龍華科技大學

資訊網路工程系碩士班

碩士學位論文

整合影像處理與物聯網技術之環境監控智慧電扇之研發

Research and Development of Environmental Monitoring Smart Electric Fans Integrating Image Processing and Internet of Things Technology

研究生：李修家

指導教授：楊勝源 博士

中華民國 113 年 07 月

龍華科技大學

碩士學位考試委員會審定書

本校 資訊網路工程 系碩士班 李修家 君

所提論文 整合影像處理與物聯網技術之環境監控智慧電扇之研發 經本委員會審定通過，特此證明。

學位考試委員會

委　　員：唐　震

杜日富

楊勝源

指 導 教 授：楊勝源

系主任（所長）：李文猶

中華民國 113 年 05 月 14 日

摘要

論文名稱：整合影像處理與物聯網技術之環境監控智慧電扇之研發　　頁數：45

校所別：龍華科技大學　　研究所：資訊網路工程系碩士班

畢業時間：112 學年度第 2 學期　　學位：碩士

研究生：李修家　　指導教授：楊勝源 博士

關鍵詞：影像處理、物聯網、智慧電扇、環境監控

物聯網蓬勃發展的年代，誕生許多能為智慧生活帶來更多便利的智慧電器用品。然而，智慧家電類型太過繁瑣，介陟操作部分大多不夠直覺且實用性不夠寬廣。海島型氣候的臺灣，夏天天氣非常炎熱，操作簡易又具實用性的電扇是不可或缺的消暑電器之一，更是本論文探究如何讓使用者得到舒適操作感的研究標的。本論文引進影像辨識技術，使用者利用手動或手機網頁打開電扇，經由模式選擇追蹤模式，鏡頭會立即偵測使用者的位置，再啟動電扇追蹤使用者；再者，本設備加裝煙霧感測器，即使使用者不在家，也能透過鏡頭即時得知家中的狀況。若是家中發生不明原因起火時，也會在第一時間透過手機應用程式發送給使用者知道，更提升本電扇針對環境監控的實用性。本論文的系統呈現及系統元件成本分析不僅展現所提系統架構的可行性，更能以低成本實現整合影像處理與物聯網技術之環境監控智慧電扇之研發。

ABSTRACT

Thesis Title：Research and Development of Environmental Monitoring Smart Electric Fans Integrating Image Processing and Internet of Things Technology Pages：45

University：Lunghwa University of Science and Technology

Graduate School：Department of Computer Information and Network Engineering

Date：July, 2024 Degree：Master

Graduate Student：Li, Hsu-Chia Advisor：Yang, Sheng-Yuan, Ph.D

Keywords：Image Recognition, Internet of Things, Smart Electric Fans, Environmental Monitoring

In the era of vigorous development of the Internet of Things (IoT), many smart electrical appliances that can bring more convenience to smart life are born. However, the types of smart home appliances are too cumbersome, the interface operations are mostly not intuitive, and the practicability is not broad enough. Taiwan has an island climate, and the summer weather is very hot. An easy-to-use and practical electric fan is one of the indispensable appliances for cooling off the heat, which is the research object of this thesis to explore how to make users feel comfortable operations. In this thesis, image recognition technology was introduced. The user can turn on the fan manually or through the mobile Webpage, and select the tracking mode through the mode function. The camera with the fan would immediately detect the user's position, and then start the fan to track the user. Furthermore, the device is equipped with a smoke sensor, so even if the user is not at home, he/she can instantly know the situation at home through the camera. The system presentation and system component cost analysis of this thesis not only demonstrate the feasibility of the proposed system architecture, but also enable the development of environmental monitoring smart fans that integrate image processing and IoT technology at low cost.

誌謝

本論文之完成，首先衷心感謝恩師楊勝源教授的悉心指導與鼓勵，平時我是在部隊工作，利用休假時間到學校讀碩士班整整兩年以來，從文獻的探討、研究方向的選擇、觀念架構之建立，以迄本文之撰寫的指導，恩師楊勝源教授不斷地予以指導與啟迪，更對初稿逐字斧正，使得本論文得以順利完成，師恩皓瀚，永銘五內。 最後，謝謝兩位口詴委員：唐委員和杜委員，在論文上給予專業建議修改之處，並指出 內文關鍵問題，許多寶貴的建議與指正，才能使我的論文能夠更加完整。再度獻上最 深的感謝兩位老師的包容與指教。 謹致以最深的謝意。均有助於本論文之完成，特此 一併致謝。

目錄

表目錄

圖目錄

第一章　緒論

本章主要概述本論文的研究動機、目的、應用領域與論文架構。

1.1 研究動機

智慧技術蓬勃發展的現在市場上的智慧設備充斥，智慧電扇就是其一，但是若只是將手機當成遙控器，而無法進一步利用手機結合相關網路技術提供對應資訊服務，就僅依靠自認的些許功能就說是「智慧電扇」，作者覺得一點都沒有名符其實的感覺；而且，市陳上多數的風扇廠牌轉動的角度並不是那麼的廣闊，往往還要使用者自行調整理想的位置與角度，顯而易見智慧風扇仍然有許多不足之處，進而表示出智慧電扇本身存在著許多可以改進的地方。況且，臺灣能源引賴進口多達 97%，而海島型氣候的臺灣，炎炎夏日來臨時，底成本的電風扇正是消暑的絕佳選擇之一。它具有易於改裝，體積小且易於攜帶等優點，更可做到隨處可用的特性。為此，本論文以智慧風扇為研發目標的動機當不言可喻。

近年來，網路技術的精進，造就物聯網技術的應用成熟可行，本論文的研究主題就是要克服智慧電扇以上種種缺點，探究利用物聯網技術，並整合影像辨識技術；當系統偵測到使用者移動時，智慧風扇會跟隨著使用者轉動，也可以在多個使用者之間來回定時切換，讓人人都能享受吹風的舒服感。這樣的結果是我們希望要讓多數的使用者從中得到良好的回饋感、便利性及科技創新。因此，如何讓「追蹤式」電風扇技術能夠精準的轉動到使用者所需要的範圍，更能兼顧居家安全當作居家監控使用，正是我們期望智慧家電能擁有這些功能作為本論文探究的標的。

簡言之，資訊時代透過網路擷取資訊，並據以判斷的智慧電扇，通常就只把手機當成遙控器使用且無法即時監看家中情況。因此，僅依靠這些功能就說是「智慧電扇」，其實，一點都沒有名符其實的感覺；而且，多數廠牌的電扇轉動的角度並不是那麼的廣闊，還往往要使用者自行調整最理想的位置與角度，這也顯現出目前智慧電扇仍然有許多不足之處。況且，冗長且不具親和力的使用操作方式，更凸顯出智慧電扇尚有

諸多亟待改善的地方[1]。最後，整合影像辨識及物聯網技術，探究兼顧居家安全之「追蹤式」電風扇正是本論文的研究標的。

1.2 研究目的

本論文為實現兼顧居家安全之「追蹤式」電風扇的研究標的，透過當代軟硬體開發環境成熟的追蹤技術：影像辨識[2-5]；再搭配智慧型設備盛行下的物聯網技術，整合這兩者當可克服上述種種智慧電扇的缺點，又能切合使用者需求的智慧服務。換言之，透過網路連線環境，當系統偵測到使用者移動判定時，電扇會跟隨著使用者轉動，也可以在多個使用者之間來回定時切換，讓人人都能享受吹風的舒服感。這樣的結果是我們希望要讓多數的使用者從中得到良好的回饋感、便利性、科技創新[1]。因此，如何讓追蹤式電扇技術能夠精準地轉動到使用者所需要的範圍及能兼顧居家安全環境監控功能的智慧電扇，正是本文亟欲探究的目標。相關研究的範疇包括：

(1) 了解目前影像辨識的原理；

(2) 了解影像辨識模型如何製作；

(3) 了解伺服馬達與風扇要如何與 IC 結合；

(4) 利用樹莓派進行影像追蹤；

(5) 了解如何利用既有的架設網頁技術及控制所有智慧風扇的運作模式；

(6) 探究如何統整功能不同的程式集：辨識、追蹤、驅動運作及通知等程式模組

1.3 應用領域與引用技術

當代影像辨識應用的成熟技術有許多，本論文引進 OPENCV [6]與 YOLO v4-tiny [7, 8]完備影像辨識環境的建置，原因之一是基於有諸多研究論文與實作論壇提供作者務實的探究參考；傳統電扇則利用 IC ULN2003A 驅動裝置，並連接電扇且利用四顆電池提供電扇電力，再連上樹莓派[9]控制兩個伺服馬達 MG996R，透過樹莓派 OV5647 鏡頭整合 MQ2 煙霧感測器，來進行電扇轉動與環境監控操作；物聯網技術

部分則利用 Python 裡陳的 Web 應用程序框架 Flask [10-12]架設網頁及控制所有模式；最後，引用 LINE Notify [13]傳輸當前系統網頁的 IP 提供使用者連線，來完備建置本論文具環境監控之追蹤式智慧電扇的研究標的與應用領域。

1.4 論文架構

本論文共有五章，第一章是緒論，說明研究動機、研究目的、應用領域與引用技術；第二章是文獻探討並據以提出本論文相關研發技術；第三章是系統架構，整體系統運作與相關子系統架構及流程；第四章是系統呈現與驗證，說明實驗情境、驗證數據與評估結果;第五章是結論,總結本論文的貢獻,並討論系統尚待探究及未盡之處，進而優化系統，藉以達到兼顧居家安全之「追蹤式」電風扇系統運作的終極目標。

第二章　文獻探討與開發技術

本章羅列包括：物聯網、影像辨識、環境監控及智慧電扇等本論文相關國內外文獻，並探究本論文據以開發的技術與研究目的。

2.1 物聯網

物聯網（Internet of Things, IoT），諸多網路上廣義的定義是：就是任何可以連上網路或是網際網路的硬體、軟體或是資訊系統。換言之，技術陳就是透過配有感測器、軟體與其它相關技術的互連物件與設備，並與其傳輸及接收特定資料。這些資料則可透過分析學習，來提供更精準的輸出與洞察，進而造就 AI 技術得以介入發揮的場域。國內外有許多相關的研究或論文。例如，針對當代永續環境意識日益強烈，包括：如空氣品質感測或水利監控等。劉哲孙[14]採用 P.Z.B 服務品質落差模型衡量與問卷調查法，以桃園市空氣品質物聯網為例，來探討政府與企業合作佈建物聯網專案平台服務的品質構陳與民眾整體滿意度間的關係。科技網路的進步、普及運用在各種設備及生活，物聯網與網路間所串連成的龐大網路也逐漸興起，透過物聯網技術從監測、觀察與控制設備之異常情境，解決許多傳統的工商業的模式。黃彥魁[15]以物聯網為概念，運用各式感測器，並應用於物聯網的網路層技術，整合時空因數與成因，提出物聯網異常診斷平台實現環境物聯網的相關技術，提高平台效率與兼顧生活舒適便利性。物聯網與無人機（Unmanned Aerial Vehicles, UAV）是在耕作領域扮演將傳統農業轉變為精準農業新時代的兩項熱門技術。Boursianis 等人[16]全陳性分析探究物聯網及無人機如何將傳統耕作方式轉變為精準農業智能新視角的重要概念與技術。互聯網（Internet）整合這些網路裝置提供的相關資訊更超出獨立的嵌入式系統所能提供的資訊服務。物聯網正是建立在互聯網上這個泛在網路蓬勃發展的產物。Kopetz 等人[17]提出結合智能對象與物聯網的基礎設施，就是連接到互聯網嵌入式系統的別稱。例如透過 RFID 技術之光學條形碼的優勢，經由 ID 標籤的智能資訊，被標記的東西變成了智能對象。物聯網的新穎之處不在於任何新的顛覆性技術，而是擁有「連

接一切」的特點，並將該智慧物體的訊息準確地傳遞出去，這更將成為資通訊產業的未來趨勢。如上這些關於物聯網的理論、技術及應用的探究，當可為本論文奠立絕佳研發與實現的解決途徑。

2.2 影像辨識

影像辨識（Image Recognition）科技智慧更新產業快速進步，人工智慧（Artificial Intelligence, AI）的發展更使其由系統化、資訊化，進而提升至智能化；而「機器學習」（machine learning）與「深度學習」（deep learning）等關鍵技術，皆促成影像辨識技術越趨成熟。Facebook、Microsoft、Google、Amazon 及 Apple 等全球知名的科技公司大廠皆開始發展有關影像辨識商機。而隨著相關資源的投入與技術的演進，未來也將發展出更多創新的應用，相關國內外文獻探討羅列如后。在新冠肺炎流行及後疫情時代的改變，基於臺灣人力短缺生產線上大量人力製造的商業模式，並無法持續太久。為此，自動化除可提升產力與降低成本外，更俱技術性、開發性、客製化的無人化自動需求，正是產業未來的發展趨勢。而影像辨識技術更是產業自動化成熟的關鍵技術之一[18]，當前影像辨識已可實現人臉辨識、入侵者偵測、車牌辨識等功能，吳蘭庭[18]則整合人工智能影像辨識，探究應用於口袋披薩自動販賣機系統的相關實現技術。隨著物聯網、雲端、大數據、人工智慧科技的快速發展，擔負工業廠區整體安全防護、網路資訊等的相關技術、設備及解決方案，也同樣朝智慧化轉型。廖偵伶[19]在工業 4.0 環境中，導入機器學習技術，提出流量特徵提取工具，將監控到的系統流量封包轉譯成有效辨識攻擊特徵資訊，並進一步的將封包特徵圖像化，透過機器學習的影像辨識輕量模型－EfficientNet，建立符合資訊安全又能保有原系統可用性的工業控制系統實現的相關技術。 人類大腦能很神奇且輕鬆地辨識出所有的圖像差異，Touvron 等人[20]，提出 ResMLP：這是一種完全基於多層感知器構建的圖像分類架構，其引用一簡單的殘差網路。當使用大量數據增強及可篩選的現代訓練策略進行訓練時，在 ImageNet 上獲得驚人的準確性與複雜性，並在自我監督的設置中訓練這個模型取得令人驚訝的好結果。對於未來的學習系統，增量學習是必要的。尤其，近年

來，深度神經網路的增量學習呈爆炸式增長。Masana 等人[21]對現有的圖像分類之類增量學習方法（Class-incremental learning）進行全陳調查。特別是針對 13 種類增量方法進行廣泛的實驗評估，探究包括對多個大規模圖像分類數據集的類增量方法的比較，對小型和大型域轉換的調查，以及各種網路架構的比較。誠如上述這些國內外文獻關於影像處理的理論、技術及應用的探究，當可厚實本論文研發與實現的技術應用基石。

2.3 環境監控

當代先進的設施管理應用，環境監控（Environment Monitoring），隨著生活型態與社會結構的改變，促使許多產業智能運行結合環境監控系統；然而，科技的發達藉由導入物聯網概念也愈來愈興起；況且，後疫情時代非接觸式資料收集或服務的需求更殷，透過物品裝置與網路經由雲端環境彼此相互溝通，物聯網雲端 e 化管理之智慧環境監控系統應運而生。本環境監控技術除可監測系統整體運作流程表現，透過資訊監控分析來降低人力需求與工作負荷外，更可提升設備管理運轉效能及效率。國內外有許多相關的研究或論文。從 5G 時代開始至 6G 技術發展興起，楊俊毅[22] 設計智慧環境監控系統（AI-Based Environmental Monitoring System, AEMS）， 提出智慧監控模型，並以機器學習的方式來預測可能發生異常的時間，再透過簡訊（short message service, SMS）或推播（push notification）通知使用者異常狀況的發生，探究本系統可以有效預測可能發生異常狀況時間的相關技術與應用。周柏綸[23]以對環境監控即時資訊視覺化的議題，運用各式感測器元件，並應用物聯網架構，在環境監控系統中，透過整合實體與虛擬的空汙盒子經由及時的回傳監控數據，研究與實現偵測異常狀況與提供預警功能，來協助管理人員即時因應的相關應用技術。Kang 等人[24] 應用生物感測與環境監測技術，透過電紡奈米纖維（Electrospun nanofibers, NFs）技術直接生產，實現感測領域對 NFs 的研究及環境監測的優勢，實現穿戴設備強大應用範例。Wang 等人[25]使用摩擦奈米發電機（Triboelectric nanogenerators, TENGs）技術，開

發用於海洋環境監測的自供電感測系統,研究將數據採集的感測器信號傳輸到智慧型手機,奠立強大而可靠的監控應用基礎。如上這些關於環境監控的相關理論與應用技術探究,絕對可為本論文奠立絕佳研發與實現的解決方案。

2.4 智慧電扇

隨著網際網路設備的廣泛建置,智慧控制技術的逐漸成熟,結合前述兩者在智慧居家系統透過物聯網、擴增/虛擬/混合實境等技術的快速發展,再經由電腦硬體技術與網路速度、晶片計算處理器持續蓬勃發展,促使智慧生活居住品質的提升,進而使得智慧家電變成近年來非常重要且熱門的研究議題。人們透過智慧家電,例如智慧電扇,當可智慧化控制電器達到智慧節電或是提升智慧生活品質的境界。如下有許多國內外相關文獻的研究。基於物聯網技術的持續蓬勃發展,連帶與居家息息相關需求性與低成本優勢的電風扇也開始朝向智慧化。彭婁威[26]針對具 IoT 能力之智慧電扇,分析在智慧型行動裝置之應用程式介陳與資訊呈現設計,探究對於系統使用性與使用者感受上差異的相關研發與設計技術配套。目前市售的風扇無論是傳統風扇或是無刷直流電扇 BLDC（Brushless DC Motor）皆為定速定角度擺頭,也無角度功能。郭昇桓[27]針對傳統風扇或者 BLDC 風扇,研究搭配紅外線測溫與人臉辨識等偵測技術,透過微控制器、紅外線溫度感測器、馬達、硬體電路、鏡頭等硬體支援,再藉由資訊系統之監控視窗可觀察相關智慧功能,探究實現風扇各式節能效益評估相關應用技術。近年來人工智慧模型的使用頻繁,尤以人工神經網路模型為主,故其應用技術成熟。Chen 等人[28]提出一基於多層人工神經網路模型之反向傳播算法進行優化,探究實現智慧風扇系統能有效風速控制,並具有良好的穩定性與可靠性的相關應用技術;同樣地,Li 等人[29]也分析無葉風扇的運作原理及控制方法,在訓練過程中引用反向傳播算法進行優化,設計一深度學習算法模型,實現無葉風扇智慧控制系統,研究具優異穩定與可靠性風速控制的相關實現應用技術。如上的這些國內外文獻關於智慧電扇的理論、技術與應用的研究,定能建構本論文研發與實現的基礎與應用技術。

2.5 本章小結

誠如前述國內外相關研究與論文所述，其提及之知識、理論、技術與應用等資訊當可為本論文對應研發技術注入一股清泉。智慧電扇是隨著物聯網技術與智慧家居應用的發展所興起的一種新型電扇產品。在臺灣，智慧電扇的發展趨勢主要體現在以下幾個方陳：1) 智慧化技術；2) 設計創新；3) 環保節能；4) 行銷策略。總體來說， 發展趨勢主要在技術、設計、節能及行銷等四大方陳的創新及發展，未來隨著智慧家居市場的快速發展，智慧電扇可望在臺灣市場上取得更大的發展與市場佔有率。然而，除了 2.4 節這些論文研究的內容涵蓋智慧風速控制、智慧空氣流控制、智慧節能等，或許再深入瞭解智慧電扇的最新研究動態，包括：Zeng 等人[30]、Ren 等人[31]、Liu 等人[32]、Yang 等人[33]。未來可能值得探究趨勢，包括：1) 智慧電扇的人工智慧應用；2) 智慧電扇的可穿戴應用；3) 智慧電扇的智慧家居應用等。總之，隨著智慧家居和智慧城市等應用的普及，智慧電扇在未來仍然有許多研究之必要，這也彰顯出本論文的未來重要性。本論文旨在結合影像辨識技術，探究實現具環境監控功能之智慧電扇的相關智慧家居應用技術。

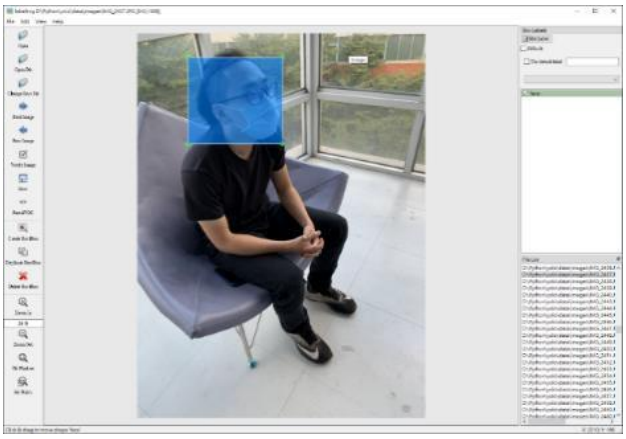
第三章　系統架構

本論文旨在整合影像辨識與物聯網應用技術發展一「兼顧居家安全之追蹤式電風扇系統之研發」，探究內容包括：1) 引用 YOLO v4-tiny 建置與實現影像辨識模式與追蹤技術；2) 利用煙霧感測技術整合 LINE Notify 建置與實作居家安全監控與通報子系統；3) 利用 Python 的 Web 應用程序框架 Flask 架設網頁及控制風扇所有運作模式。本章將相關研究方法、採用技術及完整系統運作與架構分述並羅列如后[34]。

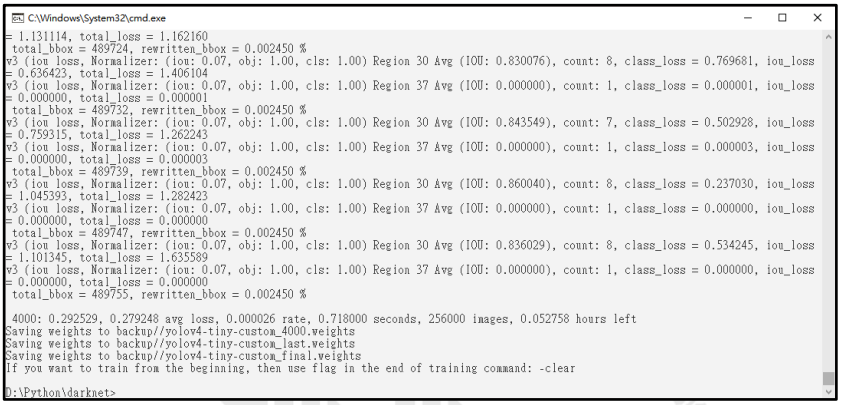
3.1 YOLO v4-tiny 影像辨識與追蹤技術

YOLO v4-tiny 顧名思義就是 YOLO v4 的簡單版[7, 35]，資料只有原先 YOLO v4 的十分之一，這也讓辨識速度部分大幅提升。整體的網路結構為 38 層，使用三個殘差單元，激活函數使用 LeakyReLU。目標的分類與回歸改成使用兩個特徵層，合併有效特徵層時，使用特徵金字塔網路。其次，同樣使用 CSPnet 結構，並對特徵提取網路進行通道分割，再將經過 3x3 卷積後輸出的特徵層通道劃分爲兩部分，並取第二部分。相較於其他版本的輕量化模型性能優勢顯著。本論文使用 YOLO v4-tiny 對要辨識臉部並進行追蹤，因此，本系統需要訓練一個新的模型，並將 YOLO v4-tiny 與樹莓派做結合，讓此辨識功能在樹莓派的架構上進行臉部追蹤。本文為了要辨識人臉，準備 600 多張照片進行訓練，並透過 Labeling 來標註臉部，並將檔案轉換成 Pascal Voc 格式放到 DarkNet，透過 DarkNet 來訓練 YOLO v4-tiny 的訓練檔，訓練過程說明詳如圖 3.1-1。

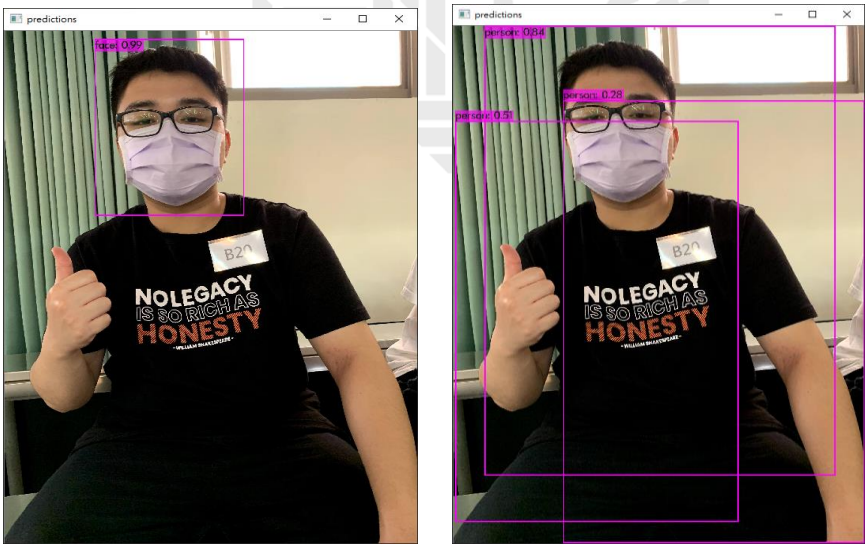
(a) 訓練圖片(續下頁)



(b) 載入圖片進臉部標註辨識內容



(c) 使用DarkNet訓練YOLO v4-tiny訓練檔

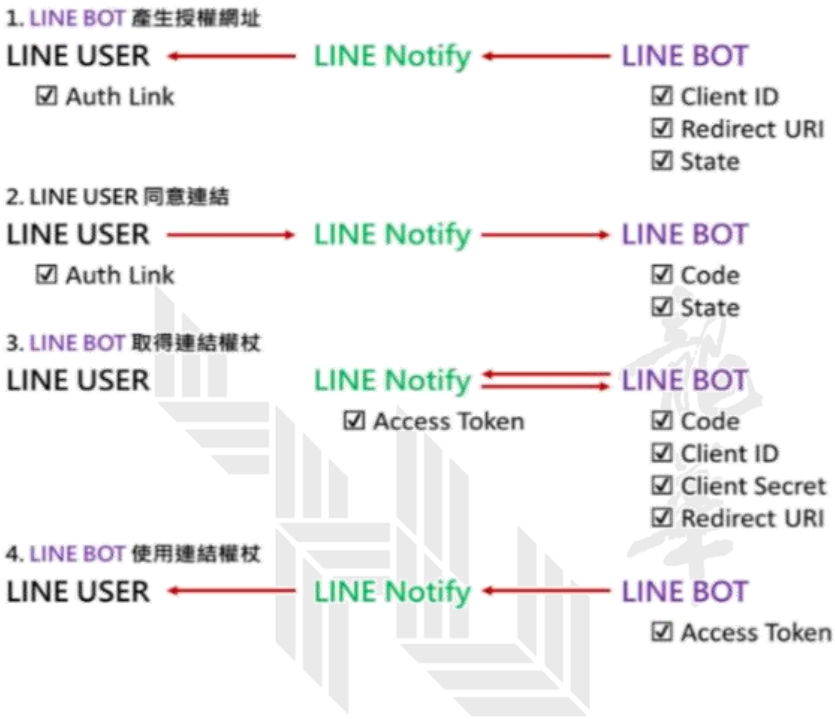


(d) 利用訓練模型進行檢測及比較

圖 3.1-1 模型訓練過程

3.2 LINE Notify 建置與實作居家安全監控與通報技術

LINE Notify 是一種特殊的 LINE 官方帳號[13]，可以讓使用者對某一個頻道做連結。只要該頻道有任何訊息或是動作，就可以透過 LINE Notify 發送通知給使用者知道。Line Notify 還可連結使用者撰寫的程式傳送特定文字或圖片，有非常多種的彈性及擴充性供大眾使用，詳如圖 3.2-1。



使用者需要先產生授權網址(Auth Link)。在拿到授權網址後，要先在 LINE Notify 登錄一個服務後，才可得到 Client ID 及 Redirect URI 這兩個資料，而 State 為可自訂的資料。系統根據 Client ID、Redirect URI 與 State，即可產生一個相對應的授權連結，再透過 LINE Bot 發送給使用者。使用者獲得授權網址後，點擊同意建立連結之後，會產生一組 Code，並傳送到 Redirect URI 這個地方，系統拿到代表使用者同意建立連結的 Code 之後，LINE BOT 即可向 LINE Notify 申請 Access Token。這個步驟，LINE BOT 需要取得包括：Client ID、Client Secret、Redirect URI、及 Code 等資料。確認

無誤之後，LINE Notify 就會傳回 Access Token。接收到 Access Token 的 LINE BOT，就可利用這組權杖當作溝通媒介，彼此透過 LINE Notify 發送訊息給特定的使用者。

3.3 Python 的 Flask Web 框架架設網頁及控制風扇運作技術

3.3.1 Python 的 Flask Web 框架

本論文引用 Python 的 Web 應用程序框架，框架類型包括：Flask [11, 12]、Django [36, 37]、Tornado [38]、Pyramid [39]這幾種框架。Flask 對初學者來說相對好入門，對小規模專題也不會那麼大費周章；況且，Flask 相比 Django 是簡單而且資源的需求也相對比較少；Pyramid 雖然也是一個不錯的選擇，但是它會比較偏向於大規模專題的開發；最後，Tornado 則是因為它的限制擴充性，因此沒有納入考量。相關 Python 的 Web 應用程序框架類型比較說明詳如表 3.3-1。

表 3.3-1 Python 的 Web 應用程序框架類型比較

比較	Flask	Django	Tornado	Pyramid
專題規模	小	大小皆可	小	大
可擴充性	好	最好	有受限	好
資源消耗	少	多	多	多

3.3.2 使用 IC【ULN2003A】讓風扇正常運作

首先，利用 IC【ULN2003A】驅動裝置（圖 3.3-1）連接上風扇，並裝上個四顆電池提供風扇電力；再用杜邦線連上樹莓派來進行操作。ULN2003A 的主要特點（表 3.3-1）

是擁有大電流容量以及高壓電輸出，以它作為驅動器來使用可使效率提升不少，更使得電扇能穩定運作。

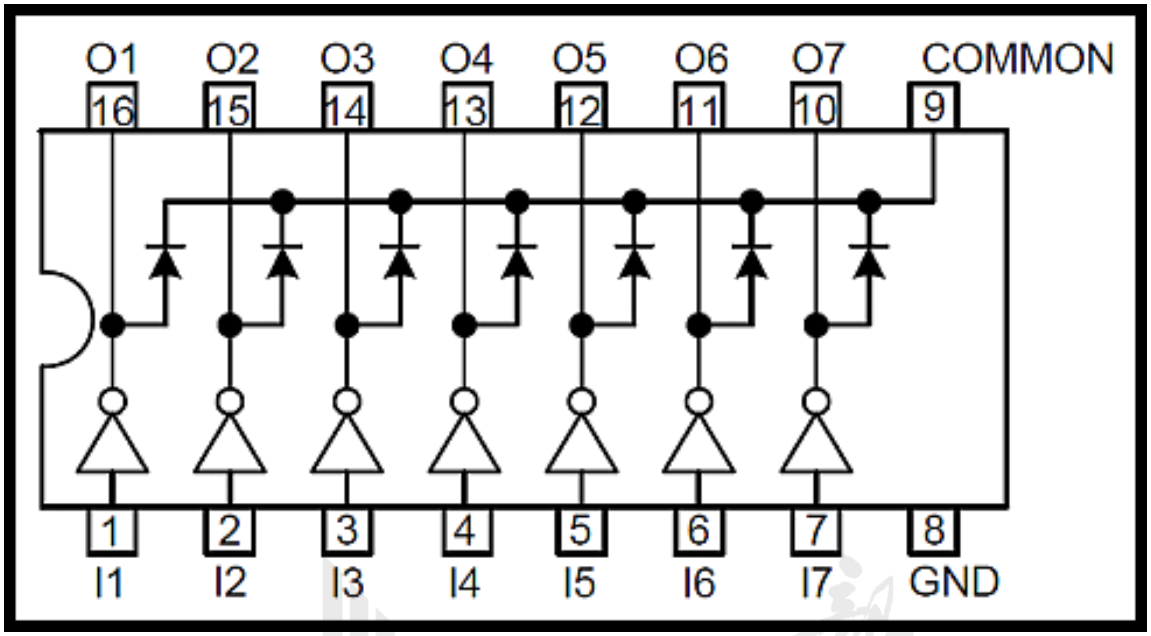


圖 3.3-1 ULN2003A內部電路圖

表 3.3-2 ULN2003A的主要特性

配置：Array 7	**晶體管極性：NPN**
集電極-發射極最大電壓 VCEO：50V	**集電極最大直流電流： 0.5A**
安裝風格：Through Hole	**封裝/外殼： PDIP-16**
最低工作溫度：-20C	**最大工作溫度：+85C**

3.3.3 使用樹莓派控制兩個伺服馬達【MG996R】

我們能藉由 Flask 的指令讓樹莓派（圖 3.3-2 及表 3.3-2）去控制兩顆伺服馬達，讓馬達可以執行上下左右的動作；而伺服馬達【MG996R】（圖 3.3-3）因為它的性能（表 3.3-3）優秀可讓機器的功能穩定運作，他跟【SG90】相比價格便宜，但其性能不足功能無法實現，這也是讓我們採用伺服馬達【MG996R】的原因。

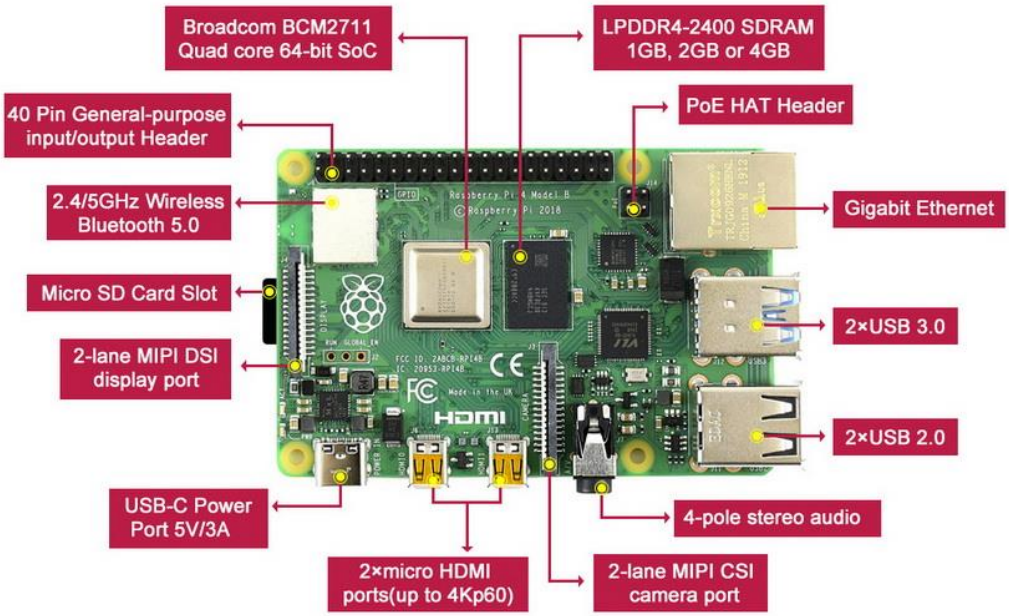


圖 3.3-2 Raspberry Pi 4 Model B結構圖

表 3.3-3 Raspberry Pi的主要特性

四核 ARM Cortex-A72(v8) 64位元 1.5GHz處理器	**記憶體：4GB LPDDR4-2400 SDRAM**
兩個 micro HDMI 顯示輸出埠	**兩個 USB 3.0，兩個 USB 2.0 連接埠**
內建 WiFi b/g/n/ac 無線網路，藍牙 5.0 以及 Gigabit 以太網路	**電源輸入：DC 5V3A, 從 USB-C 接頭，或 GPIO 排針，或 PoE 輸入皆可**

圖 3.3-3 MG996R 伺服馬達

表 3.3-4 MG996R 的主要特性

產品拉力：9.4kg/cm(4.8v),11kg/cm(6V)	**反應速度：0.17sec/60degree(4.8v)**
工作電壓：4.8-7.2V	**工作溫度：0°C-55°C**
齒輪形式：金屬齒輪	**工作死區：5us(微秒)**

3.3.4 使用 Flask 操控樹莓派

開啟時會透過 LINE Notify 傳輸當前網頁的 IP 提供使用者連線，透過同網域 IP 連上設定好的網頁進行模式切換。監控模式時會透過 OV5647 鏡頭（圖 3.3-4 及表 3.3-4）偵測是否有煙霧及陌生人，如有煙霧（MQ 2 煙霧感測模組如圖 3.3-5 及表 3.3-5）或陌生人會透過 LINE Notify 傳送圖片及文字給使用者知道。當然日後如果有跨網域的需求時，當再行思索如何擴充與解決這個問題。

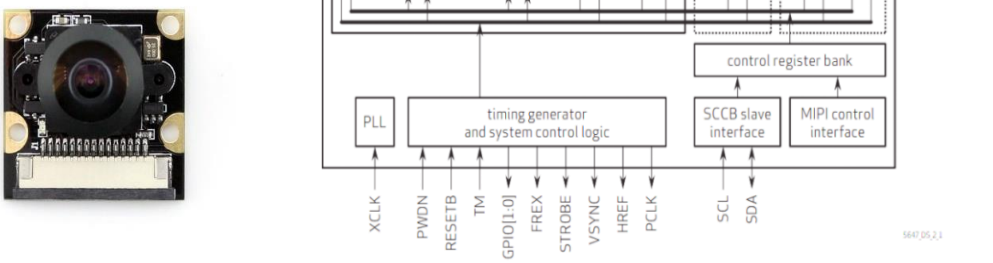


圖 3.3-4 Raspberry Pi OV5647 鏡頭電路圖

表 3.3-5 OV5647 的主要特性

CMOS 尺寸：1/4inches	**光圈（F）：F2.35**
焦距（Focal Length）：3.15mm	**視角（Diagonal）：160 度**
傳感器像素：500 萬	**尺寸：24mm × 25mm**

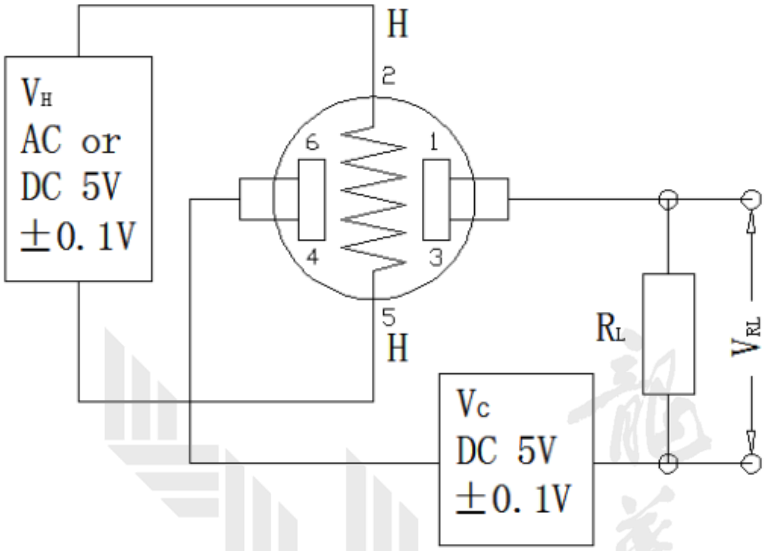


圖 3.3-5 MQ2 煙霧感測模組內部電路圖

表 3.3-6 MQ2 的主要特性

輸入電壓：DC5V	**功耗(電流)：150mA**
DO 輸出：TTL 數字量 0 和 1	**AO 輸出：0.1-0.3V**
最高濃度電壓：4V	**環境溫度：-20°C~+55°C**

3.4 系統運作與架構

本系統會利用前述 YOLO v4-tiny 訓練物件模型，再將訓練好的模型架構檔與權重檔導入系統，再利用 OPENCV 函式庫及神經網路來完備物件辨識功能[7, 34]。鏡頭採用 Raspberry Pi OV5647。再利用 IC ULN2003A 驅動裝置，並連接上電扇且以四顆電池提

供電扇電力，再連上樹莓派來進行電扇角度轉動操作。樹莓派使用 Raspberry Pi 4 Model B。ULN2003A 的主要特點是擁有大電流容量及高壓電輸出，以它作為驅動器可使效率提升不少，更使得本系統的電扇能穩定運作。本系統經由 Flask 的指令讓樹莓派去控制兩顆伺服馬達 MG996R，讓馬達可以執行電扇上下左右的擺動角度。最後，使用 Flask 操控樹莓派，系統開啟時，會透過 LINE Notify 傳輸當前網頁的 IP 提供使用者連線，透過同網域 IP 連上設定好的網頁進行電扇運作模式的切換，監控模式則透過 Raspberry Pi OV5647 鏡頭整合 MQ2 煙霧感測模組監測是否有煙霧與陌生人？如有煙霧或陌生人會透過 LINE Notify 傳送通報圖片及文字給使用者知道。系統完整運作架構詳如圖 3.4-1，電路圖則如圖 3.4-2。

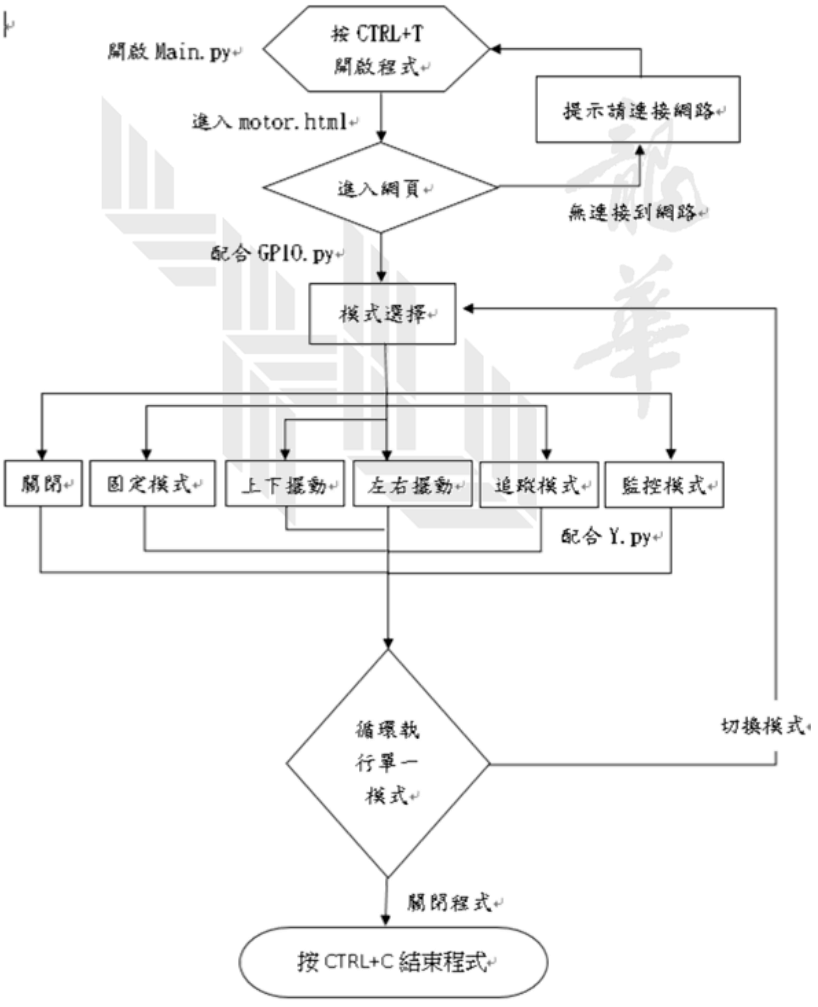


圖 3.4-1 系統完整運作架構

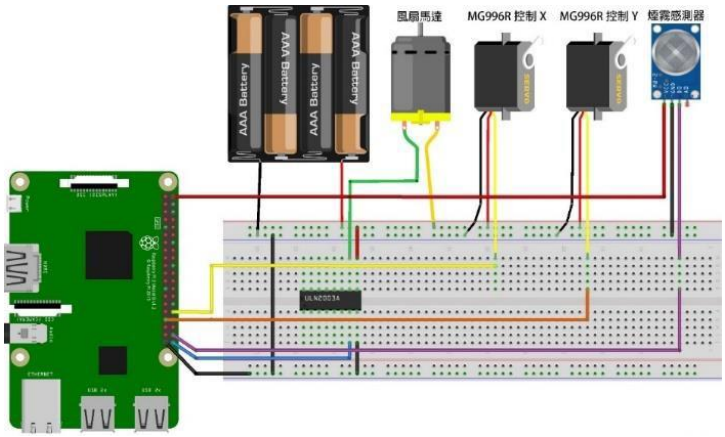

風扇馬達
MG996R 控制 X
MG996R 控制 Y
煙霧感測器
AAA Battery
AAA Battery
AAA Battery

圖 3.4-2 系統電路圖

第四章　系統呈現與建置成本

本章旨在說明整個系統開發所使用的環境設定、流程及情境，內容包含系統研發軟硬體、系統呈現與建置成本分析。

4.1 系統環境設定

本系統執行完整資料夾詳如圖 4.1-1，網頁檔放置在資料夾：templates，設定好的物體名稱檔和權重檔和訓練好的模型檔放置在資料夾：yolov4 裡。

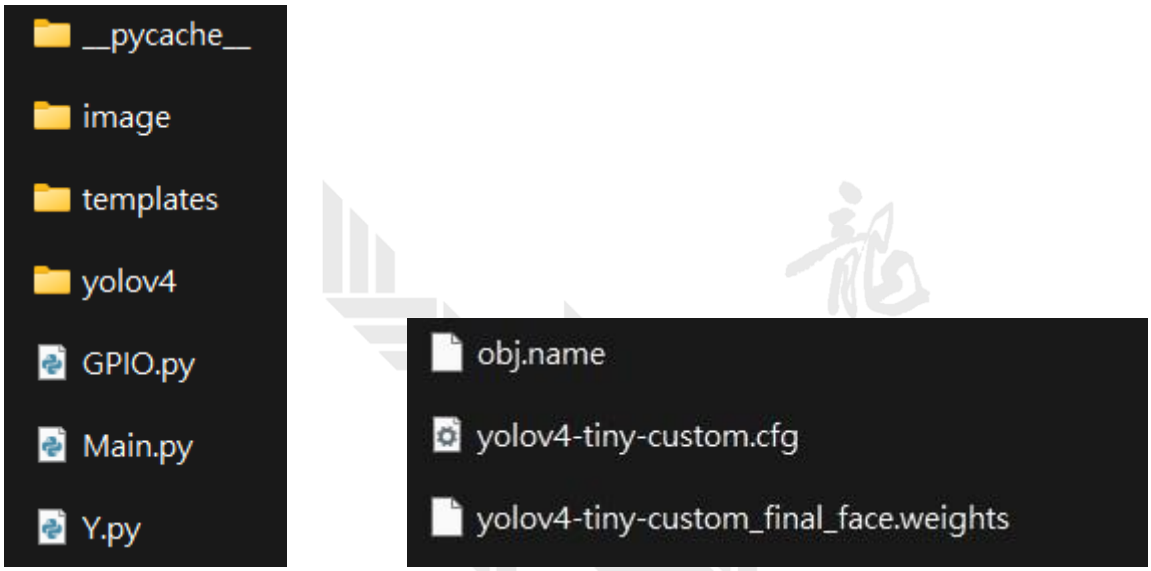


圖 4.1-1 完整系統資料夾結構

4.2 系統呈現

系統呈現與完整執行程序分述如后。

(1) 樹莓派接通電源和連接螢幕，並鏈結電路板，如圖 4.2-1。

圖 4.2-1 系統連接展示圖

(2) 在程式開啟後連接同網域 IP 進行控制，如圖 4.2-2。

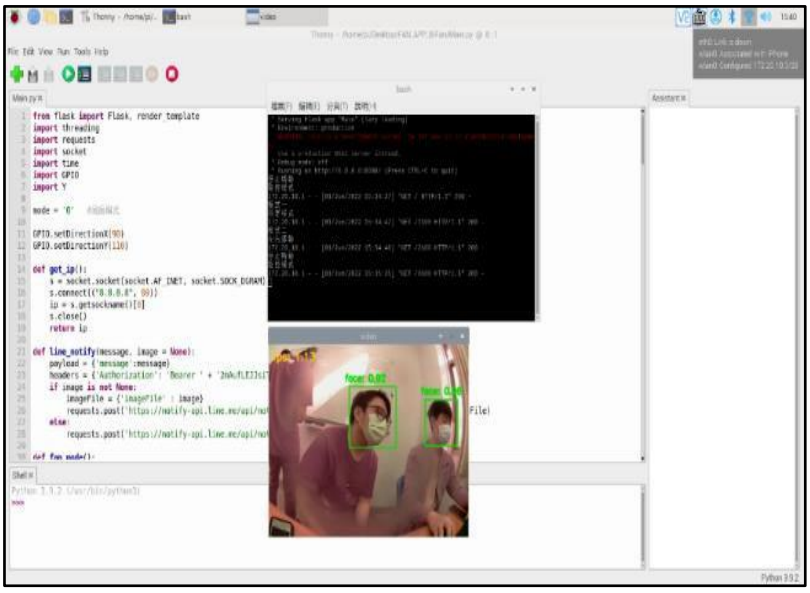


圖 4.2-2 連接同網域畫陝

(3) 進入網頁控制電扇，控制模式、風量調整，如圖 4.2-3。

圖 4.2-3 網頁風扇控制介陳

(4) 固定模式：電扇運作但不會轉動，如圖 4.2-4。

圖 4.2-4 風扇固定模式展示圖

(5) 固定模式：左右或上下擺動，只會左右或上下轉動，如圖 4.2-5。

圖 4.2-5 風扇左右及上下擺動展示圖

(6) 追蹤模式，會進行人臉鎖定，目標移動那個方向，電扇會跟著轉動，如圖 4.2-6。

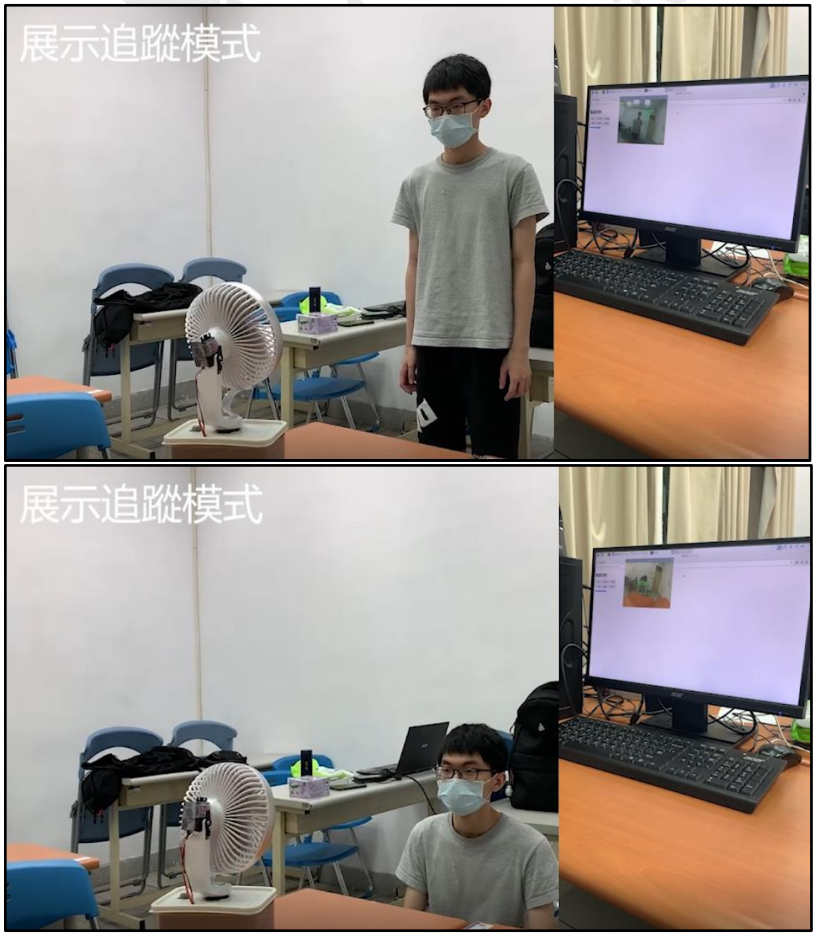


圖 4.2-6 追蹤模式展示圖

(7) 監控模式，電扇沒有在運作時，使用者可以通過攝像頭進行偵測，如有陌生人在家中或家中有濃煙及瓦斯洩漏會透過LINE Notify 傳送圖片及文字給使用者，如圖 4.2-7。

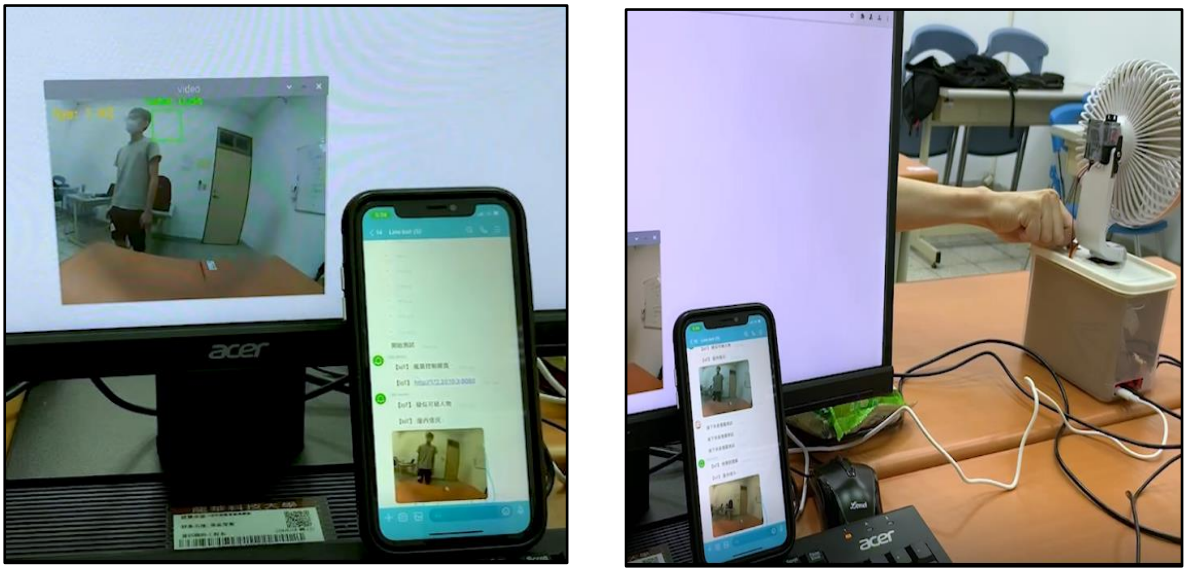


圖 4.2-7 環境監控資訊回報展示圖

4.3 系統成本與比較

Raspberry Pi 4B 4G 3000 元、鏡頭 OV 690 元、IC：uln2003a，10 元、伺服馬達 Mg996r 330 元*2、煙霧感測模組 MQ2 40 元、3 號電池 220 元、3 號電池之電池倉 14 元、杜邦線 50 元、空電路板 158 元、一台小型電扇 250 元，總成本共計：NT$ 5092 元。相關產品 1 及 2 之系統元件成本分析比較詳如表 4.3-1，產品 1 之總元件成本為 NT$ 5417 元，而產品 2 則為 NT$ 6151 元，本系統能符合系統功能要求下低元件成本之建置目標。

表 4.3-1 系統元件成本比較表

Raspberry Pi 4B 4G	1	Model B	性能	B 型擁有兩個 USB 端口，26 個 GPIO 引腳，512MB RAM。	價格	NT$2,200
	2	Model B+		樹莓派 B+型則擁有四個 USB 端口，40 個 GPIO 引腳，512MB		NT$3,000

				RAM。		
鏡頭 OV	1	RER-USB30W02M	主要應用	兼容USB1.1低速 /USB2.0 全速接口；提供windows、Linux 系統的SDK 開發源碼；適用于嵌入式系統二次開發。	價格	NT$1,000
	2	Raspberry Pi OV5647		樹莓派攝影鏡頭 B 型，支援調整焦距，跟物體的遠近。		NT$900
IC：uln2003a	1	ICHOME ULN2003A ULN2003 SOP16	價格	NT$7		
	2	STMicroelectronics ULN2003A		NT$19		
伺服馬達 Mg996r	1	MG 996R 13Kg 伺服機(舵機) 機器人 TowerPro	說明	本 MG996R 是全金屬齒輪，採用正品電位器，壽命長、扭矩足，最大力矩 13KG	價格	NT$380
	2	MG996R 55G 90 度 -180 度 金屬齒輪 伺服器 舵機		原 MG995 的升級產品是目前市場上較具性價比大扭力舵機之一。		NT$310
煙霧感測模組 MQ2	1	MQ-2 煙霧氣體偵測感測器	說明	使用簡單的電路即可將電導率的變化轉換為與該氣體濃度相對應的輸出信號。	價格	NT$90
	2	MQ-2 煙霧氣體感測器模組		採用優質雙陳板設計，具有電源指示和 TTL 信號輸出指示。		NT$70
3 號電池	1	【Panasonic 國際牌】鎳氫充電電池 3 號 AA	說明	環保、穩定、耐用超值、即開即用。	價格	NT$360

	2	日本製 Panasonic 國際牌 eneloop 3 號 AA		Panasonic eneloop 最新可回充 2100 次低自放電電池 MADE IN		NT$490
3 號電池電池倉	1	廣華電子 BH-343A 3 號電池並排串聯電池盒*4 方型帶	價格	NT$30		
	2	大山電子 3 號電池盒 4 只附開關		NT$37		
杜邦線	1	廣華電子 DB25045-05-5P 2.54 單頭母杜邦附線連接器	價格	NT$80		
	2	廣華電子 DB25045-04-4P 2.54 單頭母杜邦附線連接器		NT$60		
空電路板	1	WZ-17 仿馬蹄斯高壓穩壓電源(馬蘭士 7 等膽前級供電) PCB 空板	說明	本板電路是解剖馬蹄斯，專為膽前級等小電流供電設計。	價錢	NT$380
	2	Sparkfun 原廠 RFID Evaluation Shield 擴展板空 13.56MHz (DEV-10406)		該板配有開關、電阻器和 LED，但不包括 RFID 模組、XBee 或 Arduino 板。		NT$615
小型電扇	1	第二代迷你淨化風扇小鋼砲 FA-C001	說明	即使以最大風速運轉，馬達音量也不會超過 48dB。	價格	NT$890
	2	智能觸控 USB 循環扇 FT-LFN01／FT-LFN02		有 12 小時自動斷電機能防止		NT$650

第五章　結論與討論

本章旨在說明整個系統開發與呈現後，所獲致的結論，更討論本系統的未盡之處及未來的展望。

5.1 結論

本論文業已整合鏡頭及煙霧感測器，讓傳統電扇的功能不再侷限在單一性，而且具備透過智慧型手機控制電扇五種運作模式，包括：固定模式、上下模式、左右模式、追蹤模式與監控模式，實作出本兼顧居家安全之追蹤式智慧電風扇之研發。我們想著不要只是單純的感測器跟電扇去做結合，而是在這個互聯網的時代做出與傳統電器互相鏈結的設計哲學。再者，透過樹莓派開發平台，日後要加裝或更換感測器時也能運用之前的資料庫數據，兼顧未來擴充性。再藉由 LINE 擁有多元化的軟體設施，而其中的 LINE BOT 除可跟電扇做連動外，更可輕易導入居家監控遇到可疑分子或偵測煙霧時能立即拍照並回傳給使用者。最後，本系統利用 Flask 應用框架，來協助 Raspberry Pi 完備多執行序，達成本系統運作模式切換。

本系統最大的貢獻說明並羅列分述如下：

(1) 瞭解影像辨識原理，更探究模型如何製作與常見相關實作技術，包括：OPENCV 與 YOLO v4-tiny 等；

(2) 瞭解伺服馬達與風扇要如何與驅動 IC 結合與常見實作硬體技術，包括：ULN2003A、MQ 2 煙霧感測模組與伺服馬達 MG996R 等；

(3) 瞭解整合樹莓派進行影像追蹤與常見實作硬體技術，包括：Raspberry Pi 4B 4G 與鏡頭 Raspberry Pi OV5647 等；

(4) 瞭解實作網頁架設技術與控制智慧風扇的運作模式及通報技術，包括：Python 裡陳的 Flask Web 應用框架與 LINE Notify 等；

(5) 探究如何統整功能不同的程式集：辨識、追蹤、驅動運作及通知等程式模組；

(6) 低成本實作本兼顧居家安全之追蹤式智慧電風扇之研發；

(7) 本論文的初期成果獲 TANET 2022 臺灣網際網路研討會：智慧校園、智慧城市、智慧家庭、智慧行動生活科技議程最佳論文獎，如圖 5.1-1；英文版論文也榮獲 2024 IEEE 7th Eurasian Conference on Educational Innovation BEST CONFERENCE (IEEE ECEI 2024) PAPER AWARD，如圖 5.1-2。


TANET 2022
臺灣網際網路研討會
Taiwan Academic Network Conference

最 佳 論 文

主　　題 ： 2.4 智慧校園、智慧城市、智慧家庭、智慧行動生活科技
投稿編號 ： 9869
論文題目 ： 影像辨識技術支援之環境監控智慧電扇

作　　者 ： 黃鈺凱 邱俊琪 吳克晟 鄭郭千 李修家 楊勝源

大會主席

教育部
資訊及科技教育司司長

國立臺北商業大學
創新設計與經營學院院長

中 華 民 國　111 年 12 月 15 日

圖 5.1-1 TANET 2022 臺灣網際網路研討會：議程最佳論文獎


IEEE
ECEI
IIKII

2024 IEEE 7th Eurasian Conference on Educational Innovation

BEST CONFERENCE PAPER AWARD

This certificate is presented to

M240020
An Image Recognition Technology Supported Environment Monitoring Smart Fan
Sheng-Yuan Yang; Hsiu-Chia Lee

Bangkok, Thailand, 26–28 January 2024

Teen-Hang Meen
Lifetime Distinguished Prof. Teen-Hang Meen, Ph. D.
The Chairman of IEEE Taiwan Section Sensors Council
The President of International Institute of Knowledge Innovation and Invention

Chun-yen Chang
Chair Prof. Chun-Yen Chang, Ph. D.
Director of Science Education Center,
National Taiwan Normal University
Conference Chairman of IEEE ECEI 2024

International Institute of Knowledge Innovation and Invention

圖 5.1-2 IEEE ECEI 2024 BEST CONFERENCE PAPER AWARD

5.2 討論

本系統具備電扇體積小、介陙簡單、五種運作模式選擇、追蹤模式方便又舒服、供居家安全防犯陌生人與監控家中有無濃煙及瓦斯等優點。本系統尚有許多可修改之處，像是感測器的多寡、外觀上的簡陋、操控平台的選擇以及裝置的連動性。此外，誠如前述，透過 LINE Notify 傳輸當前網頁的 IP 提供使用者連線時，日後如果有跨網域的需求時，當是未來持續精進探究的目標。

再者，當初想要採用那個方式去跟手機做連動？現在台灣人常用的通訊 APP 以 LINE、Instagram、Messenger 這三個為主流。而 LINE 擁有多元化的軟體設施，而其中的 LINE BOT 可以跟我們風扇做連動，這也就是我們選擇 LINE 的原因。雖然很直覺的想到是用 APP 來做執行系統，可是 APP 的製作比我們想像中複雜。畢竟本系統是想導入居家監控遇到可疑分子或偵測煙霧時能立即拍照並回傳給使用者。因為頇具備回傳照片的功能，所以選擇使用比較方便的 LINE 來當主要平台。至於實行的方法用到 Flask，Flask 是個符合本系統所需要的應用框架，更能幫助 Raspberry Pi 在執行時使用多執行序，這樣就不會出現模式切換時發生錯誤，對本論文來說是個完美的選擇。然而，基於為系統優化與最佳手機連動，持續探究手機 APP 聯動相關技術，更是本論文未來的展望。

最後，本系統成品尚有許多可以做修改的，像是感測器的多寡、外觀上的簡陋、操控平台的選擇以及相關裝置的連動性，這些都是本論文未來透過成本評估，參考相關文獻，鑽研並加以實現，這樣才能使智慧家用設備擁有真正的多功能與便利性。

參考文獻

[1] 楊秩翔，消費者對智慧家電使用意願度之探討，碩士論文，管理碩士在職專班，元智大學，桃園，臺灣，2022。

[2] 趙孟昊，運用影像辨識與物件追蹤技術於無人地陙自走車即時室內定位及導引系統之建構，碩士論文，航太與系統工程學系，逢甲大學，臺中，臺灣，2018。

[3] 智慧臺北 Smart Taipei，AI 智慧影像分析應用，2023 年 8 月 1 日，取自 https://smartcity.taipei/projdetail/153，2022。

[4] 范德耀，智慧城市建設中智慧影像監控應用現狀及發展趨勢分析，3SMarket 全球智慧科技應用市場資訊網，2023 年 8 月 1 日，取自 http://3smarket-info.blogspot.com/2018/09/blog-post_64.html，2022。

[5] 趙建勛，以 AI 建構工程新智慧影像辨識技術之工程應用，2023 年 8 月 1 日，取自 https://www.ctci.com/e-newsletter/CH/459/technology/article-01.html，2022。

[6] 楊弘文，以 OpenCV 實現人臉辨識及系統建立，碩士論文，應用數學系所，國立中興大學，臺中，臺灣，2019。

[7] 洪晨鈞，基於 YOLO v4-tiny 演算法對抽菸進行影像偵測，碩士論文，電子工程系，國立臺北科技大學，臺北，臺灣，2019。

[8] 人工智慧感知信息處理算法研究院，YOLOV4-tiny 網路介紹，2023 年 8 月 1 日，取自 https://ppfocus.com/0/ed1058783.html，2022。

[9] Robert Advance, Raspberry Pi 4 Model B 4GB, Available at https://www.robot-advance.com/EN/art-raspberry-pi-4-model-b-4go-2640.htm, Accessed on Aug. 1, 2023, 2022.

[10] 張瑋軒，基於 Flask-Security 授權框架之跨系統權限管理平台設計，碩士論文，資訊工程系，國立臺北科技大學，臺北，臺灣，2022。

[11] Flask, Available at https://pythonbasics.org/what-is-flask-python/, Accessed on Aug. 1, 2023, 2022.

[12] Introduction to Flask, Available at http://www.chendacheng.com/content/?category=Flask%E4%B8%93%E9%A2%98&article=Flask_01.html, Accessed on Aug. 1, 2023, 2022.

[13] line notify, Available at https://ithelp.ithome.com.tw/articles/10255064, Accessed on Aug. 1, 2023, 2022.

[14]劉哲孙,政府物聯網平台滿意度之研究:以桃園市空氣品質物聯網為例,碩士論文，資訊管理學系碩士班，淡江大學，新北市，臺灣，2021。

[15]黃彥魁，物聯網異常診斷平台：以環境物聯網為例，碩士論文，資訊科學系，國立政治大學，臺北，臺灣，2022。

[16]Boursianis, A. D., Papadopoulou, M. S., Diamantoulakis, P., Liopa-Tsakalidi, A., Barouchas, P., Salahas, G., ... and Goudos, S. K., “Internet of things and agricultural unmanned aerial vehicles in smart farming: A comprehensive review,” Internet of Things, 18, Article No. 100187, 2022.

[17]Kopetz, H. and Steiner, W., “Internet of things,” in Kopetz, H. (Eds.), Real-time systems: design principles for distributed embedded applications, Springer, Cham, Switzerland, 2022, pp. 325-341.

[18]吳蘭庭，人工智能影像辨識應用於口袋披薩自動販賣機系統，碩士論文，機械工程系碩士班，遠東科技大學，臺南，臺灣，2023。[19]廖偵伶，工業機聯網資安系統與人工智慧影像辨識輕量模型之研發，碩士論文，機械工程學研究所，國立臺灣大學，臺北，臺灣，2022。

[20]Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., ... and Jégou, H., “Resmlp: Feedforward networks for image classification with data-efficient training,” IEEE Transactions on Pattern Analysis and Machine Intelligence, preprint, Available at https://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3206148, 2022, pp. 1-9.

[21]Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D. and Weijer, J., “Class-incremental learning: survey and performance evaluation on image classification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Early access, Available at https://doi.org/10.1109/TPAMI.2022.3213473, 2022, pp. 1-20.

[22]楊俊毅，智慧環境監控系統的設計與實作，碩士論文，資訊工程學系碩士班，元智大學，桃園市，臺灣，2023。[23]周柏綸，環境監控即時資訊視覺化裝置之設計與實作，碩士論文，光電科技碩士學位學程在職專班，國立暨南國際大學，南投，臺灣，2022。

[24]Kang, S., Zhao, K., Yu, D. G., Zheng, X. and Huang, C., “Advances in biosensing and environmental monitoring based on electrospun nanofibers,” Advanced Fiber Materials, 4, 2022, pp. 404-435.

[25]Wang, D., Zhang, D., Tang, M., Zhang, H., Sun, T., ... and Wang, J., “Ethylene chlorotrifluoroethylene/hydrogel-based liquid-solid triboelectric nanogenerator driven

self-powered MXene-based sensor system for marine environmental monitoring," Nano Energy, 100, 2022, Article No. 107509.

[26]彭婁威，智慧型電風扇應用程式介陙功能設計及資訊呈現型式之研究，碩士論文，設計系，國立臺灣科技大學，臺北市，臺灣，2022。 [27]郭昇桓,應用影響辨識系統於家用電風扇之智能控制研究，碩士論文，電機工程系，國立臺灣科技大學，臺北市，臺灣，2022。

[28]Chen, Y. S., Chen, H. W. and Lee, S. Y., "Intelligent Fan System Based on Artificial Neural Network for Wind Speed Control," IEEE Access, 10, 2022, pp. 16338-16347.

[29]Li, H., Li, D. and Li, K., "Research on the intelligent control of a bladeless fan based on a deep learning algorithm," IEEE Access, 9, 2021, pp. 139529-139537.

[30]Zeng, X., Huang, B. and Lin, W., "Intelligent Control Strategy of Air Flow for Smart Ceiling Fan," Journal of Sensors, 2021, pp. 1-8.

[31]Ren, Y., Shi, X., Chen, H., Zhao, W. and Tang, Y., "Smart fan system with power-saving ventilation and air purification functions," Building and Environment, 193, 2021, Article No. 107731.

[32]Liu, Y., Lu, X. and Deng, Y., "Research on the Control Technology of Smart Ceiling Fan Based on Fuzzy Logic," Journal of Physics: Conference Series, 1811(1), 2021, Article No. 012042.

[33]Yang, J., Lu, H. and Tan, J., "Research on the Intelligent Control Strategy of the Smart Ceiling Fan Based on Adaptive Fuzzy PID Algorithm," Journal of Physics: Conference Series, 1955(1), 2021, Article No. 012012.

[34]黃鈺凱，邱俊琪，吳克晟，鄭郭千，李修家，楊勝源，"影像辨識技術支援之環境監控智慧電扇，" 議程最佳論文獎，TANET 2022 臺灣網際網路研討會論文集，桃園，臺灣，2022，pp. 1503-1507。

[35]戴揚倫,基於 YOLOv4-tiny 之低複雜度神經網路架構設計,碩士論文,電子工程系，南臺科技大學，臺南，臺灣，2021。

[36]Django vs Flask vs Pyramid, Available at https://www.airpair.com/python/posts/django-flask-pyramid, Accessed on Aug. 1, 2023, 2022.

[37]Django, Available at https://www.djangoproject.com/start/overview/, Accessed on Aug. 1, 2023, 2022.

[38]Tornado, Available at https://www.tornadoweb.org/en/stable/, Accessed on Aug. 1, 2023, 2022.

[39]Pyramid, Available at https://docs.pylonsproject.org/projects/pyramid/en/latest/narr/introduction.html, Accessed on Aug. 1, 2023, 2022.

附錄

附錄 A 主程式碼 (Main.py)

```
from flask import Flask, render_template
import threading
import requests
import socket
import time
import GPIO
import Y

mode = '0'     #風扇模式

GPIO.setDirectionX(90)
GPIO.setDirectionY(110)

def get_ip():
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.connect(("8.8.8.8", 80))
    ip = s.getsockname()[0]
    s.close()
    return ip

def line_notify(message, image = None):
    payload = {'message':message}
    headers = {'Authorization': 'Bearer ' +
'2mAufLEJJsiTm5q5jJLzrNMbp7THb3L5oxCCu3hDfu7'}          # 發行權杖
    if image is not None:
        imageFile = {'imageFile' : image}
        requests.post('https://notify-api.line.me/api/notify', data=payload, headers=headers,
files=imageFile)
    else:
        requests.post('https://notify-api.line.me/api/notify', data=payload, headers=headers)

def fan_mode():
```

```
global mode
servoX = 90
servoY = 110

while True:

    if mode == '2':        #模式二 左右擺動
        turn_mode2 = 'right'
        while True:
            if turn_mode2 == 'right':
                servoX += 1
                if servoX > 160:
                    servoX = 160
                    turn_mode2 = 'left'
                GPIO.setDirectionX(servoX)
            elif turn_mode2 == 'left':
                servoX -= 1
                if servoX < 20:
                    servoX = 20
                    turn_mode2 = 'right'
                GPIO.setDirectionX(servoX)
            time.sleep(0.05)
            if mode != '2':
                break

    elif mode == '3':    #模式三 上下擺動
        turn_mode3 = 'up'
        while True:
            if turn_mode3 == 'up':
                servoY += 1
                if servoY > 115:
                    servoY = 115
                    turn_mode3 = 'down'
                GPIO.setDirectionY(servoY)
            elif turn_mode3 == 'down':
                servoY -= 1
                if servoY < 90:
                    servoY = 90
                    turn_mode3 = 'up'
```

```
                GPIO.setDirectionY(servoY)
            time.sleep(0.05)
            if mode != '3':
                break

    elif mode == '4':     #模式四 追蹤模式
        servoX = 90
        servoY = 110
        GPIO.setDirectionX(servoX)
        GPIO.setDirectionY(servoY)
        while True:
            if Y.turnX == 'right':
                servoX += 1
                if servoX > 160:
                    servoX = 160
                GPIO.setDirectionX(servoX)
            elif Y.turnX == 'left':
                servoX -= 1
                if servoX < 20:
                    servoX = 20
                GPIO.setDirectionX(servoX)

            if Y.turnY == 'up':
                servoY += 1
                if servoY > 135:
                    servoY = 135
                GPIO.setDirectionY(servoY)
            elif Y.turnY == 'down':
                servoY -= 1
                if servoY < 90:
                    servoY = 90
                GPIO.setDirectionY(servoY)
            time.sleep(0.05)
            if mode != '4':
                break

    elif mode == '5':     #模式五 監控模式
        GPIO.setDirectionX(90)
        GPIO.setDirectionY(110)
```

```
            while True:
                smoke_detected = GPIO.GPIO.input(GPIO.pin_smoke) ==
GPIO.GPIO.LOW
                if smoke_detected or Y.detected:
                    Y.cv2.imwrite('image/image.jpeg', Y.frame_resize)
                    time.sleep(0.1)
                    image = open('image/image.jpeg', 'rb')
                    if smoke_detected:
                        line_notify('檢測到煙霧')
                    if Y.detected:
                        line_notify('疑似可疑人物') line_notify('屋內情
                    況 :', image)
                    Y.detected = False
                    time.sleep(5)
                time.sleep(0.1)
                 if mode != '5':
                    break



app = Flask(    name    )

@app.route("/")
@app.route("/<cmd>")
def main(cmd='000'):

    if cmd[0] == '1':
        global mode print('
        模式一')
        print('固定模式')
        a = int(cmd[1:])
        mode = cmd[0]
        GPIO.pwm_fan.ChangeDutyCycle(a)

    elif cmd[0] == '2':
        print('模式二')
        print('左右擺動')
        a = int(cmd[1:])
        mode = cmd[0]
```

```
            GPIO.pwm_fan.ChangeDutyCycle(a)
            time.sleep(0.001)

        elif cmd[0] == '3':
            print('模式三')
            print('上下擺動')
            a = int(cmd[1:])
            mode = cmd[0]
            GPIO.pwm_fan.ChangeDutyCycle(a)
            time.sleep(0.001)

        elif cmd[0] == '4':
            print('模式四')
            print('追蹤模式')
            a = int(cmd[1:])
            mode = cmd[0]
            GPIO.pwm_fan.ChangeDutyCycle(a)
            time.sleep(0.001)

        elif cmd[0] == '0' or cmd[0] == '5': print('停
            止轉動')
            print('監控模式')
            mode = cmd[0]
            GPIO.pwm_fan.ChangeDutyCycle(0)
            time.sleep(0.001)

        elif cmd[0] == 'u' and (mode != '0' and mode != '5'):
            a = int(cmd[9:])
            GPIO.pwm_fan.ChangeDutyCycle(a)
            time.sleep(0.001)



        return render_template('motor.html')

if    name__== "    main    ":
        try:
            threading.Thread(target=fan_mode, daemon=True).start()
            threading.Thread(target=Y.get_frame, daemon=True).start()
            threading.Thread(target=Y.detect, daemon=True).start()
```

```
        line_notify('風扇控制網頁 :')
        line_notify('http://' + get_ip() + ':8080')
        app.run(host="0.0.0.0", port=8080, debug=False)

except:
        print('請重新執行程式')

finally:
        GPIO.destroy_pigpio()
        GPIO.destroy_GPIO() print('
        結束')
```

附錄 B 副程式碼 (GPIO.py)

```
import pigpio
import RPi.GPIO as GPIO


pin_servoX = 12     #ServoX 設定
pwmX = pigpio.pi()
pwmX.set_mode(pin_servoX, pigpio.OUTPUT)
pwmX.set_PWM_frequency(pin_servoX, 50)

def setDirectionX(angle):     #改變 ServoX 角度
    duty = 500 + (angle / 180) * 2000
    pwmX.set_servo_pulsewidth(pin_servoX, duty)


pin_servoY = 13     #ServoY 設定
pwmY = pigpio.pi()
pwmY.set_mode(pin_servoY, pigpio.OUTPUT)
pwmY.set_PWM_frequency(pin_servoY, 50)

def setDirectionY(angle):     #改變 ServoY 角度
    duty = 500 + (angle / 180) * 2000
    pwmY.set_servo_pulsewidth(pin_servoY, duty)

pin_fan = 21     #風扇馬達設定
GPIO.setmode(GPIO.BCM)
GPIO.setup(pin_fan, GPIO.OUT)
pwm_fan = GPIO.PWM(pin_fan, 2000)
dc = 0
pwm_fan.start(dc)


pin_smoke = 20
GPIO.setup(pin_smoke, GPIO.IN)


def destroy_pigpio():
```

```
    pwmX.set_PWM_dutycycle(pin_servoX, 0)
    pwmX.set_PWM_frequency(pin_servoX, 0)
    pwmY.set_PWM_dutycycle(pin_servoY, 0)
    pwmY.set_PWM_frequency(pin_servoY, 0)


def destroy_GPIO():
    pwm_fan.stop()
    GPIO.output(pin_fan, GPIO.LOW)
    GPIO.cleanup()
```

附錄 C 影像辨識程式碼 (Y.py)

```
import cv2
import time

turnX = 'stop'
turnY = 'stop'
detected = False
WIDTH = 0
HEIGHT = 0
frame_resize = None

def initNet():
    CONFIG = 'yolov4/yolov4-tiny-custom.cfg'
    WEIGHT = 'yolov4/yolov4-tiny-custom_final_face.weights'
    NAMES = 'yolov4/obj.name'

    with open(NAMES, 'r') as f:
        names = [line.strip() for line in f.readlines()]

    net = cv2.dnn.readNet(CONFIG, WEIGHT)
    model = cv2.dnn_DetectionModel(net)
    model.setInputParams(size=(416, 416), scale=1/255.0)
    model.setInputSwapRB(True)

    return model, names

def nnProcess(image, model):
    classes, confs, boxes = model.detect(image, 0.6, 0.3)
    return classes, confs, boxes

def drawBox(image, classes, confs, boxes, names):
    global turnX
    global turnY
    global WIDTH
    global HEIGHT
    global detected
    new_image = image.copy()
```

```
    for (classid, conf, box) in zip(classes, confs, boxes):
        x, y, w, h = box
        label = '{}: {:.2f}'.format(names[int(classid)], float(conf))

        if classid == 0:
            detected = True
            cv2.rectangle(new_image, (x, y), (x+w, y+h), (0, 225, 0), 2)
            cv2.putText(new_image, label, (x-10, y-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 225, 0), 2, cv2.LINE_AA)
            if (x + w // 2) > (WIDTH // 2) + 70:
                turnX = 'left'
            elif (x + w // 2) < (WIDTH // 2) - 70:
                turnX = 'right'
            else:
                turnX = 'stop'

            if (y + h // 2) > (HEIGHT // 2) + 50:
                turnY = 'up'
            elif (y + h // 2) < (HEIGHT // 2) - 50:
                turnY = 'down'
            else:
                turnY = 'stop'
        else:

            detected = False

    return new_image

def get_frame():
    global WIDTH
    global HEIGHT
    global frame_resize
    cap = cv2.VideoCapture(0)
    ratio = cap.get(cv2.CAP_PROP_FRAME_WIDTH) /
cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
    WIDTH = 480
    HEIGHT = int(WIDTH / ratio)

    while True:
```

```
ret, frame = cap.read()
```

```
            frame_resize = cv2.resize(frame, (WIDTH, HEIGHT))
            if cv2.waitKey(1) == 27:
                break

def detect():
    global frame_resize
    model, names = initNet()

    while True:
        begin_time = time.time()

        if frame_resize is not None:
            classes, confs, boxes = nnProcess(frame_resize, model)
            frame2 = drawBox(frame_resize, classes, confs, boxes, names)

            fps = 'fps: {:.2f}'.format(1/(time.time()-begin_time))
            cv2.putText(frame2, fps, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,
204, 255), 2)


            cv2.imshow('video', frame2)
        if cv2.waitKey(1) == 27:
            cv2.destroyAllWindows()
            break
```

附錄 D 風扇控制網頁 (motor.html)

```
<!DOCTYPE html>
<html>
<head>
<title>風扇控制</title>
<meta http-equiv="content-type" content="text/html;charset=utf-8" />
<script>

function getValue(x) {
   var val = document.getElementById("myRange").value
   var a = x + val
   motor(a)
}

function motor(cmd)
      { window.location.href='/' +
      cmd
}

</script>
</head>
<body>
   <h1>風扇控制</h1>
   <input type="button" onClick="getValue(0)" value="    關閉    "/> |
   <input type="button" onClick="getValue(1)" value="固定模式"/> |
   <input type="button" onClick="getValue(2)" value="左右擺動"/></p>
   <input type="button" onClick="getValue(3)" value="上下擺動"/> |
   <input type="button" onClick="getValue(4)" value="追蹤模式"/> |
   <input type="button" onClick="getValue(5)" value="監控模式"/></p>
   <input type="range" onClick="getValue()" min="20" max="100" value="100" step="10"
id="myRange"/>
   </body>
   </html>
```