

嘉義市第 38 屆中小學科學展覽會
作品說明書

國中組 生活與應用科學(一)科

作品名稱：物體影像辨識方法在機器人物體夾取之影響研究探討

編 號：

關鍵詞：視覺辨識，深度學習，物聯網

摘要

在高齡化及少子化雙重影響下，老年安養問題已浮上檯面，尤其行動不方便與健忘的老人，常常想要一個物品，經常想不起來，加上行動不便找個東西都花非常多的時間，結果還是找不到。如何開發一個機器人來協助老人、並讓機器人有一個敏銳的眼睛能很清楚地找到物體。本研究嘗試結合今火紅 AI 技術以不同樣的方法的切入，嘗試研究一好的物體辨識方法讓機器人可以有效率的辨識出他要的物體。另一方面近年來，觸控已然成為主流，隨處可見這樣的應用，再加上在人工智慧中語音辨識技術的提升，聲控系統也雨後春筍般的湧現在家電與機器中，所以本研究嘗試研究一以軟硬體開發技術自行開發一結合觸控與聲控的雙控互動式人機介面共使用者方便操控。

壹、研究動機

為了解決高齡化及少子化雙重因素影響下，台灣面臨人口高齡化的到來，加上許多高齡化的人常伴隨著行動不方便與健忘的問題，尤其是獨居老了他們在生活上極度的不方便，往往想要一個物品常常忘記放在哪裡或者不方便去尋找，所以本研究希望能開發一個雲端智慧語音機器人(ihound)如圖一，可以協助他們尋找物品，那就可以解決生活上不方便問題。



圖一：雲端語音尋物獵犬機器人示意圖

再加上我們國小到國一所學習相關技術表一，如Scratch、藍芽車、Arduino、機器人與Python程式設計。尤其在國中一整年的Python程式課程，讓我們對程式開發充滿的興趣。除了自我已具備能力外、並想嘗試研究與結合現今火紅AI技術如電腦視覺、語音辨識、IoT(物聯網)、深度學習與控制版(Arduino or Raspberry pi)技術，來研究開發一個雲端語音尋物機器人ihound。其系統架構如圖二，然而在尋物機器人尋物物品中最重要就是給機器人敏銳的眼睛，可以很清楚地看到，並找到物體。所以如何讓機器人正確的辨識物體變成一個很重要的課題，加上與學校老師討論，決定想挑戰這個課題，進行這方面的科展研究與開發。

表一： 研究相關技術盤點與規劃

分類	技術項目	相關性/對策
已具備	Scratch	低/用不上
	Arduino 開發	高 /複習
	樂高機器人程式設計	中/複習
	python 程式設計	高/複習
	藍芽車	中/複習
需挑戰	Raspberry pi 與 Webcam 開發	高/自學與老師、專家討論
	Python OpenCV (顏色辨識,樣板匹配)	高/自學與老師、專家討論
	自走車	高/自學與參加研習
	Python Google API(語音辨識)	高/自學與老師、專家討論
	IOT	高/自學與老師、專家討論
	多執行續	高/自學與老師、專家討論
	Python 互動式人機介面	高/自學與老師、專家討論
	圖片增生系統	高/自學與老師、專家討論
	深度學習	高/自學與老師、專家討論



圖二： ihound 系統架構圖

貳、研究目的

一、研究目的

本研究會先開發一結合電腦視覺、語音指令辨識、IoT(物聯網)、影像處理 (OpenCV)、深度學習、自走車與 Wifi 控制版 (arduino and Raspberry pi) 相關技術協助行動不方便與健忘的老人的語音尋物 AI 服務獵犬機器人。並且研究探討不同物體辨識方法，讓機器人找尋物體時可以很清楚的辨識出他要的物體，並對於所要尋找物品位置定位及路徑導引，然後結合機器人(車)的語音播放系統通知使用著並以夾取機構將物品夾取，進而解決行動不便者在尋找物體不便的問題。

根據上述主要的目的，將其拆為幾個子目標依序來完成:

- (一) 研究以樹莓派、麥克風、按鈕、led 燈，開發獵犬控制盒(h-Box)
- (二) 研究 Arduino 與樹莓派的串列通訊開發控制獵犬機器人主體(h-BoT)
- (三) 研究以 Window Arduino IDE 與 Matrix mini Lib 進行車體方向控制
- (四) 研究在樹莓派以 Python OpenCV 與 VNC 開發攝影機影像擷取之即時監控
- (五) 研究在樹莓派進行語音辨識中的語音轉文字(STT)與文字轉語音(TTS)
- (六) 研究以自然語言處理(NLP)關鍵字提取技術進行語音控制
- (七) 研究以 IoT 無線通訊技術進行 h-BoT 與 h-Boxy 雙向無限通訊
- (八) 研究以多執行序同時進行影像辨識與語音處理流程
- (九) 研究樹莓派以 OpenCV 之可縮放式影像模板匹配 (matchTemplate)方法進行物體偵測
- (十) 研究樹莓派以 OpenCV 之 HSV 色彩模型方法進行物體色彩特徵偵測與辨識
- (十一) 研究以使用 Keras+Tensorflow 訓練深度學習卷積神經網路模型進行物體分類辨識
- (十二) 研究模擬環境因素之深度學習訓練圖片自動生成技術
- (十三) 開發一圖形化互動式人機觸控介面方便使用者操作

綜合以上課題，本研究這次的研究希望也能產出以下成果:

- (一) 實作出一 Raspberry pi 監控攝影機，以結合 WebCAM、OpenCV、Wifi 技術可以透過網路遠端影像監控。
- (二) 即時生成逐字稿系統；以結合 Google API(STT (Speech -to- Text) 與 Python 檔案管理技術將語音轉成文字，再將文字存成文字檔
- (三) 實作出語音聲控與播音系統(智慧家庭,聲控汽車)、結合 Google API (STT(Speech -to- Text))、TTS (Text-to-Speech))、Raspberry pi 與 Arduino 上控制元件(攝影機，繼電器)與感知設備如攝影機、繼電器、溫度計..，連接家用電器進行智慧控制與回報。
- (四) 即時圖片生成系統；將基本圖片資料上下翻轉、左右翻轉鏡像、放大縮小、亮度與與透過高斯雜訊將每個圖片各個區域添加雜訊使其變模糊，藉此使機器能夠學習在惡劣環境辨成功辨識。
- (五) 智慧門禁_即時人臉辨識系統:結合 OpenCV 人臉偵測再加上本研究探討 IoT 與深度學習卷積神經網路進行智慧門禁人臉辨識。

參、研究設備及器材

一、硬體設備

本研究嘗試開發一互動式人機觸-聲雙控介面之雲端智慧語音獵犬機器人(ihound)作品相關硬體有兩大部分(表二)，分別為智慧語音獵犬機器人控制盒 (h-BOX)與智慧語音獵犬機器人控制盒本體 (h-BOT)其設計圖如下：

(一) h-BOX 硬體包含: 樹梅派、按鈕、Led 燈，RGB led 燈與行動電源:

(二) h-BOT 在硬體:包含樹梅派、喇叭、攝影機、行動電源、測距超音與 Matrix BOT 其中在機器人方面主要是以 Matrix Mini 機器人套件作為本實驗之機器人、Matrix Mini 機器人套件包含:2 個 DC 馬達接口，4 個 RC 馬達接口，4 個數位和 3 個類比輸入/輸出端口，2 個 RGB LED 燈，3 個 Onboard 按鈕以及 4 個 I2C 接口。

1. M1 and M2:以 2 個 DC 馬達進行控制車子行進方向
2. D4: 1 個數位輸入/輸出端口進行超音波測距
3. USB :1 個 USB 進行與 raspberry pi 進行資料通訊。
4. RC1: 以 1 個 RC 馬達進行夾具控制

表二：開發硬體列表:

名稱	圖片	數量	說明
Raspberry pi 3 控制板		2	設備一:獵犬機器人控制盒 設備二:獵犬機器人主體
ARDUINO UNO 制板		1	獵犬機器人、超音波感測器與夾具控制
USB麥克風		1	Raspberry pi 聲音輸入
Analog3.5 mm喇叭		1	Raspberry pi 聲音輸出 機器人偵測距離回報
WebCAM攝影機		1	Raspberry pi物件影像擷取
按鈕		1	語音辨識啟動
3D列印機		1	機器人部分零件、結構與載具外殼開發
超音波感測器		1	距離偵測
RGB led燈		1	物體追蹤狀態即時顯示
Led燈		1	語音狀態顯示
Matrix Robot		1	物體追蹤、物體夾持
Mobile power		2	設備一:獵犬機器人控制盒電源 設備二:獵犬機器人主體電源

二、軟體或應用程式

本作品相關軟體技術都是本組人員自行研發，在軟體發系統平台組要分為兩個部分在 Raspberry pi 的 Linux 系統與 PC windows (win10)系統，這兩個系統分別進行開發相關軟體技術簡介如下 (表三)：

(一) Raspberry pi 的 Linux 系統：

1. 主要開發語言與平台：語言為 python、平台 Thonny Python IDE。
2. 安裝 API 套件: 語音辨識、OpenCV、Socket 無線通訊 (兩個 Raspberry pi)、Serial 有線通訊 (Raspberry pi 與 Arduino)、Tkinter (圖形化)、threading(多執行緒)、GPIO (樹莓派輸入輸出) 等等
3. 主要開發軟體技術與整合：語音轉文字技術、文字轉語音技術、NLPC(自然語言關鍵字控制)技術包含影像 HSV 色彩模型方法，影像模板匹配進行物體偵測、IoT 無線通訊技術、Serial 有線通訊技術、影像處理與辨識技術、圖形化程式介面技術、多執行緒技術，樹莓派輸入輸出技術，尤其將所有開發技術整合成一個系統。

(二)PC windows (win10) 系統：

- 1.主要開發語言與平台:語言為 C++、平台 Arduino IDE (內建 Arduino 輸入輸出與 Serial 有線通訊)。
- 2.安裝 API 套件: Matrix mini Lib。
- 3.主要開發軟體技術與整合: Arduino 輸入輸出技術(超音波)、Matrix Robot 控制技術與 Serial 有線通訊(Raspberry pi 與 Arduino)、尤其將所有開發技術與上述 Raspberry pi 整合成一個系統。

表三：開發使用軟體技術列表：

名稱	語言/套件	說明
Thonny python IDE	Python	Raspberry pi 開發平台
ARDUINO IDE	C++/Matrix mini Lib	ARDUINO UNO 輸入輸出 Matrix Robot 控制 Raspberry pi 與 Arduino 訊息相互傳遞
OpenCV	cv2	WebCAM 影像擷取與顯示 影像處理 顏色辨識 影像模板匹配
Google API	import speech_recognition import gTTS	語音轉文字 文字轉語音
Python 套件	import serial import numpy import Image, ImageTk import tkinter import threading Import Socket GPIO Keras and Tensorflow	Raspberry pi 與 arduino 訊息相互傳遞 多維度陣列與矩陣運算 圖片編輯與儲存 視窗圖形化人機介面 多執行序 雙向無線通訊(TCP/IP) 樹莓派輸入輸出 深度學習物體辨識

肆、研究過程與方法

一、既有技術之整理

當我們決定要研究這個主題的時候，從整個流程圖（圖三）我們發覺需要開發技術非常的多，包含如電腦視覺、語音辨識影像處理 Arduino 跟樹莓派的相關的技術。



圖三：ihound 互動式人機觸-聲雙控介面之雲端智慧語音獵犬機器人

據上述整理一下我們團隊從國小到國一所以學習的技術的技術盤點予研讀相關欠缺的技術，發現絕大部的技術 python 語言套件都有支援如 OpenCV、語音辨識、多執行序與視覺化介面..等等，自走車廠商有開放提供開發者開發控制車子，加上學校創客中心相關樹梅派、Arduino 與 3D 列印等等資源，並與指導老師討論後。於是在老師指導下開始相關議題的研究過程如下。

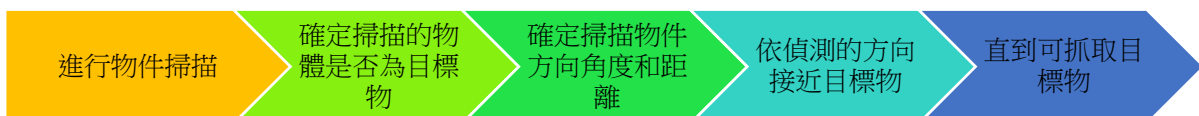
二、掃描策略研究

我們尋物機器人有一個很重要的課題，就是要如何找到我們要的目標物。老師也建議我們去研究網路上的線上的遊戲 Blockly Games (圖四)，其中 Pond 是 Blockly Games 第六個主題是池塘導師，畫面中會有兩隻小鴨，黃色的是自己，紅色的是要攻擊的目標，必須調整射擊角度和距離，攻擊到紅色小鴨並且消失後才能闖關成功。其中一個很重要的課題就是怎麼去找到我們的目標紅色小鴨。



圖四：Blockly Games Pond 掃描策略遊戲

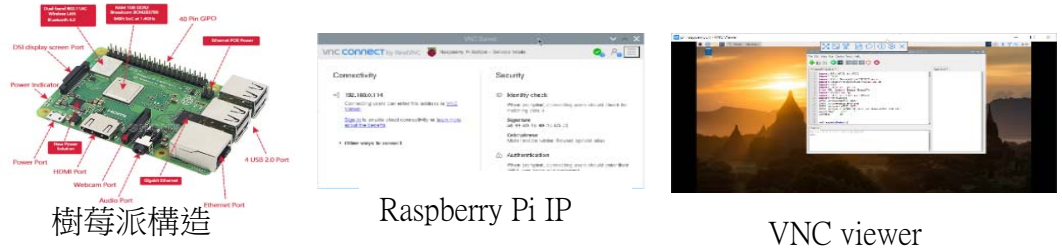
在這遊戲中我們可以控制的是黃色小鴨的方向與速度還有掃描的角度，在這過程當中黃色小鴨方向的控制去好像我們對機器人車的方向控制一樣，黃色小鴨的掃描線控制就跟我們的控制鏡頭一樣可以旋轉，紅色小鴨就像是我們要找的目標。然後我們再進行目標物(顏色)辨識，當確定目標物無誤之後逐漸接近目標物距離一定時開始抓取目標物。整個流程如圖五所示：



圖五：掃描策略流程

三、樹莓派 (Raspberry pi)實作練習

本研究主要是以 Raspberry Pi 3B v1.2 Wifi 如圖六。使用 Micro SD 安裝 RASPBIAN DESKTOP 作業系統來進行研究。透過 VNC viewer 連線到 Raspberry Pi 來進行相關研究程式撰寫與套件安裝。



圖六：Raspberry Pi 3B 結構圖

我們以樹莓派內附 Thonny python IDE 以 Python 進行程式撰寫。安裝相關技術套件如 OpenCV，然後以其所功能進行相館影像擷取與處理，其他相關技術如表四也如此開發。

四、主題研究

接下來開始本科展相關主要研究如圖七，相關研究如下所示:



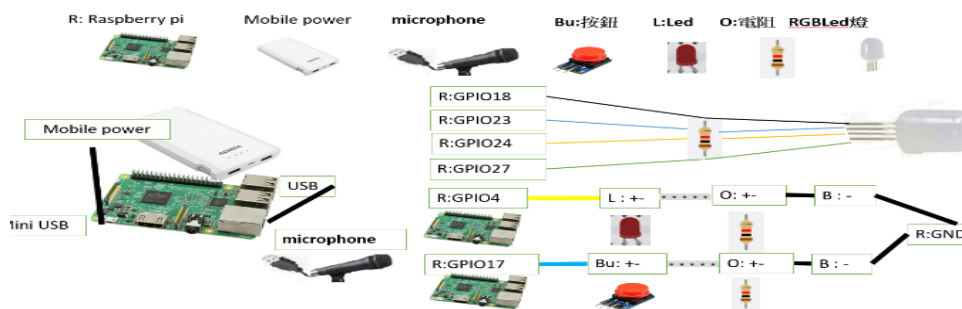
圖七：各主題研究流程

(一)、研究一、雲端智慧語音獵犬機器人(ihound)

本研究嘗試開發一互動式人機觸-聲雙控介面之雲端智慧語音獵犬機器人 (ihound)相關硬體研究有兩大部分:分別為智慧語音獵犬機器人控制盒 (h-Box) 與智慧語音獵犬機器人控制盒本體 (h-BOT) 其設計如下：

1.h-Box 硬體包含: 樹莓派 Raspberry pi 3 、按鈕、單色 Led 燈，RGB led 燈與行動電源:

(1)在 h-BOX 接線如圖八所示，以一樹梅派作為 h-BOX 硬體主體連接各控制元件連接端口說明如表四，以 python 結合 GPIO 套件開發



圖八：h-box 接線圖

表四：h-box 端口與功能說明

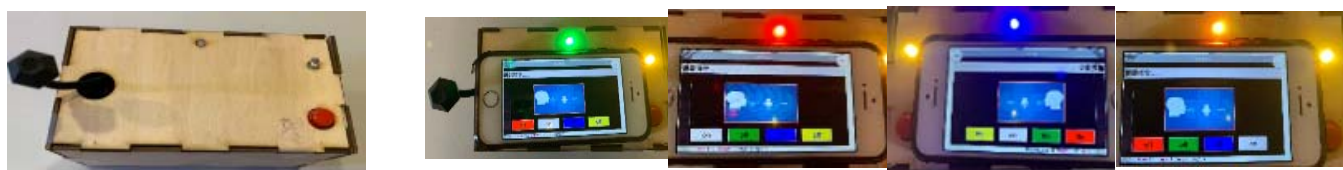
名稱	端口	功能
Raspberry pi 3 控制板		1 語音轉文字 2 將文字無線傳輸至獵犬機器人 3 接受獵犬機器人回傳訊息
USB 麥克風	樹莓派 USB	接受語音
按鈕	樹莓派 GPIO17	1 啟動麥克風接受語音 2 開啟語音辨識 Led 燈。
單色 Led 燈	樹莓派 GPIO4	1 亮: 語音辨識中 2 暗: 語音辨識結束
RGBLed 燈	樹莓派 GPIO18 樹莓派 GPIO23 樹莓派 GPIO24 樹莓派 GPIO27	1 亮紅燈:獵犬機器人正尋找紅色物體 2 亮綠燈:獵犬機器人正尋找綠色物體 3 亮藍燈:獵犬機器人正尋找藍色物體 4 亮黃燈:獵犬機器人正尋找黃色物體 5 燈暗:獵犬機器人正停止尋找物體
行動電源:	樹莓派電源	提供控制盒電力

(2) h-BOX 軟體介面設計： 語音控制啟動按鈕，觸控控制啟動按鈕、h-BOT 尋找物體即時狀態顯示與 IoT 雙向無線資訊傳送顯示區(如圖九)。



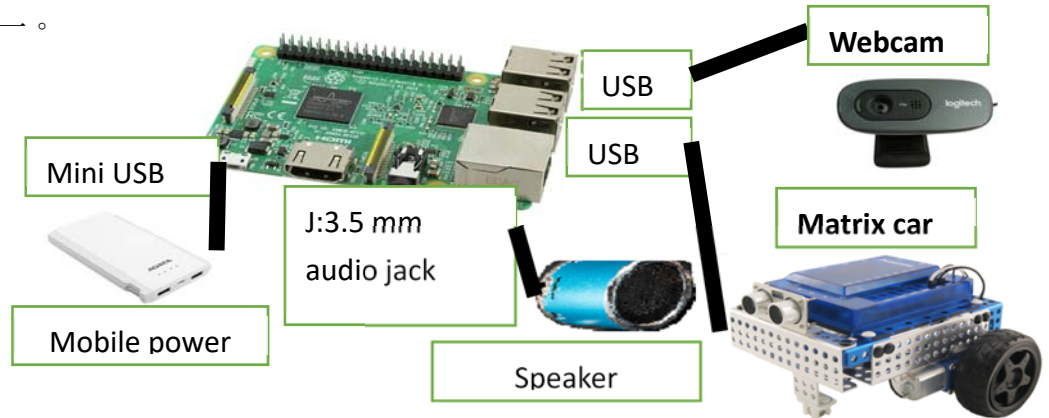
圖九：語音控制 h-BOT 尋找物體即是狀態顯示

(3) 如圖十所示，為 h-box 原形成品，在 h-box 中間加入樹莓派專用觸控螢幕，本研究先以手機觸控螢幕暫代，使用者可以使用 iBox 控制盒，透過按鈕啟動麥克風進行語音控制或者透過觸控螢幕進行觸控控制，LED 燈會立即反應力技能目前正在尋找的物體顏色。



圖十：h-Box 成品與尋找不同物體顏色燈展示

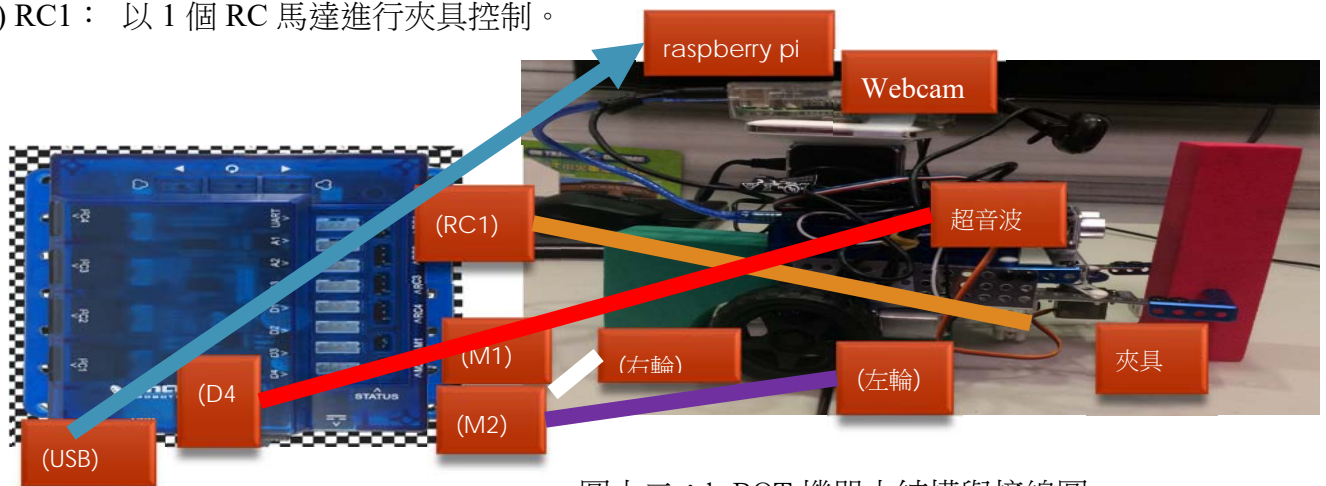
2. h-BOT 在硬體:包含樹梅派 Raspberry pi 3、喇叭、攝影機、行動電源、測距超音與 Matrix BOT 接線圖如圖十一。



圖十一：h-BOT 接線圖

在機器人方面主要是以 Matrix Mini 機器人套件作為本實驗之機器人、Matrix Mini 機器人套件包含:2 個 DC 馬達接口，4 個 RC 馬達接口，4 個數位和 3 個類比輸入/輸出端口，2 個 RGB LED 燈，3 個 Onboard 按鈕以及 4 個 I2C 接口如圖十二。

- (1) M1 and M2：以 2 個 DC 馬達進行控制車子行進方向
- (2) D4：1 個數位輸入/輸出端口進行超音波測距
- (3) USB：1 個 USB 進行與 raspberry pi 進行資料通訊。
- (4) RC1：以 1 個 RC 馬達進行夾具控制。



圖十二：h-BOT 機器人結構與接線圖

本研究自走機器人系統是以 Win 10 之 Arduino IDE 結合 Matrix mini Kit 進行自走機器人系統開發。以 Raspberry pi 之 WebCam 影像進行物體尋跡偵測，再經由 serial 傳訊及給 Matrix mini 機器人進行尋跡控制 (圖十三)。



圖十三：Matrix Mini 機器人開發流程

3. h-BOT 軟體介面設計：包含影像顯示區、語音啟動鈕、尋找物體設定、物體顏色校準，影像顯示亮度與對比調整，HSV 人機介面進行顏色校準、機器人回報執行結果顯示區(圖十四)。



圖十四：h-BOT 互動式人機介面設計

(二) 研究二、基於樹莓派之語音辨識 STT、TTS 語音控制研究與開發

本研究主要的目的是希望行動不方便的人，可以透過語音來控制之器人來尋找他們要的物品或者是幫他們取物，語音辨識為本研究一個主要的課題，語音辨識主要分為兩個部分文字轉語音 (Text to Speech) 與語音轉 (Speech to Text) 文字。

1. 研究文字轉語音 (TTS)

TTS 全稱是文字轉語音 (Text To Speech)，是指將輸入的文字或儲存於電腦中的文件，並模擬人聲發出語音的技術。

2. 研究語音轉文字 (STT)

STT 是以演算法，將語音內容轉換為相對應的文字。透過聲音特徵比對、足夠的語料收集，來建立龐大的語言資料庫，系統接收語音後立即比對語料庫，並將語音內容轉換為可能的文字。STT 目前的應用範圍非常廣泛，包括 Microsoft、Google Assistant、甚至 Apple Siri，都有採用 STT 技術，STT 技術因而變得普及，比 TTS 更廣為人知！比如說，記者採訪完畢，很多報館都會以最快的速度將新聞寫好推出，當中最常見的手法便是利用 STT 向手機讀出稿件內容，省卻了打字的時間。在語音辨識方面，本研究是在樹莓派中以 python 開發使用 Google API (圖十五(a)) 所提供的語音轉文字(STT)組還有文字轉語音(TTS)這兩個套件作為本研究主要的工具。

3 研究語音控制_NLP 關鍵字提取技術

我們可以藉由上述的功能將麥克風音源轉成文字，再藉由自然語言關鍵字提取技術如圖十五 (b)進行文字的關鍵字的分析，啟動 ihound 只令動作。本 ihound 系統語音控制總共分為 4 大類如下所示:

- (1)獵犬機器人自我介紹
- (2)獵犬機器人尋找物體

- (3) 獵犬機器人行走控制
- (4) 獵犬機器人夾爪控制



圖十五(a)：Google API 所提供的 STT 與 TTS

```

elif (sr_txt.find("關")!=-1) and (sr_txt.find("夾")!=-1):
    port1.write(b'c')
    text1=("獵犬機器人正關夾爪..")
    sr_txt2="close"
elif (sr_txt.find("開")!=-1) and (sr_txt.find("夾")!=-1):
    port1.write(b'o')
    text1=("獵犬機器人正開夾爪..")
    sr_txt2="open"

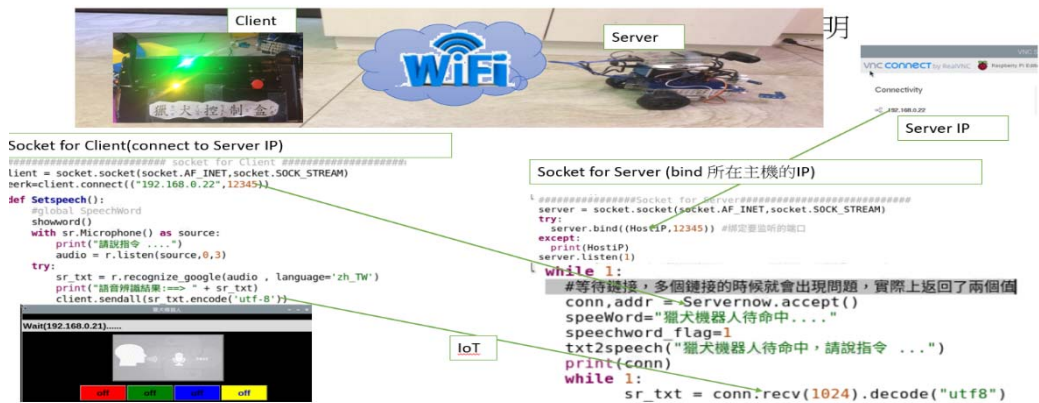
```

圖十五(b)：獵犬機器人夾爪控制關鍵字提取

(三)研究三、開發 IoT 技術進行兩樹莓派系統雙向相互通訊技術

本研究以 python Socket 套件連接兩個樹莓派一個為 Client & 另一個為 Server 如圖十六及圖十七：

- (1) 樹莓派 1 為 Client 端 (h-Box)：接一麥克風負責將語音轉為文字，在將文字由 socket 傳送至 Server。
- (2) 樹莓派 2 為 Server 端(h-BOT)：接 WebCAM、Matrix mini CAR，將 Client 傳送文字轉為指令，然後執行指令。



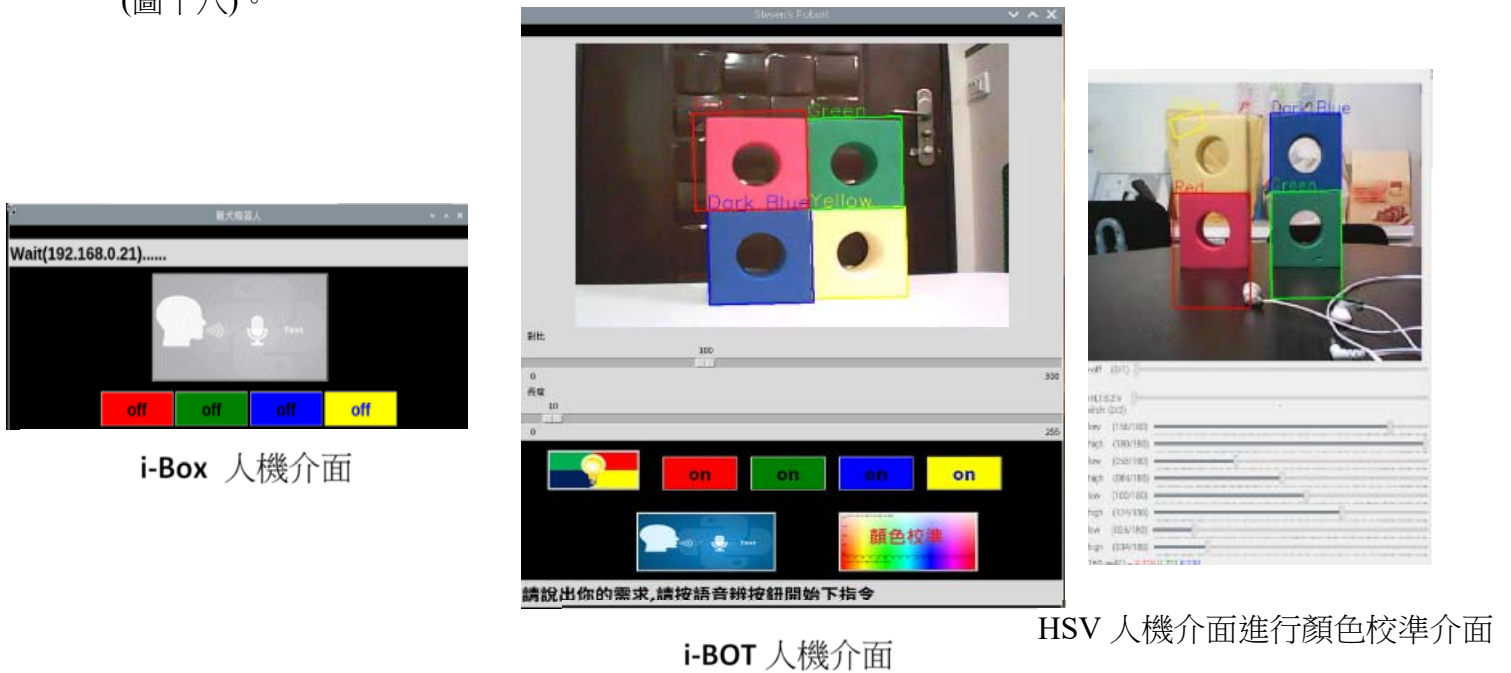
圖十六：IoT 技術進行兩樹莓派系統 h-BOX 與 h-BOT 雙向相互通訊技術



圖十七：IoT 技術進行兩樹莓派雙向相互通訊流程

(四)研究四、開發一圖形化人機觸控與聲控介面方便使用者操作

在 ihoun 開發一圖形化視窗介面[5]並整合多執行緒功能可以同時執行語音聲控和物體影像偵測與辨識。視覺化互動式人機介面包含影像顯示區、語音啟動鈕、尋找物體設定、物體顏色校準，影像顯示亮度與對比調整，HSV 人機介面進行顏色校準、機器人回報執行結果顯示區(圖十八)。

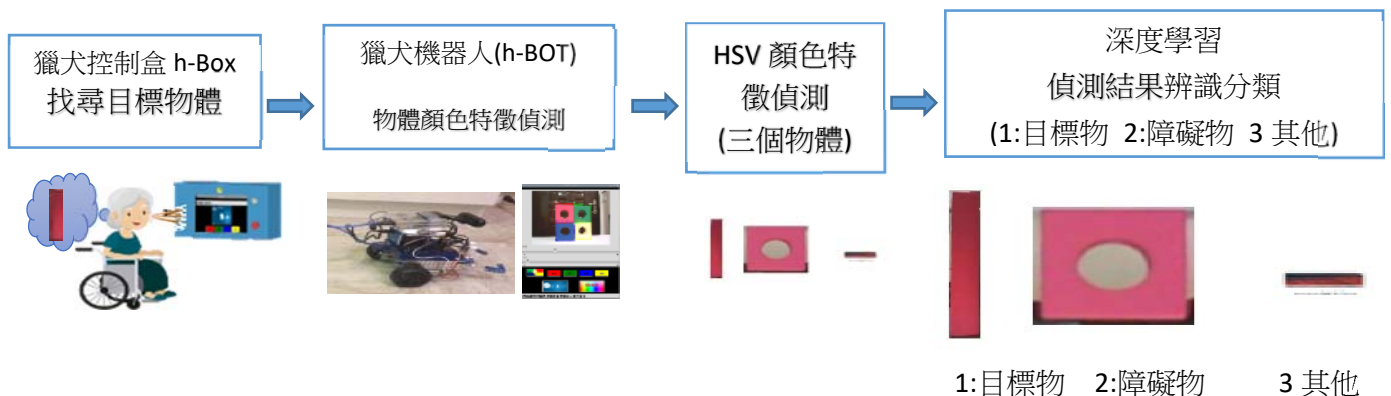


圖十八： ihound 人機介面

(五)研究五、樹莓派攝影機影像物體進行即時體偵測與辨識

本主題是要透過獵犬機器人中樹莓派攝影機偵測與辨識到目標影像，目標找尋方面，我們主要分為兩個階段，第一階段是屬於目標物偵測，第二個階段是屬於目標物辨識。在第一階段物體偵測方面本研究提出兩種方法，一個是小化影像模板匹配法另外一個是色彩特徵辨識法快速的偵測到我們所要找尋的物體。

然而在物體顏色特徵偵測法中雖然可以很快的偵測我們要的顏色物體，可是獵犬機器人在行進中可能會遭遇到同樣顏色可是不是我們的目標物的集合物體，所以在這邊本研究在過程中加入了一個的深度學習的卷積神經網路方法可以針對我們所偵測到的顏色特徵物體加以辨識分類是何種物體，最終找到我們的目標物如圖十九。



圖十九：物體即時體偵測與辨識流程

目標找尋方面我們主要分為兩個階段，第一階段是屬於目標物偵測，第二個階段是屬於目標物辨識。

1. 在目標物偵測方面本研究嘗試以兩種方法來偵測到目標影像。

其中偵測方法:

- (1) 以 OpenCV 之可縮放式影像模板匹配 (matchTemplate) 方法進行物體偵測。
- (2) 以 OpenCV 之 HSV 色彩模型方法進行物體偵測。

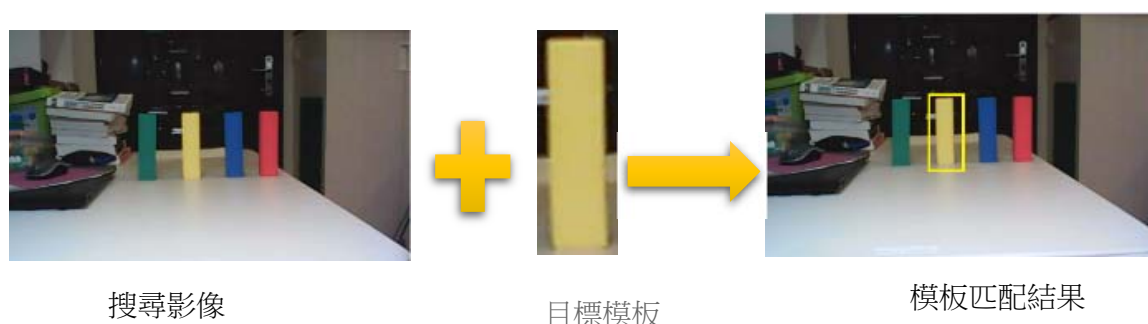
2. 目標物辨識方面本研究嘗試一個深度學習方法來辨識影像。

- (1) 使用 Keras 與 Tensorflow 卷積神經網路 (CNN) 進行物體辨識。

以下為各點詳細敘述。

3. 以 OpenCV 之可縮放式影像模板匹配 (matchTemplate) 方法進行物體偵測

如圖二十時，首先我們是以 OpenCV 所提共的 match Template[4] 的演算法進行影像比對，這個演算法需要 2 張影像: (a)搜尋影像 (b)目標模板。

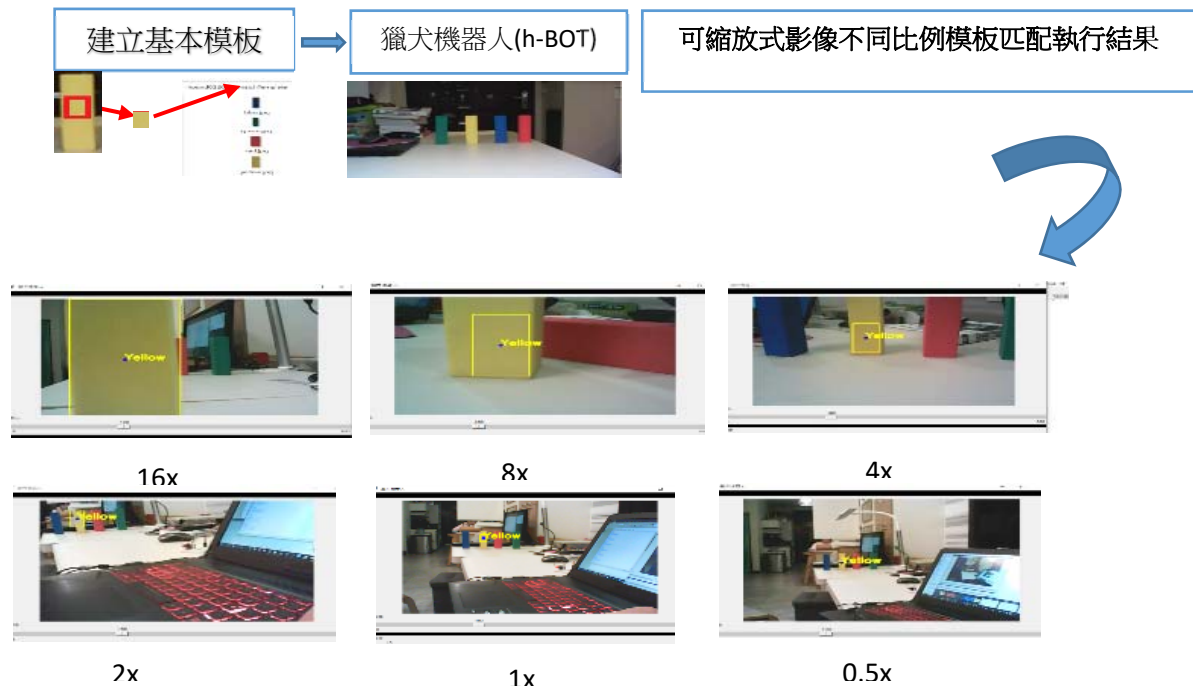


圖二十：物體偵測

模板匹配方法是通過目標模板在採集到的搜尋影像進行滑動尋找與目標模板圖像相似的目標。模板匹配不是基於直方圖的方式，而是基於圖像的灰度匹配。其基本原理是逐像素的把一個以一定大小的實時圖像窗口的灰度矩陣與參考圖像的所有可能的串口灰度陣列，按照某種相似度量方法進行搜索比較的匹配方法。

因為目標物大小會隨獵犬機器人距離遠近大小而改變，所以固定大小之樣板在獵犬機器人距離遠近上會出現問題。故本研究提出一可縮放式影像模板匹配(matchTemplate)方法進行物體偵測。

因為研究目標物為四個不同顏色積木，但在一個積木中本身顏色與材質(紋路)相同，故先擷取目標物一部分作為基礎模板，然後再物體偵測如圖二十一所示，過程縮放比例由 16、8、4、2、1、0.5 最大比例依序縮小模板進行匹配如下流程：



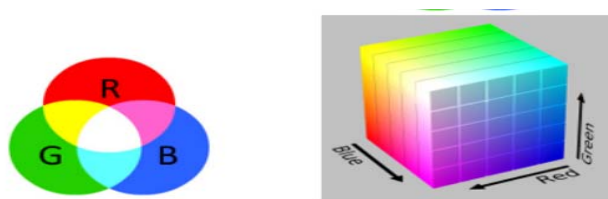
圖二十一:可縮放式不同模板縮放比例 16x、8x、2x、1x、0.5x 進行影像偵測結果

4. 以 OpenCV 之 HSV 色彩模型方法進行物體偵測：

本研究以物體顏色[3]作為尋找目標影像的特徵，所以我們會以攝影機所擷取的物體做影像的顏色辨識與分析。為了提升物體偵測的精確度和速度，我們比較了多種色彩空間的特性，發現 HSV 色彩空間受光線影響較小且色彩區辨力較強，因此我們將數位系統初得的 RGB 色彩空間數據轉為 HSV 色彩空間資訊並將之呈現在程式畫面上，而我們這裡是以 OpenCV 所提共的功能來做顏色辨識。接下來我們針對 OpenCV 的顏色來做一個說明。

(1) RGB 色彩模型:

在數位影像處理中常用的採用模型是 RGB（紅，綠，藍）模型和 HSV（色調，飽和度，亮度），RGB 廣泛應用於彩色監視器和彩色視訊攝像機，我們平常的圖片一般都是 RGB 模型，RGB 影像使用三種顏色。影像中每一個像素的 RGB 分量分配一個 0~255 範圍內的強度值，此模型是建立在直角座標系統的基礎上如圖二十二所示。RGB 是從顏色發光的原理來設計定的，它的顏色混合模式就好像有紅、綠、藍，當它們的光相互疊合的時候，色彩相混，而亮度等於兩者亮度之總和，越混合亮度越高。但此色彩空間容易受到物體的陰影、光線的明亮度有所影響，而轉換色彩空間之目的就是為了減少這些環境因子對影像所造成的影響。RGB 的影像顏色容易因明亮度的強弱影響產生顏色深淺的變化，就是說想要抽離出光線對於色彩的影響是比較困難的，比較不適合直接進行影像處理



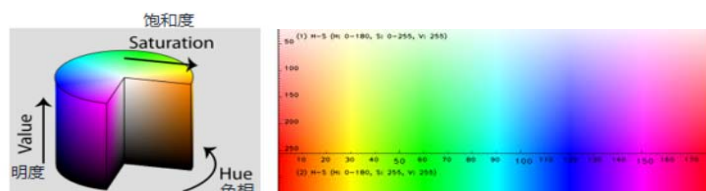
圖二十二：RGB 色彩模型

(2) HSV 色彩模型:

然而 HSV 模型更符合人描述和解釋顏色的方式，HSV 的彩色描述對人來說是自然且非常

直觀的。HSV 模型中顏色的引數分別是：色調（H：hue），飽和度（S：saturation），亮度（V：value）。由 A. R. Smith 在 1978 年建立的一種顏色空間，也稱六角錐體模型(Hexcone Model)。色調（H：hue）：用角度度量，取值範圍為 $0^{\circ} \sim 360^{\circ}$ (opencv $0^{\circ} \sim 180^{\circ}$)，從紅色開始按逆時針方向計算，紅色為 0° ，綠色為 120° ，藍色為 240° 。它們的補色是：黃色為 60° ，青色為 180° ，品紅為 300° ；飽和度（S：saturation）：取值範圍為 $0.0 \sim 1.0$ ，值越大，顏色越飽和。亮度（V：value）：取值範圍為 0 (黑色) ~ 255 (白色)。

為了減少環境光線變化對物體辨識的影響，本實驗採 HSV 色彩模型因為它對光源強弱較不靈敏，進行物體辨識。



圖二十三：HSV 色彩模型

(3) RGB 轉換 HSV

HSV 是將 RGB 轉換成為色度(hue)、飽和度(saturation)、亮度(value)也可稱“強度”(intensity)所組成，色相(H)是色彩的基本屬性，就是平常所說的顏色名稱，如紅色、黃色等。飽和度(S)是指色彩的純度，越高色彩越純，低則逐漸變灰，取 $0-100\%$ 的數值。明度(V)，亮度(L)，取 $0-100\%$ 。如圖 2.3 所示。其 RGB 彩色座標系統到 HSV 彩色座標系統轉換公式如下：

```

max=max(R,G,B)
min=min(R,G,B)
if R = max, H = (G-B)/(max-min)
if G = max, H = 2 + (B-R)/(max-min)
if B = max, H = 4 + (R-G)/(max-min)

H = H * 60
if H < 0, H = H + 360

V=max(R,G,B)
S=(max-min)/max
    
```

圖二十四：RGB 彩色座標系統到 HSV 彩色座標系統轉換公式

(4) OpenCV 中 RGB、HSV

OpenCV 中有個函式可以直接將 RGB 模型轉換為 HSV 模型。而在 OpenCV 中 HSV 個別值域分別為 $H \in [0, 180)$ ， $S \in [0, 255]$ ， $V \in [0, 255]$ 。

(5) OpenCV 中 HSV 顏色辨識

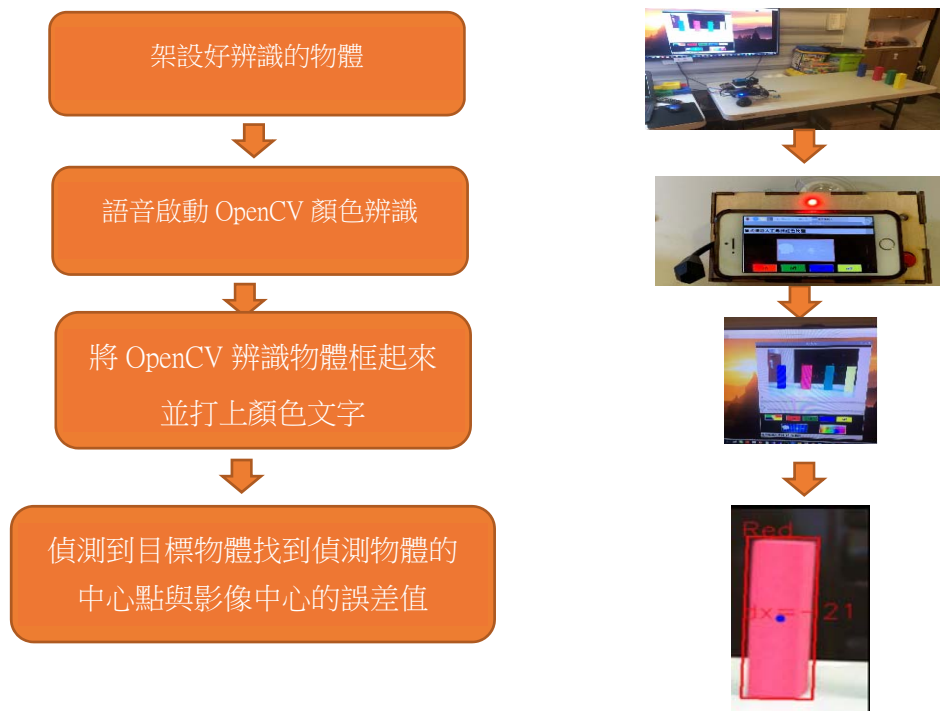
我們知道 H 分量基本能表示一個物體的顏色，但是 S 和 V 的取值也要在一定範圍內，因為 S 代表的是 H 所表示的那個顏色和白色的混合程度，也就說 S 越小，顏色越發白，也就是越淺；V 代表的是 H 所表示的那個顏色和黑色的混合程度，也就說 V 越小，顏色越發黑。經過實驗，識別藍色的取值是 H 在 100 到 124，S 和 V 約在 45 到 255 之間。

	黑	灰	白	紅	橙	黃	綠	青	藍	紫	
hmin	0	0	0	0	156	11	26	35	78	100	125
hmax	180	180	180	10	180	25	34	77	99	124	155
smin	0	0	0	43	43	43	43	43	43	43	43
smax	255	43	30	255	255	255	255	255	255	255	255
vmin	0	46	221	46	46	46	46	46	46	46	46
vmax	46	220	255	255	255	255	255	255	255	255	255

圖二十五：OpenCV_HSV 色彩空間表

(6) OpenCV 顏色辨識實作

我們將樹莓派攝影機所取得影像，根據上述所敘述以 OpenCV 功能來做影像偵測，這裡我們首先針對四種顏色積木來做探討就是紅色、藍色、黃色與綠色如圖所示。實驗流程如圖所示，整個樹莓派系統流程如圖所示



圖二十六：OpenCV 顏色辨識實驗流程

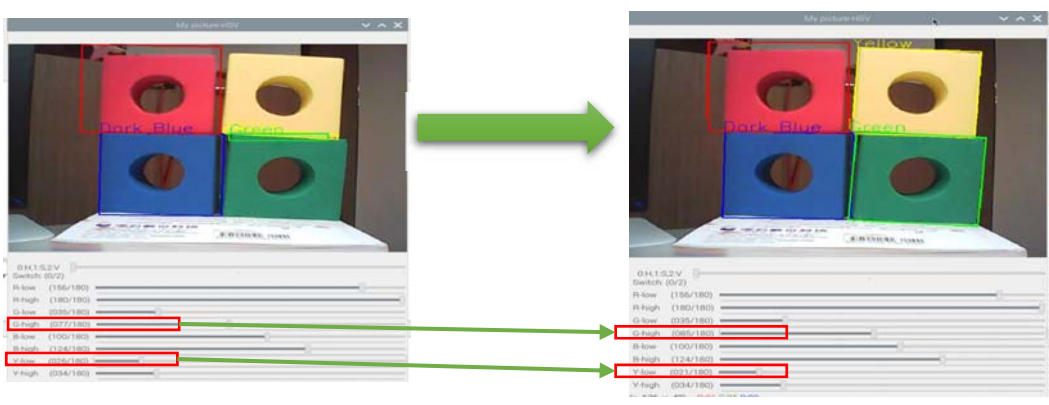
(7). OpenCV 顏色辨識顏色校準

由於物體的顏色在不同的燈光下仍然會有少許的誤差產生如圖二十七，所以必須要針對 OpenCV 個別的顏色的擷取範圍做些部分的調整，所以在這個的因素下我們開發了一個 HSV 人機介面可以針對 H、S、V 三個數值做調整如圖二十八，方便我們在做實驗的時候，可以針對不同的環境光來做 HSV 的參數校準，校準之後就可以完整的來做我們的顏色的辨識跟物體的尋找與取物如圖二十八。

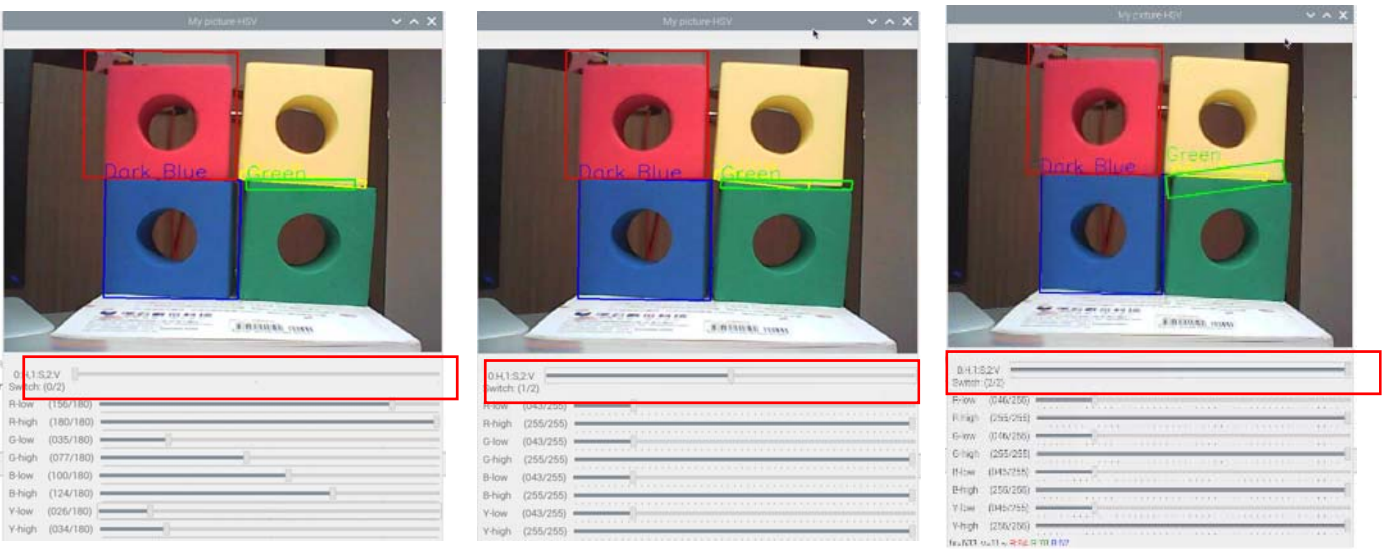
	紅		綠	藍	黃
hmin	0	156	35	100	26
hmax	10	180	77	124	34
smin	43		43	43	43
smax	255		255	255	255
vmin	46		46	46	46
vmax	255		255	255	255

```
H_lower_valu,H_higher_valu = np.array([156, 35, 100,26]),np.array([180, 77, 124,34])
S_lower_valu,S_higher_valu = np.array([43, 43, 43,43]),np.array([255, 255, 255,255])
V_lower_valu,V_higher_valu = np.array([46, 46, 46,46]),np.array([255, 255, 255,255])
```

圖二十七：系統以文獻資料為依據設定個物體



以系統界面進行 H(Hue)色相 微調結果



H(Hue)色校準介面(0~180)

系統 S 飽和度校準介面(0~255)

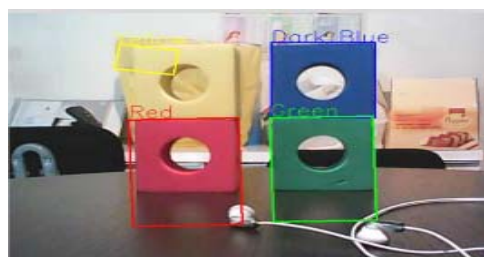
系統 V(Value)明度校準介面(0~255)

圖二十八：HSV 人機調整介面

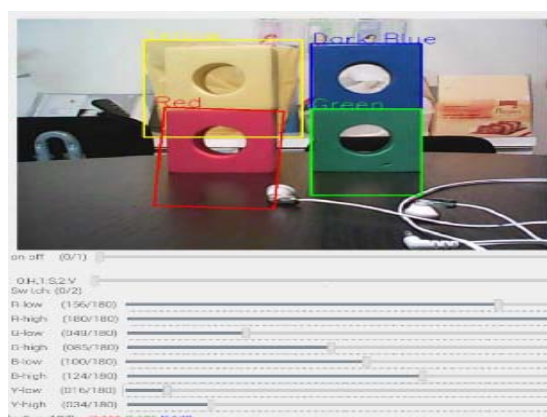
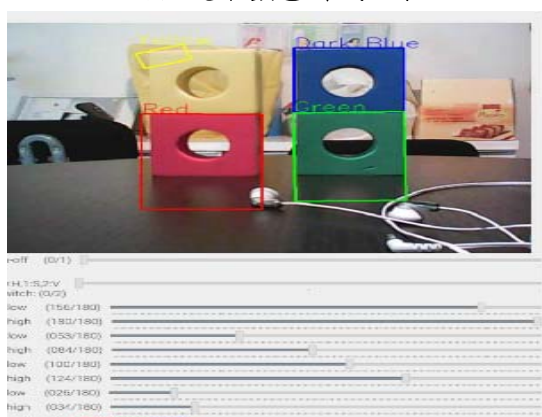
另外如圖二十九所示，原本在自然光下顏色辨識結果是正確無誤，但如果我們在環境開啟不同光源時，發現黃色積木辨識有些誤差，經以自行發 HSV 人機介面進行顏色校準後黃色積木辨識有些無誤。



自然光下顏色辨識結果



環境光下顏色辨識結果



圖二十九：自行開發 HSV 人機介面進行顏色校準

將以上完成之研究進行不同環境光對聲控尋物機器人對物體顏色辨識的影響。

結合上述結果，我們進行三種不同環境光對顏色辨識的影響實驗，實驗結果如下表五，由表格的數據可以得知黃色的物體對於光源的影響最大。

根據實驗結果我們分析或許是我們紅色、藍色、綠色的物體比較偏暗色系，所以對於環境光的影響比較小，黃色是屬於亮色系，對環境光的影響或許比較大，因為我們知道暗色系比較會吸收光源，亮色系比較偏反色光源，換句話說白色物體應該受環境光的影響會更大。

表五紅色、藍色、綠色與黃色在不同環境光的影響實驗值

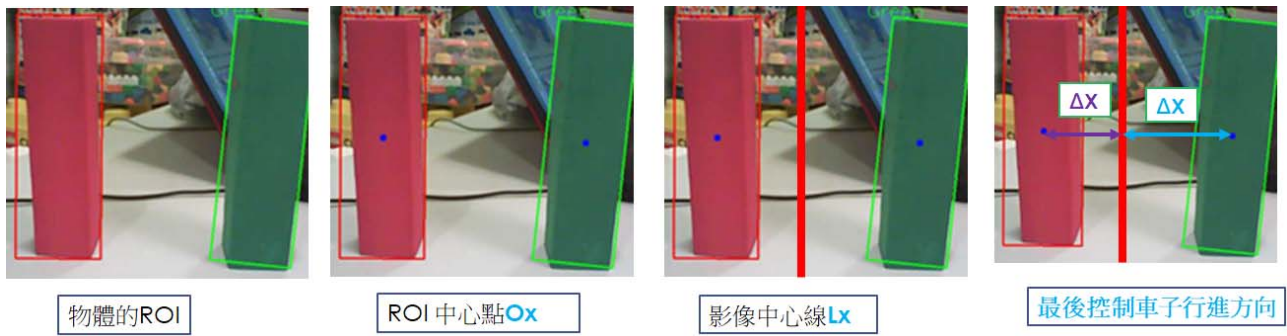
顏色	紅		綠		藍		黃	
	Hmax	Hmim	Hmax	Hmim	Hmax	Hmim	Hmax	Hmim
OpenCV	180	156	35	77	124	100	34	26
藍光	180	156	35	83	124	100	33	22
白光	180	156	35	83	124	100	30	20
黃光	180	156	35	83	124	100	16	30

(六)研究六、物體追蹤演算法技術開發

本系統在物體追蹤是以 Raspberry pi 之 WebCam 影像進行物體尋跡偵測，再經由 serial 傳訊息給 Matrix mini 機器人進行尋跡控制。

1.尋跡偵測：首先找出物體的 ROI(Region of Interest)感興趣區域，再找出 ROI 的中心點 o 之 X 座標 Ox 與找出影像中心線 Lx 座標，接著判斷 Ox 座標在 Lx 左右邊與 Lx-Ox 距離 ΔX，最後控制車子行進方向 (圖三十。公式如下。

$$\Delta X = Lx - Ox$$

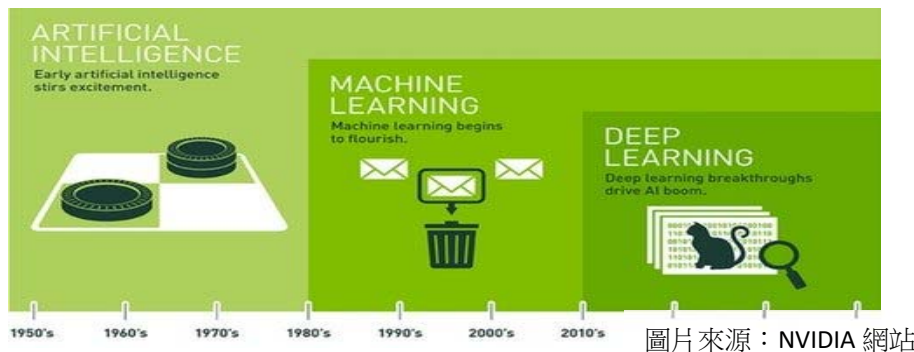


圖三十：物體追蹤

研究七、人工智慧、機械學習、深度學習介紹:

人工智慧/人工智能 (Artificial Intelligence, AI [Wiki, 1]) :是指由人製造出來的機器所表現出來的智慧。通常是指電腦模擬/模擬人類思維過程以模仿人類能力或行為的能力。

在圖靈所提出的圖靈測試中，如果一台機器與人類對話，而不被辨別出己方的機器身分時，便能宣稱該機器擁有智慧。「機器學習」是一門涵蓋電腦科學、統計學、機率論、博弈論等多門領域的學科，從 1980 開始蓬勃興起。電腦從大量的資料中找出規律來「學習」，稱為「機器學習」，也是「資料科學」(Data Science) 的熱門技術之一。最早研究機器學習的理論都是統計學家或機率學家所發表的，可以說是「資工+統計」的雙重 Domain Knowhow。另外，這也是為什麼「機器學習」會被稱之為人工智慧領域的一支。機器學習是由統計機率學、加上電腦科學方法的一個延伸學科，在數理邏輯研究方向失敗後，機器學習在 1980 年代到 2006 年間成為研究主流。但機器學習的理論有很多，包括支援向量機 (SVM)、決策樹 (Decision Tree)、AdaBoost、隨機森林等等。其中有一個分支在 1980 年初的時候一度興起，叫「類神經網路」(Artificial Neural Network)。



圖三十一：人工智慧、機器學習與深

類神經網路也就是深度學習的方法。也就是說人工智慧包在最外層、機器學習包在第二層、深度學習是第三層如圖圖三十一。

總而言之，三者的包含關係為：人工智慧 > 機器學習 > 深度學習

(一) 使用 Keras 卷積神經網路 (CNN) 進行物體辨識:

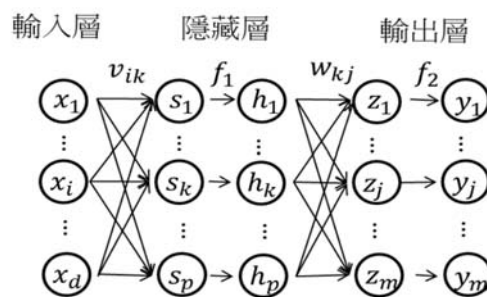
因為在執行物體顏色辨識過程，獵犬機器人上的攝影機可能找到同顏色但不是積木的物體，本文研究一深度學習方法，可以針對攝影機所偵測到的物體作辨識，來辨隊所偵測到的物體是否是要尋找的積木。

本研究在物體辨識方面主要是使用上深度學習方法，使用 Keras 多層感知器 (MLP) 與卷積神經網路 (CNN) 進行物體辨識

1. 多層感知器 (MLP)：多層感知機 (MLP, Multilayer Perceptron) 也叫人工神經網路 (ANN, Artificial Neural Network)

Artificial Neural Network)，除了輸入輸出層，它中間可以有多個隱層，最簡單的 MLP 只含一個隱層，即三層的結構，如下圖三十二：

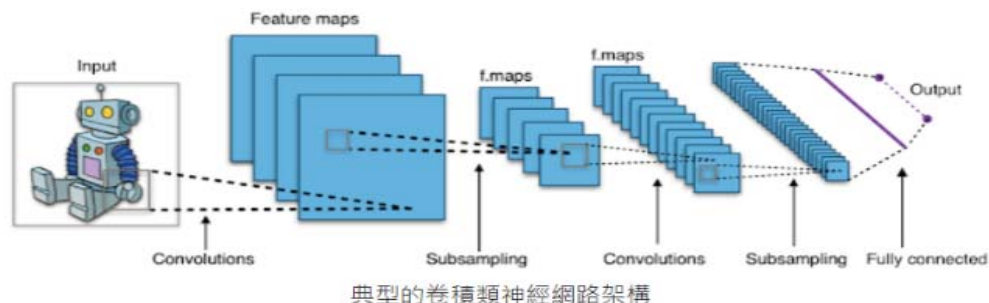
從上圖可以看到，多層感知機層與層之間是全連線的（全連線的意思就是：上一層的任何一個神經元與下一層的所有神經元都有連線）。多層感知機最底層是輸入層，中間是隱藏層，最後是輸出層。



圖三十二：神經網路

2. 卷積神經網路 (Convolutional Neural Network)

一般使用縮寫 CNN 來稱呼，它與傳統的多層感知網路最大的差異在於多了卷積及池化這兩層，也就是因為這兩層而讓 CNN 擁有能夠「看」到圖像（或語音）細節的能力，而非像其它神經網路只是單純的提取資料進行運算，無怪乎近年所舉辦的視覺競賽，優勝者幾乎都是採用深度學習的 CNN 架構。CNN 的基本思想非常的簡單直觀，利用多樣化的影像資料庫做為訓練影像，將影像利用數以百萬計的神經網路參數(一群具有特定功能的參數我們稱為 model)向網路輸出端傳遞，在輸出端計算目標與預測的誤差，藉由反向傳播學習不斷更新神經網路的權重值，造就卷積類神經網路可解決大量資料的問題，因此對於高變化性、大量且高維的影像辨識而言，具有很大的應用與研究價值，網路架構常包含單個或多個卷積層(convolution layer)、池化層(pooling layer, subsampling)，並在再輸出端連結全連接層(fully-connected layer, 原始的類神經網路)。典型的 CNN 如下圖三十三。全連接層神經網路包含平坦層 (Flatten)、隱藏層 (Hidden)、輸出層 (Output)。



典型的卷積類神經網路架構

圖三十三：圖源：https://en.wikipedia.org/wiki/Convolutional_neural_network

(二) 研究模擬環境因素之深度學習訓練圖片自動生成技術

在考量辨識物體資料收集時間與複雜度。本研究自行開發資料增量系統，透過 OpenCV 的功能將圖片上下翻轉、左右翻轉鏡像、放大縮小，亮度與透過高斯雜訊將每個圖片各個區域添加雜訊使其變模糊。再實作上本研究透過上下與左右翻轉、亮度明暗將資料集增量的方式，將原始單一圖片增加其資料量。尤其在車子行走過程遇到視線不佳的情況或行進間，使得動態視訊過於偏向模糊等，為了解決此問題，本研究參考 Zhang [2]的作法將高斯雜訊方法加入資料集增量。高斯雜訊是一種透過高斯分佈添加雜訊技術，通過常態分佈設定平均數與標準差，將圖片中每一點的像素添加隨機數值，形成高斯雜訊的圖片，也因此改善視訊模糊辨識不良並且增加資料量，解決訓練資料量不足的問題。轉化後高斯雜訊圖，原始圖片是張較清晰的圖片而透過高斯雜訊將圖片各個區域添加雜訊使其變模糊，藉此使機器能夠學習在惡劣環境辨成功辨識。

(三) Keras 卷積神經網路 (CNN) 分類器模型建立

本研究是在 PC 以 Python 語言 keras 套件進行深度學習進行卷積神經網路 (CNN) 分類器模型建立，在將模型載入樹梅派中進行辨識。由於 keras 是以 Tensorflow 為底層技術，所以必須在 PC 與樹梅派中同時安裝 keras 與 Tensorflow。

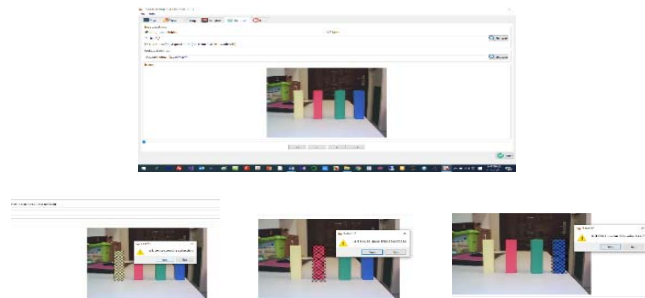
本研究在搜尋物體共有四個(紅色長方形積木、綠色長方形積木、黃色長方形積木與綠色長方形積木)，其中在物體偵測方面本系統是以 OpenCV 之 HSV 顏色作為搜尋物體 ROI 偵測，然後再以深度學習進行物體 ROI 辨識是否為長方形積木、障礙物或是其他。而深度學習分類器模型建立步驟如下。

1. 蒐集各積木正向資料

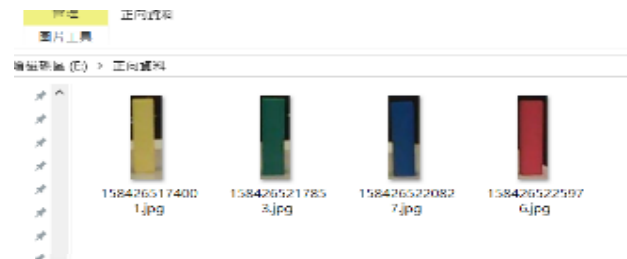
進行積木目標物 ROI 標記，在積木 ROI 標記方面本研究使用網路上所提供免費 OpenCV Cascade GUI 進行標記截圖。



圖三十四：實驗目標物截圖



圖三十五：OpenCV cascade GUI 進行標記



圖三十六：積木 ROI 標記截圖結果

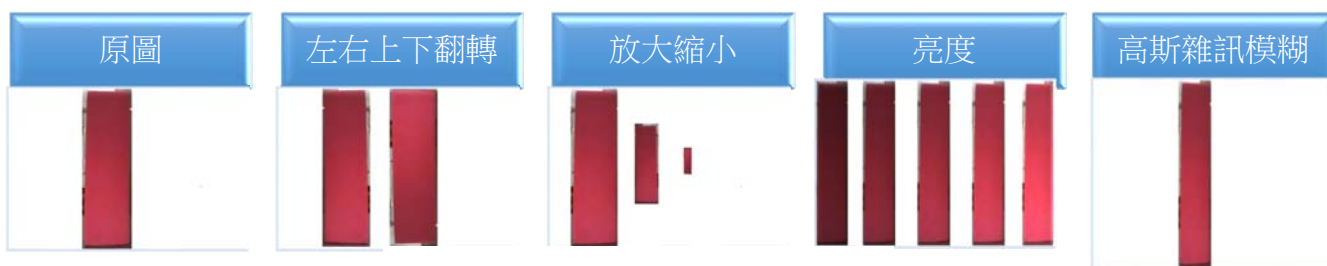
2. 資料訓練:本研究將訓練資料分為三類 $labs\{0:Goal,1:blocks,2:other\}$ ，其中:

(1) 0:Goal(目標物)以 OpenCV cascade GUI 手動標記產生 22 檔案



圖三十七：手動標記目標圖

接下來用自行開發的 Picture generator 系統將 22 圖片產生各自產生上下翻轉、左右翻轉鏡像、放大縮小、亮度與透過高斯雜訊模糊圖片產生 4031 (圖三十八)



圖三十八：圖像生成結果

(2) Label:block (障礙物)以 OpenCV cascade GUI 產生 24 檔案，如圖三十九。



圖三十九：障礙物手動標記

接下來用自行開發的 Picture generator 系統將 22 圖片產生各自產生上下翻轉、左右翻轉鏡像、放大縮小、亮度與透過高斯雜訊模糊圖片產生 2304，如圖四十。



圖四十：障礙物圖片生成結果

(3) 蒐集各積木負向資料

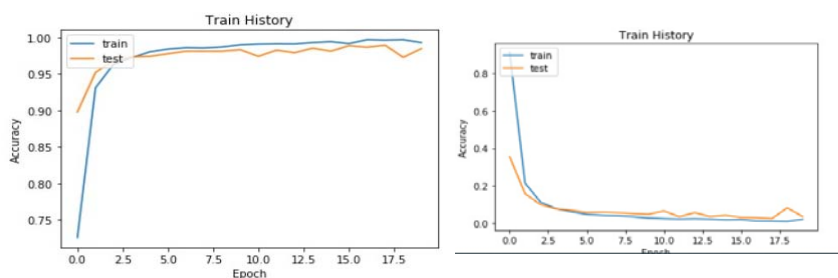
Label2:Other 以 OpenCV cascade GUI 框選不是積木的影像資料產生 900 檔案，如圖四十一。

(4)分類器模型參數研究:

接下來本研究以 Keras 卷積神經網路 (CNN) 建立物體辨識分類器模型，過程我們探討在類神經網路不同的神經元數、卷積層數與池化層數對模型辨識準確率，平均誤差的影響。本研究訓練樣本總數為(T)=目標積木圖片數(G)+障礙物圖片數(B)+負面資料圖片數(O)即: $T=G+B+O=4031+2304+900=7235$ 。

本研究訓練實作過程，會將總數 7235 大約 30%做為測試資料，總數 70%作為訓練資料。另外總數 70%訓練資料中的 20%作為訓練過程驗證資料。測試資料約 $T*30\%$ ，驗證資料= $T*70\%*20\%$ 。訓練次數 Epoch=20，每次訓練取樣數 Batch_Size 固定為 128。T=7235 樣本數。表七

中實驗 1 的訓練結果:loss: 0.0221 - acc: 0.9933 - val_loss: 0.0367 - val_acc: 0.9848，評估準確率: 0.989 平均誤差:0.03。準確率及誤差曲線圖，如圖四十二。



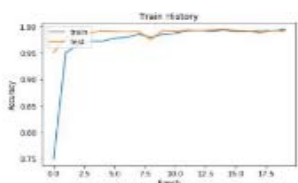
圖四十二： 實驗 1 準確率及誤差曲線圖

根據混淆矩陣分析圖 (圖四十三) 可以知道，label0 目標物體 與 label2 其他物體容易混淆辨識錯誤。測試資料準確率: 0.9904。

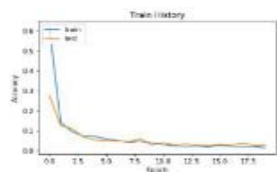
predict \ label	0	1	2
0	1275	0	22
1	0	800	0
2	1	0	313

圖四十三： 實驗 1 混淆矩陣分析圖

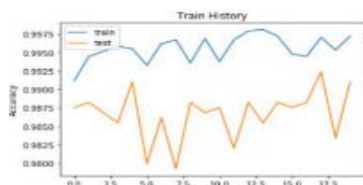
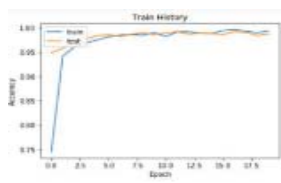
就以上實驗的結果來看 (表六)，一卷積神經網絡在本研究訓練的成果都相當的不錯準確率都接近於 1，從微觀來看實驗 4 的準確率是最高且誤差最小，但從準確率-誤差曲線圖圖如四十四來看，其訓練過程收斂不過平滑。所以本研究決定以其次且平滑收斂實驗二作為本研究實作訓練參數來建立實驗的模型訓練。



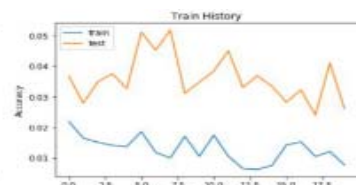
實驗 2 準確率-誤差曲線圖



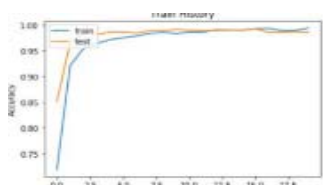
實驗 3 準確率-誤差曲線圖



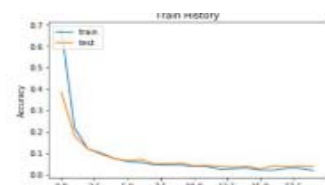
實驗 4 準確率-誤差曲線圖



實驗 5 準確率-誤差曲線圖



實驗 6 準確率-誤差曲線圖



圖四十四： 實驗 2~6 準確率及誤差曲線



圖四十五：實驗 2~6 混淆矩陣分析圖

表六：本研究相關實驗之關鍵數據整理

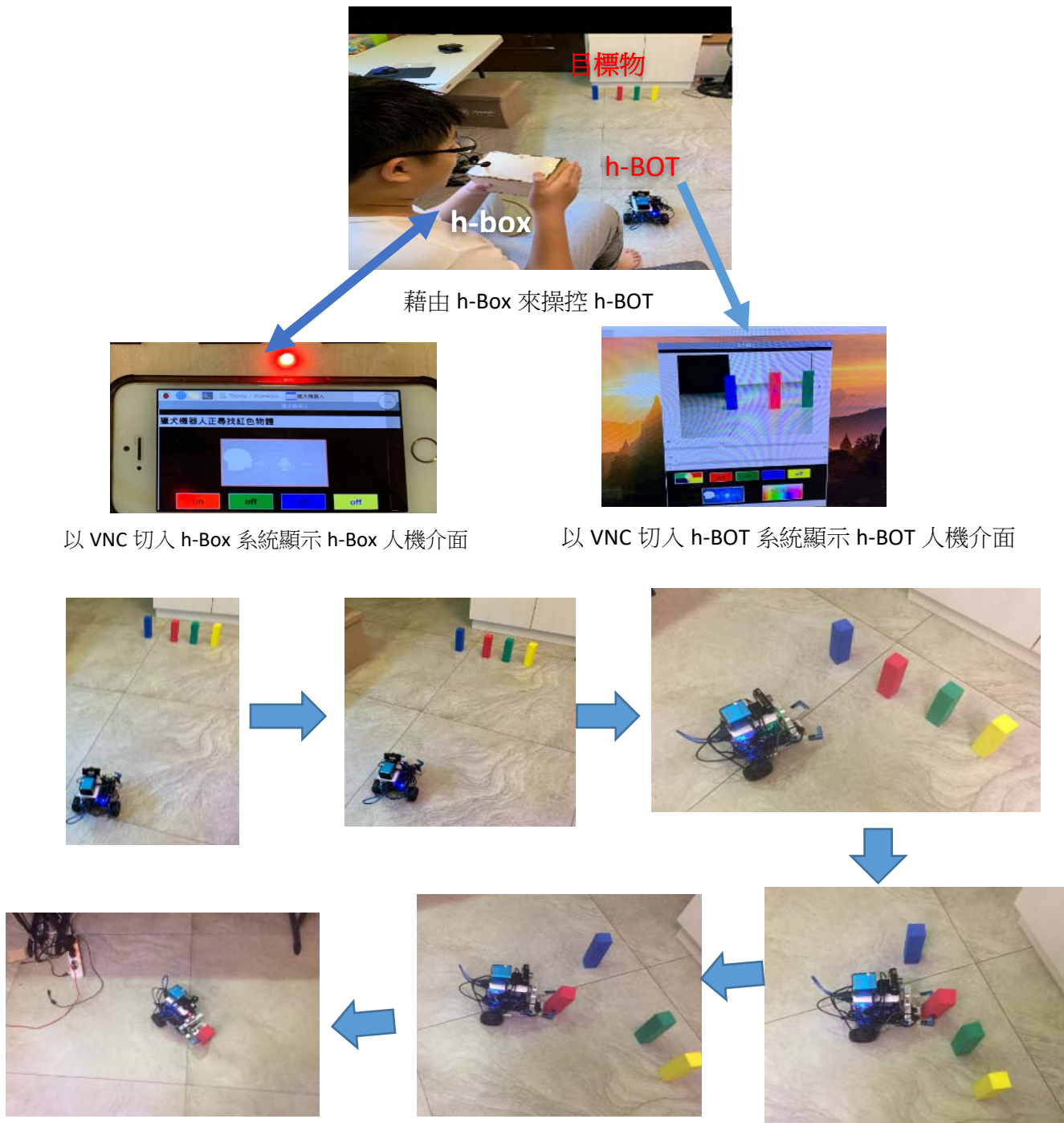
實驗編號	卷積層數(卷積個數)	池化層數(影像所檢比例)	神經元數			評估準確率	測試資料準確率	平均誤差
			平坦層	隱藏層	輸出層			
1	1(32)	1(2,2)	8192	1024	3	0.989	0.9904	0.03
2	2(64)	2(2,2)	4096	1024	3	0.9940	0.9966	0.02
3	3(128)	3(2,2)	2048	1024	3	0.990	0.992	0.025
4	1(20)	1(2,2)	5120	1024	3	0.9941	0.9975	0.017
4-2	1(20)	1(2,2)	5120	512	3	0.9894	0.9962	0.031
5	2(40)	2(2,2)	2560	512	3	0.9953	0.9962	0.0288
6	3(60)	3(2,2)	960	512	3	0.99035	0.9879	0.0280
7(2)	2(64)	2(2,2)	4096	512	3	0.9935	0.9962	0.021
8(2)	2(64)	1(2,2)	2(2,2)	1024	3	0.9945	0.9917	0.0187

另一方面從所有實驗的混淆矩陣分析圖如圖四十五來看，label0 目標物體與 label2 其他物體，容易混淆辨識錯誤，所以後續必須收集更多的 label2 來作訓練。

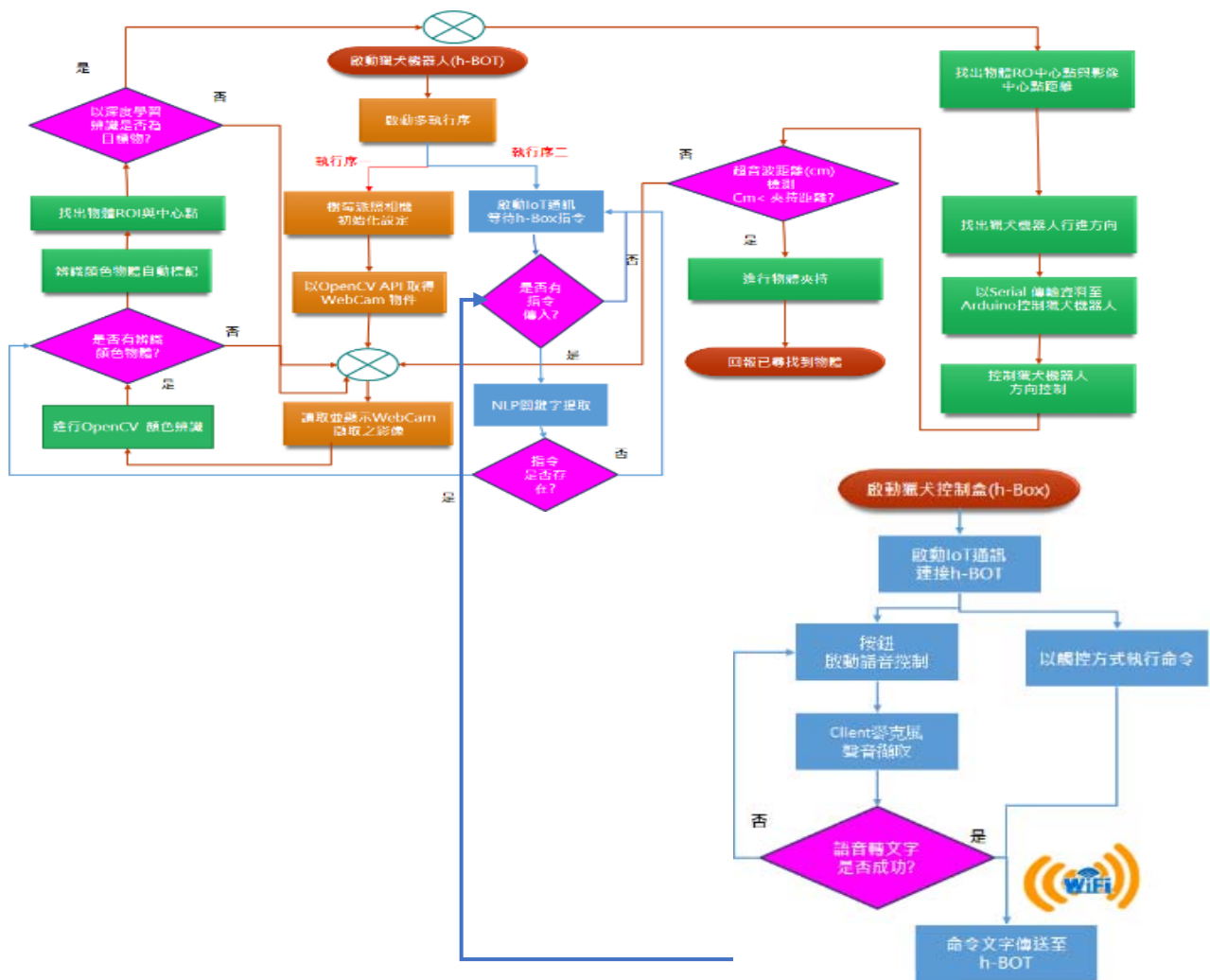
伍、研究結果

一、本研究完成結合觸聲雙控人機互動式雲端智慧語音獵犬機器人(ihound)的研究
 本研究結合樹莓派、Arduino、自走車、物聯網、語音辨識、影像處理偵測與辨識和深度學習開發一 ihound 提供使用著尋找物體。使用著可藉由獵犬控制盒(h-Box)以 IoT 無線雙向通訊技術進行國語語音或以觸控方式操作獵犬機器人(h-BOT)，實作過程如圖四十六。

如圖四十七為本系統使用流程，使用者藉由獵犬控制盒(h-Box)透過本研究開發 IoT 無線雙向通訊技術、多執行序與互動式人機介面來操作 h-BOT，h-BOT 開始以本系統所研究之語音辨識與聲控進行物體偵測與辨識技術與物體追蹤演算法技術來尋找物體。



圖四十六：h-Box 來操作 h-BOT 物體偵測與辨識實作



圖四十七：ihound 系統使用流程

二、物體影像辨識方法在機器人物體夾取之影響

本研究提出兩個偵測方法：一為以 OpenCV 之可縮放式影像模板匹配(matchTemplate)方法進行物體偵測，另一個為以 OpenCV 之 HSV 色彩模型方法進行物體偵測，這兩個方法實作結果，不管方法一或二一般而言都可以尋找到顏色目標物，但如果 h-BOT 在偵測物過程，如果環境有相同顏色物體就會造成 h-BOT 偵測誤判，使得 h-BOT 開始迷航因而失敗。所以基於上述子原因為系統研究了一個深度學習卷積神經網路 (CNN) 進行物體辨識方法，根據上設表實驗結果，我們選取實驗二作為我們的訓練參數，訓練物體辨識模型(*.h5，如圖四十八)，然後在將載入模型進行影像偵測與辨識。

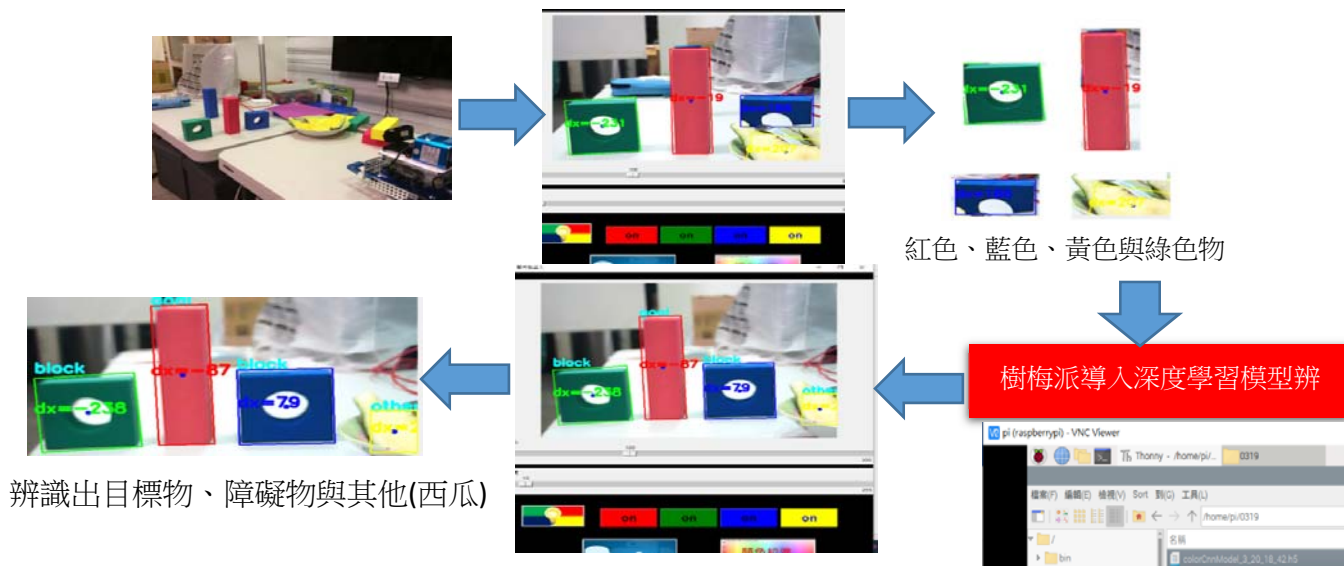
colorCnnModel_3_20_1_41.h5	H5 檔案	32,808 KB
colorCnnModel_3_20_1_38.h5	H5 檔案	8,601 KB
colorCnnModel_3_20_1_31.h5	H5 檔案	32,808 KB
colorCnnModel_3_19_23_27.h5	H5 檔案	30,771 KB
colorCnnModel_3_19_23_3.h5	H5 檔案	7,732 KB
colorCnnModel_3_19_11_51.h5	H5 檔案	8,601 KB
colorCnnModel_3_19_11_50.h5	H5 檔案	16,501 KB
colorCnnModel_3_19_11_49.h5	H5 檔案	32,808 KB
colorCnnModel_3_19_11_46.h5	H5 檔案	32,808 KB
colorCnnModel_3_19_10_45.h5	H5 檔案	8,601 KB

圖四十八：本研究所訓練出來的物體辨識模型檔案

1. h-BOT 樹莓派深度學習物體辨識實作

在深度學習實作過程，深度學習模型我們是在一般 PC 辨識模型訓練，訓練好的模型我們會

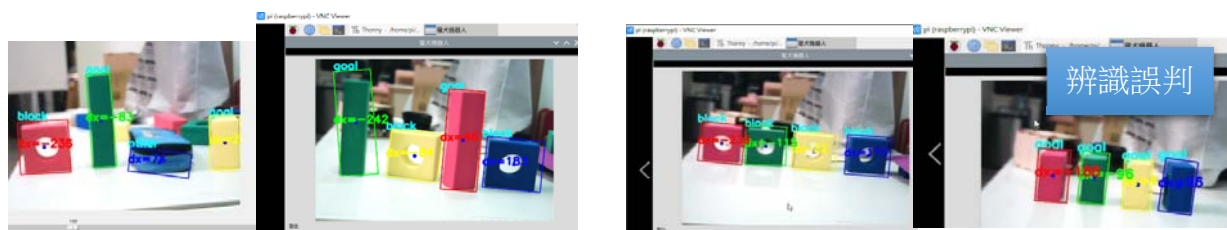
把它傳到樹莓派 (Raspberry pi) 裡面進行辨識，所以必須再 PC 與樹莓派 Python 上安裝 keras 與 Tensorflow 套件。實作如下我們目標物為共有四個 (紅色長方形積木、綠色長方形積木、黃色長方形積木與藍色長方形積木)，實驗過程前述如圖十九，一開始以 HSV 顏色辨識會發現不管何種形狀積木，只要顏色是我們找顏色都會辨識出來，黃色西瓜都會誤以為是目標 (如圖四十九)，接下來這四個被偵測而被框選的物體會進行深度學習分類辨識，經導入深度學習模型即可辨識出目標物、障礙物與其他。後續 h-BOT 只需要針對目標物做定位與追蹤即可。



圖四十九：樹莓派導入深度學習模型辨識實

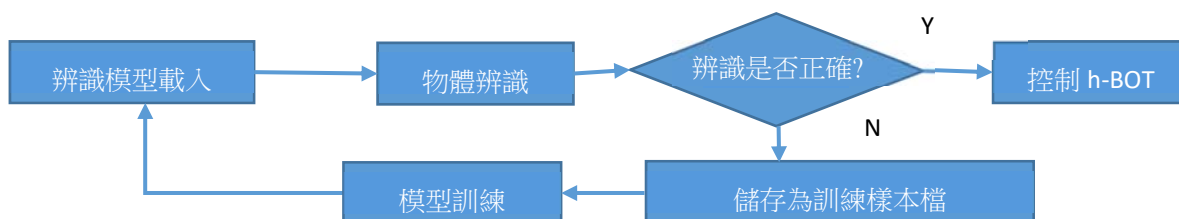
2. 物體辨識誤判

另外本研究在實作過程當中發現一個有趣的問題，就是我們的障礙物它轉 90 度之後會跟在辨識的時候有可能會跟我們的目標物幾何特徵幾乎是一模一樣，變成式的打法庭造成變成個變數位發生錯誤，如圖五十，造成機器人在尋找物體的時候辨識錯誤，後續在實作的時候盡量要找兩個特徵不相同的物體做為目標物與障礙物即可。



圖五十：深度學習物體辨識結果與物體辨識誤判

3. 辨識回饋訓練與辨識: 本研究在進行深度學習辨識的時候，會將誤判的資料存檔，作為後續訓練的樣本。以作為後續對模型加強訓練的資料，以再次提升模型的辨識能力，如圖五十一。



圖五十一：辨識回饋流程

六、討論

本科展在研究、測試與實作過程中有遭遇到其一些問題，在不斷與老師、組員討論加上閱讀一些參考資料後，本研究總共提出幾個自我解決方案如下：

一、機器人超音波距離偵測：

本研究一開始在機器人超音波距離是以 **Matrix** 所提供 **API** 來進行超音波距離測。但實作時發現 **API** 所距離偵測距離不穩定。後來就改為自行開發，後續就沒有不穩定，如圖五十二所示。



圖五十二：將廠商機器人 API 改為自行開發

二、在 **ihound** 研究上，原本只規劃以聲控為主，但後續發現在目前語音辨識系統中，大多研究主要是以中文為主，對於台語、客語目前並沒有合適語音庫提供。所以本研究，後來決定另外研究一觸控方法，以結合圖像式觸控與聲控雙控互動式人機介面系統，提供非中文以外的人使用。

三、在物體偵測與辨識實作中，研究以 **OpenCV** 的套件中提供了物體 **HSV** 顏色偵測作為目標物的外觀偵測。但在目標物偵測時會將所有同顏色物體偵測出來。所偵測未必是真正目標物，會造成 **ihound** 誤判而搜尋物體失敗。所以本研究以 **Keara+Tensorflow** 方法研究一卷積神經網路深度學習進行物體形狀的外觀分類辨識。最後並將 **OpenCV** 偵測技術與深度學習分類辨識技術整合成一系統流程。目前本方法 **ihound** 在一般正常角度皆可辨識與夾取成功。但在實作過程中，會因為角度關係障礙物辨識會跟目標物形狀辨識一樣如圖五十所示。這個問題只須後續找差異大的物體來做障礙物即可。

四、在深度學習資料收集上，往往需要大量訓練資料，如果都由人工擷取需花費大量時間。另一方面要模擬獵犬機器人在真實的環境運行中要考慮到環境的燈、光物體的遠近大小與動態影像的擷取。本研究也參考文獻[2]的研究後，決定研究以 **Python** 結合 **OpenCV** 影像處理 **API**，自行研究開發一資料增生系統。將目標物基本圖片資料進行上下翻轉、左右翻轉鏡像、不同比例放大縮小、不同亮度與透過高斯雜訊與模糊化將每個圖片各個區域添加雜訊使其變模糊，藉此使機器能夠學習在惡劣環境辨成功辨識。

柒、結論

一、本研究在物體偵測與辨識上結合影像處理、偵測與辨識。尤其研究結合深度學習上在物體辨識。整套系統從樹梅派以 **OpenCV** 影像擷取、處理與偵測到最後以 **keras** 與 **tensorflow** 進行卷積神經網路進行物體辨識分類有不錯成果。但目前只能應用在四個顏色積木物體辨識，後續擴充深度學習方法在其他物體上，或再專研其他方法如 **Yolo**、**Faster RCNN**...等。

- 二、本研究在相關的軟體技術研究上，雖然最後研究成功在樹莓派開發許多程式與技術。但有許多都是以 Python 結合既成的套件或 API 來應用開發如以 OpenCV 套件進行影像處理、偵測與辨識、語音辨識、多執行序與圖片自動增生系統等。以 keras 結合 tensorflow API 開發深度學習或機器人控制是以 C++結合廠商 lib 或 API 來開發應用。只有極少數為自行開發的演算法如追蹤演算法等。希望在後續科展(高中)可以有更多自己開發的演算法。
- 三、最後本研究從無到有研究與實作一個結合電腦視覺、語音指令辨識、IoT(物聯網)、影像擷取、處理、偵測、深度學習分類辨識、自走車與控制版(Arduino and Raspberry pi)相關技術的雲端智慧語音機器人(ihound)。讓使用者可以透過聲控或觸控來找尋物體。在此次科展研究過程中除從老師身上學習許多技術與經驗，另外也透過自我學習研究了相關硬體與軟體技術。雖然過程中也遭遇一些失敗，幸好透過問題分析與討論也解決一些問題。對我們而言本次科展是一個不錯成長與經驗。美中不足的是因為 ihound 是第一次研究，我們在 ihound 的軟體技術上花了相當的研究時間。所以在 ihound 外型開發一簡易的架構，後續研究可以列為改良項目之一讓外型更美觀與科技。

捌、參考資料

[1] 人工智慧 wiki

<https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD>。

[2]Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing, 26(7), 3142-3155. OpenCV HSV2RGB Class Reference

[3]https://docs.opencv.org/3.4/d8/d00/classcv_1_1cudev_1_1color__cvt__detail_1_1HSV2RGB.html

[4]OpenCV _template_matching:

https://docs.opencv.org/3.4/de/da9/tutorial_template_matching.html

[5]Graphical User Interfaces with Tk

<https://docs.python.org/3/library/tk.html>

[6]<https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD%E5%8F%B2#%E6%B8%B8%E6%88%8FAI>

[7] <http://www.prowesscorp.com/whats-the-difference-between-artificial-intelligence-ai-machine-learning-and-deep-learning/>

[8] Yann LeCun, Yoshua Bengio & Geoffrey Hinton, Deep learning, Nature, Vol. 521, pp. 436–444, 2015.

[9]周光中、柳世民(2017)。數位信號處理器應用於自走車之環境地圖建立。高雄應用科技大學工程科技學刊，3(1)，41-52。

[10]林志哲、劉政鑫、林啟瑞(2012)。基於實數編碼之遺傳演算法於無人自走車的路徑規劃。車輛工程學刊, (9), 67-82。

[11]徐復國、吳智謙 (2013)。多媒體服務機器人之間發。明新學報，39(2)，27-43。

[12]陳冠宇、郭瑞濠、曾彥博 (2011)，立體視覺伺服雙目標追蹤系統之發展。先進工程學刊, 6(2), 117-123

[13]廖憲正、郭志忠、林政賢 (2015)，室內長距離語音辨識技術挑戰與初探。電腦與通訊，164，42-54。

[14]蒲永仁、施安鍵、李素幸(2015)。以飽和度與亮度進行電腦視覺煙霧辨識研究，勞動及職業安全衛生研究季刊，23(1)，16-24。