

Intelligence-Closet

智慧衣櫃

解決你的一切穿衣問題!

組別：A01

指導教授：林世崧 博士

學生：黃懷萱、陳墨兮、詹惠茹

摘要

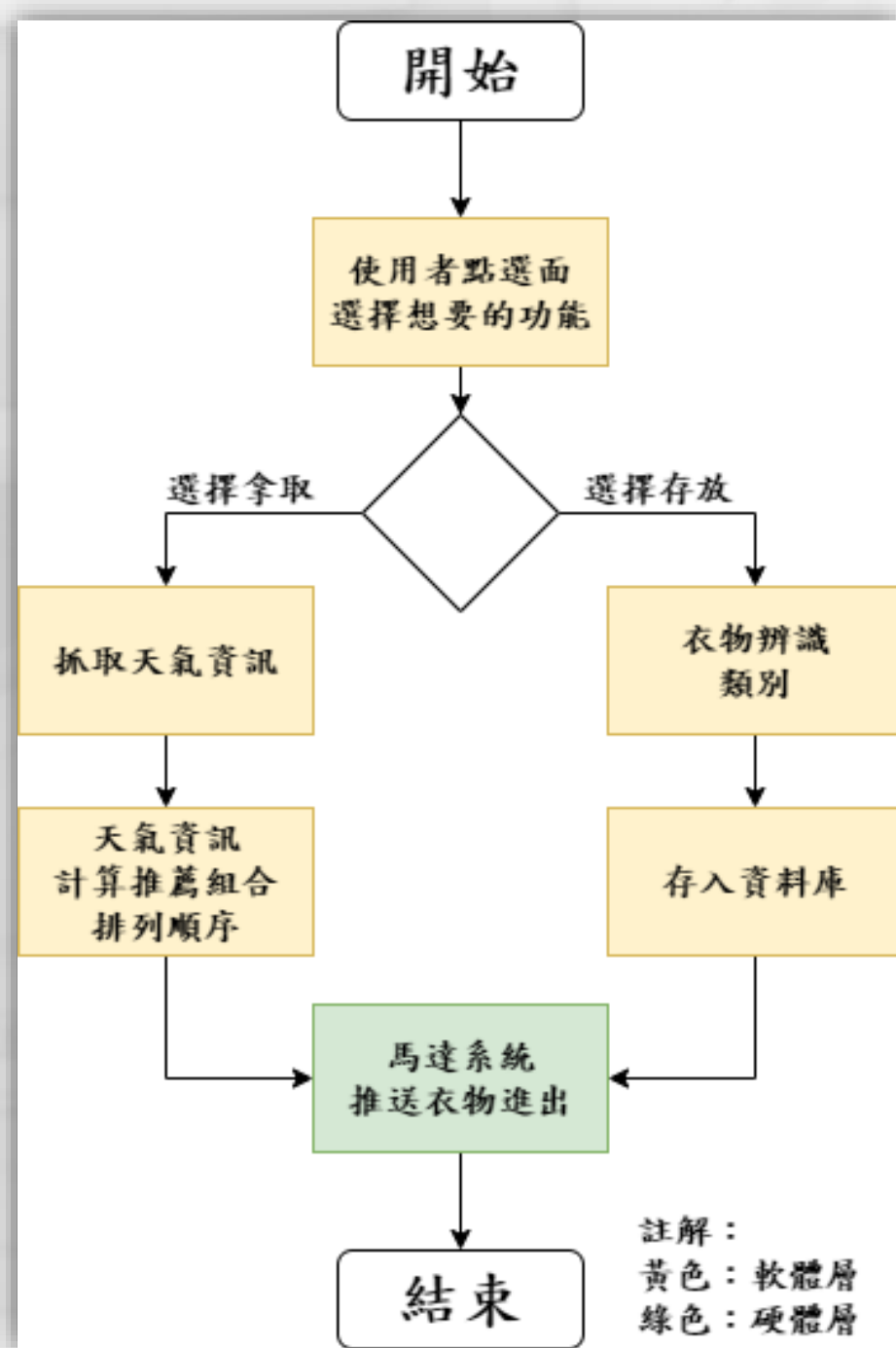
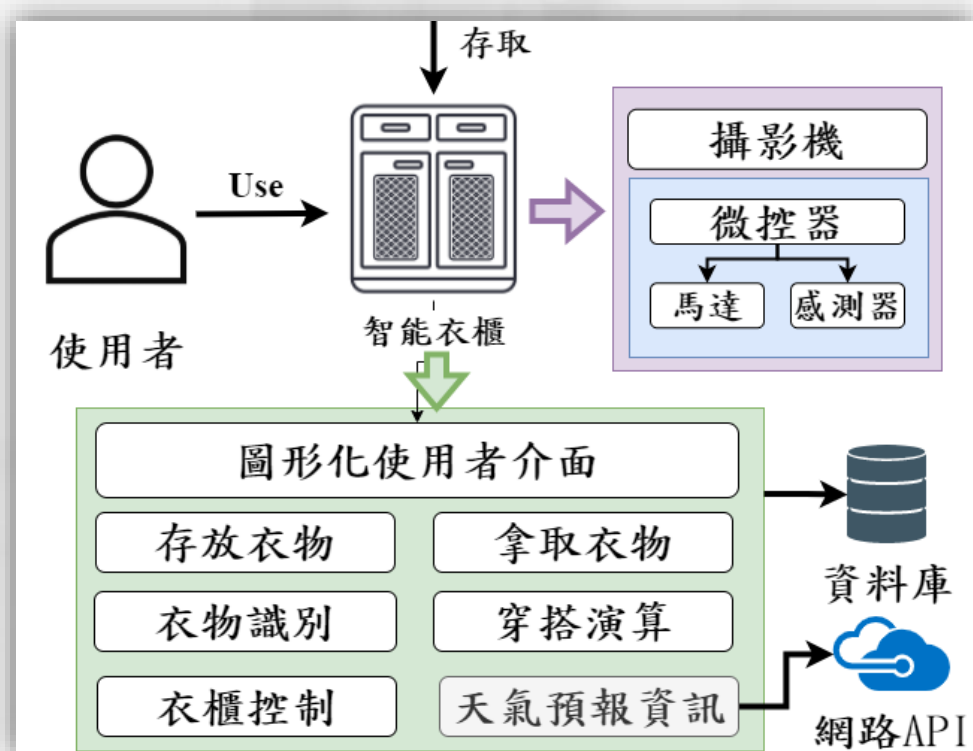
因應現代技術，大多事務都傾向於自動化趨勢，本研究計畫完成一台智能衣櫃。其中包含衣物辨識，讓使用者放入衣物至檢視口後不用手動輸入項目細節便可存入衣物資訊，完畢後再自動運轉將衣物整理擺放至位置，只需要將衣物放入衣櫃就可以在面板上查看所有的衣物資訊，再者是使用時會自動抓取天氣資訊並自動計算出當時最適合的衣物組合，透過UI介面選取心儀的衣物組合，接著會由衣櫃內部的馬達運轉自動推送到使用者面前，不需要翻箱倒櫃的搜尋衣服，利用機器來記錄並完成所有事物。

流程

當使用者點選介面功能後，若選擇存放，系統會先將衣物進行辨識動作，辨識完後的衣服類別可自動存入至資料庫當中；而當使用者選擇拿取功能時，需聯網獲取外部API得到天氣資訊，並根據該天氣資訊計算推薦衣物組合，上述這些步驟皆為軟體層可達到的目標。而最後須由馬達與各項零件配合，來將衣物進行推送進出等動作，而這些為硬體層的功能建造與配置。

系統架構圖

Web應用框架為概念基礎所建立的圖形化使用者介面負責供使用者使用，這些前端頁面將會發送的請求至後端Python來進行動作，包含資料庫的傳輸、硬體的控制、衣物的辨識、網路API請求以及推薦穿搭衣物演算法的運作等。



研究內容



自動搭配：

還要用手機查天氣太麻煩？智慧衣櫃！

＝找衣服＋查天氣＋穿什麼

＝減少思考穿搭衣物組合的時間

方法：外部天氣API→天氣推薦演算法→獲得推薦衣物

自動拿取存放：

衣物太多找不到？智慧衣櫃！

＝減少找尋衣物的時間＋可視化界面＋衣物自動送出

＝衣物一目了然

方法：資料庫＋Arduino（馬達等系統）

衣物識別：

紀錄衣服太麻煩了好懶？智慧衣櫃！

＝自動拍照＋自動辨識＋自動記錄

方法：深度學習辨識＋資料庫

Intelligence-Closet

智慧衣櫃

組別：A01

指導教授：林世崧 博士

學生：黃懷萱、陳墨兮、詹惠茹

解決你的一切穿衣問題!

機電整合 天氣推薦

機器 (Arduino)

透過單晶片，Arduino MEGA 2560，來建立一套控制硬體的流程，下圖是硬體概念設計與命名方式，如最外層為檢視口，內部存放空間為圓盤等，輸送衣物為模型車。



中央氣象局抓取天氣資料

根據濕度、溫度、使用者喜好度等演算

UWL代表使用者喜好的溫度傾向，NT代表的目前的氣溫，NH為現在偵測濕度，全身保暖程度設為TCWL，使用者喜好度設為UL，CWL代表衣物的保暖程度，UWL代表使用者喜好的溫度傾向。

$$|26 - UWL - NT - (5 - TCWL)| + \left| \begin{array}{c} (26 - NT + 5 - TCWL) * \frac{7}{5} \\ + \\ (60 - NH) * \frac{3}{5} + (5 - TCWL) \end{array} \right| + UL$$

溫度越高 / 低，濕度的權重比越高。

人體最適合的溼度為60% (氣象局新聞稿, 2019)，當溫度太高而濕度的溫度取向高會因為人體無法有效的排除熱量，反之當溫度太低時濕度越高時，人體感知的溫度將會越敏感。

供使用者參考的天氣參數

溫度、體感溫度、紫外線、最高體感溫度、最低體感溫度、天氣現象、風速、相對溼度、降雨機率、舒適度、天氣描述

機電應用流程

1. 等待使用者選擇存放，機器會將衣物先送至攝像頭前面準備拍照，拍照之後將衣物存到內部空間。
2. 等待使用者選擇拿取，機器會指定存放空間位置，將衣物從內度取出。

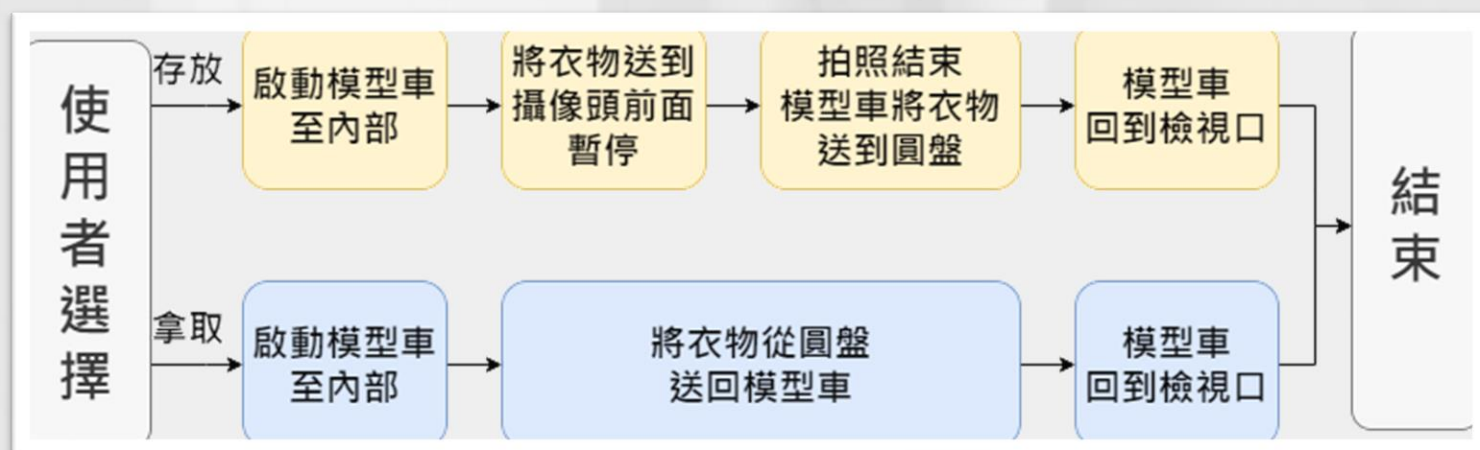


圖4 機器流程圖

機器(Arduino)與軟體(Python)

使用PySerial溝通當使用者透過HTML所渲染出來的畫面，點擊按鈕事件 (JavaScript)，傳遞請求 (Python)，當控制器單元 (Arduino) 接收成功並完成該請求時，再回傳結束訊息。

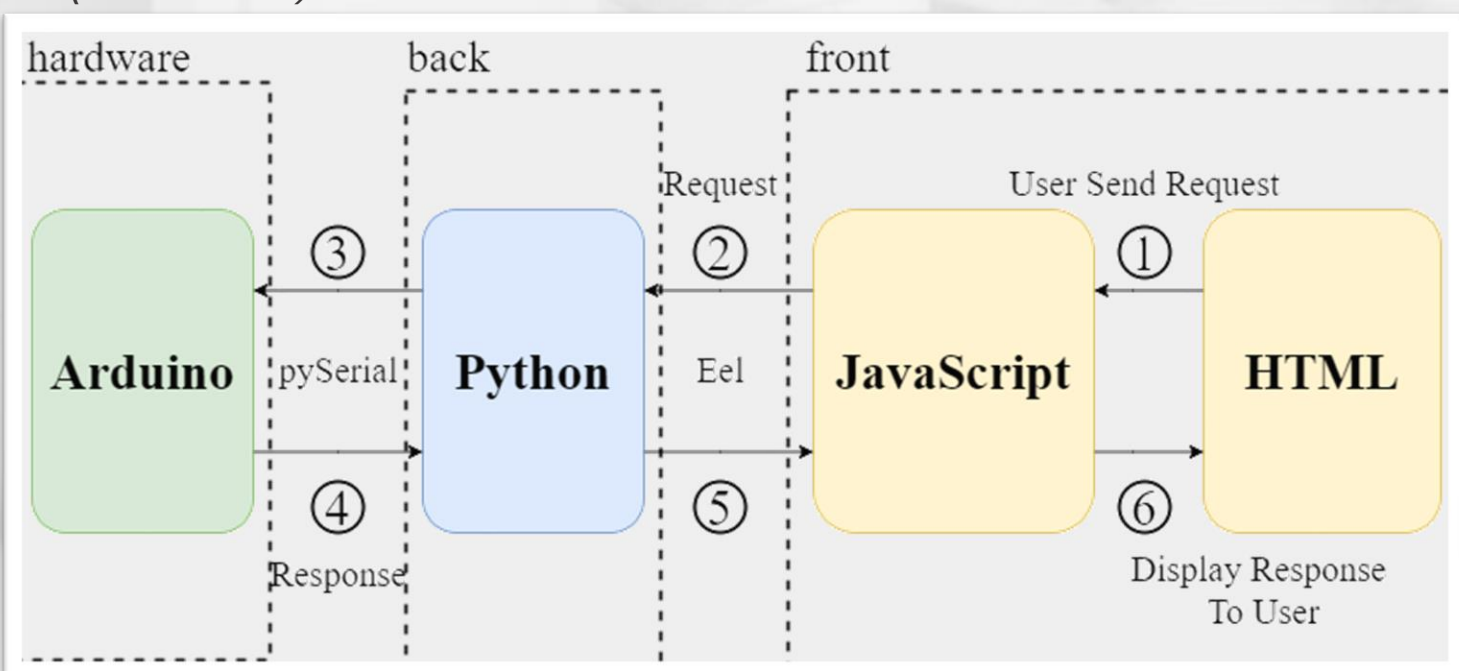


圖5 機電溝通流程圖

衣物辨識



可根據圖片辨識衣物類型

短袖、襯衫、短褲、長褲、洋裝、帽衫、外套、長袖、裙子、西裝外套及背心

黑色、灰色、白色、紅色、橙色、黃色、綠色、青色、藍色及紫色

深度學習，FastAI

ResNet18，學習預設的閾值準確值為0.5，本模型從0.2開始並測試其他閾值的結果。並創建DataLoader，需要有DataBlock來確認相同標籤 (因變量) 與不同圖像 (自變量)，使用ImageBlock有助於將圖像轉換成張量，MultiCategoryBlock將有助於進行多標籤分類。

epoch	train_loss	valid_loss	accuracy_multi	time
0	0.805188	0.412300	0.431870	04:06
1	0.403875	0.181955	0.843847	04:26
2	0.239219	0.102775	0.948567	04:00
3	0.165170	0.094429	0.957328	07:39

圖1 訓練過程

epoch	train_loss	valid_loss	accuracy_multi	time
0	0.132121	0.080975	0.966399	10:04

圖2 訓練結果