**專題技術分析報告：YOLO 模型選擇與 Raspberry Pi AI Kit 硬體整合**

**一、研究背景**

本專題旨在開發一套部署於大型客車之即時影像辨識系統，偵測車輛內輪差與視野死角區域之行人、自行車等潛在危險物件，並透過鏡頭與邊緣運算裝置實現高即時性與高準確率的辨識功能。

**二、影像辨識規劃**

**（1）任務需求**

* 同時支援多鏡頭（左、右、後）
* 每支鏡頭需達 5~10 FPS 推論速度
* 必須能辨識 person、bicycle、motorbike 等常見目標
* 系統需即時提供辨識結果並能擴充為即時警示邏輯

**（2）ROI 設計**

* 根據鏡頭位置，定義特定 ROI（Region of Interest）
* 僅針對 ROI 區域中的物件進行警示，提高準確度並減少誤報
* 使用 OpenCV 搭配相對比例設定與多邊形支持

**三、YOLO 模型選擇分析**

考量 Raspberry Pi 搭配 AI 加速器（Hailo-8 / Hailo-8L）之效能條件，針對 YOLOv5/v6/v8 系列模型進行比較分析：

| **模型** | **架構類別** | **推論速度** | **準確度** | **模型大小** | **ONNX/Hailo 支援** | **綜合評估** |
| --- | --- | --- | --- | --- | --- | --- |
| YOLOv5n | Nano | 非常快 | 較低 | ~3MB | ✅ | ⭐⭐⭐⭐☆ |
| YOLOv5s | Small | 快 | 較高 | ~7MB | ✅ | ⭐⭐⭐⭐⭐ |
| YOLOv6n | Nano | 非常快 | 中等 | ~4MB | ✅ | ⭐⭐⭐⭐☆ |
| YOLOv8n | Nano | 快 | 中高 | ~6MB | ✅（需轉ONNX） | ⭐⭐⭐⭐☆ |
| YOLOv5m/v8m | Medium | 慢 | 高 | >20MB | ❌（過重） | ⭐⭐☆☆☆ |

**建議模型：YOLOv5s 為主力部署模型，YOLOv8n 為未來升級選項，YOLOv6n 可用於快速部署驗證。**

**四、硬體規劃與搭配建議**

**（1）Raspberry Pi 5B（8GB） + Hailo AI Kit**

| **組件** | **效能概述** |
| --- | --- |
| Raspberry Pi 5B | 4核心 64-bit + PCIe Gen3 支援 |
| Hailo-8L | 13 TOPS，適用多路輕量推論 |
| Hailo-8 | 26 TOPS，適合高精度、高頻率推論任務 |

**（2）整體配套建議**

* 若僅處理單路或低頻辨識，可選擇 Hailo-8L
* 若需同時處理多鏡頭、搭配高 FPS 推論，建議升級至 **Hailo-8**
* 所有模型需轉換為 ONNX 並導入 Hailo 軟體堆疊（HailoRT、TAPPAS）

**五、總結**

本系統將以 Raspberry Pi 5 搭配 Hailo-8 為主要硬體平台，選用 YOLOv5s 模型進行即時影像辨識任務，並結合多鏡頭架構與 ROI 區域偵測邏輯以實現即時性與準確性的最佳平衡。後續亦可根據實際推論效能與警示需求，切換至 YOLOv8n 進行升級部署。