

中華民國第 59 屆中小學科學展覽會 作品說明書

高級中等學校組 工程學(一)科

052302

自動化影像辨識機械手臂

學校名稱：桃園市私立治平高級中學

作者： 高三 蔡定維 高三 歐有智 高三 鄭淳升	指導老師： 魏慶龍 張仁安
-----------------------------------------------	-----------------------------

關鍵詞：機械手臂、樹莓派、影像辨識

壹、摘要

這次專題以自動化工廠中的機械手臂為題材，並主要結合影像辨識及物聯網雲端存取資料庫系統，來製作我們的專題。影像辨識主要的工作是辨別物品是否有瑕疵，再將辨識後的結果傳至機械手臂及雲端資料庫，而機械手臂會根據影像判斷後的結果進行夾取及分類，來模擬工廠內的情形。

貳、研究動機

在這個時代，科技不斷的進步、機械技術不斷的提升之中，製造工廠內不再像以往一樣人數很多，反而大部分的工作都交給機器人來處理和製造，不用給它休息、上下班的時間，只需要給它電力、定期的維護和維修及遠端監控來確認目前情形就可以長時間、長期的工作。

這樣一來不僅能夠減少很多人力成本方面的花費，並減少人員暴露在危險的工作環境中的次數和機率，更可以增加製作過程中的精確度以及提升工作和製造的效率，進而提升產品的產量。

由於自動化花費成本少，換來的效益也大，所以這個情況在現在的工廠中也越來越普遍，而這就是所謂的工廠自動化。因此我們想研究與模擬適用於自動化工廠的產線中識別產品好壞的機械手臂。

參、研究方法

一、研究設備及材料

(一)、Raspberry pi3 model B+ (樹莓派)



圖. Raspberry pi3 model B+

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data

(二)、Raspberry pi camera V2 (攝影機)



圖. Raspberry pi camera V2

Image Sensor :	Sony IMX219 image sensor in a fixed-focus module with integral IR filter
Resolution :	800 萬像素
Still picture resolution :	3280 × 2464
Max image transfer rate :	1080p30; 720p60; 640x480p90
Interface :	15 Pin MIPI Camera Serial Interface (CSI-2)
Lens size :	1/4
Dimensions :	23 x 25 x 9mm

(三)、Arduino NANO

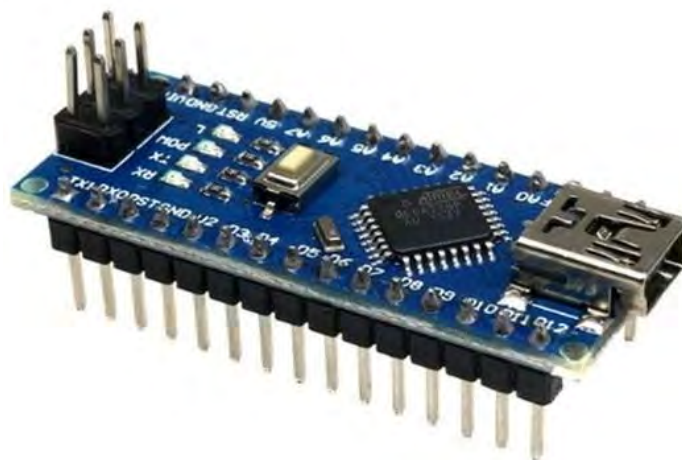


圖. Arduino NANO

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5V
Flash Memory	32KB of which 2KB used by bootloader

SRAM	2KB
Clock Speed	16MHz
Analog IN Pins	8
EEPROM	1KB
DC Current per I/O Piins	40 mA(I/O Pins)
Input Voltage	7-12V
Digital I/O Pins	22(6 of which are PWM)
PWM Output	6
Power Consumption	19 mA

(四)、伺服馬達(MG995)



圖. 伺服馬達(MG995)

- Weight: 55g

- Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6v)
- Operating speed: 0.20sec/60degree (4.8v); 0.16sec/60degree (6.0v)
- Operating voltage: 4.8~ 6.6v
- Temperature range: 0- 55deg
- Dead band width: 1us
- 轉動幅度: 180 度

(五)、降壓模組

- 輸入電壓：5-24V
- 輸出電壓：6V

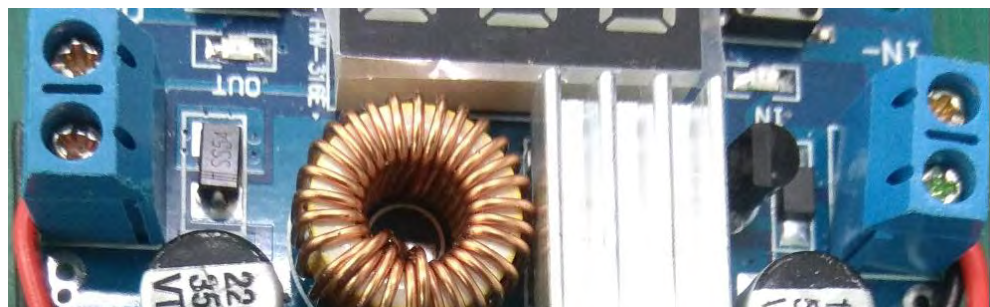


圖. 降壓模組

(六)、超音波感測器(HC-04)

偵測產品是否在機械手臂前



圖. 超音波感測器

二、研究過程

(一)、組裝機械手臂

首先買機械手臂的支架，支架是給 MG995 或是 MG996 的伺服馬達使用，而我們是使用 MG995 來進行組裝。在組裝機械手臂都有考量到重量及各顆伺服馬達會不會沒有力量去推動該關節及考量未來在做整體動時所需要動作位置規劃。組裝完後再利用 Arduino NANO 搭配可變電阻來手動改變 PWM 輸出控制各顆伺服馬達的角度，以測試馬達及整體結構是否有問題。

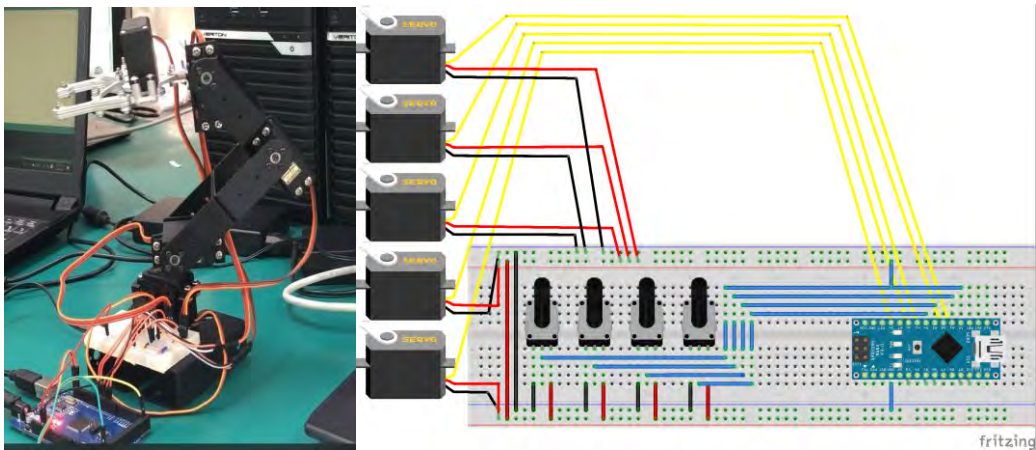


圖. 機械手臂(左) 手臂測試電路(右)

(二)、機械手臂電源電路

原本機械手臂馬達的線都是接在麵包版上，電源是使用 NANO 板上的電源，但是這樣每次都要重接，有時候手臂在動作時線會被扯掉，所以我們設計了一個電源和控制線合在一起的電路，好處是體積比麵包板小，且每條從 NANO 板上出來接伺服馬達的訊號線不容易被干擾。再利用雷射切割機製作的壓克力盒子，並將電路放置在其中，防止觸碰到電路且能小幅度的達到整理線路的效果。

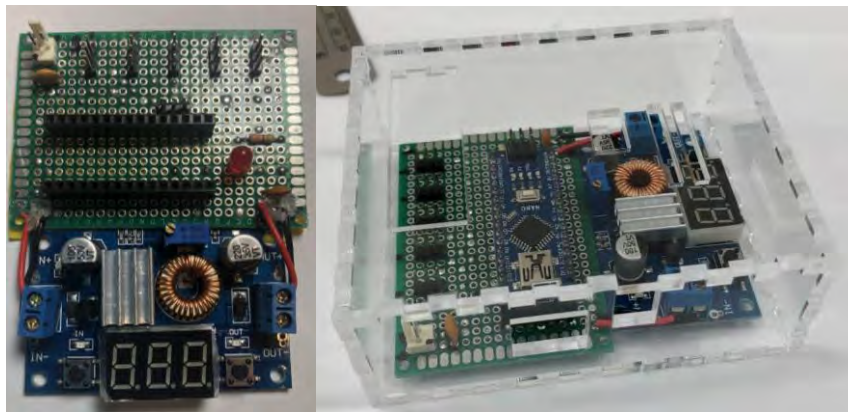


圖. 電路及盒子

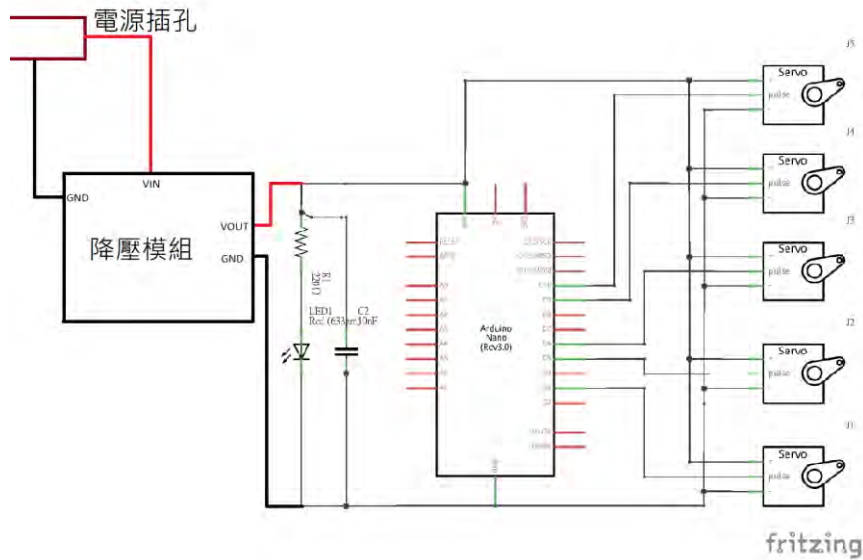


圖. 機械手臂控制電路圖

(三)、自製輸送帶

由於市面上販賣的輸送帶普遍價格非常貴，加上外面所販賣的 大小都非常大，且重量都非常重，我們就想說自己來做一個輸送帶，剛好學校有雷射切割機及 3D 列印機，藉此機會來學習如何使用。

首先我們參考外面輸送帶的外型及運作動作，之後利用 SolidWork 來設計輸送帶的外觀，將外殼部分轉成 dxf 檔給雷射切割機，兩遍的滾軸轉成 STL 檔給 3D 列印機。

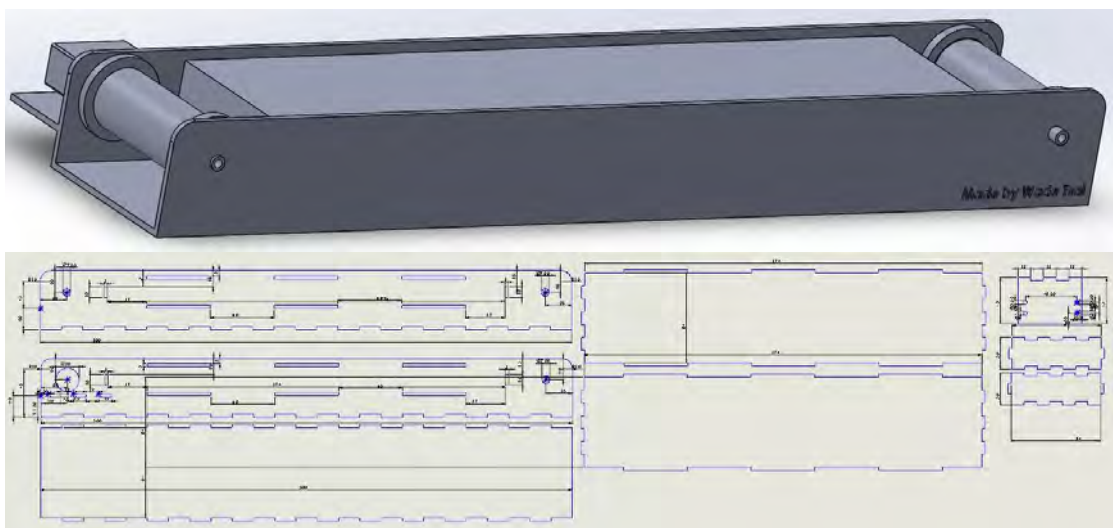


圖. 輸送帶設計圖

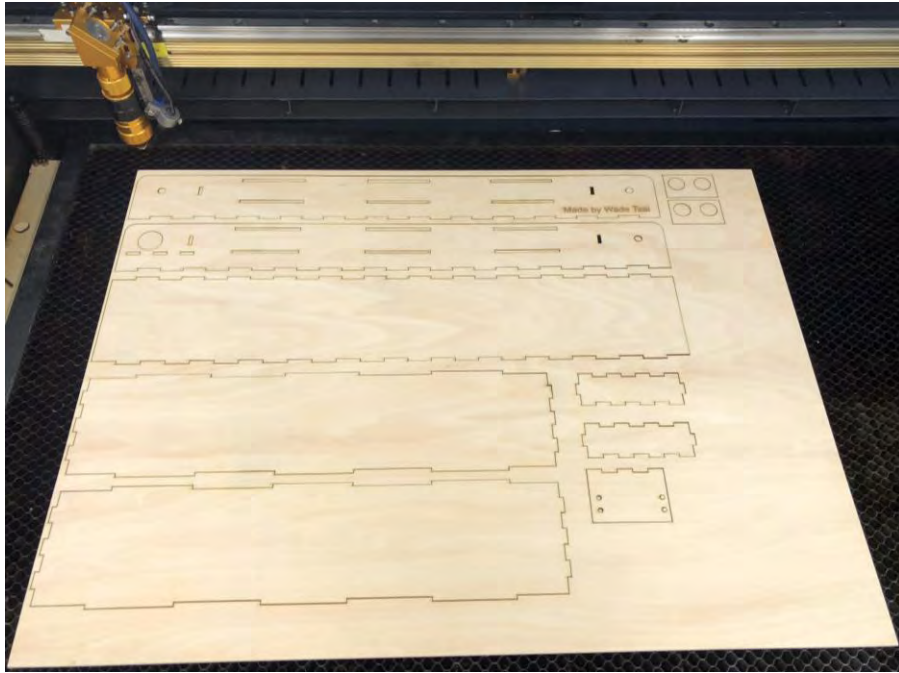


圖. 雷射切割(輸送帶)

都印好後使用保麗龍膠固定外框，再將 MG995 伺服馬達裝上。皮帶是使用一般外面家具行所賣的墊子，我們找到的墊子非常軟，買好後就先量輸送帶的長度，量完就開始裁切墊子，再來我們就裝到輸送帶上，轉動也很正常。整體大小為 500mmX90mmX50mm、重量也不會太重、成本也低，且很好操控。

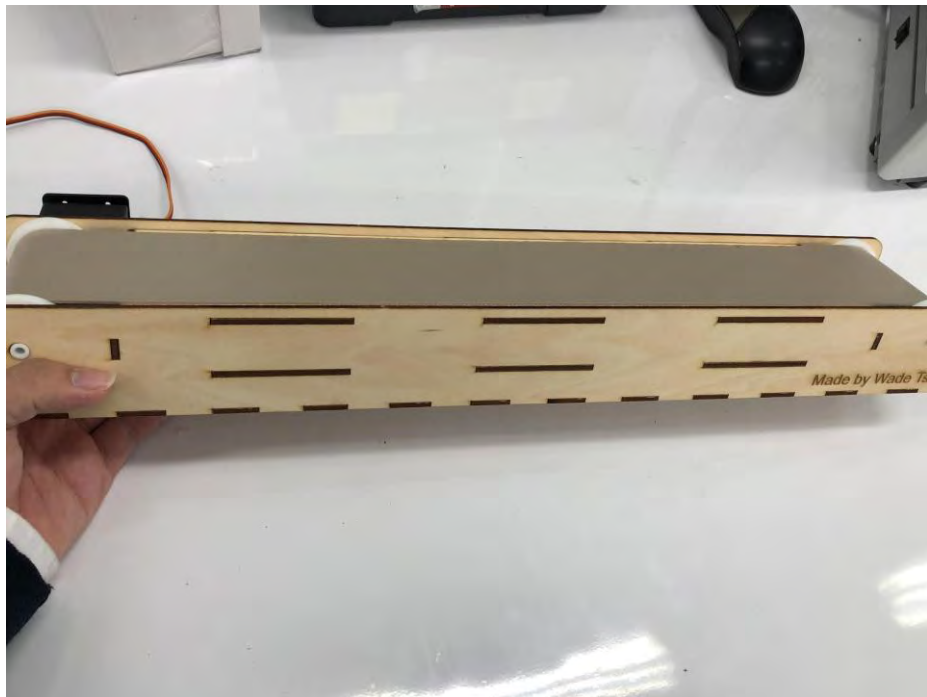


圖. 自製輸送帶

(四)、機械手臂的控制

機器手臂主要是由 Raspberry pi 發送特定指令給機器手臂，其中間溝通是使用 USB (Universal Serial Bus) 傳送字串，Arduino NANO 接收到指令後會進行判斷，並執行特定的動作。機械手臂程式內分為手動控制及自動控制，起始預設為手動模式，之後可以依照使用者進行切換。

如果接受到手動指令，手臂會先回到起始位置等待後續指令。樹莓派所送出的資料結構為(“A 度數 B 度數 C 度數 D 度數”)的字串，NANO 接收到資料會利用 index() 將 ABCD 各個度數取出給伺服馬達。除了可以控制外，還可以將各位子的度數存下形成一個連續動作。

若是接收到自動指令的話接收到壞掉物品的指令就會夾起物品並轉動馬達把東西放到壞掉物品的區域。為了要讓手臂的動作看得更加順眼，要讓每一顆馬達能同時工作，這樣也能夠節省動作的時間。

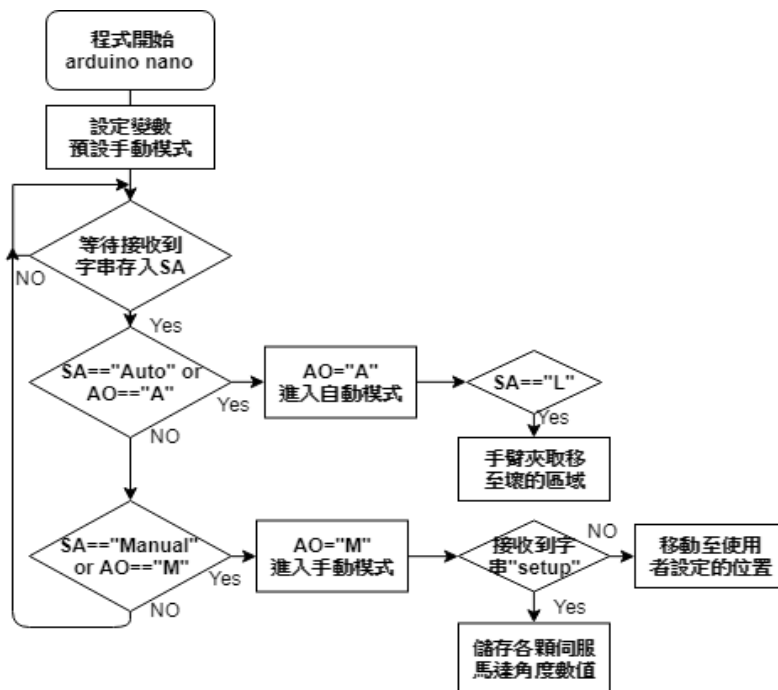


圖. 機械手臂程式流程圖

(五)、Firebase 雲端存取

為了能隨時監控工廠內辨識的狀況，需要用到雲端存取。我們就使用 Firebase 雲端開發平台中的資料庫(Realtime Database)來存取資料，只要使用手機或是其他地方的電腦上 Firebase 網站登入有權限的帳號或者擁有金鑰文件程式打開資料庫，之後我們使用 App Inventor 來至作可連結到 Firebase 的系統，即可查看資料庫中的資料，可保護好資料不隨便外流。

當樹莓派辨識結果為好的物品時，會增加資料庫中 OK 項目的資料數據，若是結果為壞的物品時，會增加 fail 項目的資料數據，如果按下監控視窗中的重置伺服按鈕，則會把資料庫中的資料清除。

wade-124678

[-] "object_detect_data"

OK: 5

fail: 2

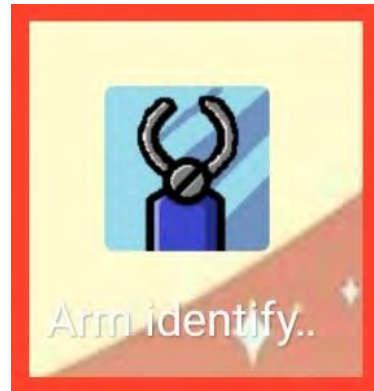


圖. 資料庫內部界面(左) APP 圖標

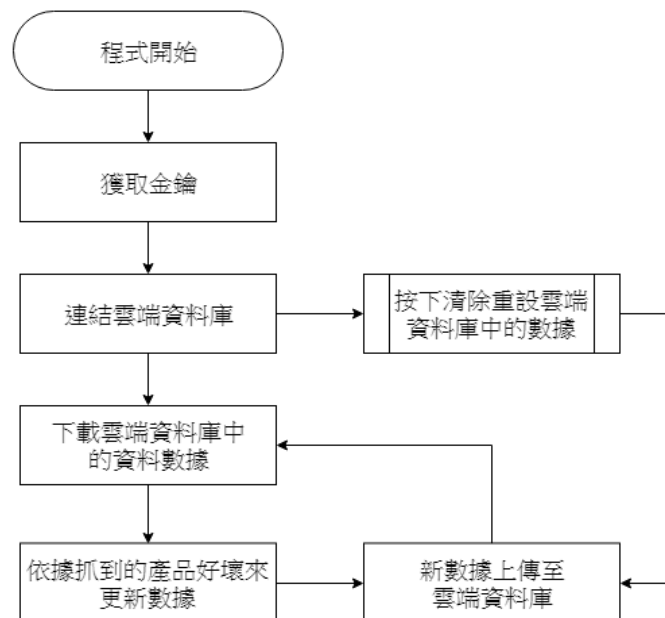


圖. 資料庫流程圖

(六)、建立 Ponyth GUI

為了方便控制機械手臂，我們在 Raspberry pi 上建立一個 GUI 界面，界面上的控件都是自己找教學定義的，GUI 界面背後的程式是連接到 Arduino NANO 傳送字串讓它去做手臂的控制。

能點選手動或自動按鈕來切換手臂狀態，控制馬達的控件由輸入框或拖動條來調整馬達角度，開關夾子的按鈕能控制夾子的開或關，重設按鈕能重置各馬達的角度，重置伺服按鈕用來重設雲端資料庫中的資料。

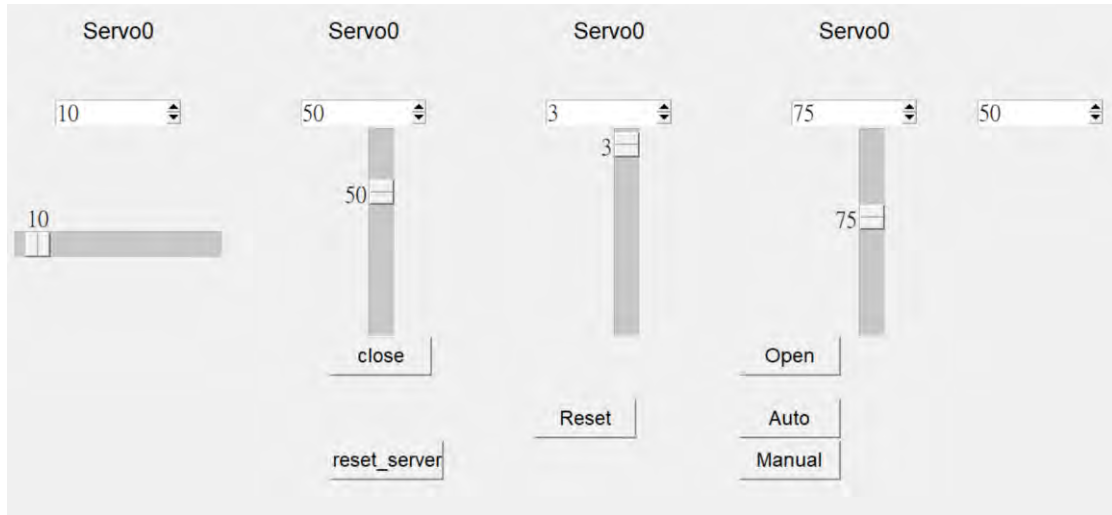


圖. GUI 界面

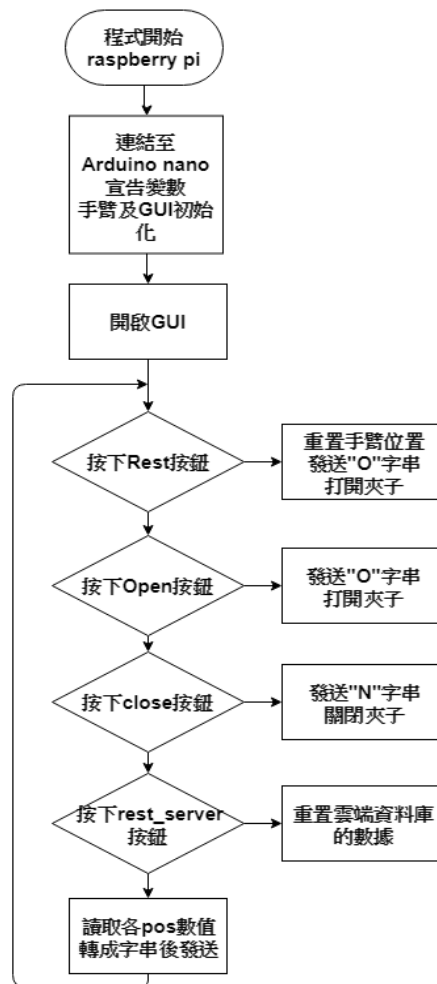


圖. GUI 測試版程式

(七)、影像辨識

現在大家常用的影像辨識程式內都有 OpenCV。OpenCV 是由 Intel 公司發起並參與開發，它是一個跨平台的電腦視覺庫，可用於圖像處理、電腦視覺以及圖形辨識程式。

我們是以 match Template 的演算法進行影像比對，這個演算法需要先給樣本圖片，之後再加上要被比對的圖，比對是依照設定的比對率進行比對。



圖. OpenCV 標誌

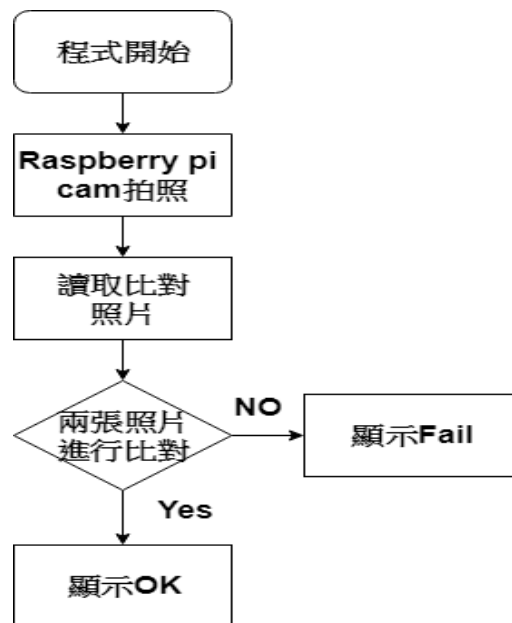
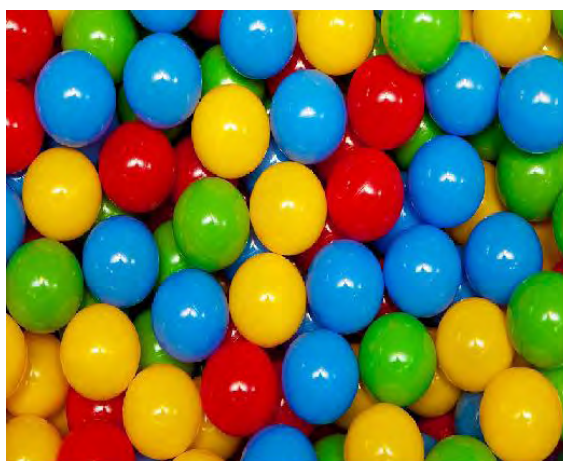


圖. 流程圖



被比對



樣本

圖. 影像辨識比對 1

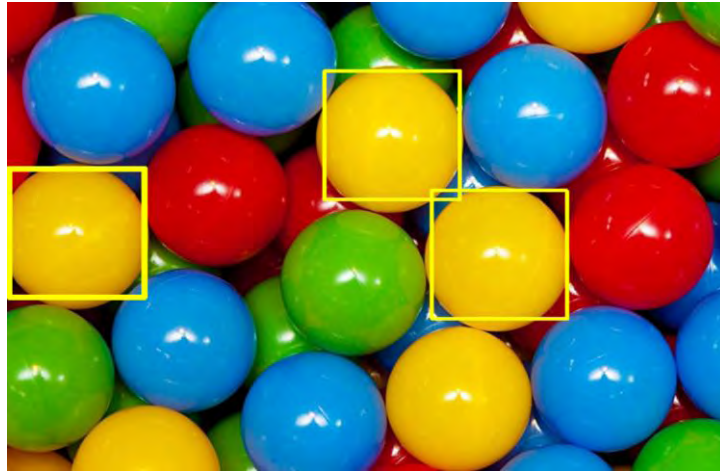


圖. 影像辨識比對 2

肆、研究結果

一、作品功能

(一)、可辨識物品好壞

當物品放置在輸送帶上，並經過機械手臂旁的超音波感測器時，輸送帶會暫停運轉且樹莓派攝影機會進行拍照及回傳至樹莓派上進行比對，並將比對結果顯示螢幕上。



圖. 預設樣本照片



圖. 比對樣本後正確
(黃框為相似處)



圖. 比對樣本後無相似處

(二)、將物品夾取後並分類

比對完後樹莓派會將結果訊號傳至機械手臂，而手臂會依據接收到的信號資料去進行夾取及分類。

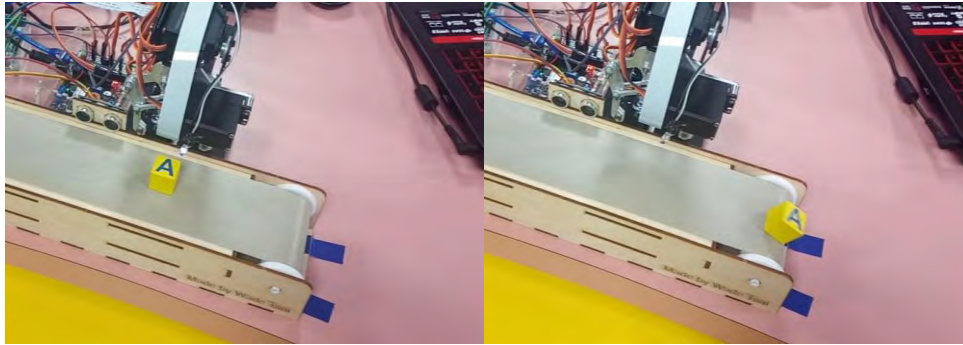


圖. 手臂分類(好)

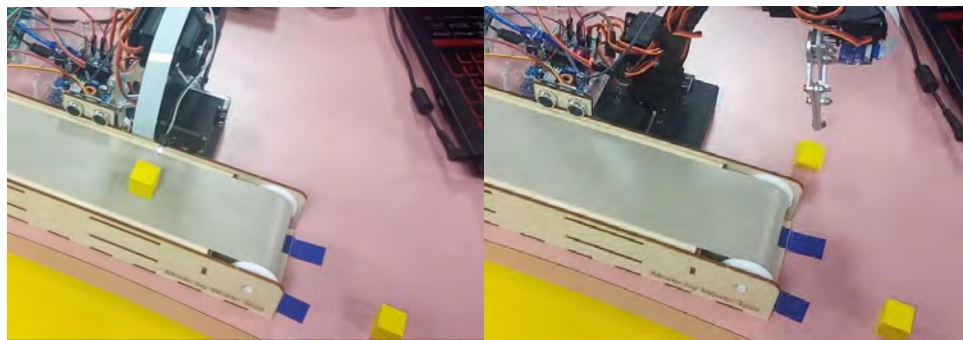


圖. 手臂分類(壞)

(三)、辨識結果能上傳至資料庫網站及手機 APP 上

樹莓派將辨識結果傳到手臂外，也會透過網路上傳至雲端資料庫更新資料庫內的資料，使用手機打開 APP 或者直接使用有權限的帳號進入資料庫網站即可得知目前累計的好壞物品數量。



OK:0

Fail:0



圖. APP 畫面

(四)、機械手臂能夠手動及自動操作

操作介面上有控制按鈕，點選自動或手動就能切換手臂的狀態，點選自動除非切換回手動，不然無法做其他手動操作，能防止誤觸導致馬達旋轉角度出錯。

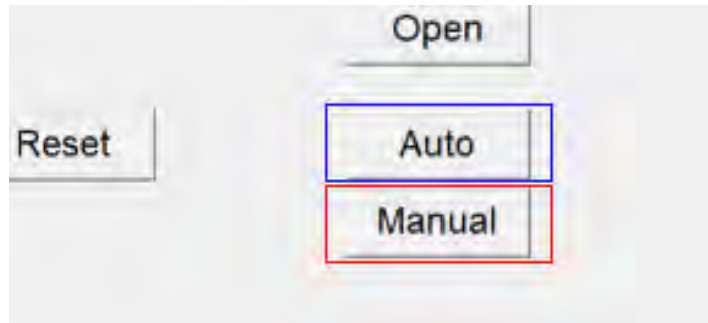


圖. GUI 介面中的手動(紅框)及自動(藍框)按鈕

伍、討論

一、操控裝置

當初原本想將機械手臂從 Arduino NANO 移至樹梅派上，但因為樹梅派只有三支 PWM 輸出，再加上以後如果要延伸或是多加機械手臂，所以經我們討論的結果，就先維持以 Arduino NANO 操控。

二、電壓太高

其中我們新增電源電路來提高驅動手臂的電壓，一開始我們將電壓調在 7 伏特，雖然低於標示最高電壓 7.2 伏特限制，但因為長時間在限制電壓的邊緣的情況下運作，及加上剛接上電源時會有突波 (Spike) 瞬間高電壓衝進降壓模組，導致轉出的電壓高於 7.2 伏特，使馬達壽命減少，甚至燒毀，所以我們將降壓模組的輸出電壓下調到 6 伏特。

三、辨識結果的正確率

在進行影像辨識時，常常因為周圍環境的光線，導致辨識結果會因為地點、時間等等而時好時壞，所以我們討論後在攝影機旁加裝了燈來減少環境影響增加了辨識結果的正確率。

陸、結論

整個做下來，學到了許多事物也發現了一些問題，例如像是在我們用攝影機進行影像辨識的時候常因為辨識問題時好時壞很容易出錯，原因是鏡頭拍出來的畫面太模糊之後才發現攝像機有個小旋鈕能夠調焦距才把問題解決，這樣可以改進辨識度的成功率，都是我們在做的過程中所遇到的問題和一些想法。

經過這次做專題的經驗，我們從中獲取了許多的知識，也遇到了許多問題並加以解決，而當每一次實驗成功或者問題解決時那種感覺真的

很開心。

柒、參考資料與其他

Jeff (2016 年 11 月 3 日)。添加線程 Thread。取自
<https://morvanzhou.github.io/tutorials/python-basic/threading/2-add-thread/>

Resnick Chang (2018 年 4 月 12 日)。建立專屬資料庫。取自
<https://wpc.page.link/c9f9de6728b6>

Resnick Chang (2018 年 4 月 18 日)。資料庫操作 — 新增資料。取自
<https://wpc.page.link/b96db385e1e4>

Resnick Chang (2018 年 4 月 24 日)。資料庫操作 — 讀取資料。取自
<https://wpc.page.link/608f49645f92>

自動化工廠參考文件

<https://ejournal.stpi.narl.org.tw/sd/download?source=10212-09.pdf&v1Id=4D3FBF97-F5E0-4EC8-AE3D-4BFB891C0A1D&nd=1&ds=1>

rasberry pi3 opencv3 python install
<https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/>

Python 多執行緒 threading 模組平行化程式設計教學
<https://blog.gtwang.org/programming/python-threading-multithreaded-programming-tutorial/>

OpenCV-Python 教程

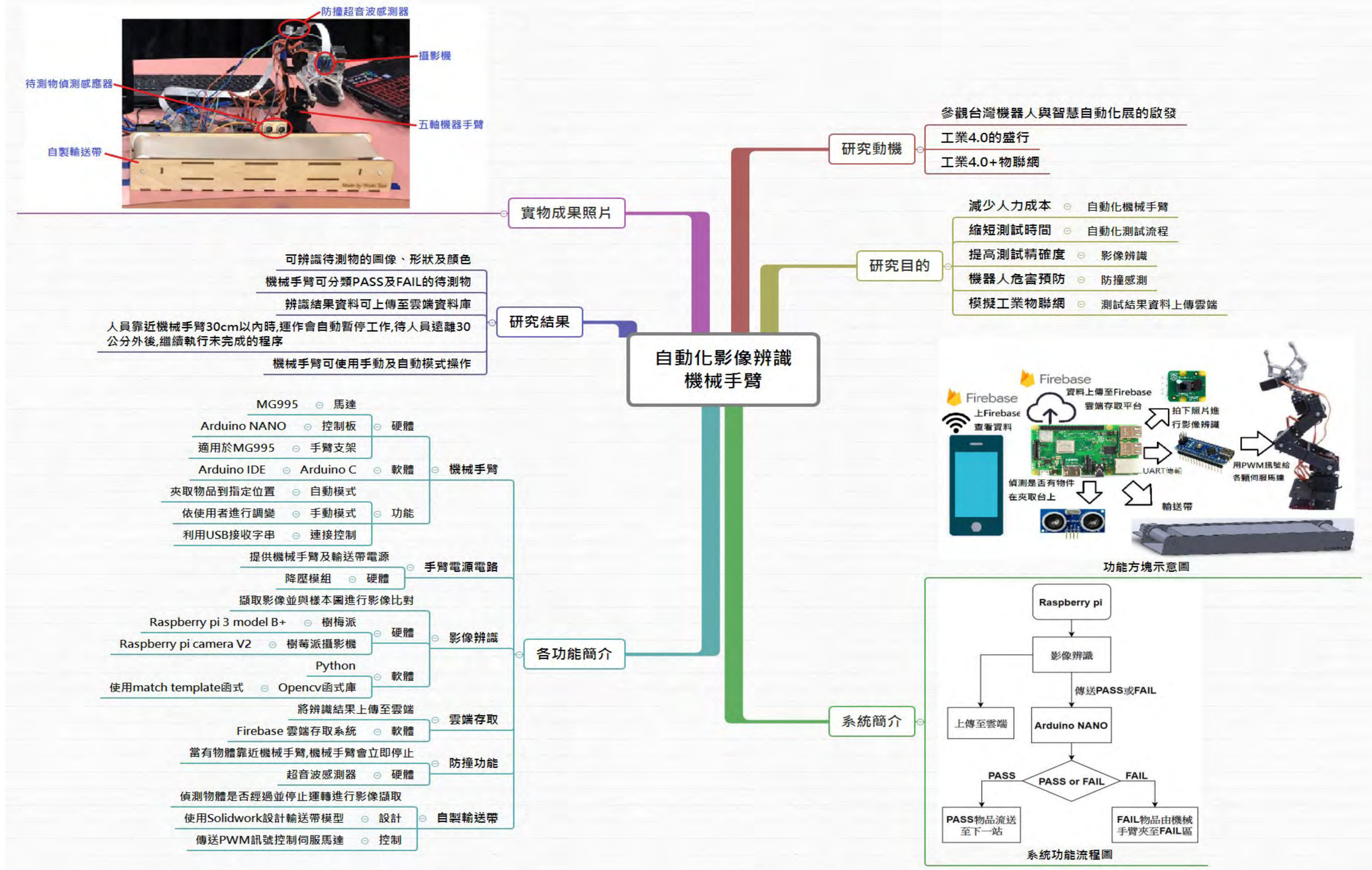
https://docs.opencv.org/3.4.1/d6/d00/tutorial_py_root.html

Raspberry gPIo

<https://learn.sparkfun.com/tutorials/raspberry-gpio/all>

【評語】 052302

本作品以樹莓派為中心，透過影像辨識辨識物品好壞，以決定驅動機械手臂來夾取分類，並整合物聯網技術，將量測取得之影像進行初步辨識影像中的英文字母，並將物件辨識資料上傳建立資料。各項科技技術整合性高且完整，作者運用軟硬體整合，製作自動化影像辨識機器手臂，實屬難得，值得鼓勵。對於所使用的關鍵技術，以及整合上述技術的工程創意與創新，以及達成的量化效益，建議可以進一步說明。



摘要

這次專題以自動化工廠中的機械手臂為題材，並主要結合影像辨識及物聯網雲端存取資料庫系統，來製作我們的專題。影像辨識主要的工作是辨別物品是否有瑕疵，再將辨識後的結果傳至機械手臂及雲端資料庫，而機械手臂會根據影像判斷後的結果進行夾取及分類，來模擬工廠內的情形。

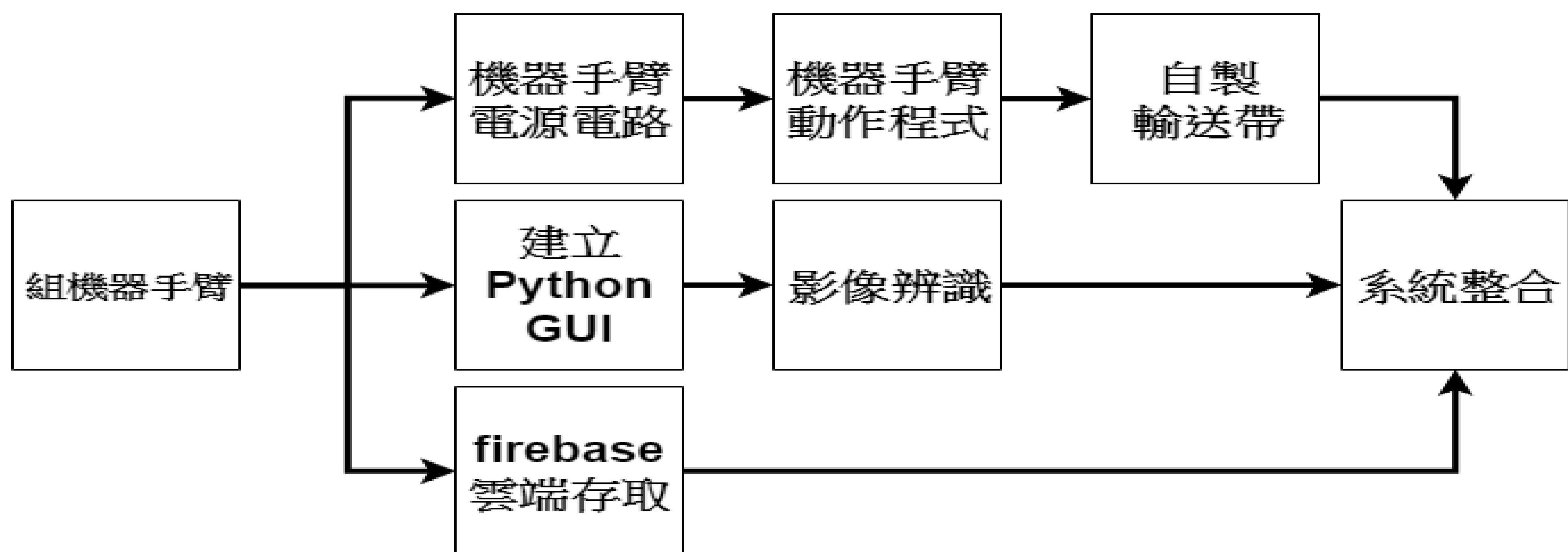
一、研究動機

在這個時代，科技不斷的進步、機械技術不斷的提升之中，製造工廠內不再像以往一樣人數很多，反而大部分的工作都交給機器人來處理和製造，不用給它休息、上下班的時間，只需要給它電力、定期的維護和維修及遠端監控來確認目前情形就可以長時間、長期的工作。

這樣一來不僅能夠減少很多人力成本方面的花費，並減少人員暴露在危險的工作環境中的次數和機率，更可以增加製作過程中的精確度以及提升工作和製造的效率，進而提升產品的產量。

由於自動化花費成本少，換來的效益也大，所以這個情況在現在的工廠中也越來越普遍，而這就是所謂的工廠自動化。因此我們想研究與模擬適用於自動化工廠的產線中識別產品好壞的機械手臂。

二、研究過程



Python GUI 人機介面

為了方便控制機械手臂，我們在 Raspberry pi 上建立一個 GUI 界面，能點選手動或自動按鈕來切換手臂狀態，控制馬達的控件由輸入框或拖動條來調整馬達角度，開關夾子的按鈕能控制夾子的開或關，重設按鈕能重置各馬達的角度，重置伺服按鈕用來重設雲端資料庫中的資料。

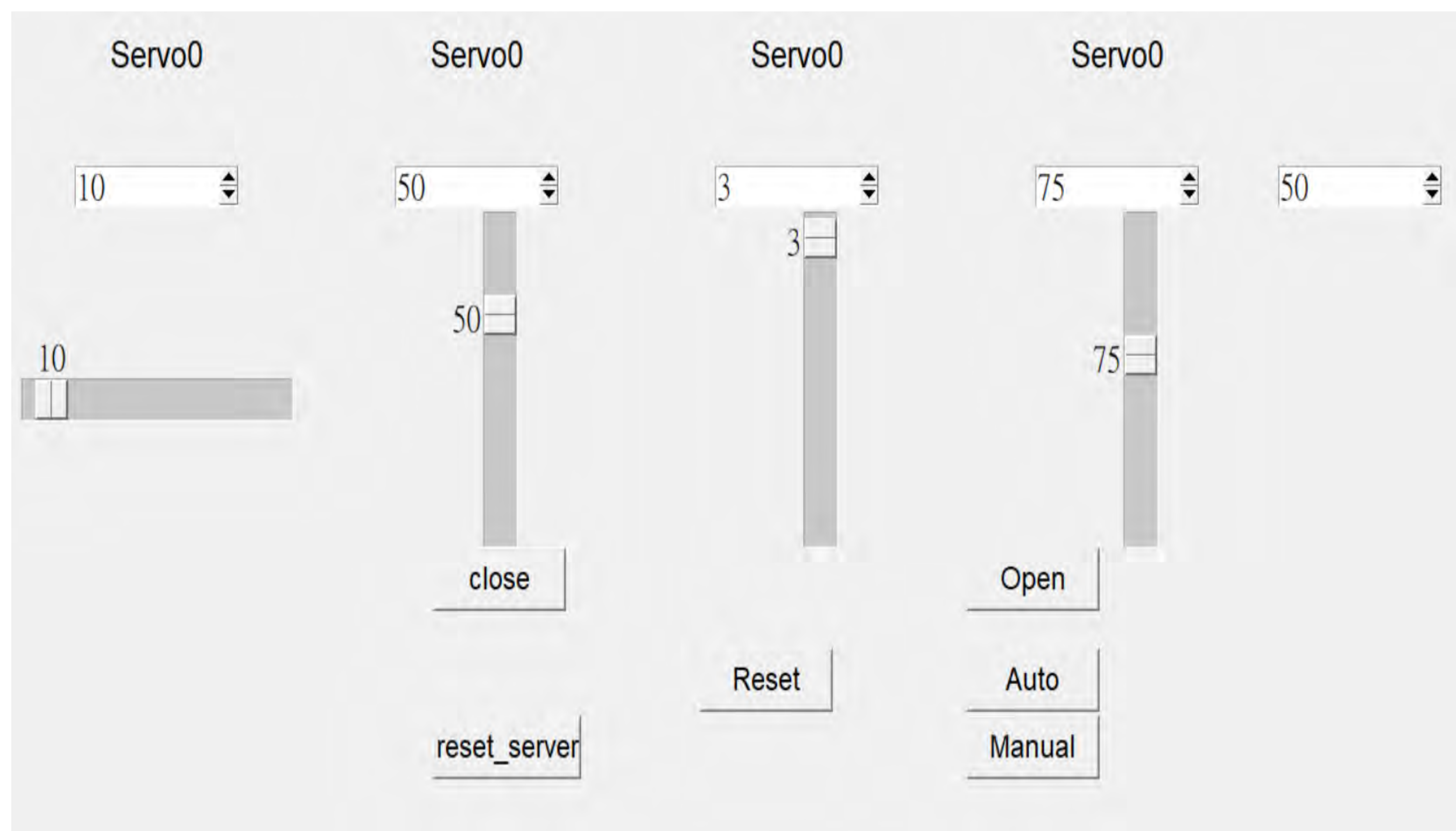


圖.Python GUI

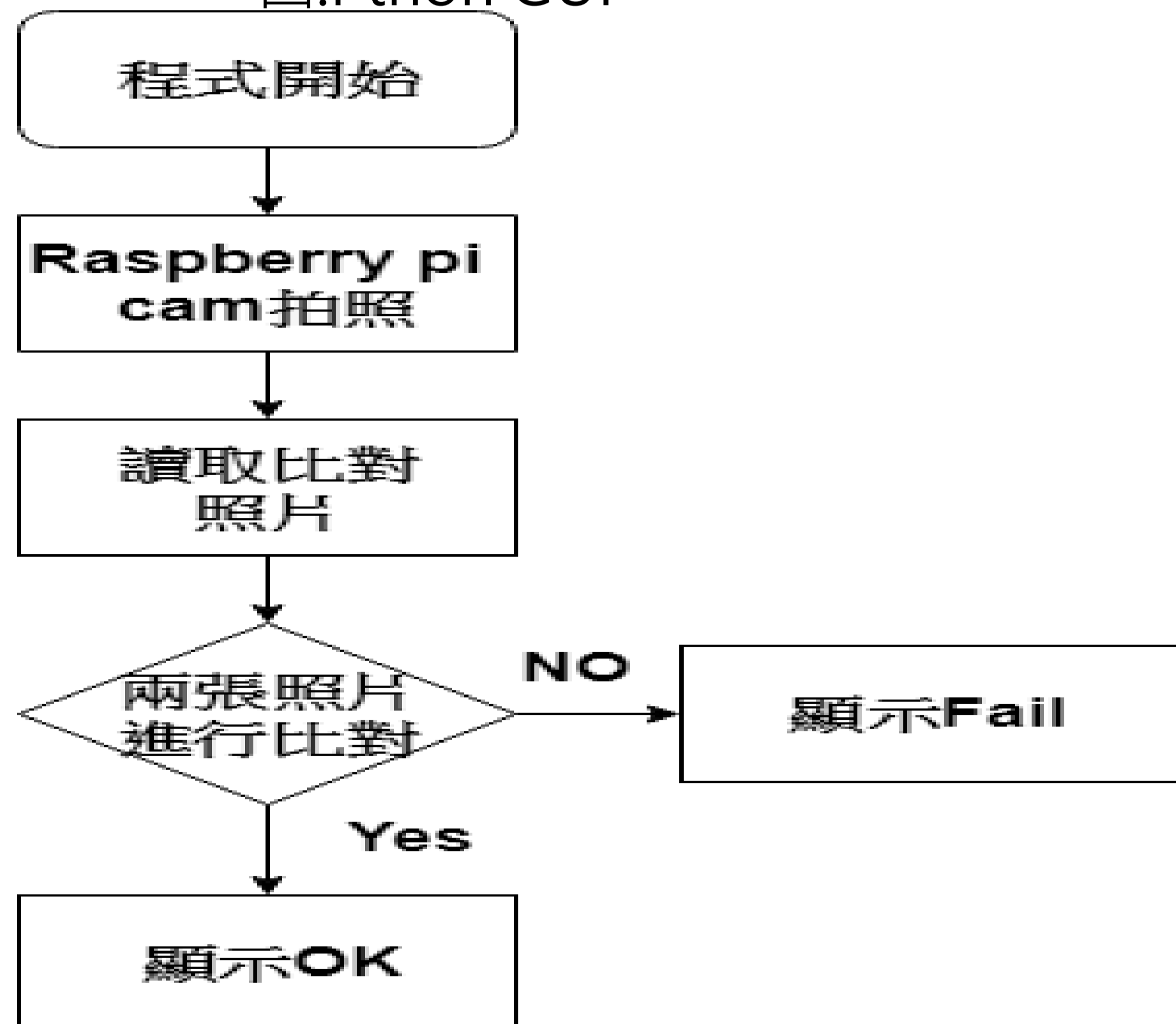


圖.影像辨識流程

影像辨識

影像辨識是採用 OpenCV 裡的 **match Template** 演算法進行影像比對，先給樣本圖片並設定比對的精確度，後將被比對圖加上比對率經過演算法運算，結果會使被比對圖上出現黃框，而黃框內為運算後相識的區域。

雲端存取系統

為了能隨時監控工廠內辨識的狀況，需要用到雲端存取。我們就使用 Firebase 雲端開發平台中的資料庫(Realtime Database)來存取資料。當樹莓派辨識結果為好的物品時，會增加資料庫中 OK 項目的資料數據，若是結果為壞的物品時，會增加 Fail 項目的資料數據。

wade-124678

```
"object_detect_data"  
├── OK: 5  
└── fail: 2
```

圖.雲端存取系統資料內部結構

自製輸送帶

由於市面上販賣的輸送帶普遍價格非常貴，加上外面所販賣的大小都非常大，且重量都非常重，所以我們自己設計一個輸送帶，整體外觀使用 SolidWorks 來設計，並利用雷射切割機及 3D 列印機。

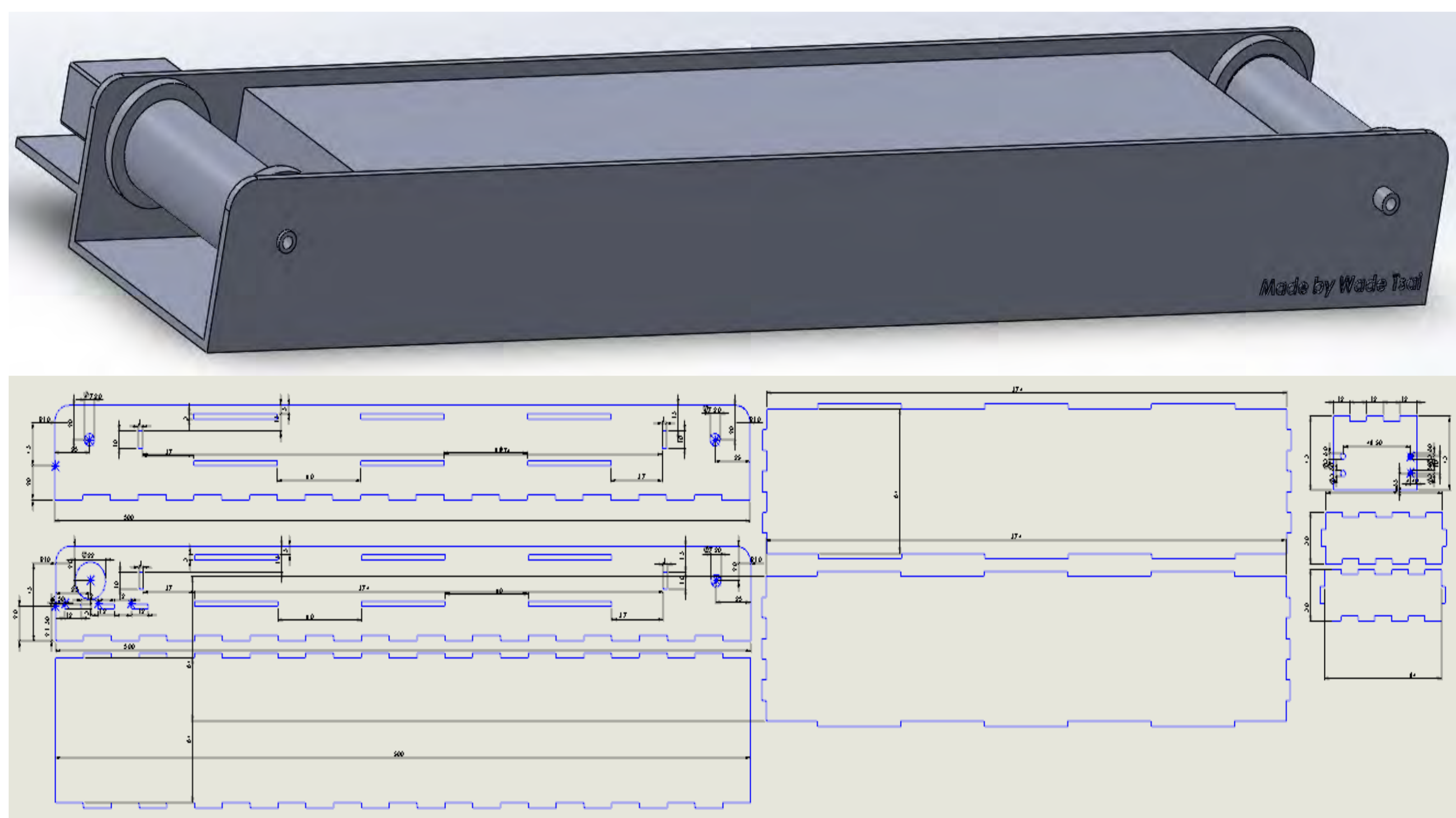


圖.輸送帶設計圖

三、研究結果

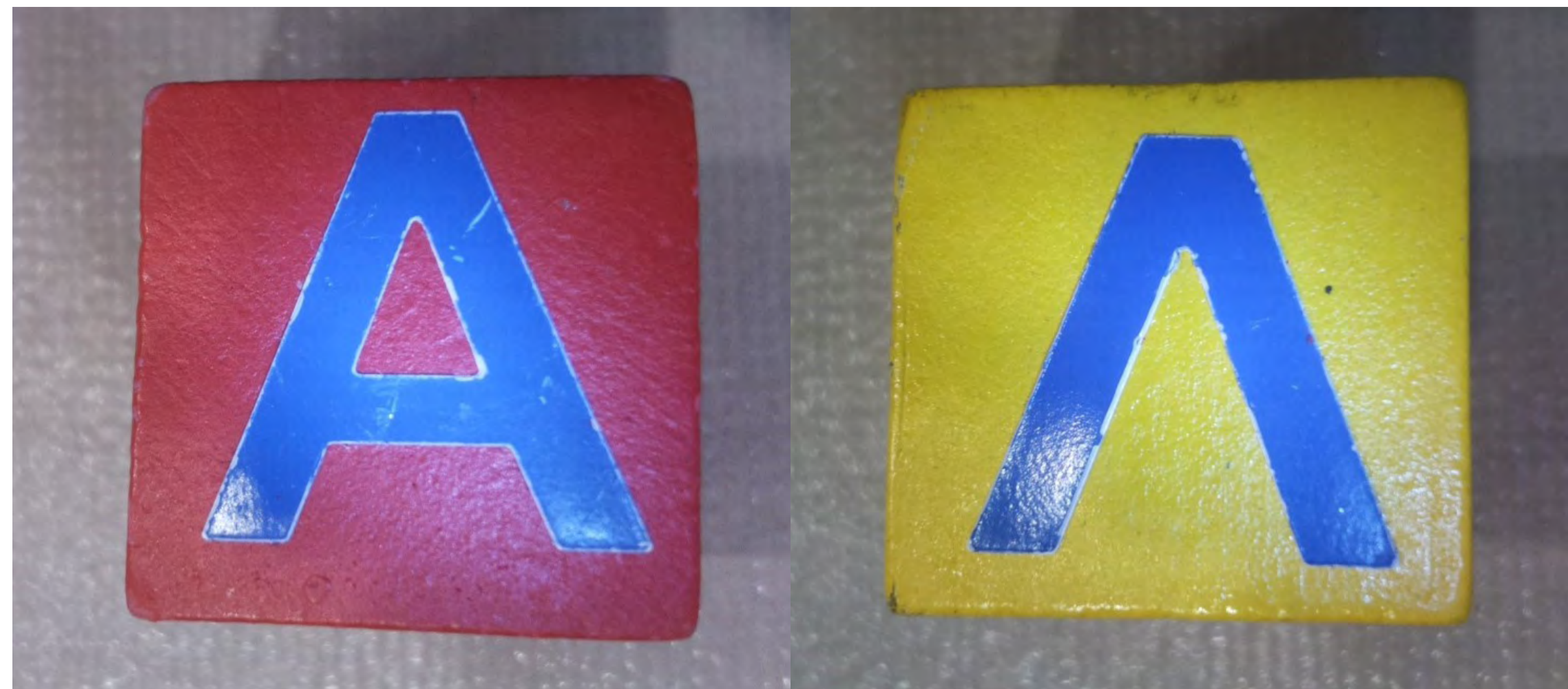


圖.預設樣本照片

圖.比對樣本後正確

圖.比對樣本後無相似處

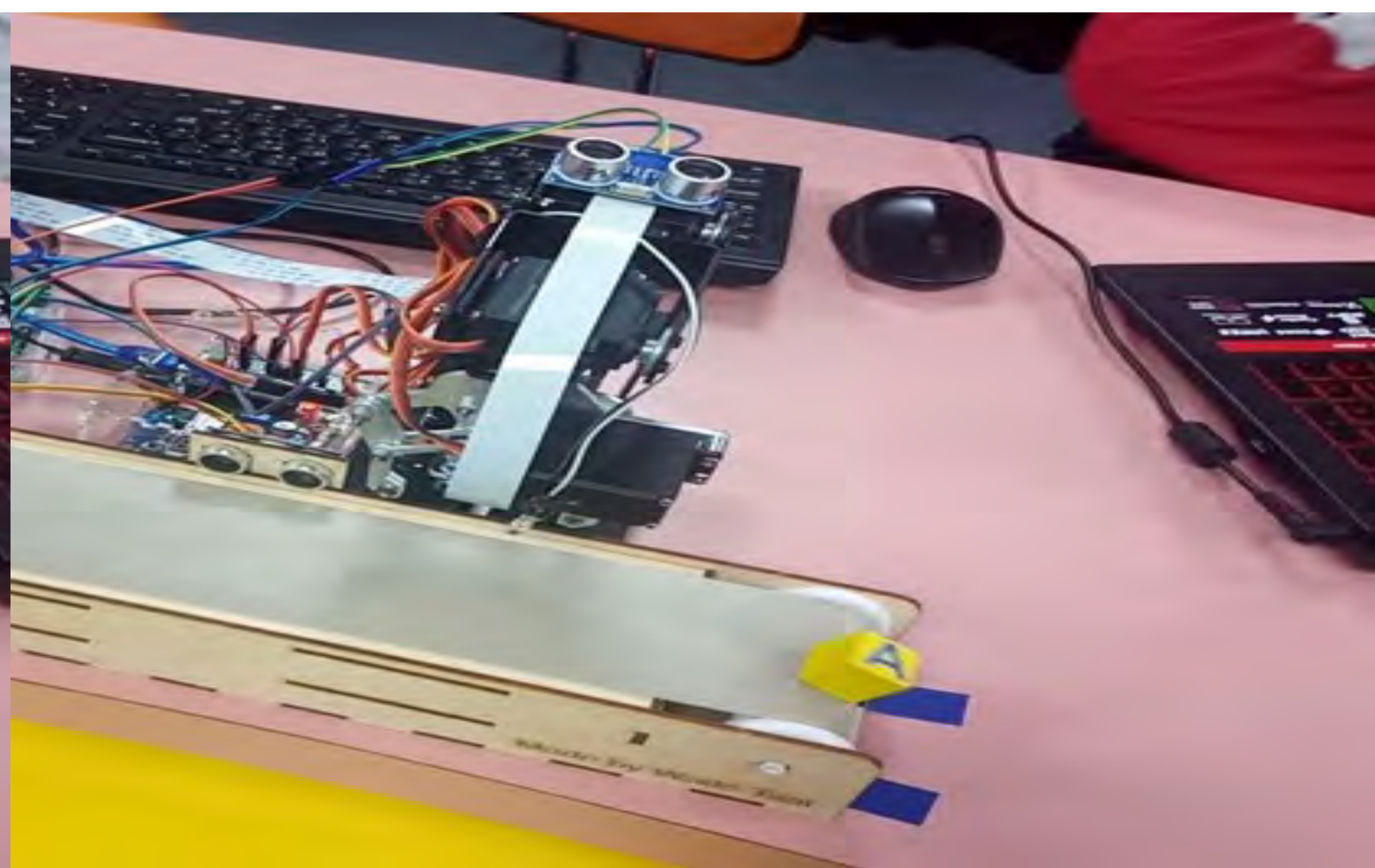
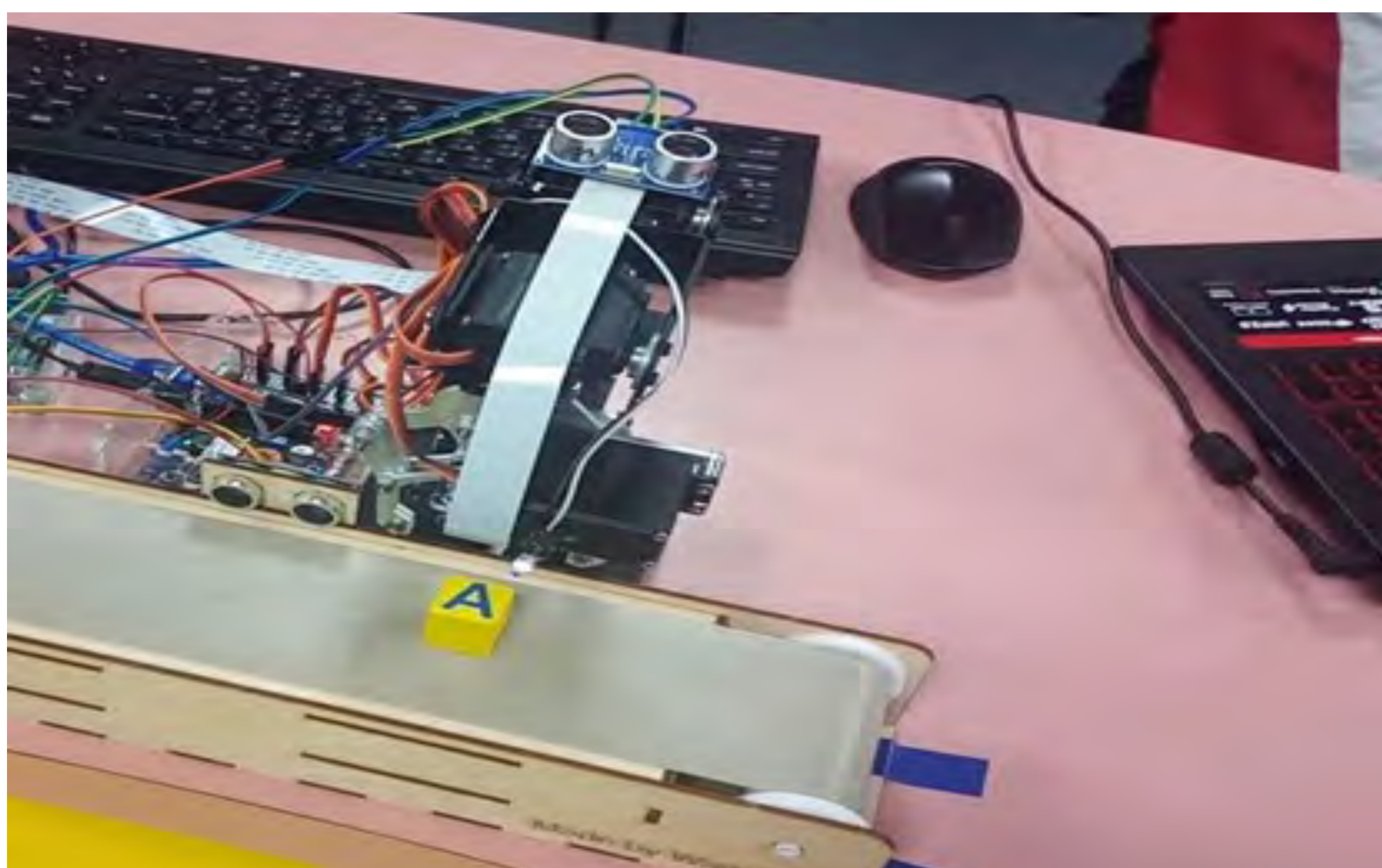


圖.手臂分類(好)

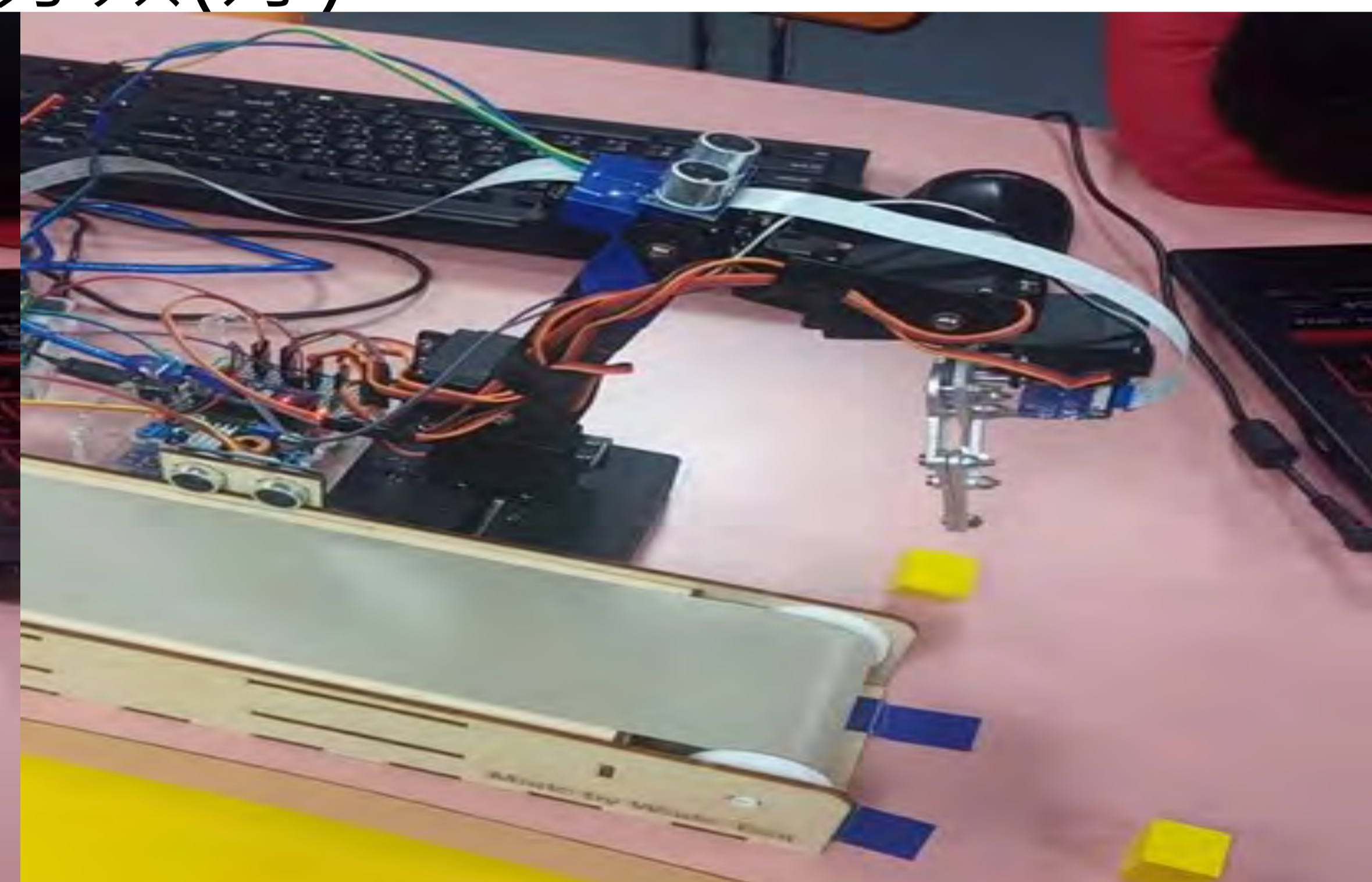
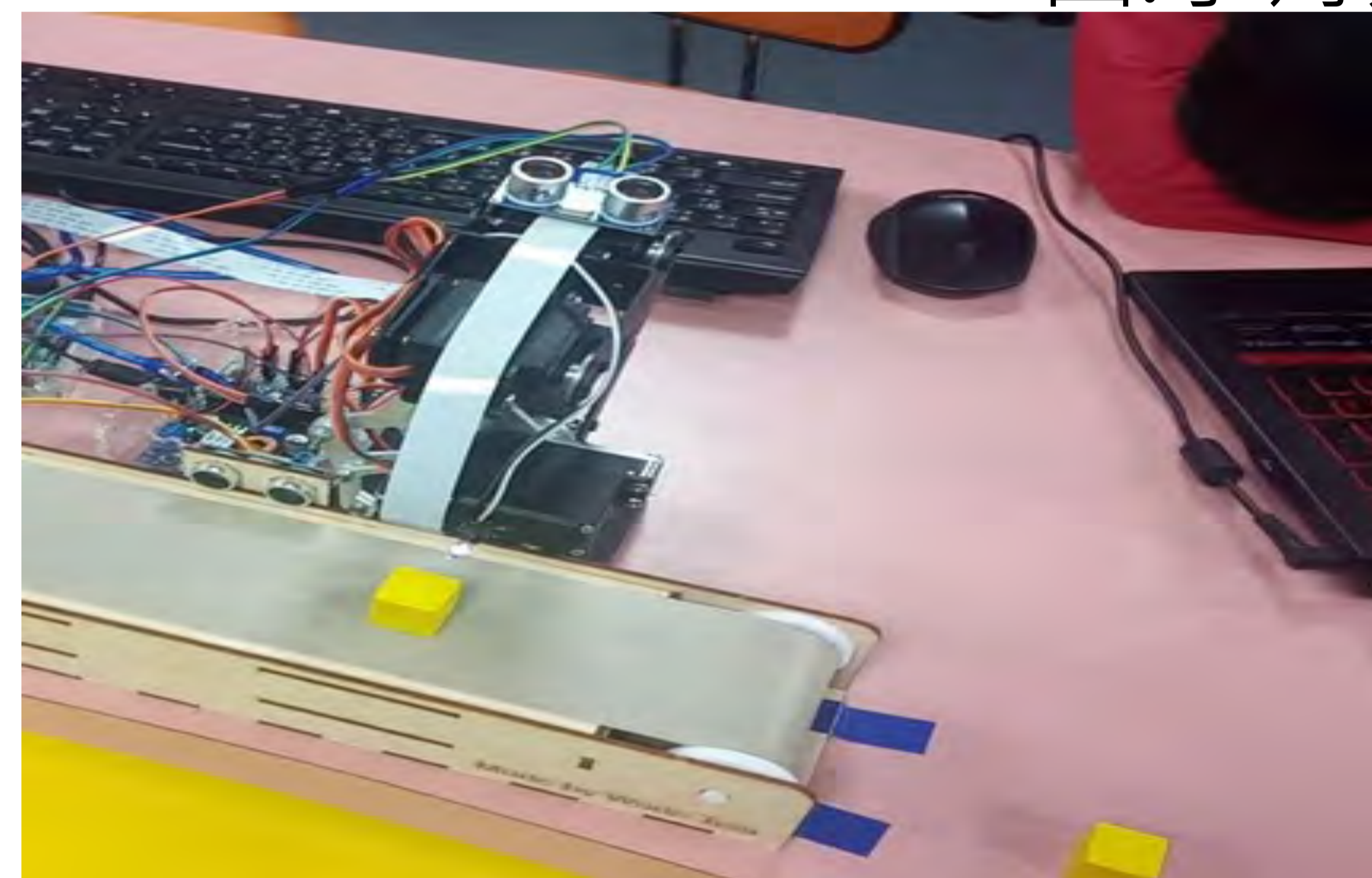


圖.手臂分類(壞)

晚上7:56 2.4KB/s 54

OK:0
Fail:0



圖.雲端監看 APP 畫面

🏠 > item > ii			
wade-124678	📁 item	☰	📄 ii
item >	ii >	OK: 0	fail: 0

🏠 > item > ii			
wade-124678	📁 item	☰	📄 ii
item >	ii >	OK: 1	fail: 1

圖.資料庫上傳前(上)及上傳