家重点实验室，北京。
10. Joseph Redmon & Ali Farhadi, 2018, YOLOv3: An Incremental Improvement
11. Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn & Andrew, 2009, Zisserman The PASCAL Visual Object Classes (VOC) Challenge

網路資料：

1.系統上傳雲端網址：

<https://thingspeak.com/channels/631960?fbclid=IwAR0cDDaJip49fRZ5v6ZJfotb6zC9KcJFJVD3p5zclIYEdxEuakxP-oCKYCM#.XNPYEWtAkwU>

2.室內空氣品質，2018/10/05：

https://en.wikipedia.org/wiki/Indoor_air_quality

3.二氧化碳，2018/10/05：

https://en.wikipedia.org/wiki/Carbon_dioxide

4.揮發性有機物, 2018/10/05:

https://en.wikipedia.org/wiki/Volatile_organic_compound

5.溫度, 2018/10/06:

<https://en.wikipedia.org/wiki/Temperature>

6.濕度, 2018/10/06:

<https://en.wikipedia.org/wiki/Humidity>

7.YOLO官方介紹網站, 2019/01/20:

<https://pjreddie.com/darknet/yolo>

8.YOLO — You Only Look Once 介紹, 瀏覽於:2019/01/20:

<https://medium.com/@chenchoulo/yolo-%E4%BB%8B%E7%B4%B9-4307e79524fe>

9.處理影像的利器 -- 卷積神經網路(Convolutional Neural Network),

2019/03/17:

<https://ithelp.ithome.com.tw/articles/10191820>

附錄

本研究所撰寫之程式碼:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys, time, urllib2
from Adafruit_CCS811 import Adafruit_CCS811
from Adafruit_BME280 import BME280_OSAMPLE_8
from Adafruit_BME280 import BME280
import SDL_Pi_HDC1000
import datetime
from picamera import PiCamera
from subprocess import Popen, PIPE
```

```

import threading
from time import sleep
import os, fcntl
import cv2
from shutil import copyfile

iframe = 0

camera = PiCamera()

camera.resolution = (2592, 1700)
#camera.rotation=180
#camera.capture('frame.jpg')
#sleep(0.1)

yolo_proc = Popen(["./darknet",
                  "detect",
                  "./cfg/yolov3-tiny.cfg",
                  "./yolov3-tiny.weights",
                  "-thresh","0.1"],
                  stdin = PIPE, stdout = PIPE)

fcntl.fcntl(yolo_proc.stdout.fileno(), fcntl.F_SETFL, os.O_NONBLOCK)

iframe = 0

def thingSpeak(CO2, TVOC, Temperature, Humidity, AtmosphericPressure,
Human):
    print 'Sending to ThingSpeak API...'
    url = "https://api.thingspeak.com/update?api_key="
    url += THINGSPEAK
    url += "&field1="
    url += str(CO2)
    url += "&field2="
    url += str(TVOC)
    url += "&field3="
    url += str(Temperature)
    url += "&field4="

```



```

url += str(Humidity)
url += "&field5="
url += str(AtmosphericPressure)
url += "&field6="
url += str(Human)
#print url
try:
    content = urllib2.urlopen(url).read()
except urllib2.HTTPError:
    print "Invalid HTTP response"
    return False
print 'Done'
#print content

ccs = Adafruit_CCS811()
sensor = BME280(
    t_mode=BME280_OSAMPLE_8,
    p_mode=BME280_OSAMPLE_8,
    h_mode=BME280_OSAMPLE_8,
    address=0x76
)
def ccs811example():
    """Get the data from the CCS811 sensor module and return it"""

    #if ccs.available():
        #temp = ccs.calculateTemperature()
    if not ccs.readData():
        co2 = ccs.geteCO2()
        tvoc = ccs.getTVOC()
    else:
        print "ERROR!"

    #Print Data information on screen
    print('CO2:{0} ppm, TVOC:{1} ppm'.format(co2, tvoc))
    return co2,tvoc

def bme280example():
    """The address parameter is important here as most BME/P280 sensors

```

occupy the default 0x77 address as noticed in the Adafruit library.

This can be easily verified by running `i2cdetect -y 1` on your Pi""

```
temp = sensor.read_temperature()
pascals = sensor.read_pressure()
pres = pascals / 100
humd = sensor.read_humidity()
#Print Data information on screen
print('Temperature:{0} degC, Pressure:{1} hPa, Humidity:{2}
%'.format(temp, pres, humd))
return temp,pres,humd
```

```
def hdc1080examples():
    hdc1000 = SDL_Pi_HDC1000.SDL_Pi_HDC1000()
    temp = hdc1000.readTemperature()
    humd = hdc1000.readHumidity()
    #Print Data information on screen
    print('Temperature:{0} degC, Humidity:{1} % '.format(temp, humd))
    return temp,humd
```

```
if __name__ == '__main__':
    THINGSPEAK = '357YZP46SGG344K5'
    while(1):
        t = time.strftime("%Y-%m-%d %H:%M:%S")
        print "Program Started at:" + t
        personCount = 0
        #camera.rotation=180
```

```
def picamera():
    #Camera
    camera.rotation=180
    global personCount, iframe
    try:
        stdout = yolo_proc.stdout.read()
        if 'Enter Image Path' in stdout:
            #camera.rotation=180
            camera.capture('frame.jpg')
            yolo_proc.stdin.write('frame.jpg\n')
```

```

        if len(stdout.strip())>0:
            #print('get %s' % stdout)
            #print()
            if 'person' in stdout:
                personCount = stdout.count('person')
            else:
                personCount = 0
            print("personCount:", personCount)

    except Exception:
        pass

#Create a subthread
t1 = threading.Thread(target = picamera)
#Start the execution
t1.start()
#picamera()
#AirSensor
ccs811 = ccs811example()
bme280 = bme280example()
hdc1080 = hdc1080examples()
Time = datetime.datetime.now()
t1.join()

#Including time, Co2, tvoc, temp, humd, pressure.
data = (str(t), "\n", str(ccs811[0]), ",", str(ccs811[1]), ",",
str((bme280[0]+hdc1080[0])/2), ",", str(hdc1080[1]), ",", str(bme280[1]), ",",
str(personCount), "\n")
f = open("AirProject_log.txt", "a")
f.write("".join(data))
f.close()
print(data)
print "-----"
if(THINGSPEAK is not False):
    thingSpeak(ccs811[0], ccs811[1], (bme280[0]+hdc1080[0])/2,
hdc1080[1], bme280[1], personCount)
#Capture Time, on the minute

```

```
now = datetime.datetime.now()
now_sec = now.second
sec_to_zero = 59 - now_sec
time.sleep(sec_to_zero)
```