

PAPAGO巡邏車

期末報告

指導老師:張顯盛老師

2020/01/07

目錄

CONTENTS

01 前言

02 車體

03 藍芽控制

04 自動移動

05 機器學習

06 結論

07 Q&A

01

前言

◆◆ 專題發想與目標

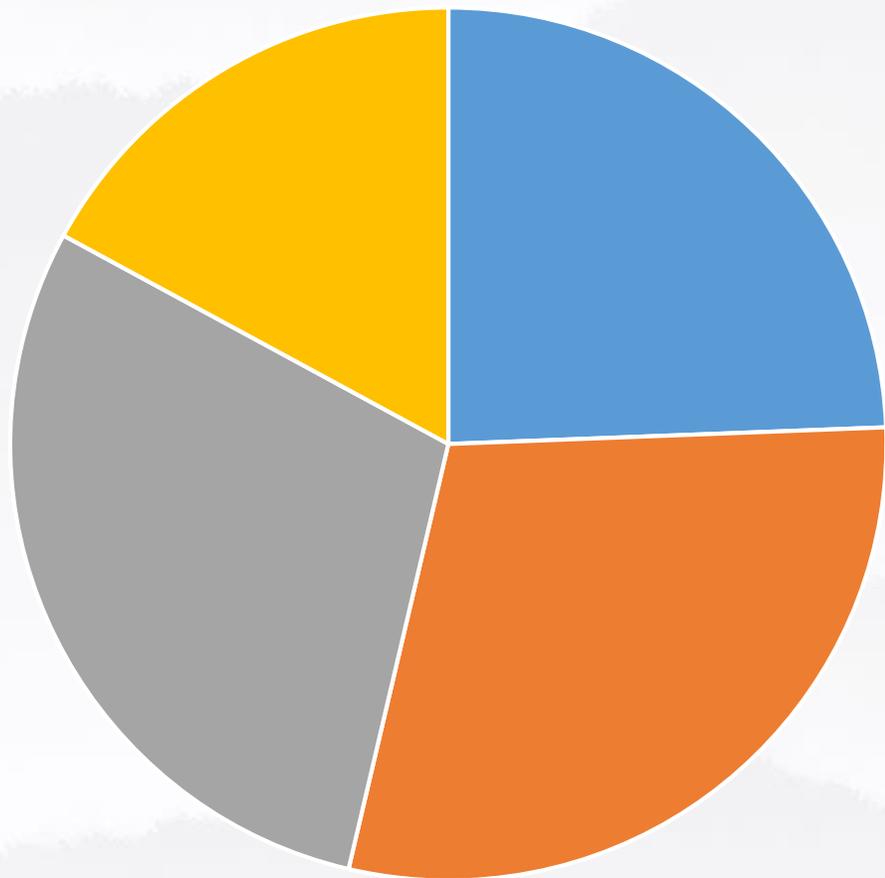
- 近年來在人力資源日漸不足的趨勢下，警員無法更專注於社會治安的維護，導致小型犯罪不斷發生，並且無餘力追緝通緝犯，因此機器輔助便成為打擊犯罪的重要一環。
- 能**自動移動**和**遠端遙控**，同時透過鏡頭跟人工智慧達到**車輛特徵**的辨識並回傳。

甘特圖

工作項目	週次 (日期)																		負責成員
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
蒐集資料	■	■	■	■	■	■	■	■	■	■	■								全員
理論探討	■	■	■	■															全員
計畫書製作				■	■														全員
車體製作	■	■	■						■	■									陳、藍、邱
外觀製作	■	■	■										■	■					陳
藍芽控制		■	■	■	■														江、邱
AI 辨識					■	■	■	■	■	■	■								陳、藍
移動軟體 製作			■	■	■														江、邱
辨識軟體 製作									■	■	■	■	■						陳、藍、邱
整體整理															■	■			全員
完成成品																■	■		全員
撰寫報告					■			■	■		■		■		■		■		全員
口頭報告						■				■		■		■		■		■	全員
預定進度	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	累 積 百分比%

成員貢獻度

貢獻度



■ 陳重佑 ■ 江權祐 ■ 邱展慶 ■ 藍國恩

人員	工作	比重
江權祐	軟體製作	30%
陳重佑	文書處理 硬體設計	27.5%
藍國恩	軟體製作 硬體設計	12.5%
邱展慶	軟體設計 硬體提供	30%



◆◆ 使用的軟體和硬體

軟體

1. Python
2. TensorFlow + Keras
3. OpenCV
4. Jupyter Notebook
5. Laserbox

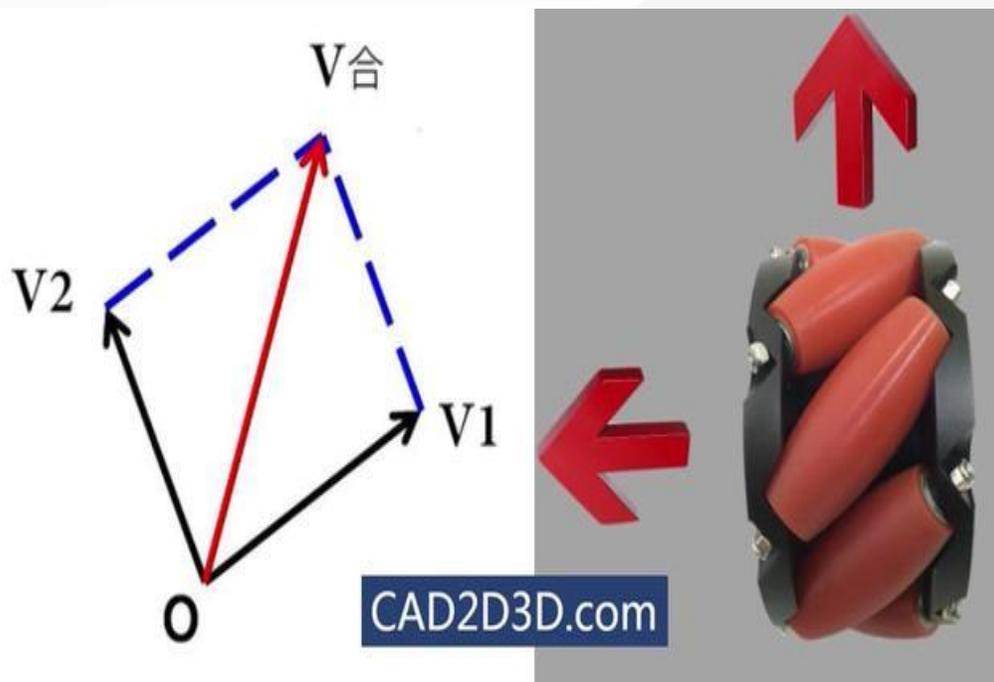
硬體

1. 樹莓派
2. 萬向輪
3. 直流馬達
4. L298N
5. 鏡頭
6. 雷射切割機

02

車體

萬向輪



原理簡介

將力量分成兩個向量，一個**前/後**和一個**右/左**。當車輪轉動時，**向量互相抵消**，使之移動。

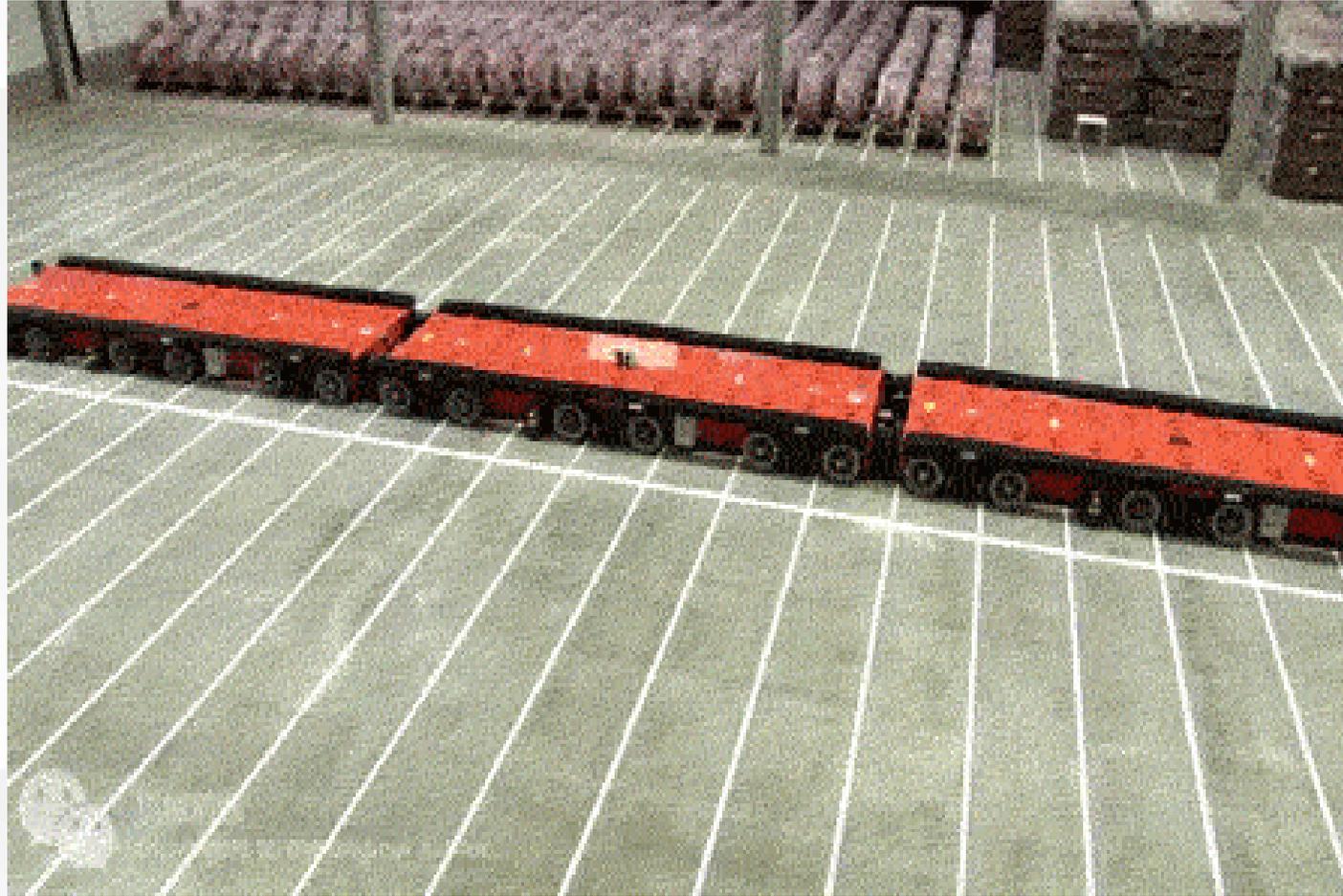
優缺點比較

優點：全方向移動，移動死角少。

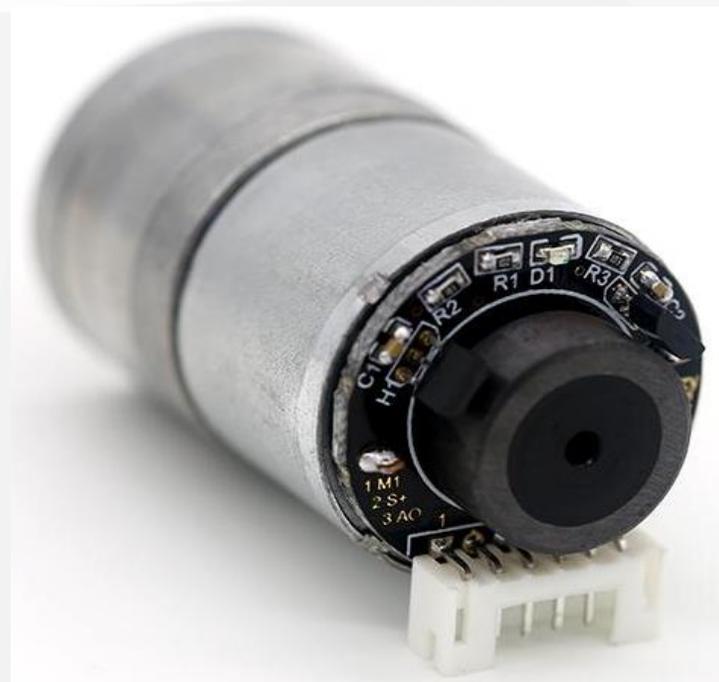
缺點：貴、加工較難、較重、壽命不長、環境適應力差以及每個輪子須要單獨驅動。

萬向輪

全方向移動示意圖



馬達



馬達產品規格:

- 馬達電壓:3V~12V(典型6V)
- 空載轉速:測試電壓6V時 35RPM

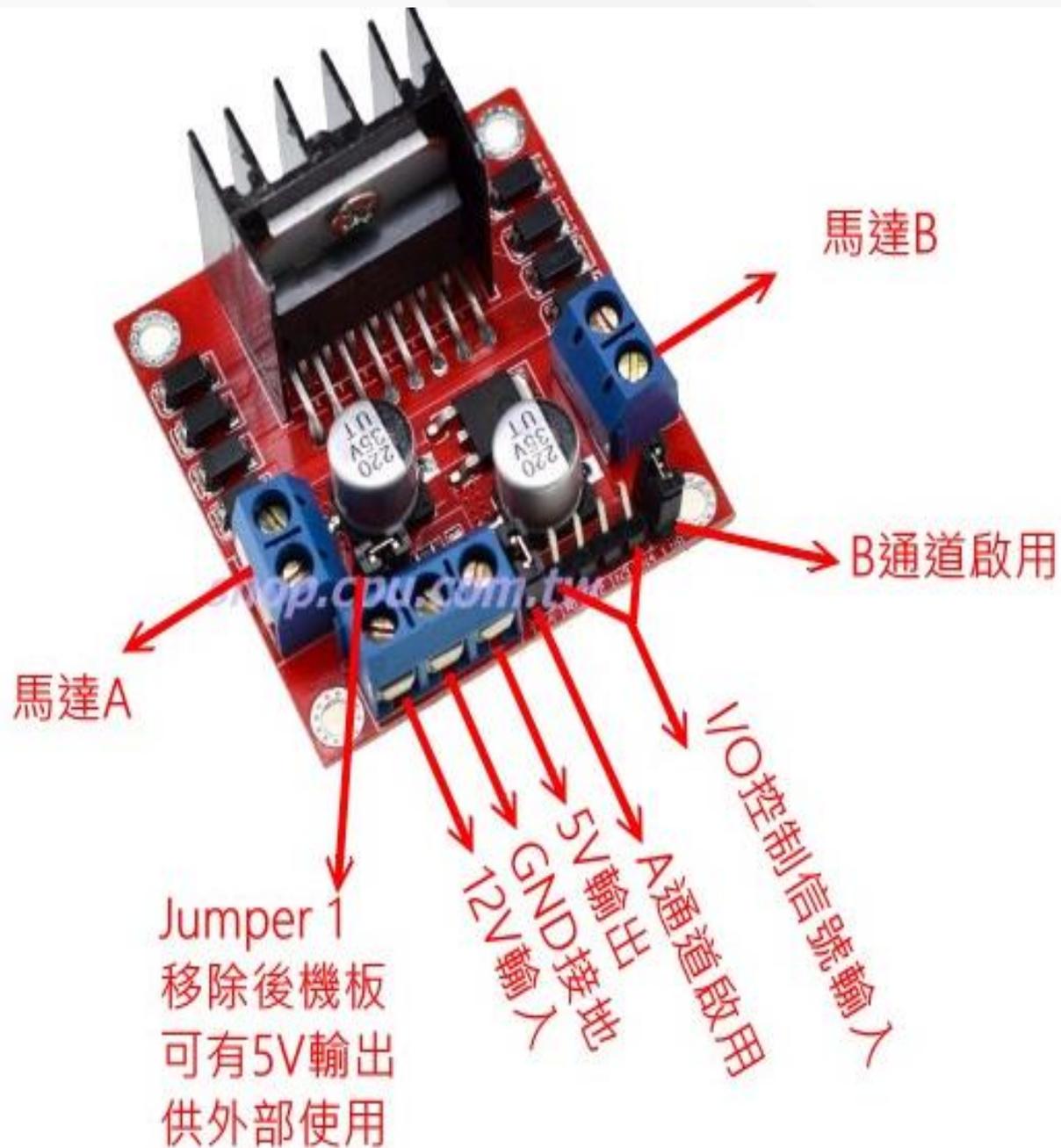
馬達接腳定義(1到6從左到右):

- pin1: + (接到L298N的IN1)
- pin2: 編碼器電源接3.3V或5V
- pin3: AO相位輸出
- pin4: BO相位輸出
- pin5: 編碼器電源接地
- pin6: - (接到L298N的IN2)

◆◆ L298N模組

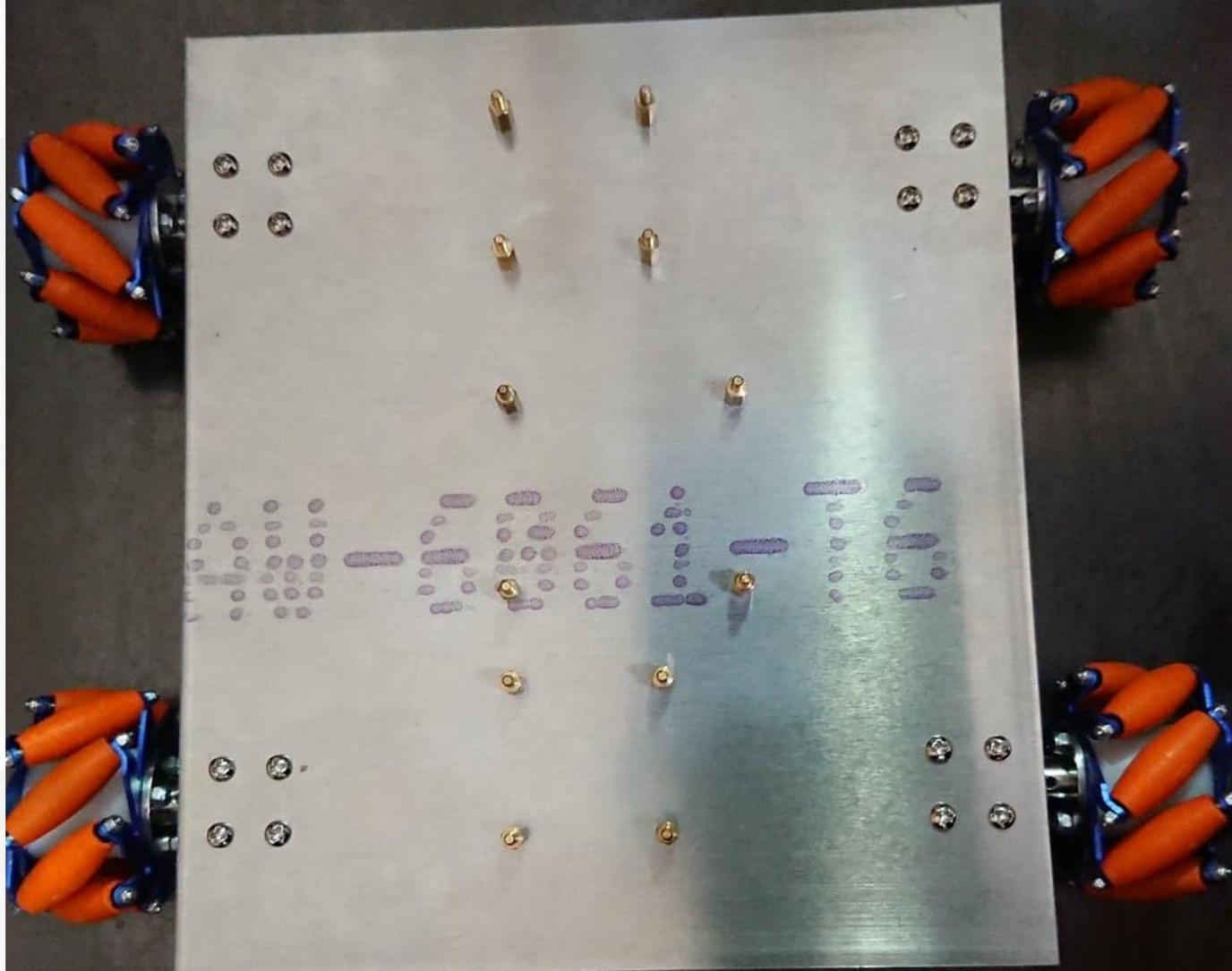
L298N馬達驅動規格:

- 主控晶片：L298N
- 電壓：5V(晶片)
- 驅動電壓：**5V ~ 35V**
- 驅動電流：2A
- 工作溫度：-20°C ~ 135°C
- 最大功率：25W
- 驅動電壓為7V-12V的時候，
可以引出5V電壓供外部使用

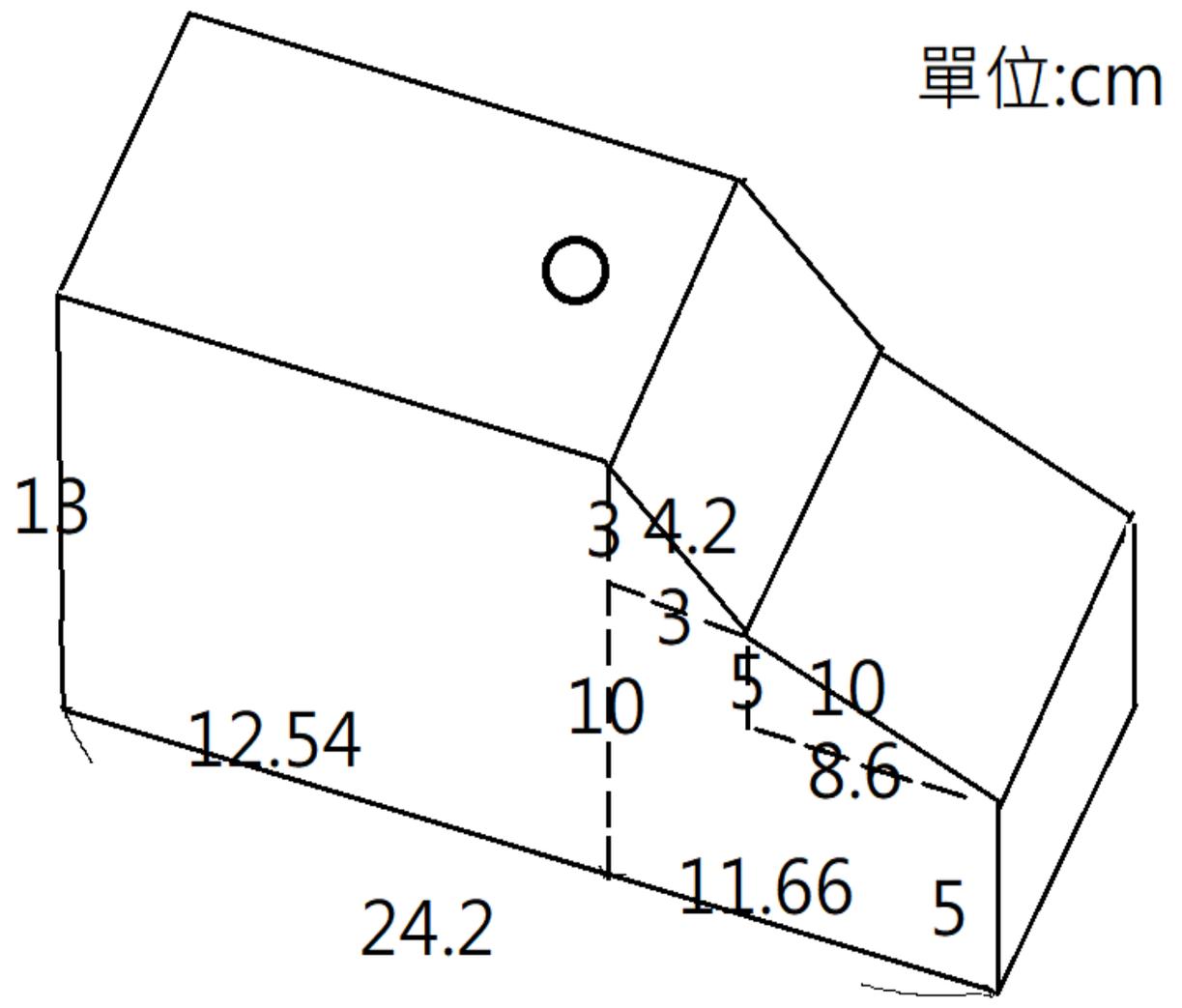




車體底盤



◆◆ 車體外觀



03

藍芽控制

樹莓派簡介

規格	
SOC	Broadcom BCM2837
CPU	1.2GHz 64-bit quad-core ARM Cortex-A53
RAM	1GB LPDDR2 (和 GPU 共享)
視訊輸出	Composite RCA; HDMI
儲存	microSD
USB	USB 2.0 x 4
Bluetooth	Bluetooth4.1;BluetoothLowEnergy(BLE)
GPIO	40-pin 2.54mm expansionheader
尺寸	85mm x 56mm x 17mm



藍芽

- 藍芽版本:藍牙4.2 :低功耗藍牙(BLE) ←
- 藍芽模組:樹莓派內建
- 藍芽使用:在樹梅派需安裝BLUEZ套件，才能使用多數藍芽功能
- 使用裝置:安卓手機
- APP撰寫: 使用APP inventor

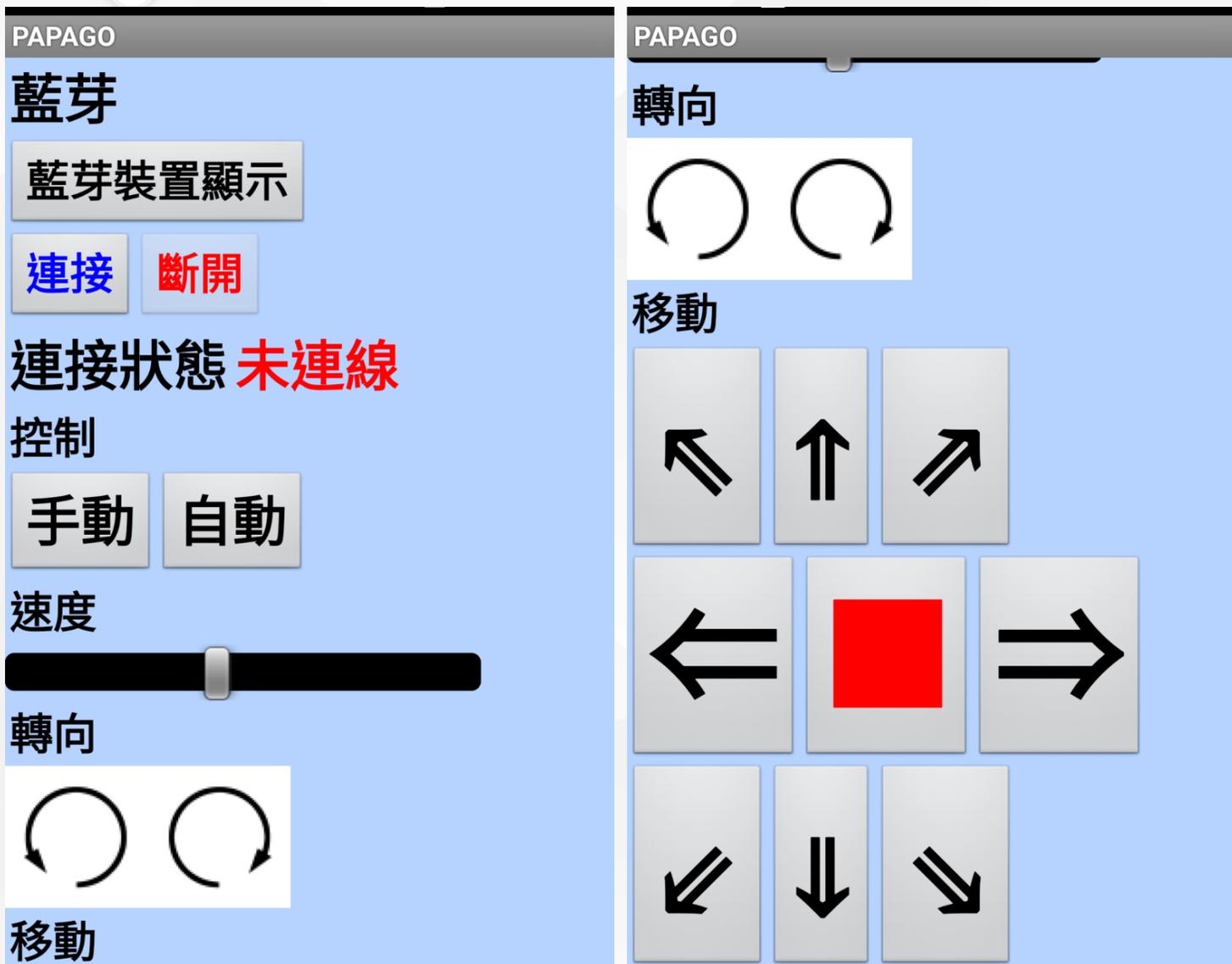
(BLE優勢)

1. 低功耗
2. 低成本
3. 多數平台支援



MIT
APP INVENTOR

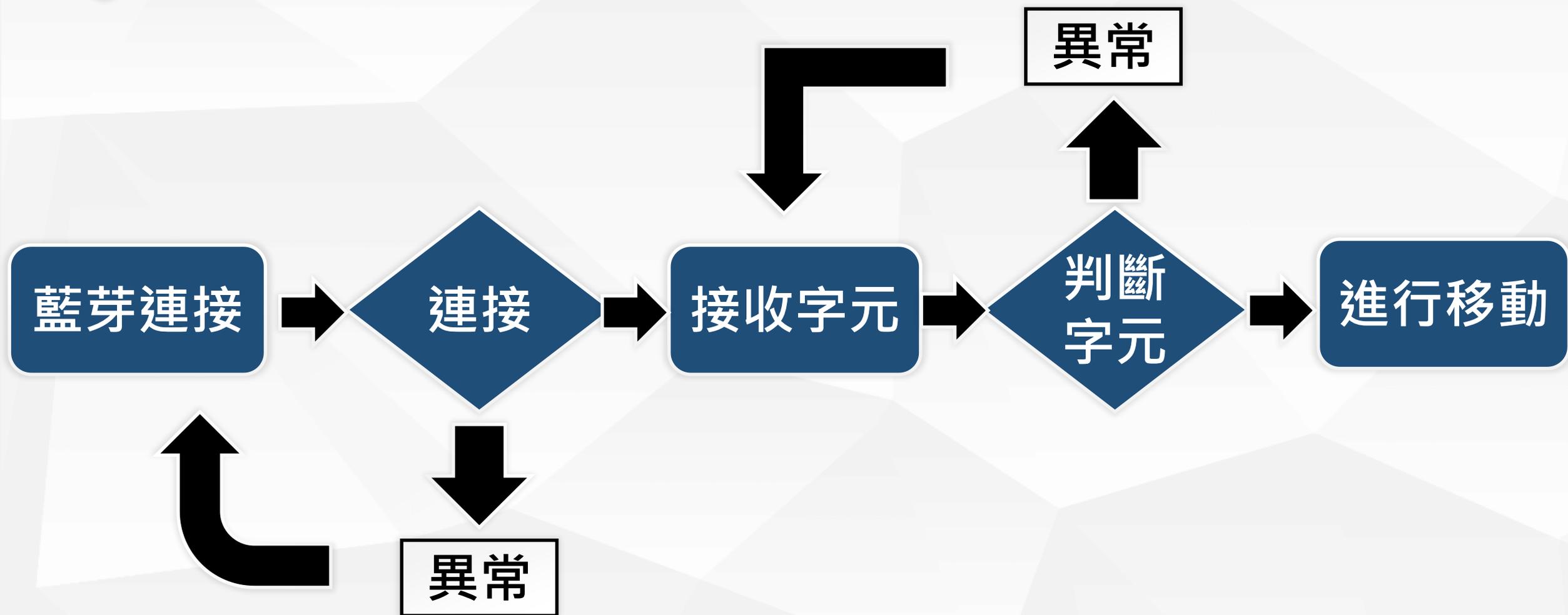
藍芽APP



APP主要功能

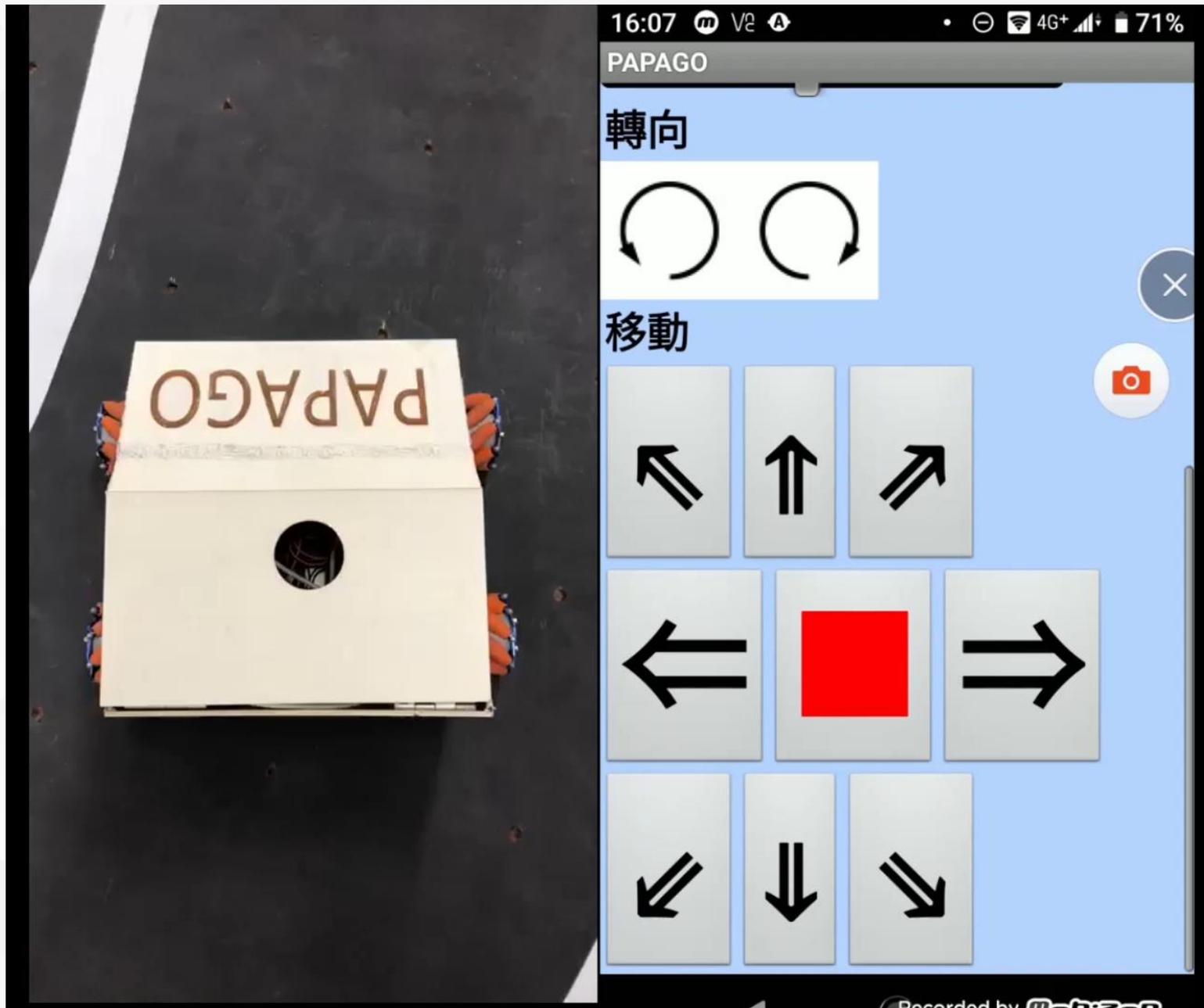
- 透過**藍芽裝置顯示**按鈕設定與樹莓派與**手機**的連線
- 透過**連結**和**段開**按鈕，建立和程式的連線
- 透過**自動**按鈕讓車體自行移動
- 透過**轉向**按鈕，設定轉動方向，按多久轉多久
- 透過**方向鍵**控制移動位置
- 手煞車按鈕()

藍芽流程



藍芽遙控成果

按鈕按下時，**程式可以接收字元**，結合移動程式完成遙控的功能。



遇到問題

問題

問題一

安裝BLUEZ問題

樹莓派有 Python2和 Python3 的差別，bluez套件如裝錯位置，會造成程式無法使用Bluetooth。

解決方法:重裝並使用pip3指令安裝

問題二

藍芽協定不同(無法和APP連接)

解決方法:需修改藍芽套件內部文件

- 新增與修改兩行

```
ExecStart=/usr/lib/bluetooth/bluetoothd -C
```

```
ExecStartPost=/usr/bin/sdptool add SP
```

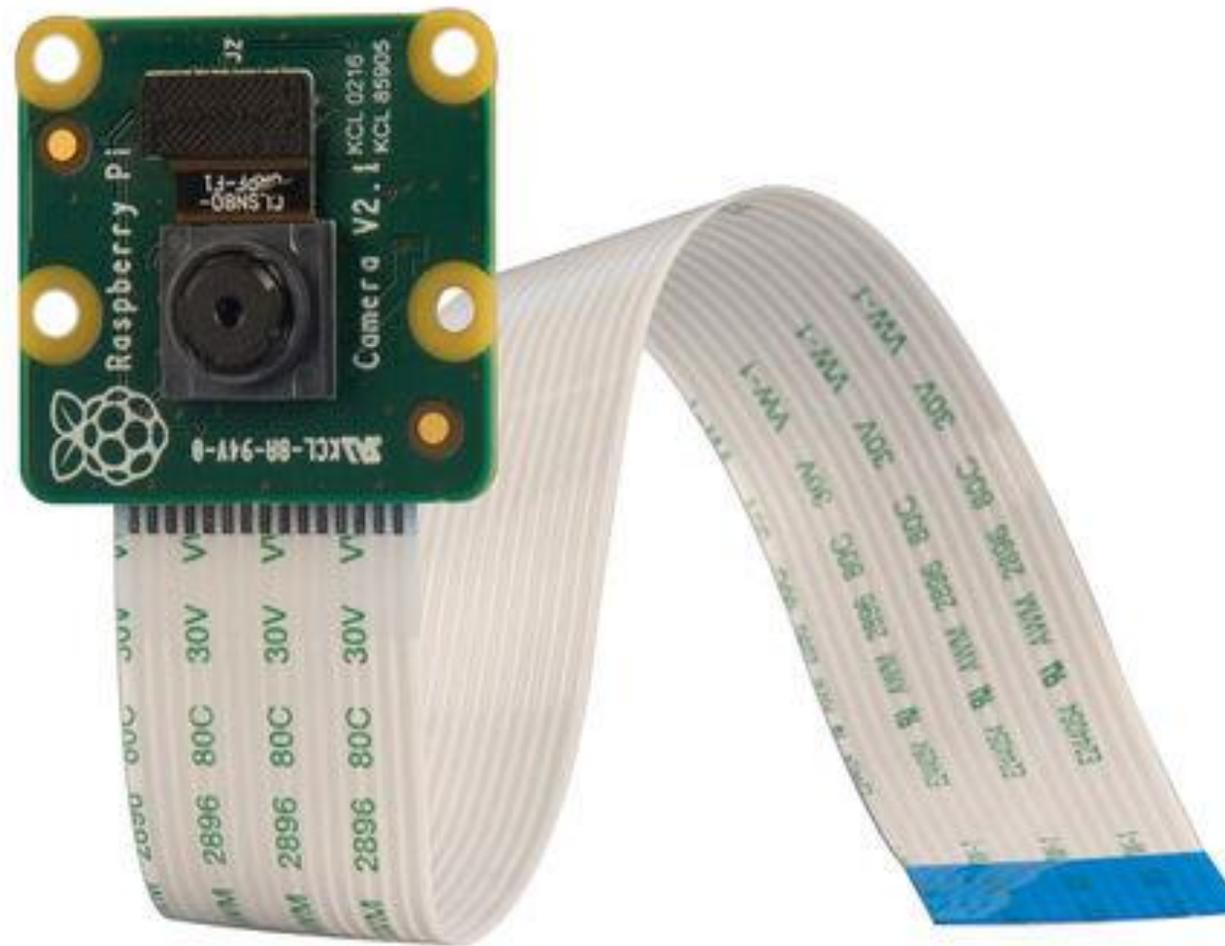
04

自動移動

◆◆ NOIR Camera V2

鏡頭規格:

- 800萬像素
- 支持1080p30，720p60錄影
- 大小25mm x23mm
- 重量3克
- 採用CSI接口



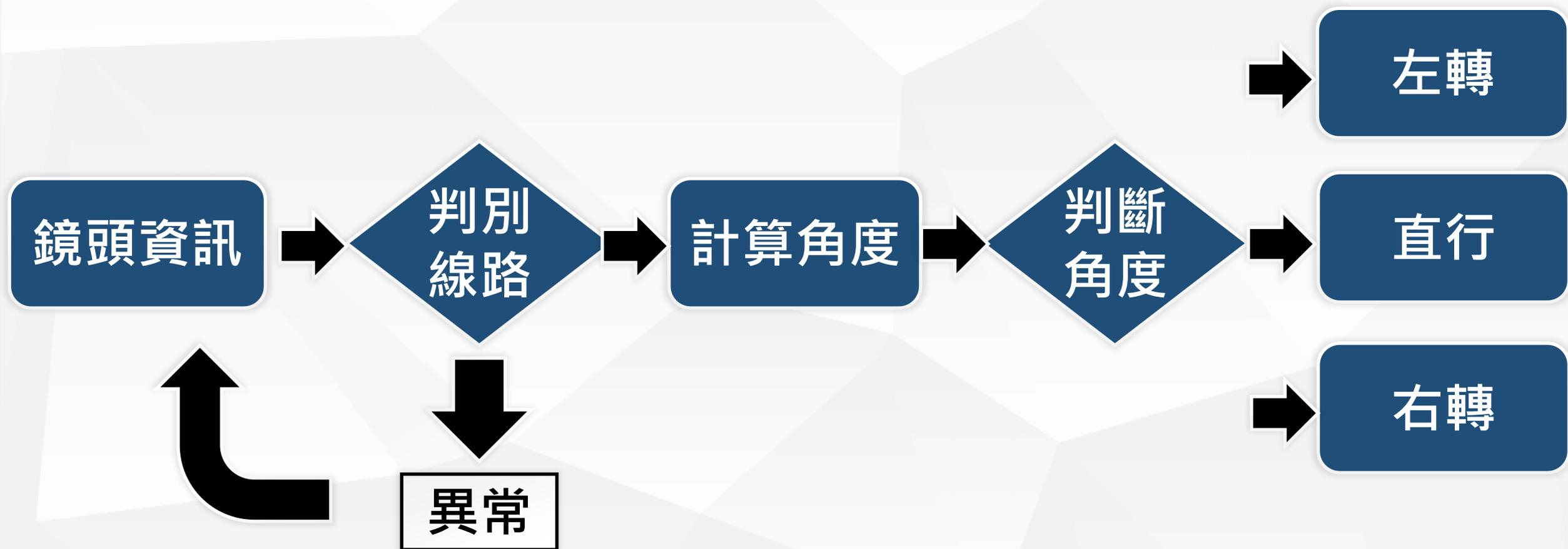
簡介

OpenCV是一個基於BSD許可發行的跨平臺電腦視覺庫，可以執行在各種作業系統上。

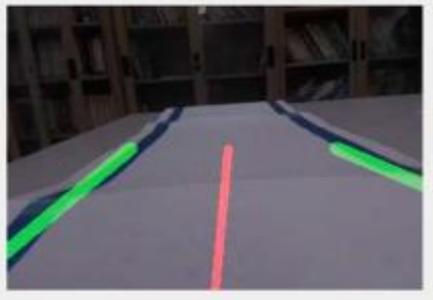
提供了Python語言的介面，實現了**影像處理**和**電腦視覺**方面很多的通用演算法。



自動流程

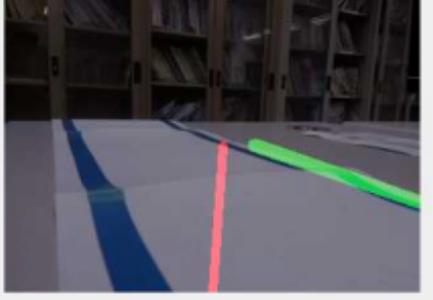


自動移動成果



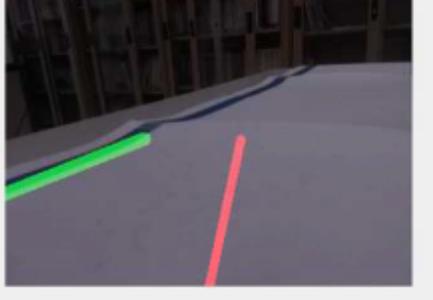
```
.video_or  
_image('D  
ect lane\  
< 12:  
+= 1  
  
115 else:  
116     image_lane  
117     angle_lane  
118     self.video  
119     show_image  
120     self.move_  
121     i = 0  
122     if cv2.waitKey
```

```
Shell  
DEBUG:2019-12-24 15:59:07,178:  
DEBUG:2019-12-24 15:59:07,180:  
DEBUG:2019-12-24 15:59:07,183:  
DEBUG:2019-12-24 15:59:07,184:  
DEBUG:2019-12-24 15:59:07,187:  
DEBUG:2019-12-24 15:59:07,238:  
DEBUG:2019-12-24 15:59:07,243:  
DEBUG:2019-12-24 15:59:07,245:  
INFO :2019-12-24 15:59:07,246:  
INFO :2019-12-24 15:59:07,247:  
angle:95  
front
```



```
6 from hand_coded_lane_follo  
7 from objects_on_road_proce  
8 from move import Movepapag  
9 from time import sleep  
10  
11 _SHOW_IMAGE = True  
12  
13
```

```
Shell  
DEBUG:2019-12-24 16:02:29,010:  
DEBUG:2019-12-24 16:02:29,611:  
DEBUG:2019-12-24 16:02:29,614:  
INFO :2019-12-24 16:02:29,699:  
DEBUG:2019-12-24 16:02:29,702:  
DEBUG:2019-12-24 16:02:29,707:  
DEBUG:2019-12-24 16:02:28,860:  
INFO :2019-12-24 16:02:28,861:  
INFO :2019-12-24 16:02:28,862:  
angle:36  
left  
DEBUG:2019-12-24 16:02:29.442:
```



```
6 from hand_coded_lane_follo  
7 from objects_on_road_proce  
8 from move import Movepapag  
9 from time import sleep  
10  
11 _SHOW_IMAGE = True  
12  
13
```

```
Shell  
DEBUG:2019-12-24 16:04:38,080:  
DEBUG:2019-12-24 16:04:38,687:  
DEBUG:2019-12-24 16:04:38,689:  
DEBUG:2019-12-24 16:04:38,704:  
DEBUG:2019-12-24 16:04:38,708:  
DEBUG:2019-12-24 16:04:38,710:  
DEBUG:2019-12-24 16:04:38,711:  
INFO :2019-12-24 16:04:38,712:  
INFO :2019-12-24 16:04:38,714:  
angle:157  
right  
DEBUG:2019-12-24 16:04:39,285:
```

遇到問題

問題一

鏡頭幀數與程式不符

樹莓派讀取線路效能約是1秒7幀，但程式卻是1秒取一次畫面，導致需要7秒才能反應完鏡頭的畫面。 **解決方法:計數變數直到1秒**

問題

```
if i < 12: '''等待大約1秒'''
    i += 1
else:
    image_lane = self.follow_lane(image_lane)
    angle_lane = self.angle(angle_lane)
    self.video_lane.write(image_lane)
    show_image('Lane Lines', image_lane)
    self.move_papago.angle_move(angle_lane)
    i = 0
```

遇到問題

問題

問題二

鏡頭與移動的時間差

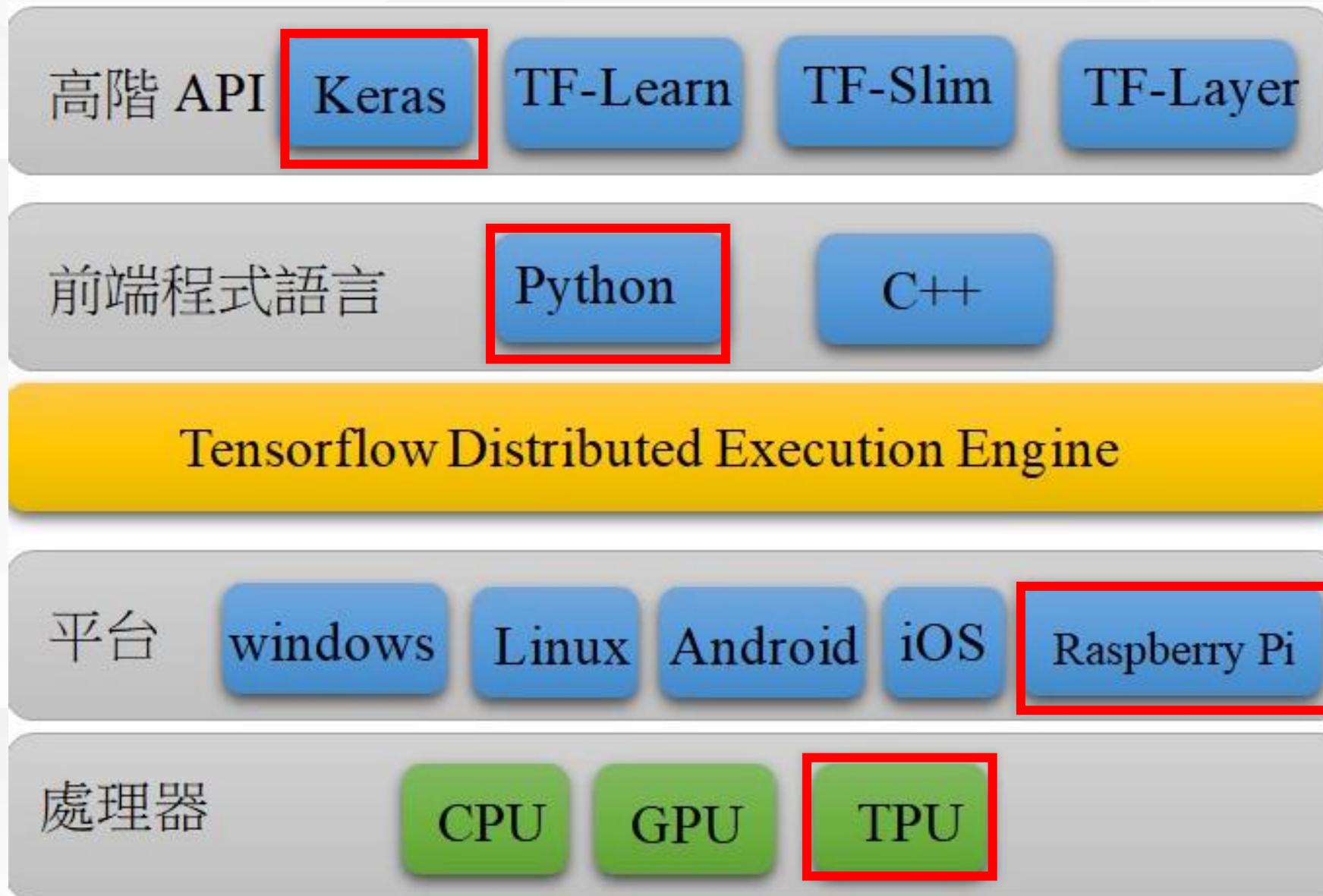
解決完上述問題後，我們發現雖然程式與鏡頭對上了，可是卻與移動有**0點幾秒**的時間差，如果只是單次判斷是沒關係，但當判斷十幾次後，就會導致移動與畫面不符。

解決方法: 無，不管如何優化這個問題都存在。

05

機器學習

架構圖



Tensorflow



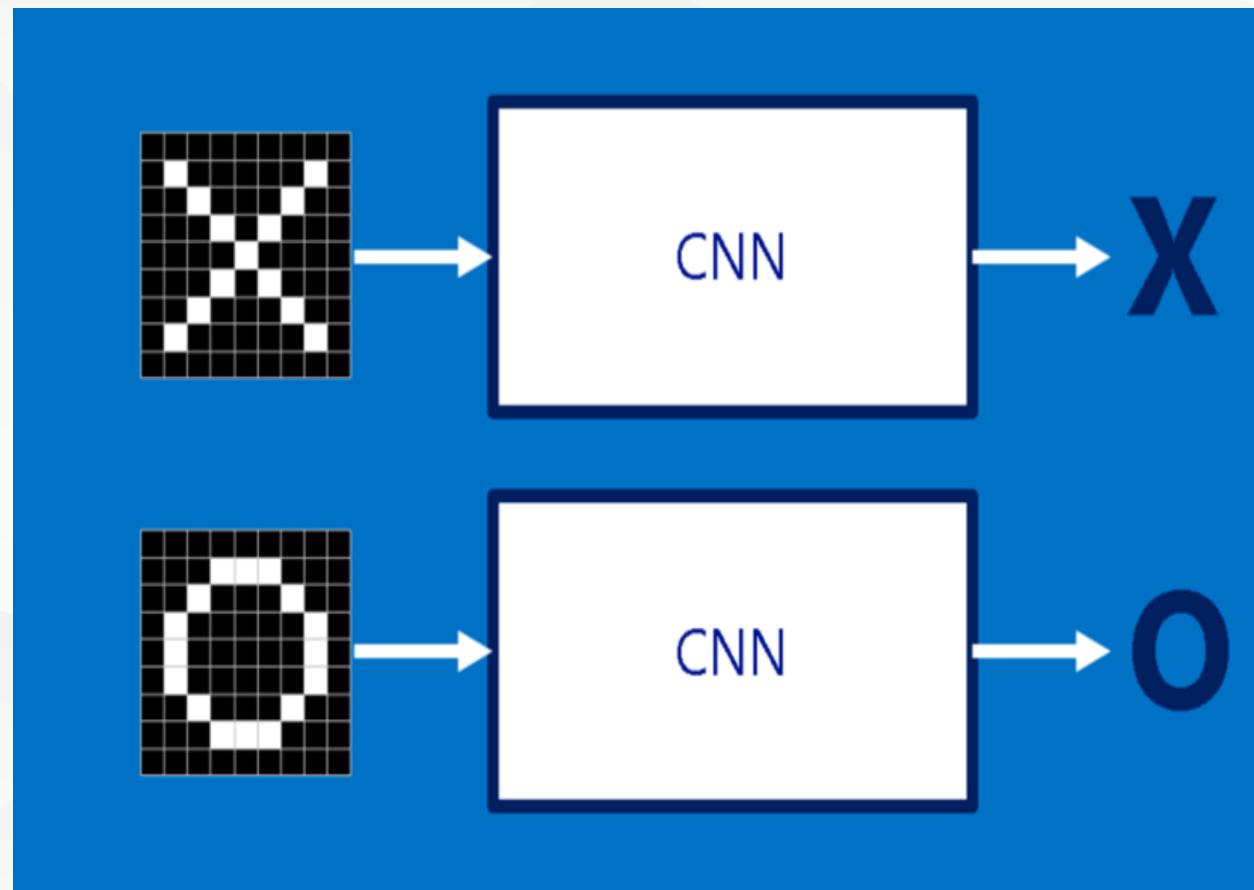
TensorFlow是由Google提供的開放原始程式庫，從2010年開始，Google建立DistBelief作為他們的第一代的機器學習系統，TensorFlow則是**第二代機器學習系統**。

TensorFlow用快速迭代不同架構的能力減少了構建所需的時間，而其**可移植**的特性，可以在不同的平台上執行模型，因此構建一次，就能隨處運行，使構建**CNN模型**變得更加容易。

CNN(卷積神經網路)

CNN是目前**深度神經網路** (deep neural network) 領域的主力，有時在圖片辨別上甚至比人類還要精準。

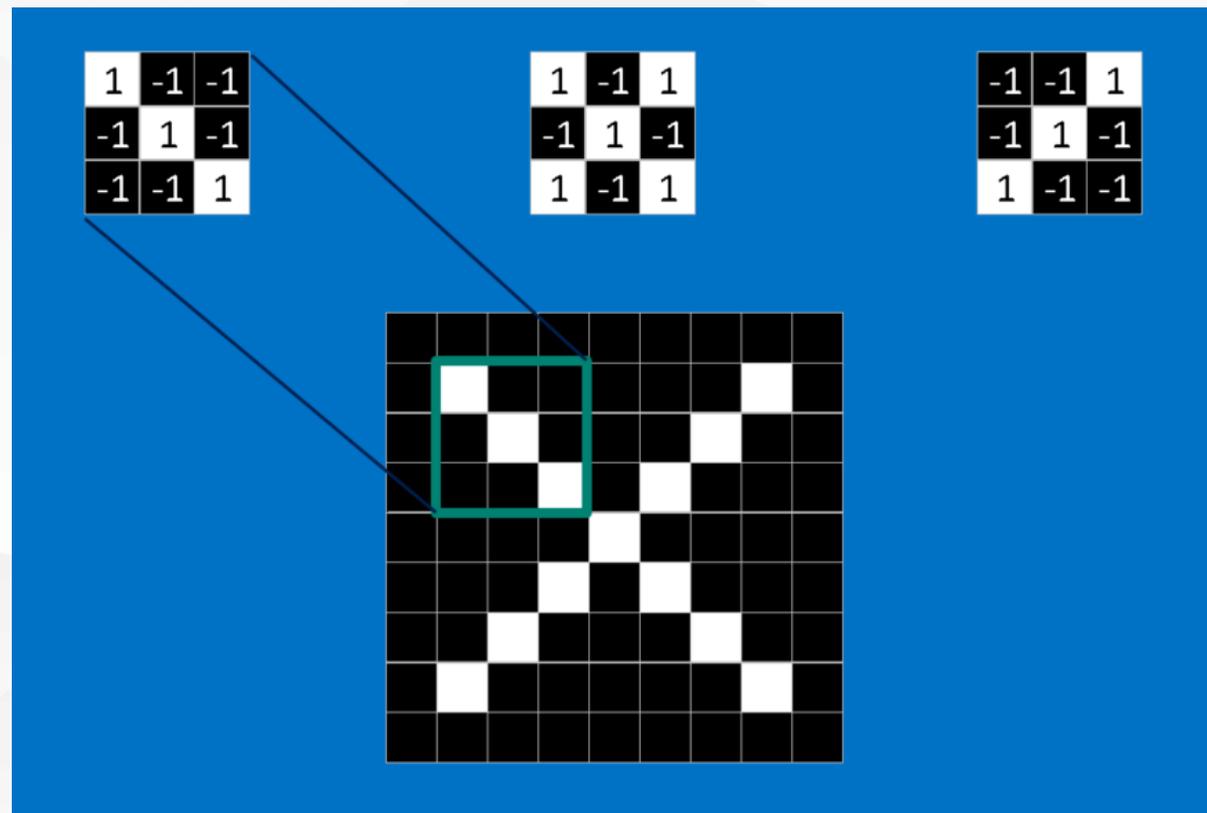
以右圖例子來說明，CNN會辨識圖片上的符號是圈還是叉，而CNN的工作，就是每當我們給它一張圖，它就會回報上面的符號是圈或是叉，而要做到這件事，需要**四個主要步驟**。



CNN(特徵)

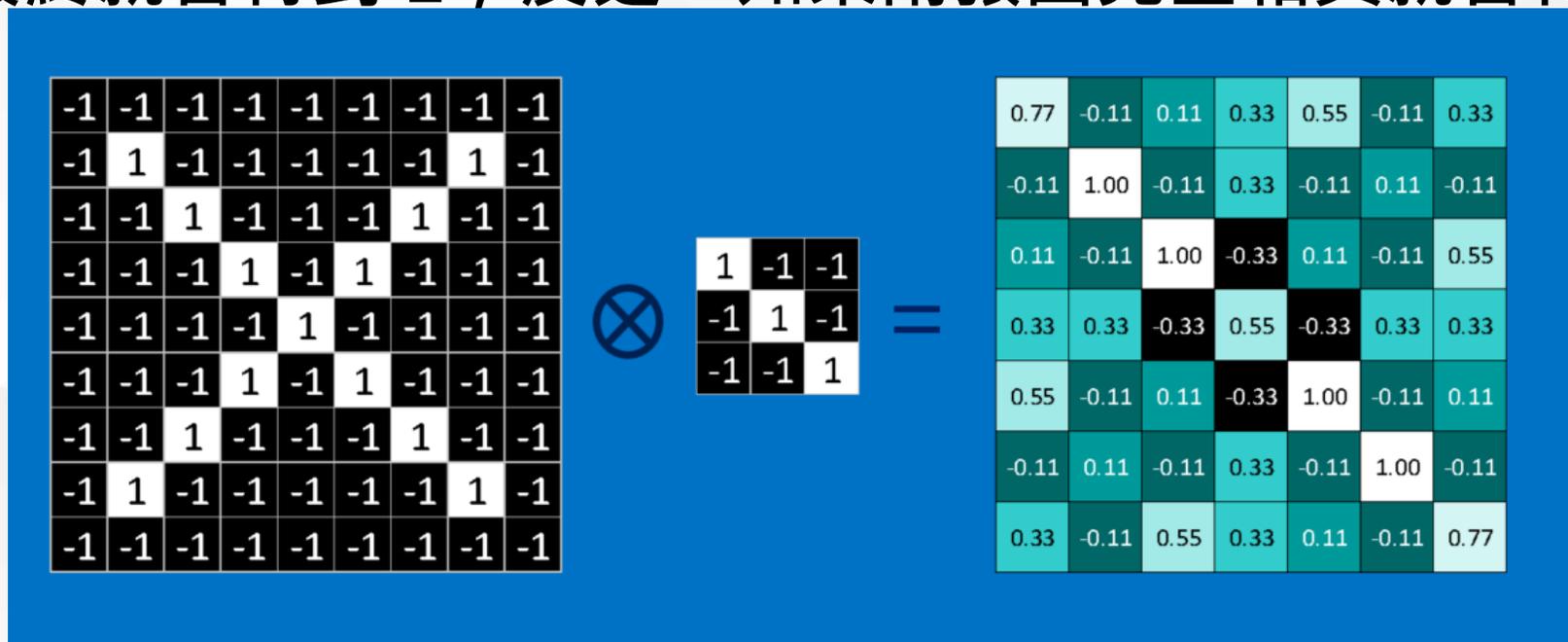
CNN會先比較兩張圖片裡的各個局部，這些局部被稱為**特徵(feature)**，而圖片裡的每個特徵都是一張更小的圖片，而特徵會**捕捉圖片中的共通要素**。

以叉叉的圖片為例，它最重要的特徵包括對角線和中間的交叉。



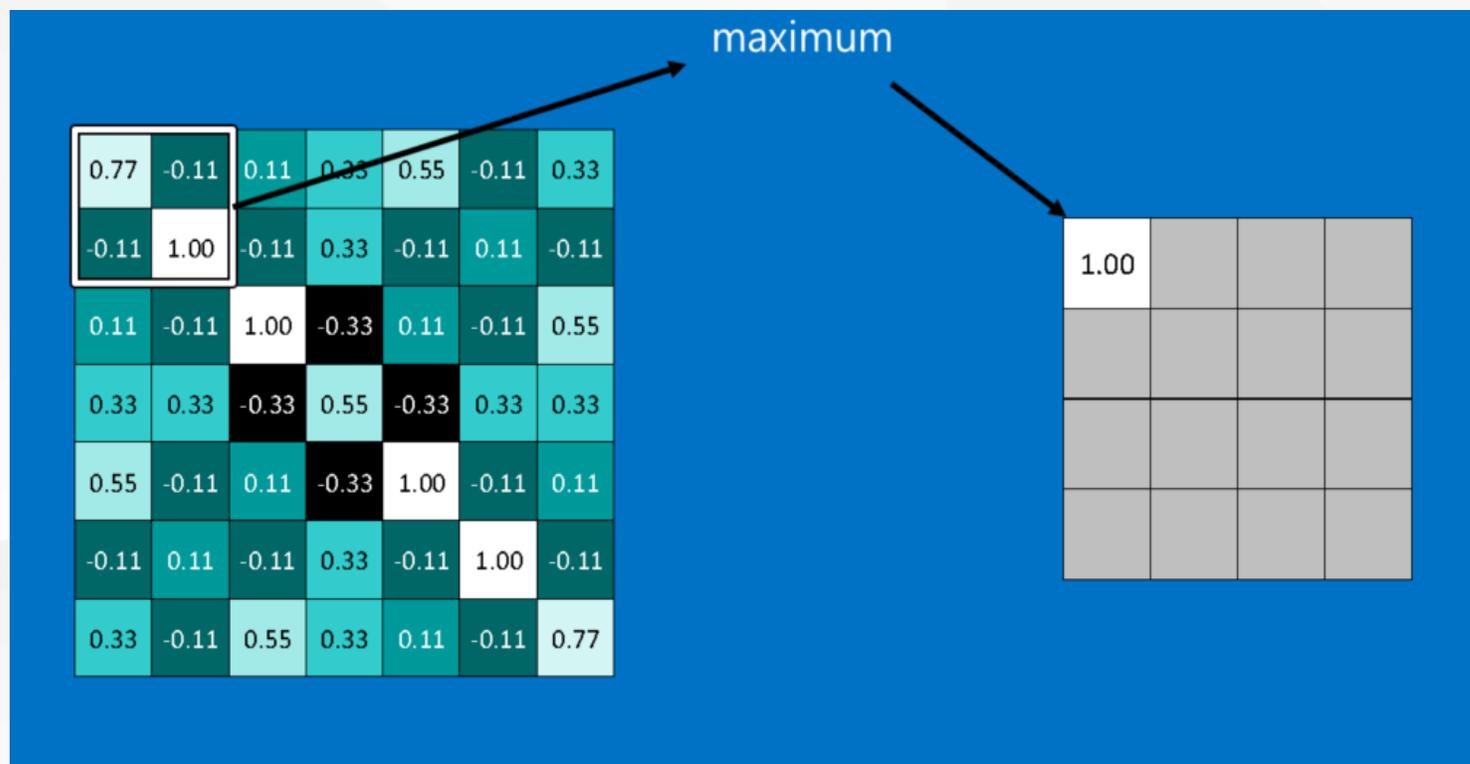
◆◆ CNN (卷積)

為了計算兩張圖片裡有多少相符的特徵，因此創造了一套篩選機制，其數學原理被稱為**卷積 (convolution)**，也就是 CNN 名稱的由來。而要計算特徵的相符程度，只要將兩者各個像素的值相乘，再將總和除以總數，也就是說像素相符的乘積為 1，像素相異的乘積為 -1，如果兩張圖完全相符最終就會得到 1；反之，如果兩張圖完全相異就會得到 -1。



CNN(池化)

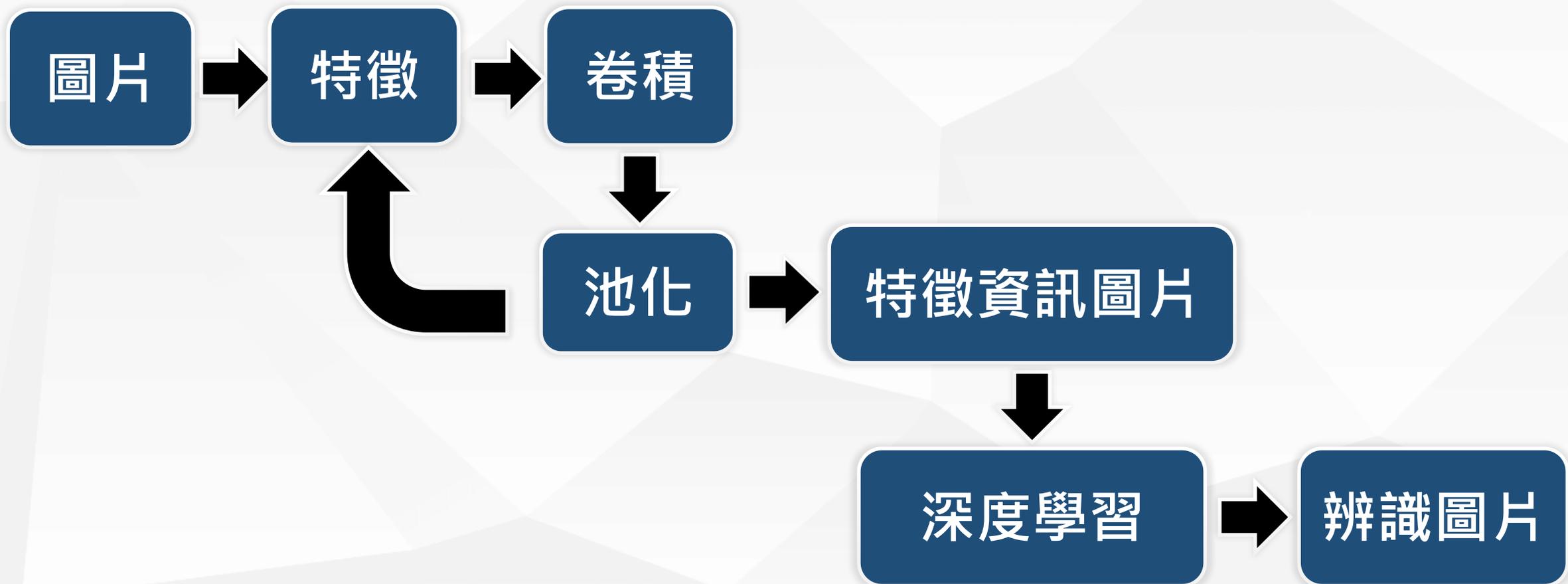
在圖片上選取不同**窗口 (window)**，並在這個窗口範圍中選擇一個**最大值**，經過池化以後，圖片所包含的像素數量會降為原本的四分之一，但還是保留了每個範圍和特徵的相符程度，所以最終我們會得到**更少像素卻不失準確度的圖片**，這也有助於讓後續工作變得更輕鬆。



◆ CNN(深度學習)

原圖在經過**卷積**、**池化**之後會變成一組更小，包含特徵資訊的圖片。接下來，這些圖片還可以再被篩選和壓縮，它們的特徵會隨著**每次處理變得更複雜**，**圖片也會變得更小**。最後，比較低階的處理層會包含一些簡單的特徵，例如稜角或光點；更高階的處理層則會包含一些比較複雜的特徵，像是形狀或圖案，而**電腦會記錄這些特徵**，**用來辨別新的圖片**。

CNN流程概述



◆◆ TPU(張量處理單元)

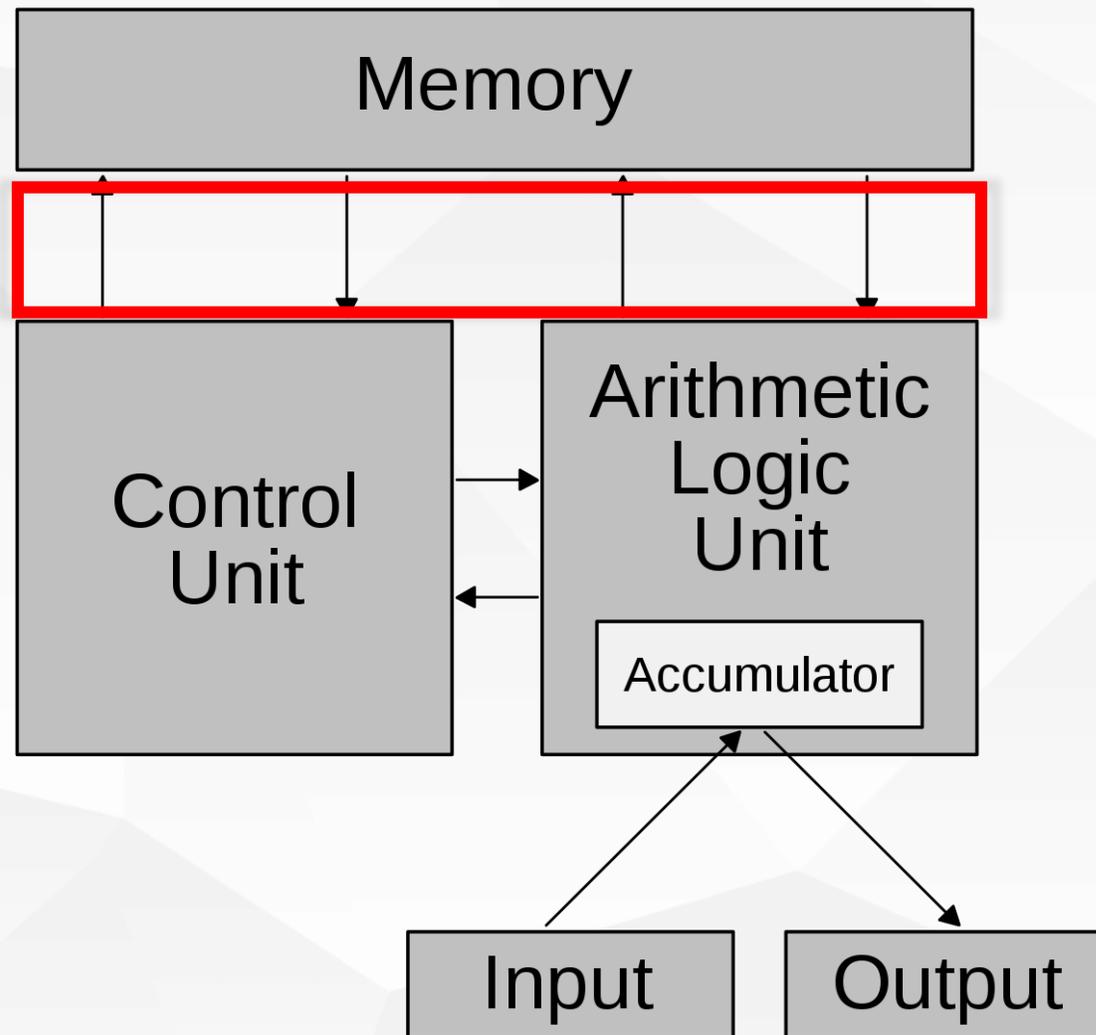
是Google推出的定製化的ASIC 晶片，專門用於機器學習工作，主要是對**馮諾依曼瓶頸**的大幅度簡化。因為該處理器的主要任務是**矩陣處理**，因此放置了成千上萬的乘法器和加法器並將其直接連接，以構建物理矩陣。



◆◆ TPU(張量處理單元)

馮諾依曼瓶頸

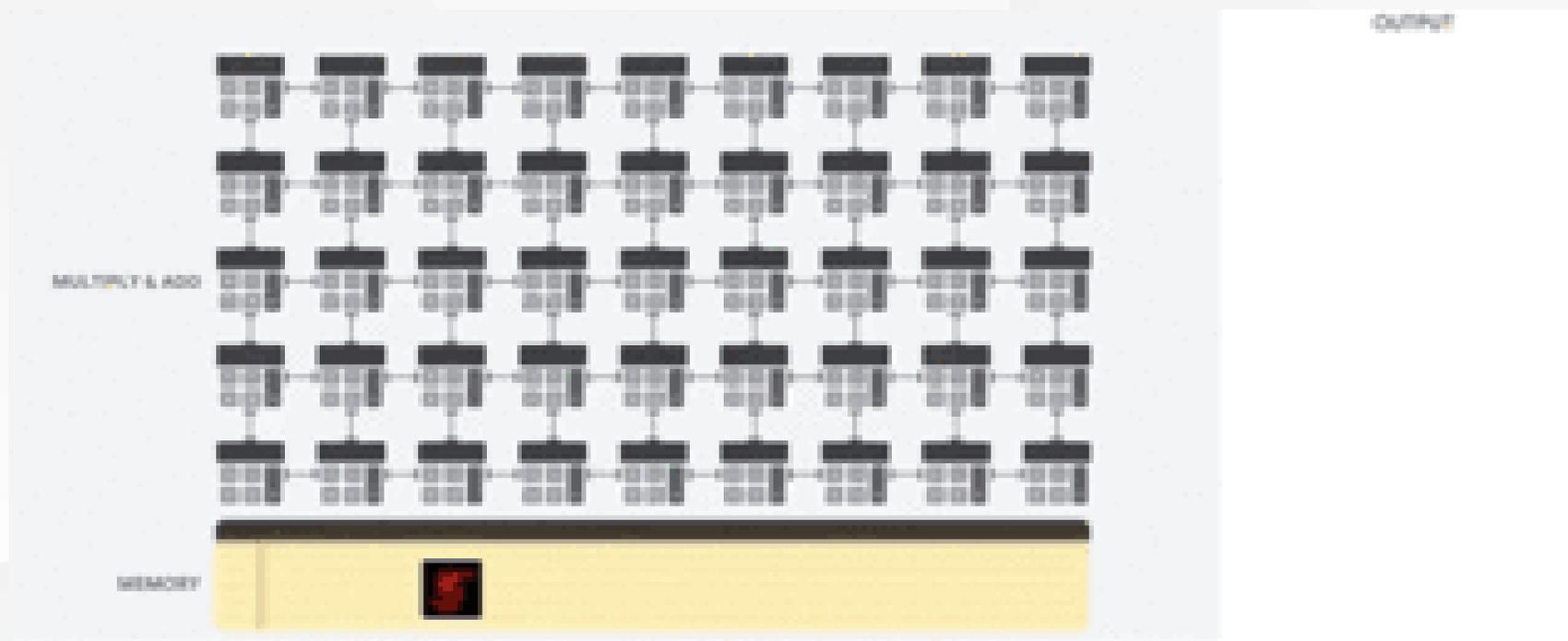
當CPU進行深度學習時，會將每次計算的結果保存，當要取用時，CPU的算術邏輯單元（ALU）只能**一次一個**的執行，因此限制了總體吞吐量，且需要**大量的能耗**，而這種情況被稱為「**馮諾依曼瓶頸**」。



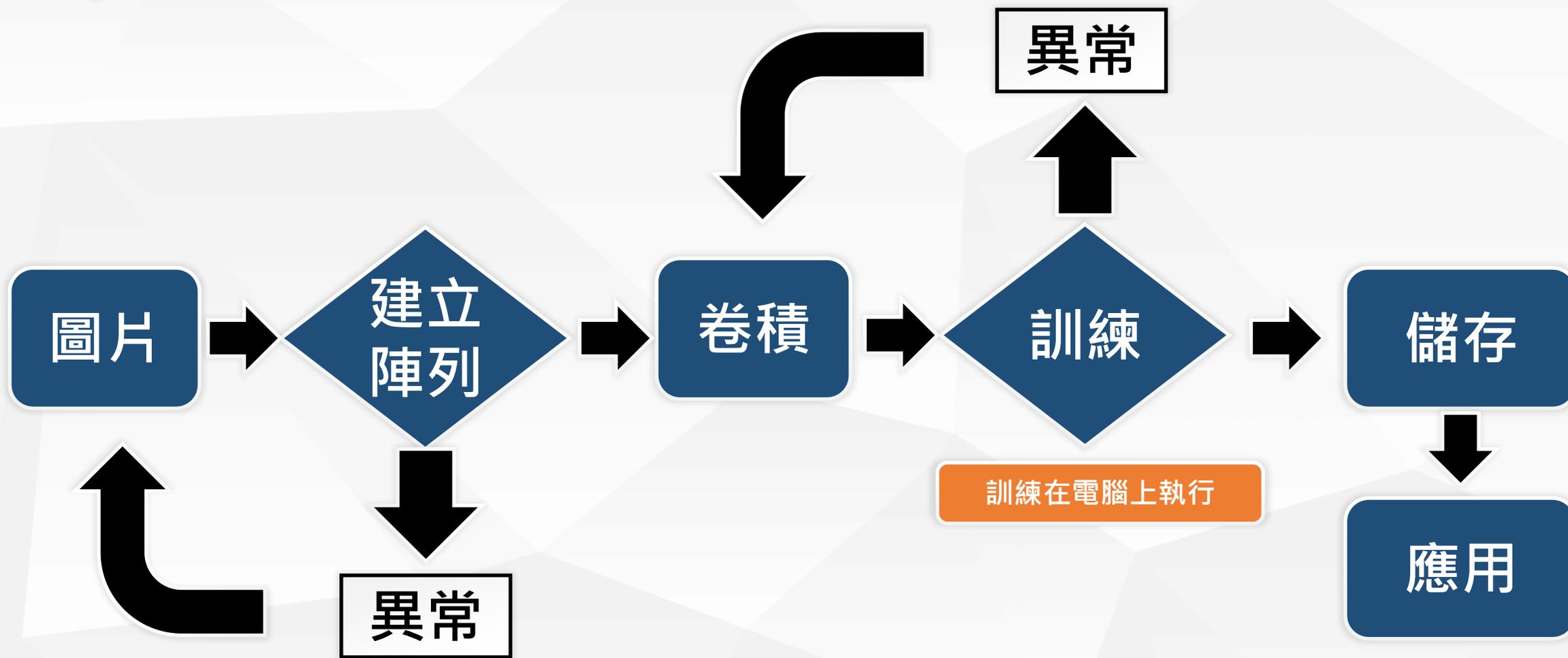
馮諾伊曼瓶頸示意圖

◆ TPU(張量處理單元)

TPU會從內存加載數據，當一個乘法被執行後，其結果將被傳遞到下一個乘法器，並同時執行加法。因此結果將是所有數據乘積的和，而在大量計算和數據傳遞的過程中，**不需要執行任何的內存訪問**，也有**省電和體積小**的優點



學習流程



程式摘要

```
#以样本数量生成乱序的list
permutation = list(np.random.permutation(x_img_train_normalize.shape[0]))
#按照随机生成的顺序重新排列数据集
x_img_train_normalize = x_img_train_normalize[permutation,:,:,:]
#训练集有4个维度，第一个维度指的是样本数量
y_label_train_OneHot = y_label_train_OneHot[permutation,:]
#开始训练
train_history=model.fit(x_img_train_normalize,y_label_train_OneHot,validation_split=0,epochs=70,batch_
```

```
Epoch 1/100
317/317 [=====] - 7s 21ms/step - loss: 1.3941 - accuracy: 0.3281
Epoch 2/100
317/317 [=====] - 6s 20ms/step - loss: 1.3563 - accuracy: 0.3281
Epoch 3/100
317/317 [=====] - 6s 20ms/step - loss: 1.3242 - accuracy: 0.3943
Epoch 4/100
317/317 [=====] - 6s 20ms/step - loss: 1.2761 - accuracy: 0.4637
Epoch 5/100
317/317 [=====] - 6s 20ms/step - loss: 1.2102 - accuracy: 0.5741
Epoch 6/100
317/317 [=====] - 6s 20ms/step - loss: 1.1234 - accuracy: 0.5899
Epoch 7/100
```

人工學習成果

Jupyter Study_Pred Last Checkpoint: an h

File Edit View Insert Cell Kernel Widgets

Code

```
elif np.argmax(y_pred1[0]) == 1:  
    print("Color = White")  
elif np.argmax(y_pred1[0]) == 2:  
    print("Color = Blue")  
elif np.argmax(y_pred1[0]) == 3:  
    print("Color = Red")  
  
if np.argmax(y_pred2[0]) == 0:  
    print("Type = vehicle")  
elif np.argmax(y_pred2[0]) == 1:  
    print("Type = van")  
elif np.argmax(y_pred2[0]) == 2:  
    print("Type = truck")
```

```
Color = Blue  
Type = vehicle
```



人工學習成果

jupyter Study_Pred  Logout

Menu Trusted Python 3

Run Code

```
print("Type = vehicle")
elif np.argmax(y_pred2[0]) == 1:
    print("Type = van")
elif np.argmax(y_pred2[0]) == 2:
    print("Type = truck")
```

Color = White
Type = van

In []:



人工學習成果

jupyter Study_Pred Python 3

Menu Trusted Python 3

Run Code

```
if np.argmax(y_pred2[0]) == 0:
    print("Type = vehicle")
elif np.argmax(y_pred2[0]) == 1:
    print("Type = van")
elif np.argmax(y_pred2[0]) == 2:
    print("Type = truck")
```

Color = Red
Type = truck

In []:



遇到問題

問題

問題一

資料庫建立

1. 蒐集資料的**準確度**和**完整性**

例如:

- 蒐集藍色汽車的照片時，由於淺藍色系過多，導致其無法分辨較深的藍色。
- 車的外型過於相似時，無法準確分辨，如廂型車和休旅車

遇到問題

問題一

資料庫建立

2. 蒐集圖片的數量

- 圖片的數量過低**準確率會降低**
- 圖片過多訓練非常耗時

問題

遇到問題

問題

問題二

樹莓派記憶體不足

1. 由於資料量太大，樹莓派記憶體無法負擔
2. 理論上可透過降低batch size(批處理參數)解決，但過低準確率會下降。
3. 樹莓派效能和記憶體皆不足

解決方法:無解

06

結論



結論

1. 功能因為**樹莓派效能**未能成功整合，但各部分都有完成。
2. 永遠要有**備份**習慣，樹莓派**容易當機**。
3. 安裝套件要非常注意**版本**、安裝的**位置**等等。

07

Q&A



Q&A



The background features a light gray geometric pattern of overlapping polygons. Several dark blue diamonds are scattered across the scene, including a large one on the left and a cluster of smaller ones in the upper right. A thin horizontal line is positioned below the main title.

THE END

THANK YOU FOR LISTENING



2020