

結合YOLO資料處理與演算法決策的即時行人號誌系統

Smart Pedestrian Signals Powered by YOLO Data

Processing and Algorithms

指導教授: 張大緯

專題成員: 林宏睿、陳品嘉、陳珮綾、駱穎玟

開發工具: C、C++、Python、YOLOv9、
libwebsocket、HTML、JavaScript、Visual
Studio Code、AnyDesk

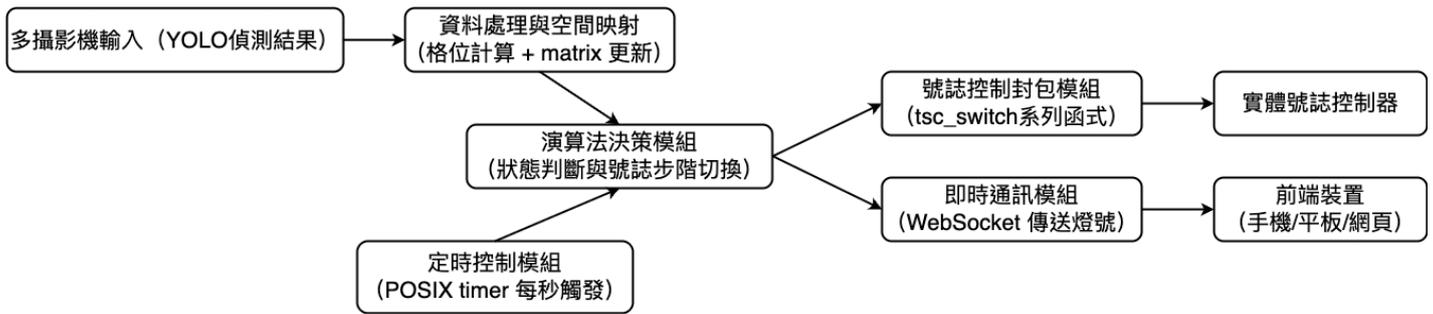
測試環境: Ubuntu (20.04.6 LTS)、晟隆號誌控
制器SLTC-9308C

一、簡介

本研究旨在解決目前行人號誌系統未能即時感知環境與行人狀態，進而導致等待時間長、燈號難配合實際人流等使效率及使用感低落的問題，因此嘗試實現一套具備即時反應能力之智慧型行人號誌控制系統，核心關注於多攝影機環境下的資料整合處理、號誌決策演算法設計、通訊模組建構及前端互動呈現。整體系統架構採模組化實作，並以 YOLOv9的行人偵測資訊為輸入，以此建構能自主運作且具擴充性的控制架構。

與傳統以固定時相或人為觸發為基礎的號控方式不同，本系統著重於即時環境偵測與自動決策演算，整合以下核心技術：

- **資料處理與空間映射**：針對多來源攝影機資料進行標準化處理，透過投影計算與格位映射機制，形成可供分析之行人格位矩陣。
- **演算法決策機制**：設計以人數累積、通行比例與時間條件為依據之狀態機制，動態調整號誌步階（綠燈、綠閃、紅燈），並依據不同場景做策略切換。
- **即時通訊架構**：採用 WebSocket 技術實作主動式訊號傳送，確保後端決策結果可以即時更新至前端顯示裝置。
- **定時控制與狀態同步**：採用 POSIX timer 機制驅動後端演算法模組，以 1 秒為週期定時觸發核心邏輯，確保演算法運作具時間一致性與狀態可預測性，並降低延遲造成之決策錯誤風險。



二、測試結果：

本次實驗模擬實際道路場景，觀察系統於行人穿越期間之號誌決策與訊號輸出流程。由圖可見，當攝影機偵測到有行人進入斜邊斑馬線，並其通行距離已超過道路長度的30%時，系統即依據演算法條件 (alg1) 綠燈階段判斷邏輯) 觸發狀態切換，進入綠燈閃爍階段 (CS = 1)。

此過程由定時模組每秒觸發一次 runPedestrianSignalControl()，持續統計通過人數與行人分布。當達成轉換條件後，系統不僅將訊號狀態更新為 GreenFlash且透過 WebSocket 模組即時廣播至前端裝置，同時呼叫 tsc_switch_strategy()、tsc_switch() 等控制函式，實際下達命令切換後端號誌控制器之步階設定，完成實體號誌狀態同步切換。

圖二 Log 檔圖片顯示，系統成功傳送 "GreenFlash" 訊息予 client，並同步顯示對應的綠人動畫圖示。此結果驗證系統能根據實際行人狀態進行即時反應，並完成從感知、判斷、控制到前端顯示的完整號誌控制流程。



(圖一)路口行人即時影像

```

ESC [msignal packet tx: SWITCH_STRATEGY
aa bb 17 ff ff 0 e 5f 10 16 0 aa cc 37
ESC [1;4;32m2025-05-19 15:08:16
ESC [msignal packet tx: SWITCH
aa bb 18 ff ff 0 f 5f 1c 0 0 0 aa cc 23
ESC [1;4;32m2025-05-19 15:08:16
ESC [msignal packet tx: 5F4C
aa bb 19 ff ff 0 c 5f 4c aa cc 71
ESC [1;4;32m2025-05-19 15:08:16
ESC [m[WS] Sent "GreenFlash"
  
```

控制封包輸出
已向號誌控制器送出
SWITCH_STRATEGY、SWITCH、
5F4C 指令

WebSocket輸出:
行人燈號狀態切換為
GreenFlash



(圖二)Log檔輸出&WebSocket即時通訊輸出