

專題名稱：無人車自動剎車系統

實驗室名稱：519

指導教師：陸子強

專題學生：

資工四 B。410822018。樂澤崴。s1082201@gm.pu.edu.tw

資工四 B。410817827。許立劼。s1081782@gm.pu.edu.tw

## ● 前言

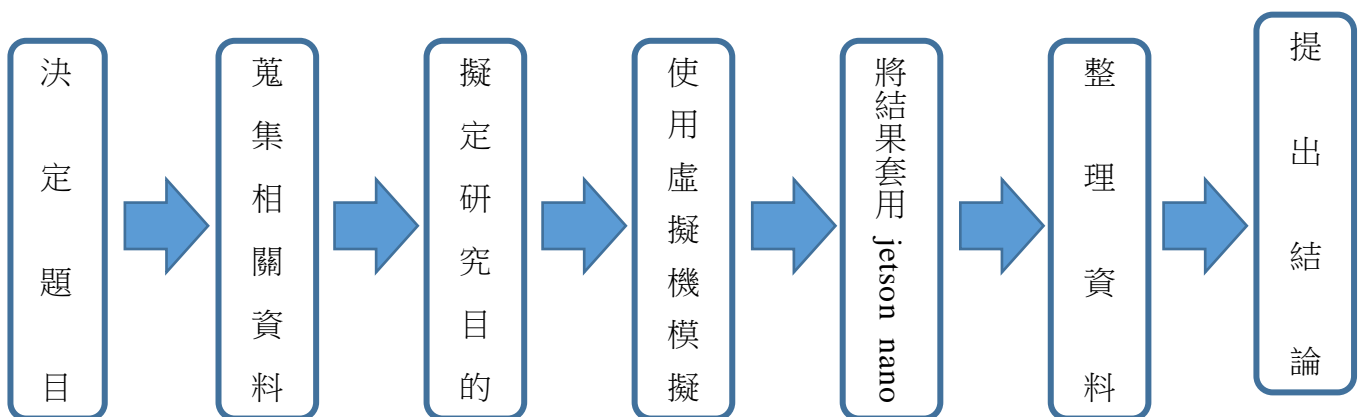
### 一、研究動機

我們認為自動駕駛這項技術的重要性，在目前的需求量也逐漸的遞增，雖然沒辦法做到全自動駕駛，但也朝著全自動駕駛逐步實現，本專題是自動煞車系統，也是屬於自動駕駛技術的一環，所以我們專題針對自動煞車系統，除了距離感測之外我們還能以視覺的角度來介入。在目前自動駕駛系統都是以雷達與感測器來進行環境偵測，而這些設備雖然足以構成一個完善的系統，但是在感測的部分還是有一些例外存在，像是一些路上出現很小的障礙物或是一些交通事故，就無法被立即被偵測或判定。所以我們打算使用深度模型影像辨識來去解決這些問題。不去使用雷達或感測器去偵測環境，而是使用影像去加強環境感測能力及穩定性。

### 二、研究構思

- (一) 運用過去製作 AI 循跡自走車的相關知識，深入分析探討是否可以使用類似的方法來完成本專題。
- (二) 研究如何使用 ROS 來傳送命令。
- (三) 當 ROS 可順利傳送命令，接著研究如何將命令由 jetson nano 傳送到 Arduino。
- (四) 決定實驗場域，收集要使用的影像，並將影像透過 labeling 做標註。
- (五) 將標註好的影像使用深度模型做訓練。
- (六) 將訓練完成的影像放進影像辨識的程式，找出 bounding box 並算出面積。
- (七) 將影像辨識的程式與 ROS 的程式做結合，並將值傳到 Arduino 以完成指定功能。

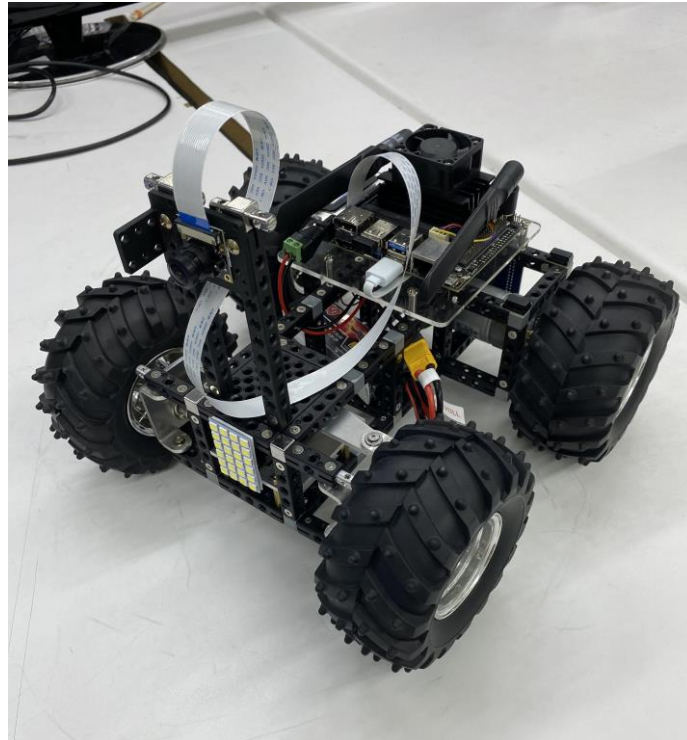
### 三、研究流程



圖（一）研究流程（研究者自繪）

## ● 研究背景

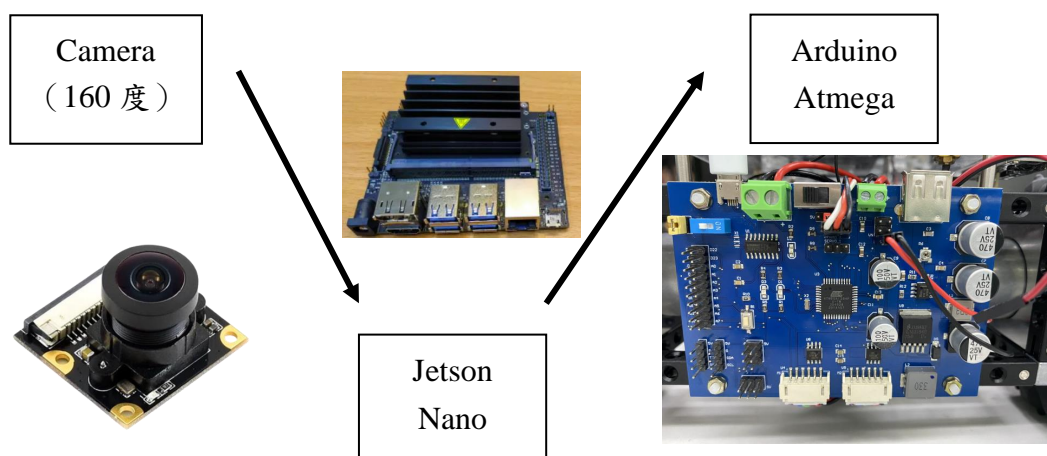
### 一、硬體架構



圖（一）實驗之無人車

圖（一）為本專題使用之無人車，硬體架構如圖（二）所示主要元件說明如下：

- （一）以 jetson nano 作為主要運算平台。
- （二）Camera 使用 160 度廣角，負責偵測目標物的影像。
- （三）使用 Arduino 控制直流馬達前進停止。
- （四）使用 ROS 傳送 Jetson Nano 的指令給 Arduino 控制馬達。



圖（二）無人車控制架構圖

## 二、Jetson Nano

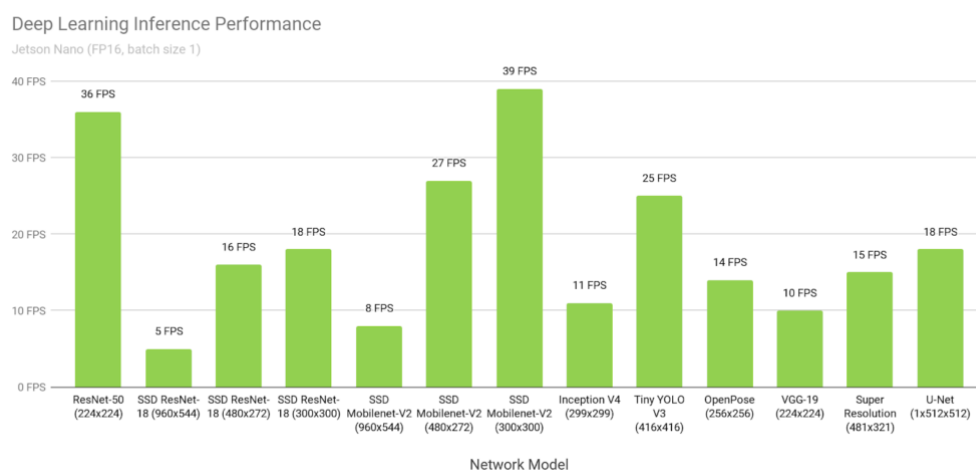
由 NVIDIA 推出的 Jetson Nano 總共有兩個版本，分別是 2GB 跟 4GB，本專題使用的是 4GB 版本，規格如圖（四），另外由圖（五）可以看到，Jetson Nano，在大部分的深度學習演算法都有不俗的運算速度，而且也支援許多深度學習演算法。



圖（三）Jetson Nano

開發套件技術規格	
GPU	配備 128 個核心的 NVIDIA Maxwell™
CPU	四核心 ARM® A57 @ 1.43 GHz
記憶體	2 GB 64 位元 LPDDR4 25.6 GB/秒
儲存空間	microSD (未含記憶卡)
影片編碼器	4Kp30   4x 1080p30   9x 720p30 (H.264/H.265)
影片解碼器	4Kp60   2x 4Kp30   8x 1080p30   18x 720p30 (H.264/H.265)
連線能力	Gigabit Ethernet, 802.11ac wireless*
相機	1x MIPI CSI-2 connector
顯示器	HDMI
USB	1 個 USB 3.0 Type A、2 個 USB 2.0 Type A、1 個 USB 2.0 Micro-B
其他	40-pin header (GPIO, I2C, I2S, SPI, UART) 12-pin header (Power and related signals, UART) 4-pin fan header*
機械規格	100 mm x 80 mm x 29 mm

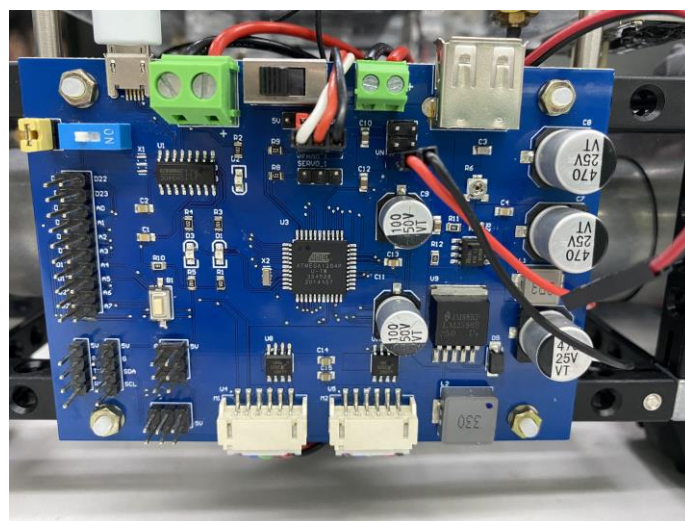
圖（四）Jetson Nano 規格表



圖（五）

### 三、 Arduino

本專題使用之 Arduino 並非常見的 Arduino UNO，而是使用 ATmega1284P 比起 UNO 有更多的腳位可以運用，未來如果要擴充模組，就不用擔心腳位不夠用的問題，雖然程式儲存空間及記憶體沒有比 ATmega2560 還多，但是考慮到 ATmega2560 的體積太大，所以還是用 ATmega1284P 比較合適。



圖（六）Arduino ATmega1284P

### 四、 IMX219 Camera

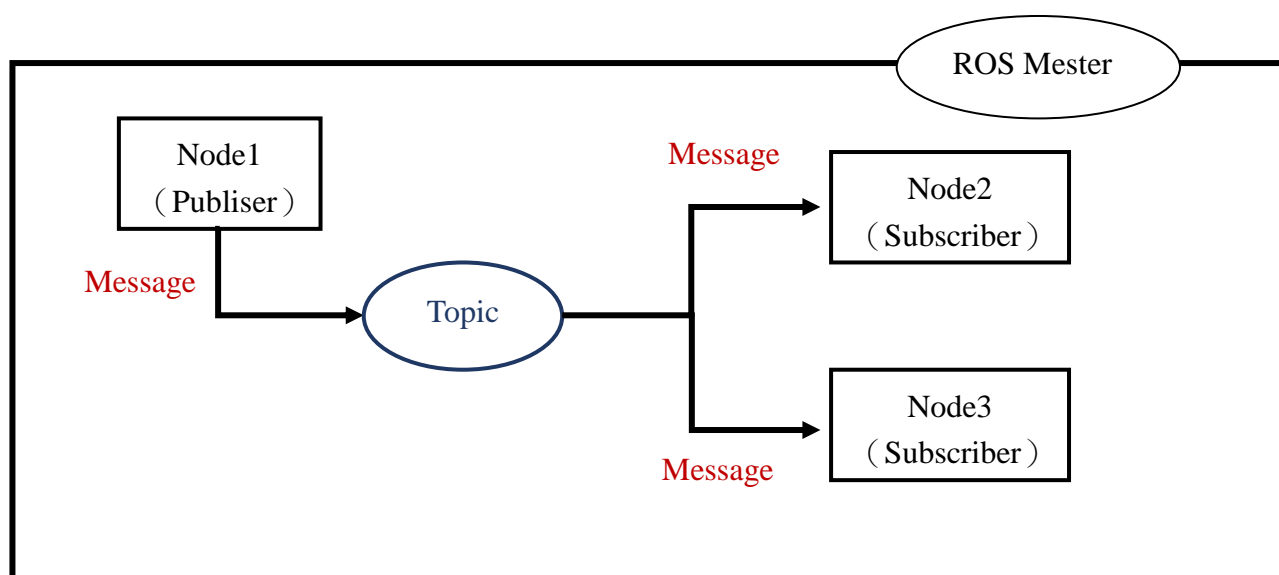
IMX219 是 CSI 攝像頭，原本是使用 USB 攝像頭，但考慮到 USB 攝像頭的像素太低，畫面會模糊不清，影響到影像的辨識結果，所以選擇使用更高像素的 CSI 攝像頭，以解決畫面不夠清晰的問題，另外為了能讓視距增加，選用了 160 度的廣角鏡頭，已加寬視距。



圖（七）IMX219

## 五、 ROS

ROS 的全稱為 Robot Operating System，顧名思義就是機器人作業系統，但是它其實不是作業系統，它是需要安裝在 Linux 的環境上，它更像是為機器人的元件進行溝通的框架，採用了分散式架構各程式獨立運行，彼此間可傳遞訊息，傳送訊息的機制採用類似訊息佇列遙測傳輸（Message Queuing Telemetry Transport, MQTT）通訊協定其通訊架構如圖（八）所示。



圖（八）ROS 通訊架構

- （一）ROS Master：主要負責節點的初始化與查詢，並且管理所有節點的 Topic 通訊。
- （二）Node：在 ROS 框架中獨立運行的程式，各節點擁有不同的功能。透過發佈(Publish)或訂閱(Subscribe)話題(Topic)與其他節點溝通。
- （三）Message：各節點通訊時所傳遞或讀取的資料。消息的資料型態除了傳統的整數、字串、浮點數等之外也可傳輸影像、點雲(Point Cloud)、光雷達地圖等較為複雜的資料。
- （四）Topic：在節點與節點通訊時需產生一個 Topic，發佈者節點可以將消息發佈到話題上，訂閱者節點則訂閱話題從而獲取消息。

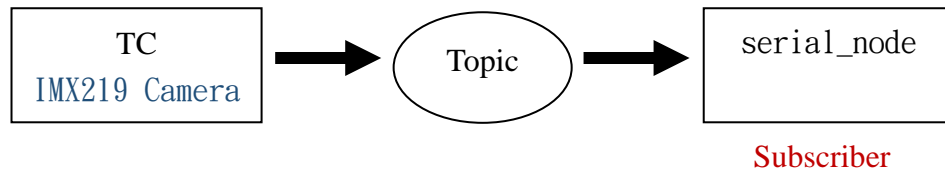
## ● 研究方法



## 一、使用 ROS 通訊

(一) 由 Jetson Nano 使用 IMX219 Camera 獲取影像，執行 TC 節點，該節點進行影像辨識，得出目標物之面積，將其面積轉換成相對應的命令，發佈到 Topic 中。

(二) 執行 Arduino 的 serial\_node 節點，訂閱 Topic，將 TC 節點發送的命令傳入 Arduino，使 Arduino 根據命令控制馬達。



(圖九) 目標物偵測 ROS 架構

## 二、模型訓練

本專題使用 Single-Shot MultiBox Detector (SSD) 深度學習框架，對目標車子進行訓練並輸出模型，其流程如圖 (十)

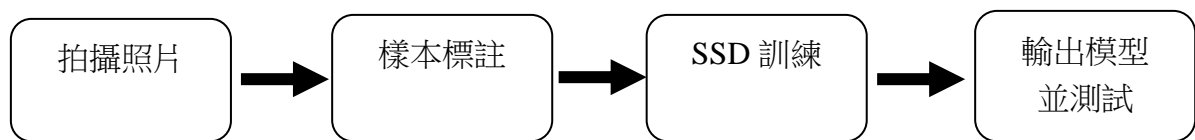


圖 (十) 模型訓練

## 三、樣本標註

本專題使用 Labellmg 對所有照片進行標註，如圖 (十一) 所示，標註完成後將標註檔儲存至 Annotations 資料夾中。



圖(十一)LabelImg 照片標註

#### 四、模型訓練

使用 SSD 模型進行訓練，總共有四百多張照片，總共訓練了三個多小時，訓練 2500 輪

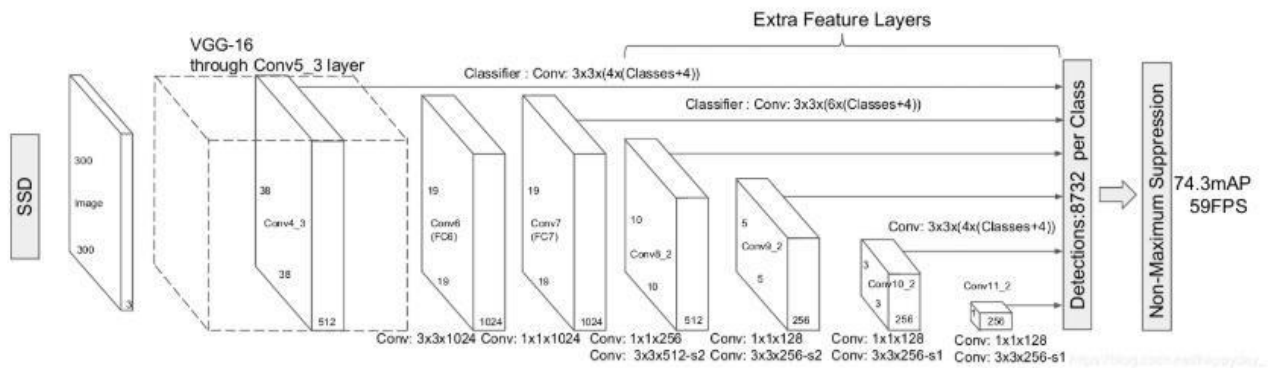


圖 (十二) SSD

```
import torch
device=torch.device('cuda') # 'cuda'/'cpu' , import torch
num_classes=2 # 物件類別數+1(背景)
train_size=428
valid_size=0
batch_size=4
learning_rate=0.0001
threshold=0.5 # 錨框匹配為物件/背景的閾值，參考值=0.5
epochs=2500
```

圖 (十三) 參數

#### 五、即時影像運算

在 Jetson nano 上使用 SSD 深度模型，載入訓練好的模型即時進行辨識並將結果回傳，本方案雖然實作上直觀容易，但執行上效能低落，SSD 深度模型辨識準確度高，速度不快，約只有 3FPS，主要原因是受制於 Jetson Nano 硬體效能，處理速度差強人意，車輛行駛時，永遠都是顯示約 7 秒前的畫面，辨識速度完全無法用於自動煞車系統，所以我們在程式中修改 CSI 攝像頭的 FPS 為 3，最後成功讓延遲時間大幅縮短，可用於自動煞車系統。

#### ● 使用環境

本專題之實驗場域為學校五樓之電梯口，利用此空間來模擬十字路口的交通情況，本專題之無人車可依路口交通情況來判斷前方車輛是否造成危險，若不安全則煞車，反之繼續前進。



## ● 成本分析

名稱	支出
無人車	\$40000
記憶卡	\$600
筆電	\$39000
人事費	\$12000
隨身碟	\$229
Jetson 教科書	\$432
CSI Camera	<b>\$1000</b>
<b>總共</b>	<b>\$93261</b>

圖（十四）支出表

## ● 結論及未來發展

本專題所提出自動煞車系統並非使用感測器來完成，而是使用影像辨識的方式，雖然排除巨大延遲問題，但還是無法達到毫秒級的延遲，未來可以加上感測器，與影像辨識一起使用，讓系統判斷的更精確，甚至利用這些方式，能夠達成自動跟車的功能，讓系統更加完善。