附件一

國立臺灣海洋大學資訊工程學系專題書面報告

**利用桌上型電腦與跨平台傳輸進行無人機之智慧視覺與自主飛行**

**作者**

00657010 顧賀翔

**參與人員**

00657010 顧賀翔

00657018 王伯倫

00657112 林揚昇

00657115 林奕寬

**指導教授：蔡宇軒 博士**

中華民國 109年10月12日

**目錄**

壹、摘要………………………………………………………3

貳、研究動機與目標…………………………………………3

參、背景知識介紹……………………………………………4

肆、設備應用….…………………………………………5

伍、研究方法及步驟…………………………………………7

陸、研究輔助工具…………………………………………13

柒、系統架構與設計…………………………………………14

捌、實驗結果與展示………………………………………14

玖、結論………………………………………………………19

拾、未來展望…………………………………………………19

拾壹、參考資料及網站…………………………………20

**壹、摘要**

近年來隨著資訊科技的進步，機器視覺(Machine Vision)已逐漸被廣泛應用在生活中，如：汽車自動駕駛、體溫檢疫等，使得相關產品發展迅速。本專題選擇大疆(DJI)無人機作為主體，利用Open Pose、 Web Socket以及Android等工具，實現了無人機自主飛行的功能，並透過跨平台傳輸讓使用者能輕易地透過桌上型電腦選擇自動追隨的目標。

**貳、研究動機與目標**

**(一)為何要製作此專題？**

近來，無人機的發展甚為迅速，而其應用則更加廣泛，除了應用在娛樂外、消防救災、智慧農業、物流等也都展現出無人機的價值。雖然無人機已廣泛應用在許多產業中，但仍有更多的發展與創新值得我們研發。起初，我們希望無人機能代替人們工作，減少勞力的支出以及節省時間的消耗但達成相同的效果，例如：保全巡邏、漁夫捕魚等等。為達成這項目標，影像處理與分析以及遠距離的跨平台傳輸是不可或缺的一環，因此，藉由人物辨識為初步目標，實現無人機的自主飛行。

**(二)預計達到的目標**

建置出一個Android應用程式，使其能基本的操控無人機並進行與桌上型電腦以及無人機的雙向溝通。利用Python中的Tkinter建置出GUI，其中顯示無人機回傳畫面並同步進行影像分析，另設有輸入欄位供使用者輸入指令。使用者能在家透過畫面中的人物框架選擇無人機欲追隨的目標，直到使用者選擇取消追蹤或是下一個目標。在達成無人機自主追蹤人的目標後，這樣的技術可以用在影片拍攝、鎖定犯人等等，皆有實作的機會。

**參、背景知識介紹**

（此章節內容將針對本文有提到的專有名詞進行解說）

* 大疆：全名為深圳大疆創新科技有限公司，英文簡稱DJI，中文簡稱大疆。以生產、研發民用無人飛行載具及航空攝影系統為主的科技公司。其全球市場占有率為70％，是世界上最大的民用無人機製造商。
* SDK：軟體開發套件（Software Development Kit）。一般是一些被軟體工程師用於為特定的軟體包、軟體框架、硬體平台、作業系統等建立應用軟體的開發工具的集合。
* Tkinter：一個python模組，它是一個跨平台的指令碼圖形介面介面。最大的特點是跨平台，缺點是效能不太好，執行速度慢。
* Open Pose：一套由卡內基梅隆大學感知計算實驗室打造可以讀懂人類肢體語言的程式庫。是世界上第一個實時多人關鍵點檢測和多線程的C++庫。
* Web Socket：一種網路傳輸協定，可在單個TCP連接上進行全雙工通訊。允許伺服器端主動向客戶端推播資料。瀏覽器和伺服器只需要完成一次交握，兩者之間就可以建立永續性的連接，並進行雙向資料傳輸。
* RTMP：即時訊息協定（Real-Time Messaging Protocol）。也稱即時訊息傳輸協定，是最初由Macromedia為通過網際網路在Flash播放器與一個伺服器之間傳輸串流媒體音訊、影片和資料而開發的一個專有協定。
* GUI：圖形使用者介面（Graphical User Interface）是指採用圖形方式顯示的電腦操作用戶介面。
* JavaScript：由工程師Brendan Eich所發明，是一種直譯式(Interpreted language)、物件導向(Object-based)的程式語言，主要用來撰寫網頁程式。隨著版本的演進，也推出許多像是JQuery、React.js等架構，通常會與HTML結合使用。
* JSON：JavaScript物件表示法（JavaScript Object Notation）是一種由Douglas Crockford構想和設計、輕量級的資料交換語言，該語言以易於讓人閱讀的文字為基礎，用來傳輸由屬性值或者序列性的值組成的資料物件。

**肆、設備應用**

**硬體:**

* DJI Mavic 2 pro



* samsung galaxy s9+



* Ubuntu18.04 LTS

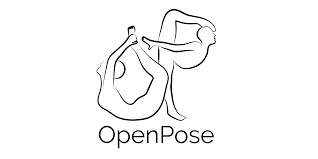


* RTX 2080



**軟體:**

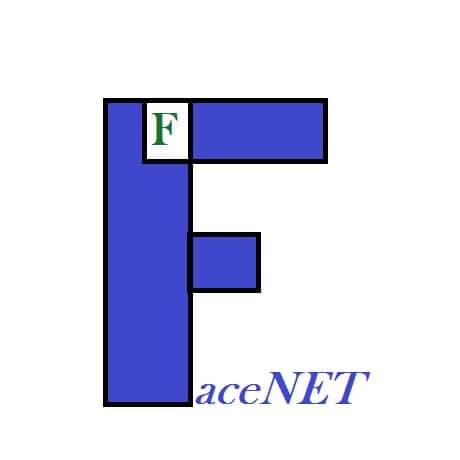
* Openpose



* SDK(Android,Google,DJI)

* FaceNet



* Socket



* Tensor flow



* NingX



**伍、研究方法及步驟**

**(一)起初的概念規劃**

在了解無人機的構造及功能後，我們想利用無人機取代目前人們付出勞力且規律的工作，想必在實現成熟後會是一個相當大的市場。首先，必須要能實現無人機自主飛行的過程，大疆原有的功能是需要手動選取範圍作為跟隨目標。為了能輕易切換目標，因此，加上人臉辨識後能顯示人物框架，便只需輸入目標的ID即可切換或取消追蹤目標。在實現後，往後的發展與應用相當豐富。

**(二)實作系統架構與方法**

有了確切的目標後，大家開始分工實作這個系統，其中主要包含了Android app設計、影像處理、跨平台資料傳輸等工作。

在App的製作上，我們參考大疆DJI GO 4的設計概念，主要使用XML以及Java實作飛機操作的介面與延伸功能。App功能包括無人機的起降落、影像顯示、影像串流、虛擬搖桿等。此部分花最多時間在研究DJI釋出的SDK，多樣複雜且環環相扣。

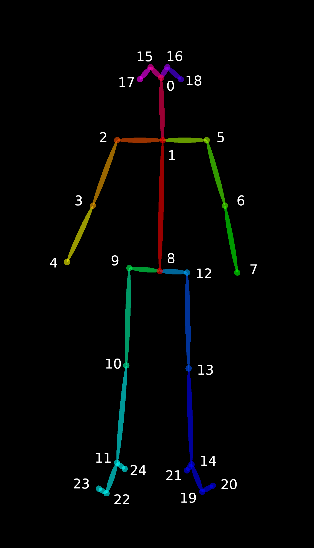
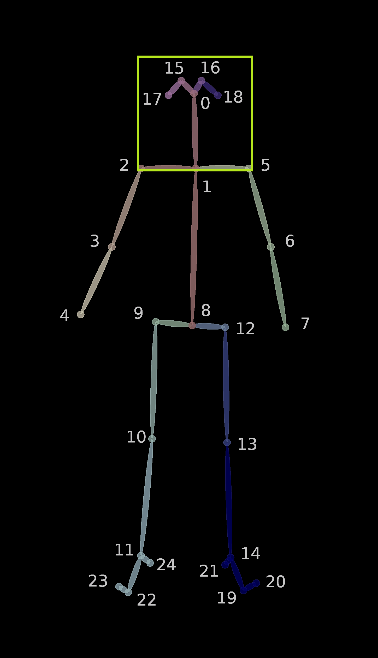
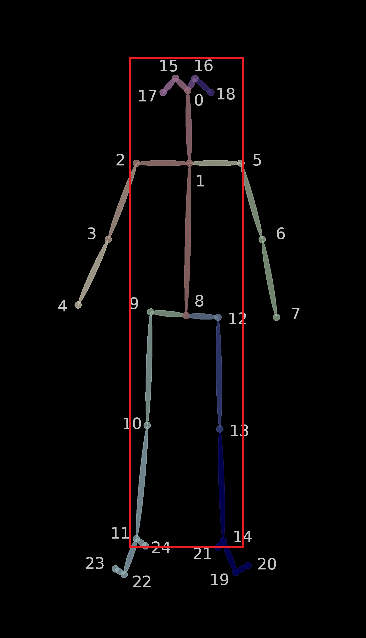
在實作無人機的Activate Track之前，我們必須先確保無人機能正常的飛行。透過Web Socket傳送指令，經由手機Java程式判斷接收到的訊息，將其轉換成控制飛機的指令。在Activate Track有誤或是無人機失控等狀況，能在最短的時間內穩定飛機，不造成危險。

DJI的SDK中的Live Stream與Activate Track的功能及其重要。Live Stream能透過Wi-Fi與RTMP進行影像串流。在App開啟此功能後，能在PC電腦接收無人機的畫面，進行後續影像處理。Activate Track則在使用者選取目標之後，能根據色塊的移動追蹤選取的目標。這兩項功能也是實現無人機自主飛行的關鍵。

為了使用者方便操作，我們在PC端使用Tkinter中的Frame、Label以及Button建置了一個使用者介面，包括影像顯示、指令輸入與資訊欄。影像中包含無人機回傳的影像、人臉辨識後的結果與人物ID而WebSocket連接狀況與指令輸入後皆會顯示在資訊欄中。

談及影像處理，此專題最重要的一環，我們利用Open Pose抓取人物的臉。首先，透過Open Pose的poseKeypoints判定在畫面中存在多少人。依據keyPoint輸出(如圖(一))判別人物的左右肩膀分別為第2點(以下簡稱2)與5點(以下簡稱5)以及臉(第15至18點)是否在畫面中。確認人物在畫面中後，以2與5的x座標判別此人物是否面向鏡頭，若2的x座標小於5則判為正面。人物面向側面時，有可能2至5的距離過小，因此在選擇人臉框架時，若2至5的距離小於鼻子(第0點(以下簡稱0))至胸前(第1點)則將兩倍鼻子至胸前的長度訂為框架寬度(border)，否則border為2至5的距離。取得框架寬度後，從胸前(第1點)往左右各半個border的距離，並往上一個border的距離即為人臉框架如下圖(二)綠色方框所示。

由於Activate Track無法只依靠人臉進行追蹤，因此我們需要找到人物身體的範圍。我們增加人物的骨盆(第8點(以下簡稱8))是否在影像中的判斷。左上點為人臉綠色方框左上點；右下點的x座標為5的x座標；y座標為0往下2倍0至8的距離，此兩點所圍的範圍即為人物身體，如圖(三)紅色方框所示。

圖(一) 圖(二) 圖(三)

我們利用WebSocket進行跨平台的資料傳輸，Server端為PC使用Python；Client端為Android使用Java。使用者選擇目標ID後，經由WebSocket將人物方框中四個角的座標傳送至App上，隨即無人機會執行Activate Track追蹤選取的人臉。WebSocket的即時性、低耗電與使用較少記憶體的特性是我們選擇其做為跨平台資料傳輸的主要原因。

整體系統而言，透過無人機拍攝影像，經由RTMP傳送至PC進行分析，再將分析結果傳回手機端進行任務飛行即實現了無人機的智慧視覺與自主飛行。

**(三)深入規劃與測試**

第一版的App與人臉辨識完成後，我們就開始陸陸續續針對許多美化、細節及效能方面的問題進行調整。

在手機操作頁面中，有大小兩個Frame分別存放地圖與無人機影像，藉由點擊小畫面能彼此切換。原本兩個Frame都接收無人機的影像，以show與hide達成大小切換的目的。經測試後發現切換後畫面會卡住不再更新，原因為無人機的圖傳影像只能提供給一個Frame，修正成delete與create後就能順利更新畫面。

每台無人機的畫面比例不同，傳至App後發現畫面扭曲，而導致後續Active Track或影像辨識選取區域與實際不符，因此改成連接時透過SDK取得該設備之無人機畫面比例再做修正，即可正常顯示人物框架的位置。

在使用虛擬搖桿操作無人機時，原始各方向飛行速度不一且操作模式複雜，因此在實際飛行上有些困難，會使飛機有亂飛的情況發生。詳細研究SDK後，發現推拉搖桿時可以設定飛行的角速度、加速度或是相對速度，因此我們在這方面修改了多次的參數設定，使得無人機能如預期的飛行。

影像經由RTMP傳輸至PC端的過程，會因為手機訊號的強弱而影響影像的畫質與傳輸速度，進而影響人臉辨識的準確性，綠色方框超出原本所預期的範圍許多。解決方法有其二，第一種為將手機移至室外，使得GPS訊號與Wi-Fi訊號通順；第二種方法為對於人物的判定更為嚴格，我們使用上圖(一)的第1點及第5點做為判別是否為正面的基準點，在測試後效果並不顯著。

在測試的過程中，因時常需要暫停修改，而每次都要將無人機暫停並飛回原點。因此我們透過大疆的SDK實作了無人機的自動返航，按下按鈕後並會反回至起飛紀錄點，這也減少了每一次測試所花的時間。在無人機起飛後，原起飛按鈕會變成降落，避免人員的操作錯誤。

Tkinter裡Label與Text中的長寬代表著能存放多少個字元，因此字元大小直接影響Label的長寬。捨棄widget.grid改用widget.place使得每個物件的長寬不因字元大小而更改，讓介面比例始終保持一致。

在python程式裡，socket未連結到client端前不會執行後續程式碼，因此GUI會顯示不出來。我們將socket連結寫成thread，連結到client端後顯示在資訊欄並結束此thread，如此GUI即可在程式執行後隨即顯示。在client端亦使用thread每1ms查看server是否發送訊息。

在Web Socket跨平台傳輸測試時，時常接收不到server傳送的資料。原因為我們在client端使用Java中Socket的ReadLine接收訊息，因此遇到換行字元才會發送一次訊息。經修改後，在每一則訊息後強制加上一個換行字元，能讓client端順利的接收訊息。

在選擇目標後，我們採用將人物身體方框的左上及右下點依比例傳送給app進行Activate Track，例如：螢幕寬度為100px，人物的左上點為(20,10)，則傳送的訊息為(0.2,0.1)。第一版測試時發生傳送至手機App的比例出現超過1的情形，研究過後發現OpenCV讀取到畫面寬高的像素為原相機的像素，因此我們原本設定Frame的像素小於相機像素，導致在相除的過程中超過1，以致無法順利執行Activate Track。我們在畫面第一次開啟時讀取相機的像素並紀錄就能順利實現Activate Track的功能。

由於無人機並非平行於人高飛行，在高處向下拍攝時，人體的骨架會被壓縮，導致在人臉辨識上border過小而無法完整取得人臉所在位置。因此我們改將border設為人物左右肩膀(2至5的距離)的1.5倍以修正人臉方框過小的情形。

**(四)系統修正與優化**

以上工作的完成後，整個系統的雛型已大致完成，針對美觀以及效能上進行修改。

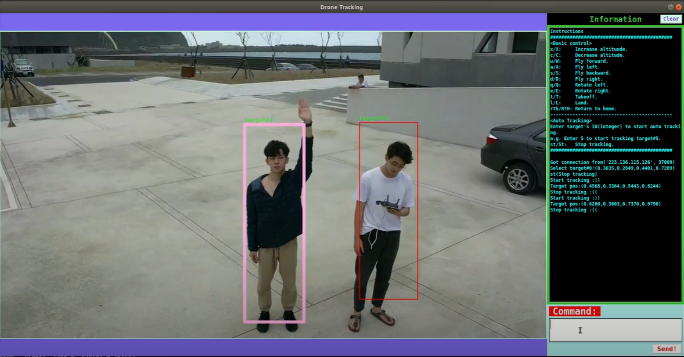
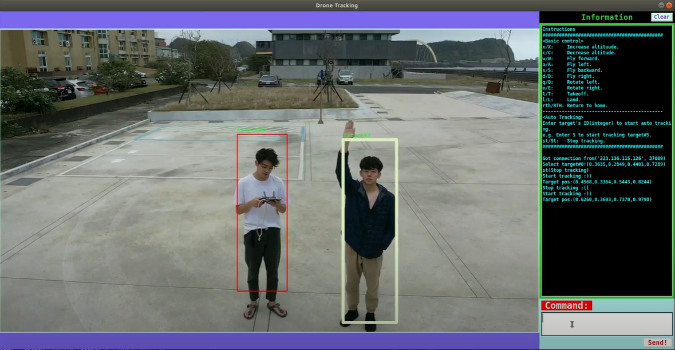
為了使用者方便操作無人機，在PC端介面右側的資訊欄顯示操作指引以及可以輸入的指令，避免操作錯誤。另外，原本只能透過按下Button才能發送訊息，我們更新為輸入指令後按下Enter鍵即可發送訊息，增加操作效率。

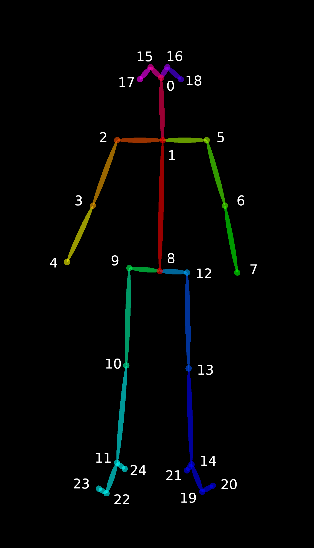
起初App畫面右側設定欄使用fragment，卻因fragment之間資料傳輸不易，並且導致activity程式量過大與複雜。因此，改用broadcast或共同的interface來傳輸，使得程式簡潔許多。

在手機App設計方面，將類似的view以相同的style來設定樣式，使得程式更簡潔易讀，在往後的更新與維護上更方便。為了符合不同手機的螢幕畫面，我們將所有物件以畫面比例顯示，使得物件以最適大小顯示。我們也善加利用thread來分攤App工作，包括無人機連線、Map讀取、與電腦端連接溝通等等，減少main(UI) thread的工作量讓程式更順暢。

一開始在處理指令時，以整數傳送目標代號並傳送座標至手機或是以英文字串控制無人機飛行，因此同時只能執行一種傳輸模式。我們將兩函式整合並在進入函式後判斷指令為整數或是字串，如此能同時進行Activate Track以及無人機的操控，程式由兩個函式縮減為一個較為整潔易懂。

在Web Socket以及RTMP訊號段開時會執行exception而關閉Socket的連線以及RTMP的VideoCapture造成Socket無法連線以及影像無法顯示。利用兩個Thread分頭執行While True迴圈，判斷訊號是否正常，若異常則不斷嘗試讀取訊號直到連接成功。修正此功能後在訊號斷開後能順利接回。

在影像辨識的過程中，若一個畫面同時存在兩個人或以上，此時我們所註記的人物編號與下一秒讀取到的編號不一致。例如：原始A註記為編號1，B註記為編號2。當A轉頭背向無人機，畫面僅剩B一人，因此B的編號會變成1。此問題會導致在進行Activate Track輸入目標編號時，與實際想追蹤的目標不符。我們原本預計解決方法為將人臉透過SVM進行訓練，並記住對應的人名，在輸入人名後即可追蹤，但後來發現人臉辨識會造成程式延遲過久而難以達成，最終我們改用openpose的骨架來判斷active track目標。(如下圖所示)，我們以舉右手為啟動姿勢，舉左手為取消姿勢，是透過左右手234、567和身體座標位置來判斷，也有確保舉雙手或拍攝角度不同時不會產生問題，而當多人同時舉手，我們會以第一舉手者為優先進行追蹤。



**(五)未解決的問題**

目前專題已完成預定的目標，但仍有一些部分是可以進行改善的，因此我們也正積極努力解決。

在command line開啟RTMP的影像時，會因網路訊號不同而造成延遲，訊號好延遲大約1秒左右，程式可以正常運行不會產生問題，但在訊號較不好時，大約會延遲3-5秒；由於桌機所顯示的畫面與實際不符，因此在傳送人物座標時，人體已不在此座標上，進而影響Activate Track容易失效。預計解決方法為提升RTMP的傳輸效能以及嘗試將影像顯示與辨識寫作兩個不同的檔案，分頭進行以利減少與實際拍攝的延遲。

還有一部分是我們有做關於人臉辨識的部分，但因openpose人臉辨識會造成延遲，且會延遲8至10秒，因此目前無法將人臉辨識應用在實際應用上。

此專題所使用的機型為DJI Mavic 2 Pro，此型號最大飛行時間約莫是30分鐘，因此在飛行上電量是一個極其重要的問題，時常在測試到一半時就顯示低電量的狀態。我們預計解決方法為嘗試修減手機App的功能，使得無人機不需接收大量的資訊，也避免傳送不必要的資訊，藉此保留無人機的電量執行自主飛行。

**陸、研究輔助工具**

「遙控無人機管理規則」於今年(2020)3月31日正式上路，需考取無人機操作證方能駕駛無人機。為了學科測驗練習方便且網路搜尋不到相關軟體，因此我們製作了「無人機學科測驗模擬試題」。主體使用HTML建置，使用JavaScript實現考題載入與答題操作並以CSS設計網頁，考題以JSON檔儲存。將使用者答題記錄存在LocalStorage方便使用者再次瀏覽題目與答案。

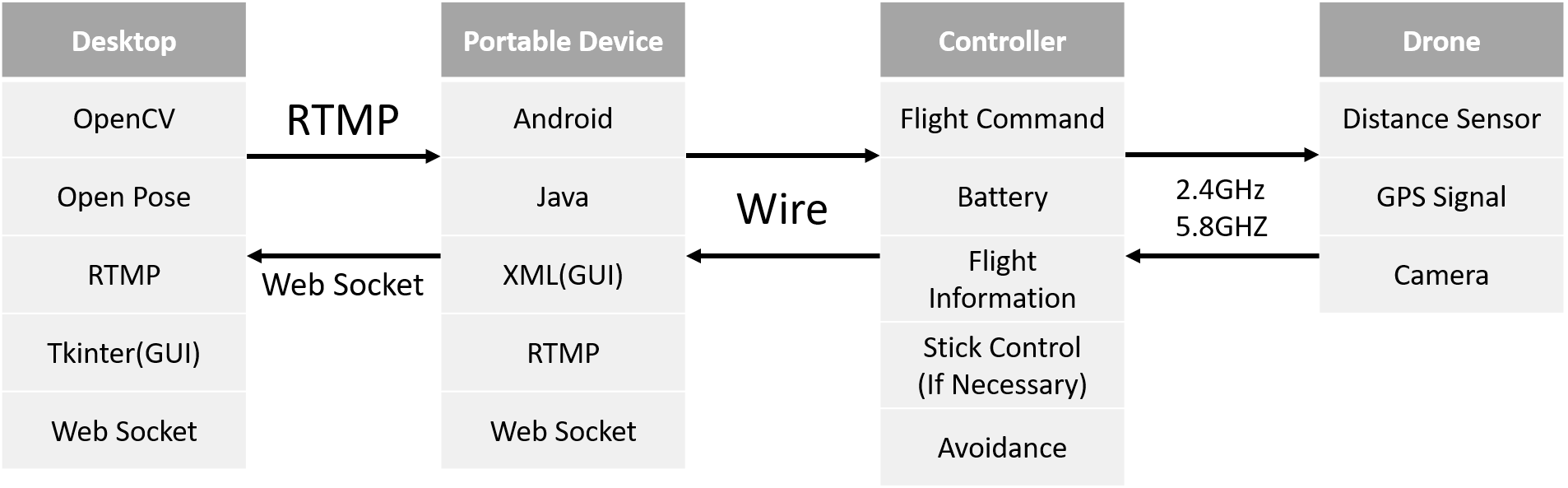
在完成第一版後，我們開始測試內容與細節。由於我們撰寫的方式為利用JavaScript生成Element、設定樣式(CSS)後再加入HTML中，因此在不同螢幕比例中的電腦圖像的大小及位置錯亂。經修正後，我們在頁面載入時讀取電腦螢幕的寬度取得像素，依設定的比例調整物件像素，如此物件比例就可以完美的固定。考慮到有多螢幕使用者，因此我們使用setInterval()，每千分之一秒更新螢幕寬度與物件。

我們在初始設定答題頁面時，將題目與正確答案紀錄在LocalStorgae中，以利往後的答題判斷與回顧。在使用者點選答案後，將使用者的答案與該題正確答案核對並顯示正確與否，也將使用者的所點選的答案加入LocalStorge。在答題結束後，使用者可以選擇回顧本次答題記錄或是回到首頁。在回到首頁後回清除所有LocalStorage的內容，正因如此答題記錄只能在答題結束後回顧並且只能瀏覽一次。在往後的修正與更新有預計加入資料庫系統，透過使用者登入即可查尋到該使用者過往的答題記錄。

題目與答案由民航局提供PDF檔，我們以手動存成JSON，並在JavaScript中以JSON Decoder將JSON轉為字串儲存。由於民航局未提供相關API因此在更新與修正題目時相當不易。

在完成成品後，經過數次的練習我們也順利通過專業級的學、術科測驗並成功取得無人機G1級別執照！

**柒、系統架構與設計**

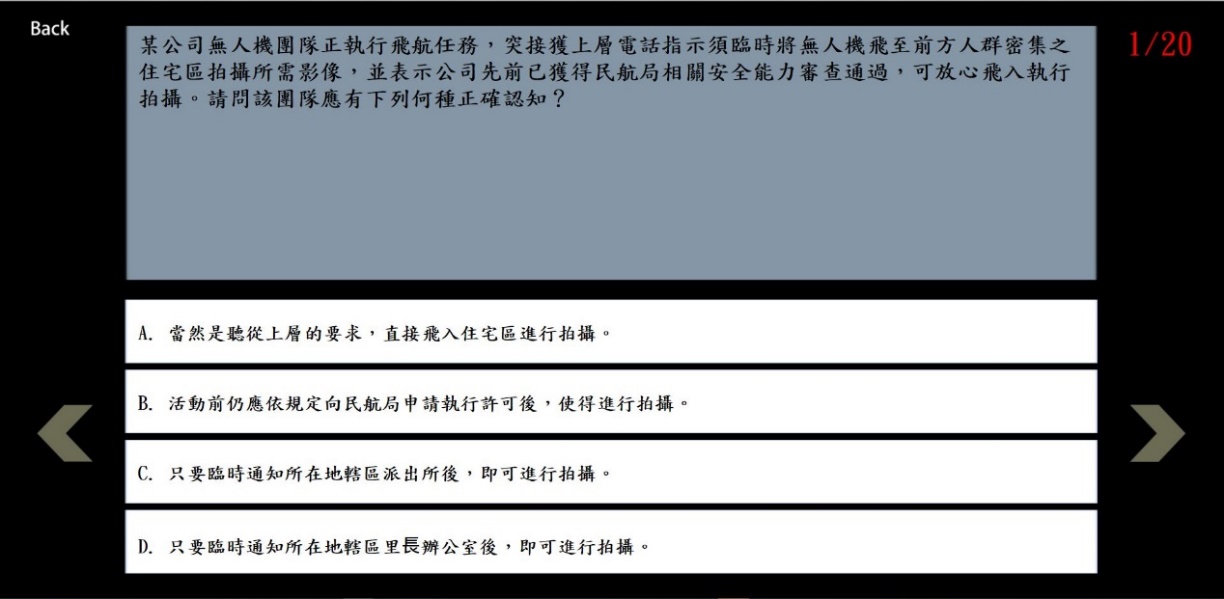


**捌、實驗結果與展示**

(一)無人機學科測驗模擬試題 - 初始頁面：可以選擇基礎級或是專業級。



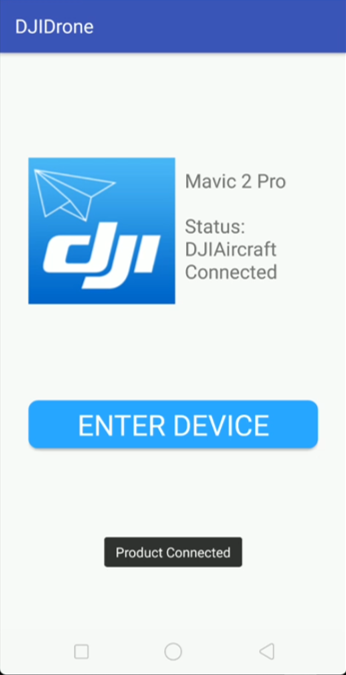
(二)無人機學科測驗模擬試題 - 答題頁面：點擊選項即可作答。



(三)無人機學科測驗模擬試題 - 結果頁面：顯示考試通過與否可以選擇回顧題目或是回首頁。



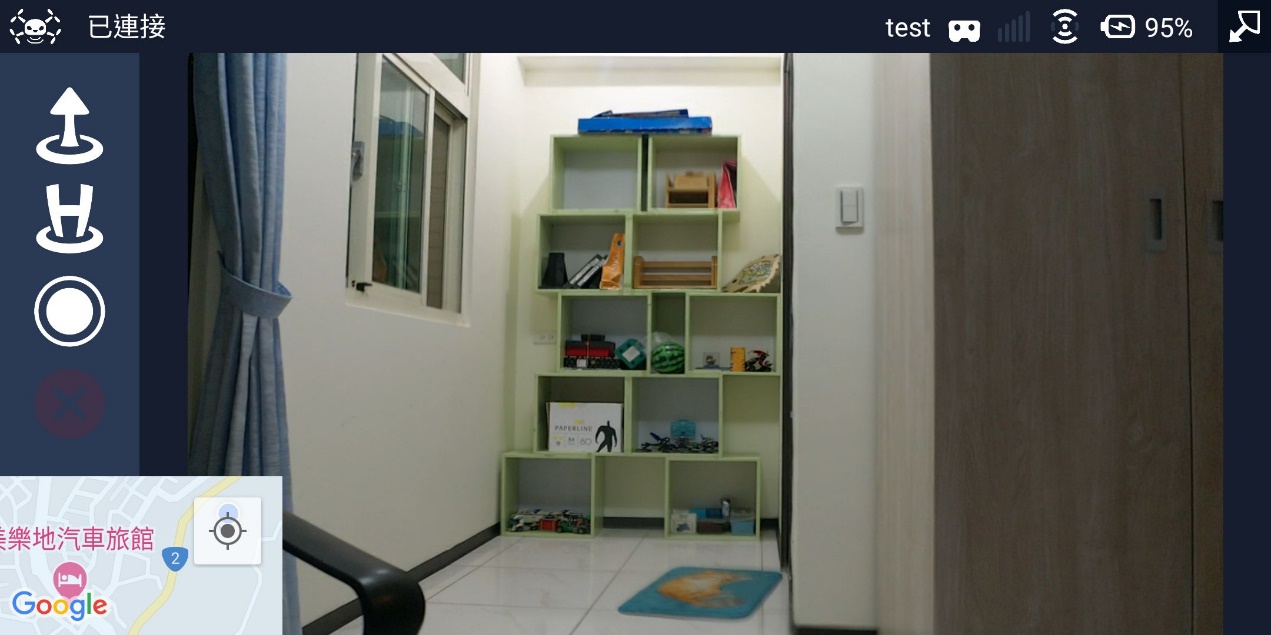
(四) Android App - 登入頁面：在連結至無人機後可以進入主畫面。

(五) Android App - 主畫面：設有無人機畫面(1)、地圖(2)、無人機設定(3)、起飛降落(4)、無人機資訊(5)以及虛擬搖桿(6)。



(六) Android App - 大螢幕：點選右上角可以隱藏或顯示設定欄以及虛擬搖桿。



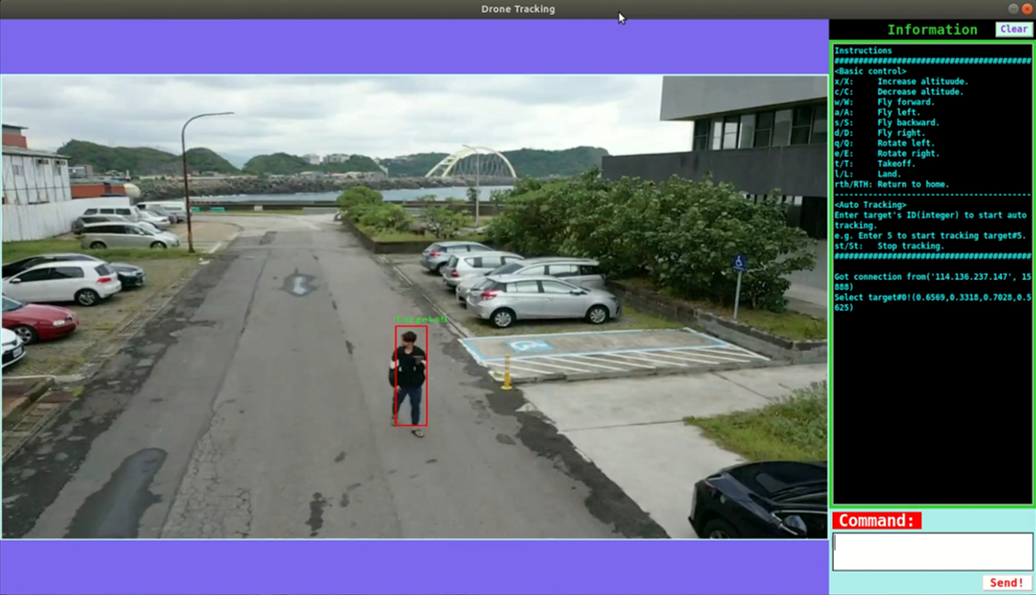
(七) Android App - Live Stream以及Web Socket：開關Live Stream以及Web Socket，開啟後可以與桌機進行溝通與傳輸。



(八) Android App - Virtual Stick以及Activate Track：開啟Virtual Stick可以使用螢幕上的搖桿操作無人機；開啟Activate Track後可以在螢幕框選目標。



(九) Desktop GUI：左側顯示無人機影像、人物框架以及目標號碼；右側上方為指令資訊，可於右下方輸入指令。



(十) 無人機實體：透過App接收指令進行基本操作以及自主飛行。



**玖、結論**

在深入無人機的功能與應用後，從無人機的操作到影像分析甚至自主飛行，每一項工作都是要動腦筋及下功夫的。

在最初了解大疆的SDK時，我們閱讀了大量的文件卻不知道應該實作在哪個部分與如何使用。因此，我們著手將一個個功能加入App實測並了解此功能飛行模式與限制，加速了我們開發手機App的效率。時常因SDK的限制導致兩個功能無法同時並用，以及很多其他的細節都是在我們在測試之後才發現的。在未來App更新與延伸應用或是SDK的變動都是會碰到的挑戰。

我們參考許多Open Pose的作品以及應用以取得最適合無人機所需的人物框架，依據無人機拍攝角度、高度以及水平距離不同改變判斷框架的參數，在經過無數次的測試以及調整後達到我們所預期的效果。在測試階段，時常受到天氣、光線以及風速等因素影響，於室內測試時GPS訊號會發生失真的情形，因此我們非常把握每一次難得陽光露頭的日子並記錄每一次的參數設定以利往後的修正。

在最後整合的部分花費不少時間，組員之間需要了解彼此平台的運作方式或限制。在串聯時因不同的限制而更改不同傳輸方式或功能啟用，在不斷的測試後才能建置出一套完整的系統。

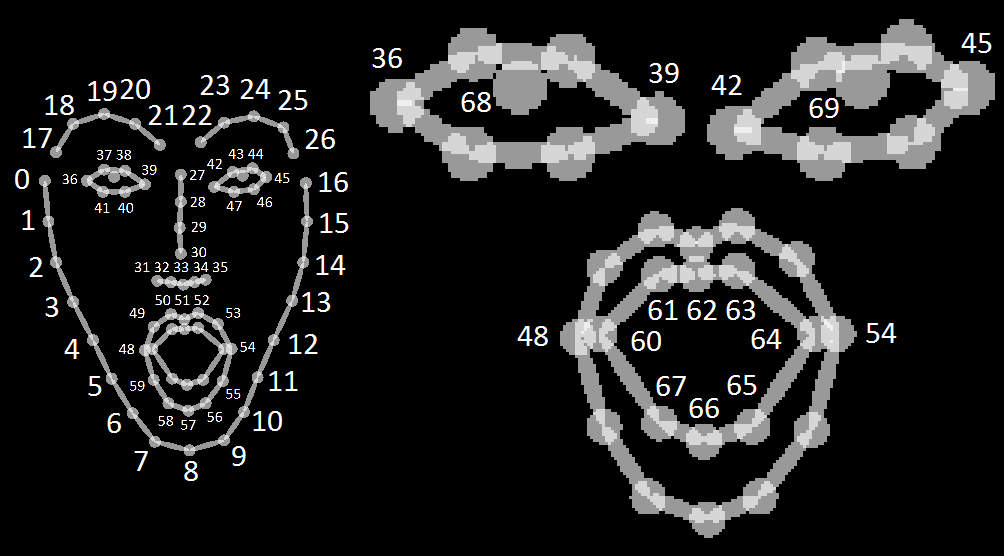
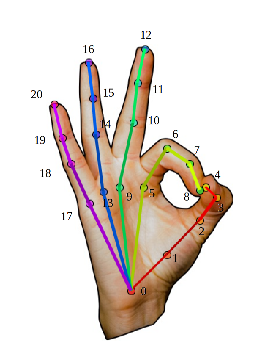
**拾、未來展望**

在實現了追蹤人物之後，我們展望未來能不使用Activate Track而是透過影像分析追隨任何目標像是野生動物觀察、巡邏查緝車輛等等以及其他任務飛行模式。以影像位置、距離等等判斷條件直接操作無人機飛行的方向。

我們也期許能實作我們起初對於無人機自主飛行的概念規劃，包括能代替警衛替夜間巡邏、結合超音波感測取代漁夫捕魚甚至自主拍攝影片等等功能。想必在智慧視覺、機器學習、影像處理與分析等技術發展成熟後，在無人機的應用上則更加廣泛，也會是無窮的商機。

除了無人機的發展外，目前也已將智慧視覺、機器學習等技術運用在智慧家電、交通工具甚至是醫療照護，實際運用像是大金智慧眼變頻空調、台北市無人公車以及照護機器人等等。智慧視覺會是未來社會中不可或缺的一環，能使得人們的生活更便利，也能省下更多的時間。

Open Pose的廣泛應用也相當值得一提。Open Pose除了能尋找身體架構以外，更可以辨識出人物臉部及手部的細節，如下圖(四)、圖(五)所見。如此，可以利用細微的表情、手勢下達指令，例如比出數字調整冷氣溫度、招手後計程車自動靠邊停車或是利用練部表情解讀唇語等等，使得生活更加的方便。

圖(四) 圖(五)

**拾壹、參考資料及網站**

* Android App
  + https://developer.android.com/docs
  + https://developer.dji.com/mobile-sdk/documentation/introduction/index.htm
  + https://developer.dji.com/api-reference/android-api/Components/SDKManager/DJISDKManager.html
* Open Pose
  + https://github.com/CMU-Perceptual-Computing-Lab/openpose
  + https://blog.techbridge.cc/2019/01/18/openpose-installation/
* Tkinter
  + https://blog.techbridge.cc/2019/09/21/how-to-use-python-tkinter-to-make-gui-app-tutorial/
  + http://effbot.org/tkinterbook/
  + https://www.tutorialspoint.com/python/index.htm
* Web Socket
  + https://blog.maxkit.com.tw/2019/01/python-websocket-server.html
  + https://websockets.readthedocs.io/en/stable/intro.html
  + https://docs.oracle.com/javase/tutorial/networking/sockets/readingWriting.html
  + https://docs.oracle.com/javase/7/docs/api/java/net/Socket.html
* 無人機模擬試題
  + https://www.w3schools.com/html/
  + https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/