

靜宜大學資訊工程學系

畢業專題報告書

作品名稱：AI 循跡自走車

指導老師：陸子強

專題學生：

資工四 B 姓名：張晁銘 學號：410817788

資工四 B 姓名：簡嘉佑 學號：410817869

資工四 B 姓名：蔡泊諺 學號：410828519

資工四 B 姓名：吳哲維 學號：410817796

中 華 民 國 一 一 一 年 十 二 月 六 日

目錄

靜宜大學資訊工程學系.....	1
畢業專題報告書.....	1
一、前言.....	3
二、文獻探討.....	4
(一) Machine Learning - Transfer Learning (遷移式學習)	4
(二) CNN model - VGG16.....	5
三、實作成果.....	7
(一) 環境.....	7
(二) 成果.....	7
三、結論及未來展望.....	10
(一) 結論.....	10
(一) 未來展望.....	10

一、前言

本組在 1101 學期時，有幸與學長一起參加 TIRT 競速自走車的比賽，在準備比賽時，對於競速自走車產生有了基礎的了解及興趣，而在學長的畢業專題時，也跟隨學長的腳步，一起在以 JetsonNano 為主體的自走車的這部分進行研究，因此本組計畫在這些研究基礎上，繼續研究如何利用深度學習的方式，使此自走車能夠更穩定、快速，最後本組也希望能夠參加之後的競速自走車比賽並且奪得好名次。

本組先前是利用影像辨識來分析路況，但此方式容易受到場地因素影響，所以在行駛時，系統非常容易做出錯誤的判斷，因此本組計畫利用深度學習的方式進行分析路況。

二、文獻探討

(一) Machine Learning - Transfer Learning (遷移式學習)

機器學習 (Machine Learning)，近年愈來愈盛行的領域，它是屬於 AI 與計算機科學的一個分支。深度學習 (Deep Learning) 則是機器學習的衍生物，簡單來說，兩者明顯的差別在於是否需要人為干涉，機器學習訓練方法需要人工輔佐，但往往可以與深度互換使用。

在機器學習、深度學習往往會遇到一些問題，比如處理 A 領域的分類問題時，缺少足夠的訓練樣本。同時，與 A 領域相關的 B 領域，擁有大量的訓練樣本，但 B 領域與 A 領域處於不同的特徵空間或樣本服從不同的分佈，此時就可以利用遷移式學習 (Transfer Learning)。

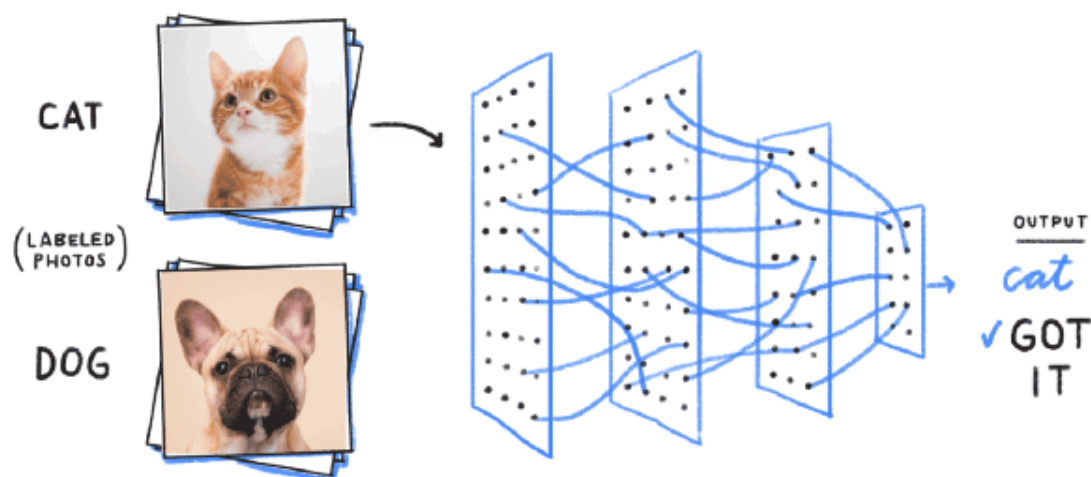
遷移式學習，為機器學習、深度學習的其中一種形式，簡單來說，就是把 B 領域中的知識遷移到 A 領域中來，提高 A 領域分類效果，不需要花大量時間去標註 A 領域數據，好比喻人類可以將以前的學到的知識應用於新的問題，更快的解決問題或取得更好的效果。

(二) CNN model - VGG16

常見的 Deep Learning model 主要分為 MLP(ANN)、RNN、CNN，下圖為三者差別。

	MLP	RNN	CNN
Data	Tabular data	Sequence data (Time Series, Text, Audio)	Image data
Recurrent connections	No	Yes	No
Parameter sharing	No	Yes	Yes
Spatial relationship	No	No	Yes
Vanishing & Exploding Gradient	Yes	Yes	Yes

而本組在這次使用的 model 為 CNN，CNN 是當今最流行的模型之一，該模型被認為比 ANN、RNN 更強大。CNN 會自動學習過濾器而無需明確提及，這些過濾器有助於從輸入數據中提取正確和相關的特徵。



與其他兩者不同的是，CNN 能從圖像中捕獲「空間特徵」。空間特徵是指圖像中像素的排列以及它們之間的關係。它們幫助我們準確地識別對象、對象的位置以及它與圖像中其它對象的關係。

CNN 衍伸出的模型眾多，如下圖，本次專題則使用 VGG16 Net 作為訓練工具。選擇其該原因為 VGG 結構非常簡單，更易於上手，且精準度也不低。

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88	79.0%	94.5%	22.9M	81
VGG16	528	71.3%	90.1%	138.4M	16
VGG19	549	71.3%	90.0%	143.7M	19
ResNet50	98	74.9%	92.1%	25.6M	107
InceptionV3	92	77.9%	93.7%	23.9M	189
InceptionResNetV2	215	80.3%	95.3%	55.9M	449
MobileNet	16	70.4%	89.5%	4.3M	55

- Top-1 表示只預測一次且正確的機率。
- Top-5 表示預測五次只要一次猜對就算正確的機率。
- Size：記憶體的最高佔據量。
- Parameters：參數的數量，愈多就須計算愈久。
- Depth：filters 的數目。

三、實作成果

(一) 環境

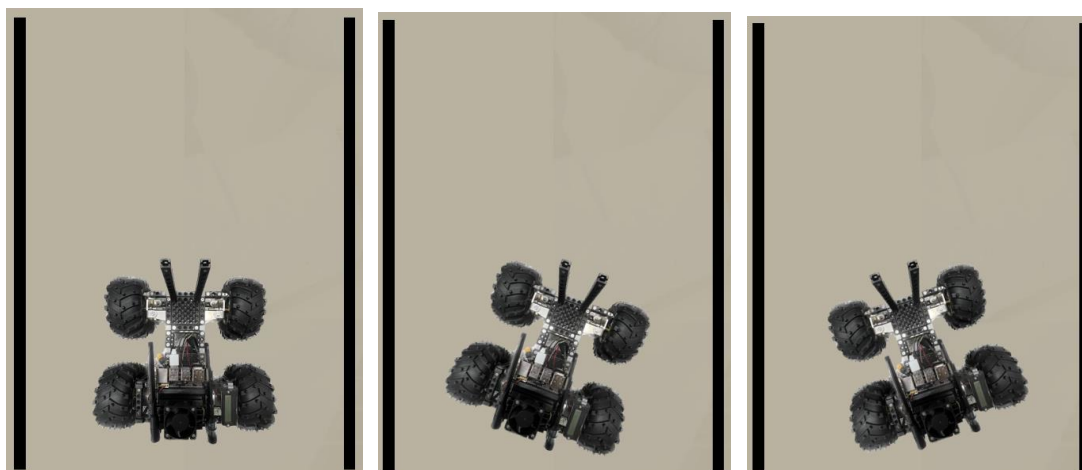
1. Ubuntu 18.04 - Jetson Nano Developer Kit SD Card Image
2. Python 3.6.9
3. OpenCV 4.1.1
4. Pytorch 1.8.0
5. Pytorch - torchvision 0.9.0
6. ROS 1.14.13

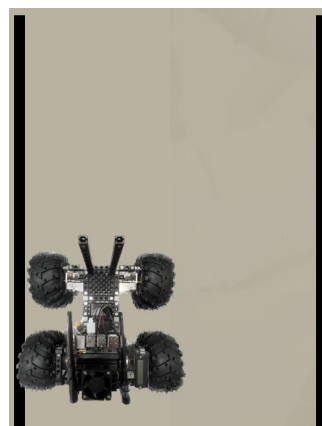
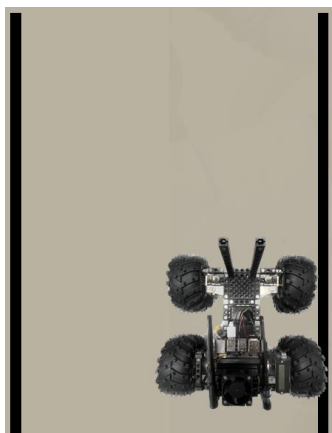
(二) 成果

在影像訓練結果，我們統整了四種模型訓練的版本，如下圖。

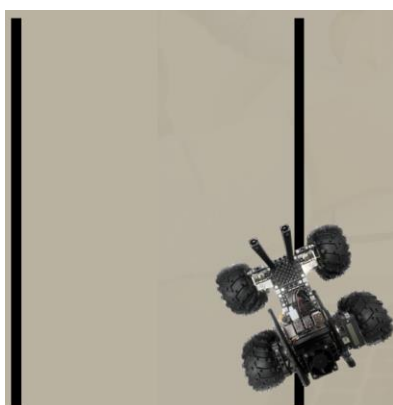
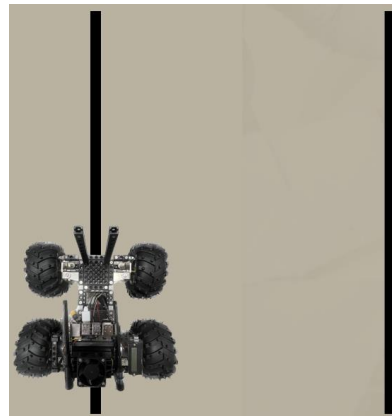
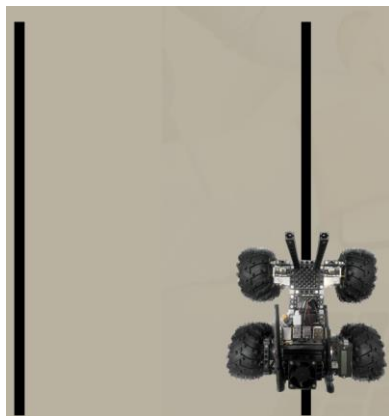
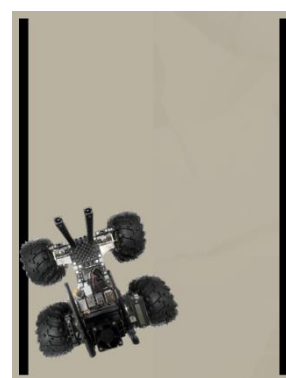
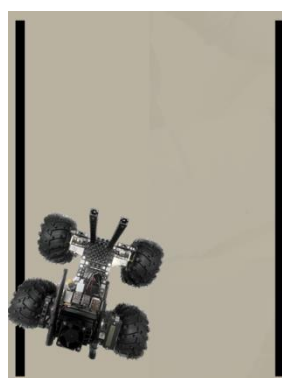
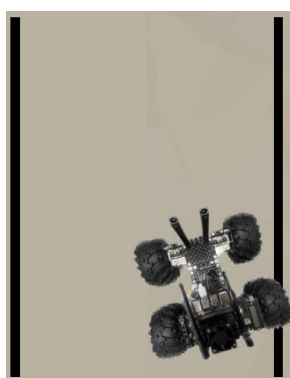


V1.0 為主要的版本，情況內容為在線內行駛。

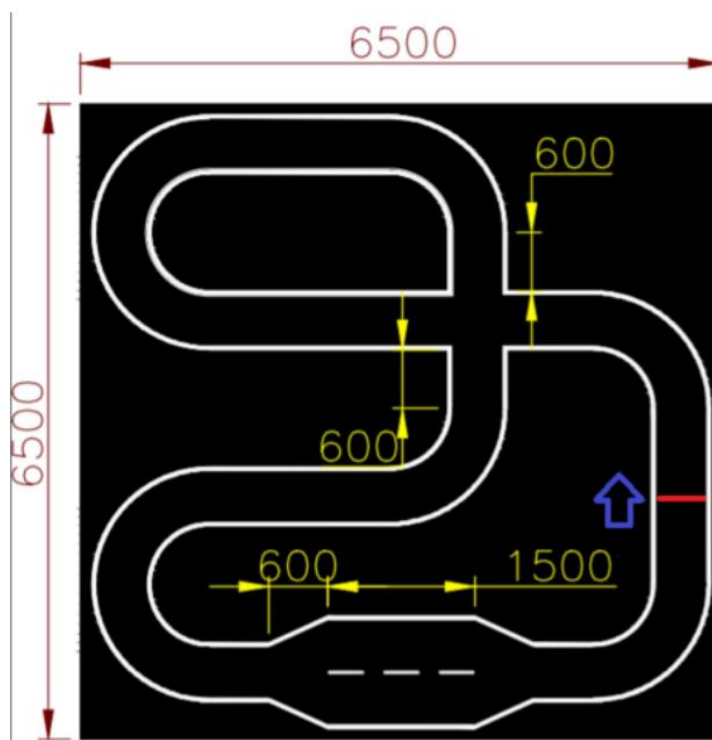




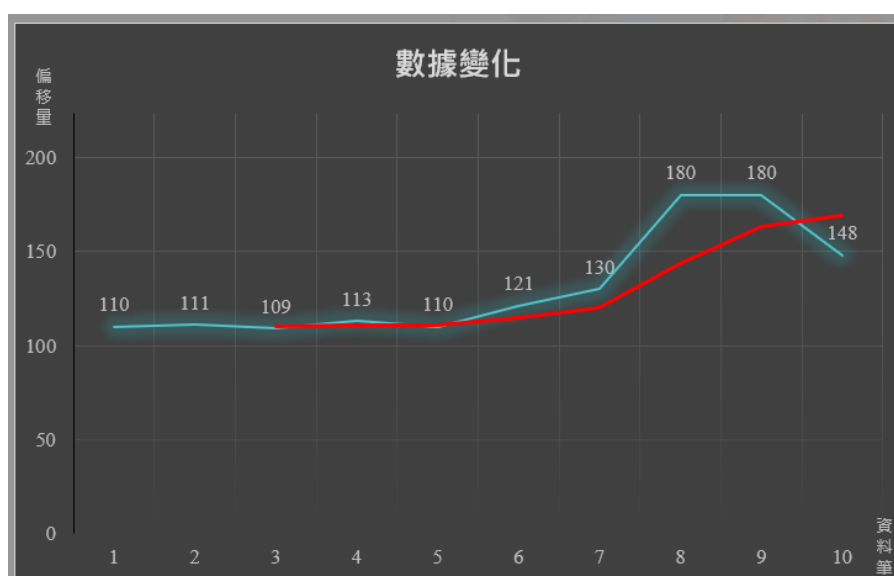
V2.0 ~ V4.0，情況內容為壓線、即將行駛出線外的情況。



根據我們行駛後得出的結果，在得知賽道狀況的情況下，如下圖，僅需要使用 V1.0 版本即可讓 AI 無人車順利行駛完整個賽道，賽道內容包含了直走、轉彎、岔路、虛線四種情況，且能夠不受外界因素影響，例如：賽道反光、賽道旁物體移動……等。



為了使 AI 無人車行駛更順利，利用了數值平滑化的形式，讓轉彎時的偏移量，不會因為突然遇到極大數值時，導致突然轉彎幅度過大而出線外。經過多項測試後，這個方法確實能使無人車更順利的行駛在線內。



三、結論及未來展望

(一) 結論

為了想要減少場地因素對自走車帶來的問題，於是利用深度學習來建立循跡模型，在這過程中，花費了我們許多心思。

深度學習對我們來說是一個全新的領域，基本上都是從頭學起，模型的建立、自定義的 dataset、模型的訓練，還必須結合影像處理，不僅如此，從一開始的燒錄 ISO 檔並建立開發環境，就開始在 debug 了。在過程中，還遇到 Linux 環境損壞，但我們還是順利克服了。

且不光是軟體的部分，在硬體の設定上也消耗了我們不少精力，最後還需要讓軟硬體結合，參數、模型不斷的一直做調整，直到最後終於成功順利行駛，從無到有，完成專題真的是慢滿滿的成就。

(一) 未來展望

在未來規劃，希望不僅僅是循跡，還希望能加入更多項目，例如：號誌辨識、防碰撞……等。並優化模型，使其能夠更為精準，除了消耗的硬體資源能夠更低，並期望參加下一屆的全國 TIRT 競賽，不再是使用影像處理，而是完全使用深度學習行駛整個賽道，獲得冠軍。