



# 基於機器學習與自主導航之多色球分類

## 搬運機器車

專題生:通訊四 楊雅筑 指導教授：劉立頌教授

### 專題簡介與目標

本專題設計並實作一台可在固定空間內自主搜尋、辨識並搬運多色物體的智慧機器車。

系統利用攝影機擷取影像，透過卷積神經網路 (CNN) 辨識球體顏色，並驅動鏟子夾取與分類放置，機器車依顏色導航至指定目標點，持續執行任務直到完成分類。

核心目標：

- 自主導航：空間移動與路徑規劃
- 物體偵測：搜尋與定位物體
- ML 分類：CNN 辨識顏色
- 抓取搬運：機構控制鏟子夾取
- 目標放置：依顏色送至終點

### 實驗系統架構

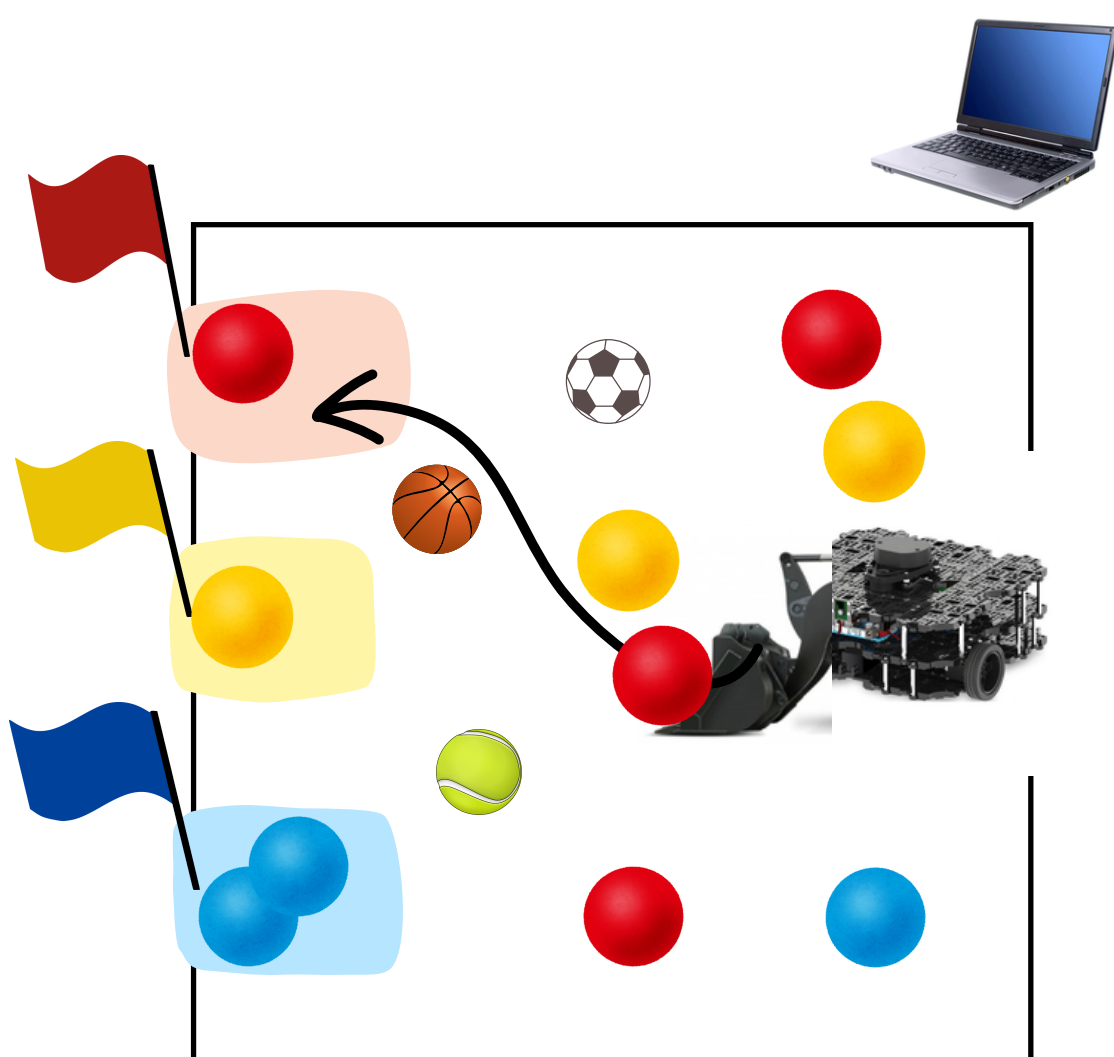
系統採主從式 (Master-Slave) 架構，分離高運算與即時控制以提升效能。

PC (Master)

- 執行 ROS 核心節點與 CNN/YOLO 推論
- 接收影像串流並進行分類與路徑規劃
- 向 Raspberry Pi 發送控制指令 (前進、轉向、夾取)

Raspberry Pi (Slave)

- 負責硬體 (turtlebot3 camera 雷達掃描 鏟子) 控制與即時影像擷取
- 驅動馬達與伺服馬達執行命令



### 目前進度

階段一：模擬與模型建構 (已完成)

- ROS Noetic 安裝與操作
- Gazebo 模擬完成 Mapping、Path Planning、Navigation
- 初步 CNN v1 訓練 (紅/藍/黃球各 60 張訓練、20 張驗證)，Accuracy > 90%

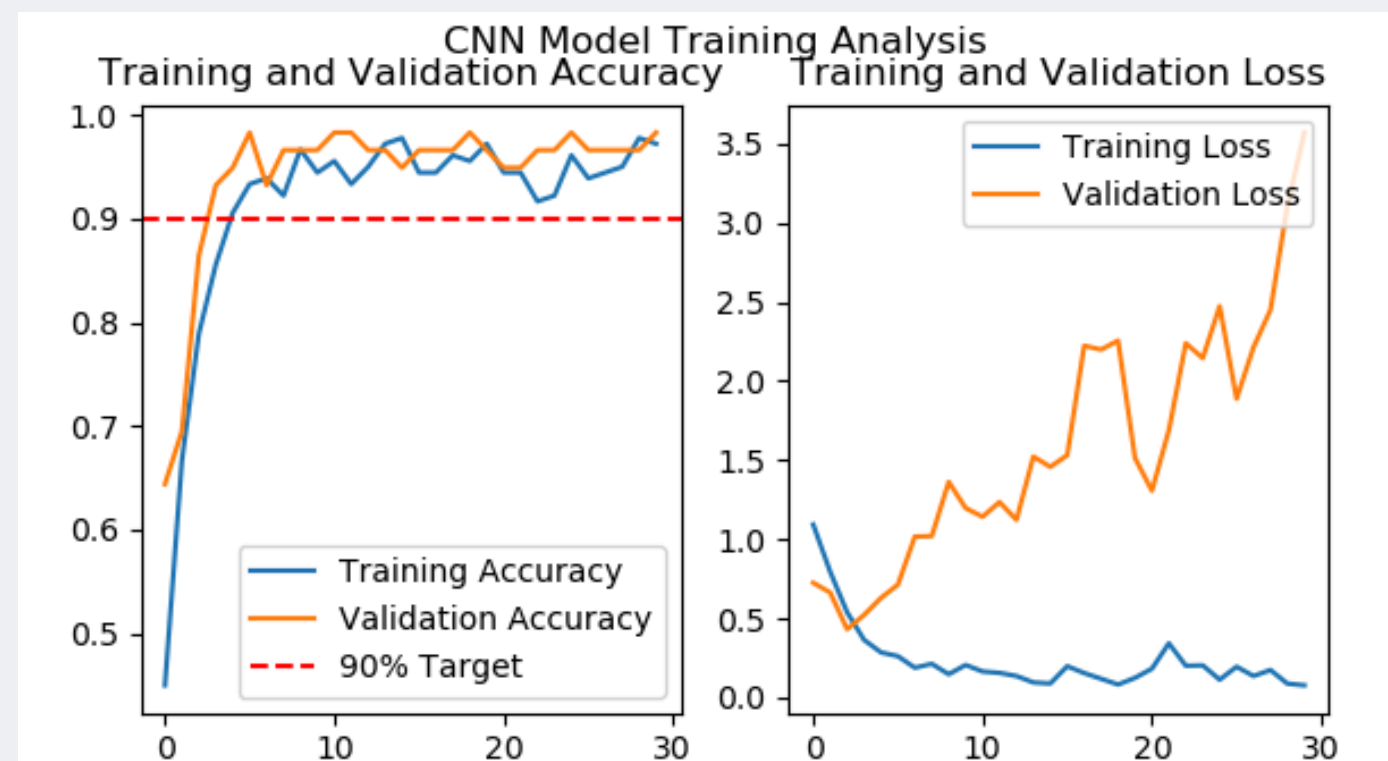
階段二：模型分析與硬體整合 (進行中)

- 分析 v1 模型結果 (發現過擬合)
- 攝影機串流測試
- 收集真實影像資料以進行微調

當前挑戰與分析：v1 模型過擬合

現象：高準確度但高 Validation Loss，出現「猜對但信心低」或「高信心猜錯」情形。

原因：模型過度學習網路圖片特徵 (背景、光線)，泛化能力不足。



解決方案與後續步驟

步驟一：真實資料收集 (以攝影機擷取多張不同光源、背景下的紅/藍/黃球影像)

步驟二：模型微調 (載入 v1 權重，以低學習率 (0.0001) 在新資料上再訓練 10-20 epochs)

步驟三：部署 v2 模型 (生成更穩健的 v2.h5，整合至 ROS 節點即時辨識)

### 未來規劃

1. 硬體整合：製作與安裝鏟子結構，並完成伺服馬達的安裝與控制。
2. 演算法整合：撰寫 ROS 節點，整合 CNN (v2) 辨識結果與 move\_base (導航) 功能，讓車子能自主開向辨識到的物體。
3. 系統實作：完成「辨識->前往->夾取->導航至目標點->釋放」的完整任務循環。
4. (進階)：若要辨識更複雜的物體 (非單色物體)，將導入 YOLO 物件偵測模型，取代目前的 CNN 分類器。