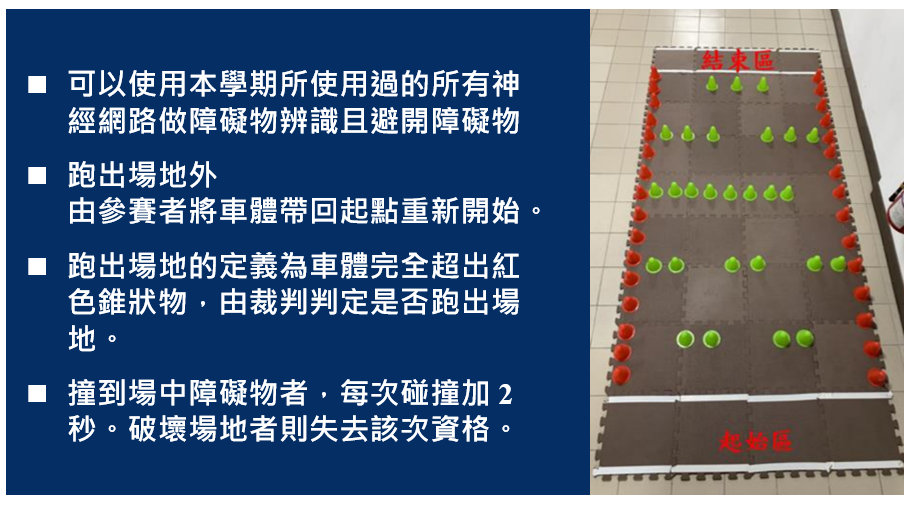
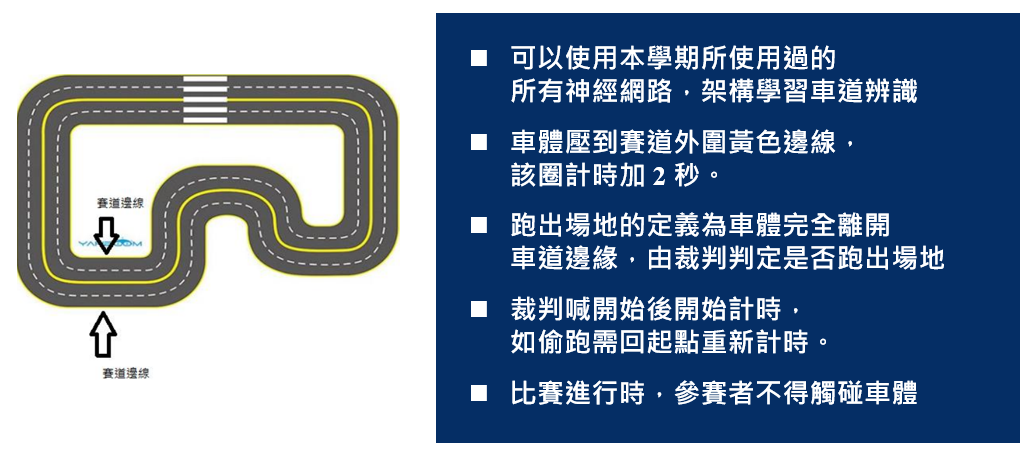
# 嵌入式智慧影像分析與實境界面

## 專案情境

* 避障賽



* 競速賽



## 二、定義問題

* 避障賽
  1. 第一個想到的是，跑到會遇到不少情況，如果全部都要label一次，怕會overfitting。
  2. 如果不要Overfitting的話會不會反而變成underfitting，因為跑道沒有那種很清晰的一定左轉或右轉的情況或邏輯存在。
  3. 如果Jetbot照著正常行走邏輯來行走的話會不會不小心回頭，反而走向起點了。
  4. 左右轉的時候會不會撞到車身旁邊的角錐。
  5. 馬達速度如何調控才不會因為地形元素而停止行走。
  6. 如何不撞上角錐。
* 競速賽
  1. 一開始看到賽道想到的第一個問題是要如何解決地面反光的問題。
  2. 選擇怎麼樣的模型比較適合競速賽。
  3. 避免模型overfitting。
  4. 怎麼避免超出賽道。
  5. 超出賽道後怎麼知道要往回走、如何往回走。

## 三、方案構思、解決方法

* 避障賽
  1. 先以基本邏輯十座為主，慢慢看Jetbot需要認得的情況還少那些再慢慢加上去。
  2. 如果有兩種一樣或相似的影像輸入，但是行走決策不相同時，看是要改變行走軌道或是故意繞遠路來改變切入正確軌道的方向。
  3. Jetbot的鏡頭調低一點比較容易看到車身附近的角錐。
  4. 測試不會被地形困住的馬達轉速。
* 競速賽
  1. 靠前處理把反光處理得比較不會影響到模型訓練，或是調整鏡頭找看看比較不會反光的角度，並用量角器將鏡頭角度記錄下來。
  2. 嘗試使用ResNet18, 34, 50，觀察記憶體用量及模型穩定度。
  3. 先以能判斷出主要前進方向的情況為主，再以實際模型表現來增加標記資料。
  4. 調整馬達轉速、延遲、敏感度、fps等等參數，一邊讓Jetbot在跑道上行走，一邊動態調整參數看看Jetbot實際轉彎和執行狀況，與小組成員討論出較好的參數配置。

## 四、個人心得

期末專案可以用的模型概括了整個學期所教的所有深度學習、強化學習等等所有模型，看到其他小組之間也都採用不同的模型，感覺期末就是一個模型大比拚，看哪個模型比較厲害，當然模型的厲害與不厲害也取決於訓練測資抓的適合不或適合該模型，我們這組原本競速友享用DQN，但是有一些延遲和覺得讓他自己學習可能會學到一些不太重要的資訊，所謂「變髒」，所以最後我們還是採用ResNet，以此來主控想要讓模型學習的側資，避障方面則是蠻前面就決定採用AlexNet來做為競賽模型，一開始是想採取V字型跑道，由起點的左邊斜切過去碰到右邊的紅錐後，看到紅錐就左轉，剛好左轉完會碰到綠錐之間的出口，通過綠錐後繼續斜切直行，直到碰到終點線的左邊，整個路線就會是很簡單的「起點-往右斜前-碰到紅錐往左轉反彈-往左斜前-終點」，不過後來遇到一些道路邏輯的問題，所以改成還是直行進入障礙賽道，以貼著右邊紅錐，遇到綠錐就左轉繞過在右轉回去貼著紅錐的策略；至於競速賽的部分我們一開始是完完整整的label了一整圈，每前進一節Jetbot車身就Label一次當下情況，也是因此估算出賽道大約是80個Jetbot車身，也就是說，每重新Label一次，我就跪著移動了Jetbot 80次，但自衛後覺得這樣有點overfitting，加上助教有告訴我們可以試試看標記一半的賽道就好，所以我們就用比較少標記資料的方式去訓練模型，果然效果好多了，很感謝助教的幫忙，我的膝蓋應該獻給助教而不是賽道；最後以上續兩種方式訓練出來的魔性在比賽的時候表現的都還算滿意，不枉費熬夜訓練模型，沒關係小石頭(Jetbot)，你想走的時候再走就可以了。

## 五、貢獻比例

|  |  |  |
| --- | --- | --- |
| 徐紹崴 | 劉文揚 | 謝狄烽 |
| **資料蒐集**  **策略設計**  **策略實現** | **資料蒐集**  **資料標記**  **策略設計** | **資料蒐集**  **策略設計**  **設計路線** |