

2025 年【科學探究競賽-這樣教我就懂】

國中組 普高組 技高組 成果報告格式

題目名稱：電腦視覺「駕道」

一、摘要

本研究使用 OpenCV 進行影像處理，以偵測車道線並計算中線來控制車輛行駛方向。首先，讀取圖片並擷取下半部作為分析範圍，透過 Canny 邊緣檢測強化輪廓，並使用霍夫變換找出可能的車道線。接著，根據左右車道線計算平均斜率與截距，過濾不合理的線條，確保車道偵測的穩定性。透過已確定的車道線，可推算出中線，並計算車輛與中線的偏移量，進而決定轉向修正，如向左或向右微調方向。最後，程式將計算出的線條與中線繪製於原圖上，並進行影像疊加以提升可視化效果。影像處理步驟可依不同圖片大小調整，確保車道偵測與車輛控制的精準度，使其能適應不同環境。

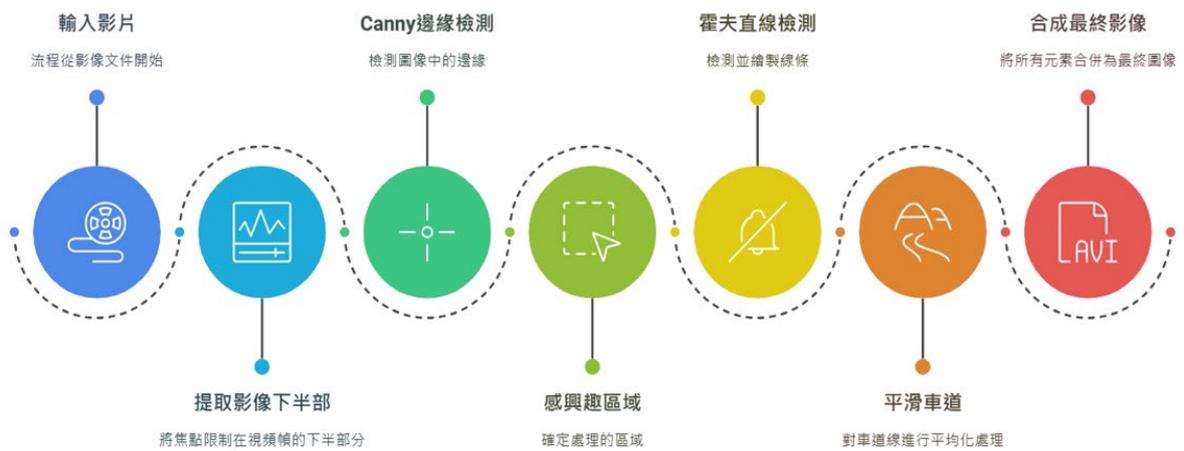
二、探究題目與動機

隨著自動駕駛技術的快速發展，車道線偵測作為實現車輛導航與安全駕駛的重要基礎技術，備受關注。在現實應用中，車道線的準確識別對於車輛的穩定行駛具有決定性影響。然而，由於環境光線變化、影像雜訊及其他車輛干擾，車道線的偵測面臨諸多挑戰。

三、探究目的與假設

本研究旨在聚焦單車道線的穩定偵測與中心線提取，提出基於影像處理與數據平滑技術的解決方案。透過高斯模糊降低影像雜訊，結合 Canny 邊緣檢測準確標定可能的邊緣特徵，並使用霍夫轉換檢測車道線，進一步分類為左側與右側車道線。為了提升偵測的穩定性，研究設計了一套平滑機制，結合歷史數據來減少車道線抖動和不連續問題。

四、探究方法與驗證步驟



圖：1 車道線流程圖

車道線步驟

(一)提取下半部

目的：提取影像的下半部分，因為車道線通常位於影像的下半部



圖：2 提取後的下半部



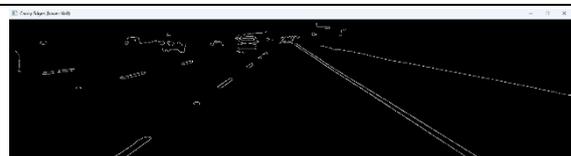
圖：3 車道原圖

(二)執行 Canny 邊緣檢測

目的：檢測影像中的邊緣，為後續的霍夫檢測提供輸入。



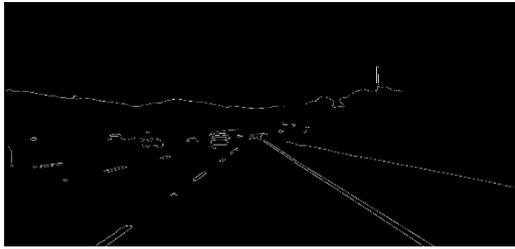
圖：4 車道原圖



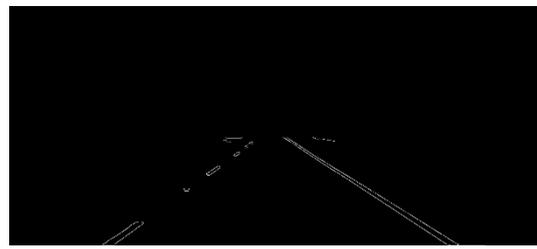
圖：5 Canny 邊緣檢測

(三)感興趣區域

目的：遮罩處理，只保留車道線所在的三角形區域，進一步減少無關資訊。



圖：6 Canny 邊緣檢測原圖



圖：7 Canny 邊緣檢測，感興趣區域

(四)霍夫直線檢測

目的：從 Canny 偵測出來的線，並加以篩選最合適的直線。

程式碼：

```
lines = cv2.HoughLinesP(canny_image, rho=2, np.pi / 180, 100, np.array([]), minLineLength=40, maxLineGap=1)
```

參數解釋：

$\rho = 2$: 霍夫空間中的距離解析度 (2 像素)。

$\theta = \text{np.pi} / 180$: 角度解析度 (1°)。

$\text{threshold} = 100$: 累積計數的閾值，僅當計數超過此值時，該點才會被判定為直線。

$\text{minLineLength} = 40$: 檢測直線的最短長度。

$\text{maxLineGap} = 1$: 允許的最大間隙，用於連接斷裂的線條。

(五)平均化和平滑車道線

目的：將檢測到的多條線段分為左右車道線，計算平均斜率與截距，並進行平滑處理。

詳細步驟：

1. 分類左右車道線：

- 通過檢測斜率的正負，將直線片段分為左側和右側車道線。
- 斜率為負值時，為左側車道線；斜率為正值時，為右側車道線。

2. 計算平均斜率與截距：

- 使用 `np.average()` 計算分類後每組線條的平均斜率與截距

3. 進行平滑處理：

如果有過去的數據，使用滑動平均公式：

$$\text{new_line} = \text{smoothing_factor} \times \text{previous_line} + (1 - \text{smoothing_factor}) \times \text{current_line}$$

4. 生成車道線端點：

根據平均斜率與截距，計算車道線的起點與終點座標。

(六)合成最終影像

目的：將車道線疊加回原始影像，生成可視化結果。

車輛控制邏輯

(一) 邏輯敘述：

車道控制邏輯的目的是根據車道線的斜率來決定車輛的行駛方向。

當系統檢測到車道線後，會根據車道線的左右偏差來控制車輛進行以下動作：

- 向右轉 (如果車道線偏向左側)
- 向左轉 (如果車道線偏向右側)
- 繼續前進 (當車道線位於中間)
- 後退 (當車道線偏離太多時)

(二) 具體步驟：

1. 計算車道線的斜率：

程式會根據車道線的兩端點來計算斜率。每條車道線會有一個斜率，該斜率決定了車道線的方向。

2. 計算車道偏差：

比較車道線的斜率（兩側車道線的斜率），來判斷車道偏差。偏差越大，表示車道線偏離的程度越高。

(三) 決定行駛方向：

根據車道線的偏差，控制車輛進行轉向、前進或後退。具體邏輯如下：

1. 轉向邏輯：

- (1) 偏差大於某個範圍時，車輛會轉向對應的方向（例如偏右則向左轉，偏左則向右轉）。
- (2) 偏差小於範圍內，車輛保持直行。

2. 前進與後退邏輯：

- (1) 當車道線在畫面中間附近且偏差較小時，車輛將繼續前進
- (2) 當車道線偏離過遠，車輛會後退並重新調整方向。

(四)控制邏輯

Move(a · b · c)函數是車到控制邏輯的核心，決定車輛的動作：

```
def move(a, b, c):
    t = (a[0][0] - b) * (a[0][0] - c)
    if t < -8000 and t > -26000:
        print("turn right")
        turn_right(0.05)
        stop(0.3)
    elif t > 8000 and t < 26000:
        print("turn left")
        turn_left(0.05)
        stop(0.3)
    elif t > -8000 and t < 8000:
        print("keep going")
        if t < 3000 or t > -3000:
            forward(0.3)
        else:
            forward([0.15])
        stop(0.25)
    else:
        print('reverse')
        reverse(0.2)
        stop(0.25)
        if t < 0:
            turn_right(0.1)
        else:
            turn_left(0.1)
        stop(0.3)
```

圖：8 控制函數程式

1. 邏輯解釋：

(1) 轉向邏輯：

- $t = (a[0][0] - b) * (a[0][0] - c)$ ：這一行計算的是車道線的斜率差異，根據偏差來判斷是否需要轉向。
- 若偏差超過某個範圍（例如小於 -8000 或大於 8000），則判斷為需要轉向，分別會進行 右轉 或 左轉。

(2) 前進邏輯：

- 當車道線斜率的偏差在 -8000 到 8000 之間時，車輛會保持在車道內並繼續前進。

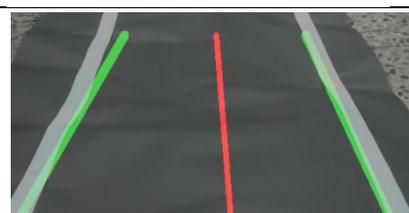
這裡的 3000 和 -3000 是調整的閾值，用來微調前進的速度。

(3) 後退邏輯：

- 當車道線偏差過大（t 值小於 -26000 或大於 26000），車輛會選擇後退，並根據偏差再進行調整方向。



圖：9 測試移動



圖：10 車道線

五、結論與生活應用

本研究基於 **OpenCV 影像處理技術** 和 **Raspberry Pi 控制** 的自動駕駛小車，透過 **車道偵測與動態方向調整**，實現小車的自主導航。在影像處理方面，我們採用了 **Canny 邊緣偵測** 與 **Hough 變換** 來精確辨識車道線，並透過 **高斯模糊** 來平滑影像、降低雜訊干擾，確保邊緣檢測的準確性。

(一)自主導航系統的應用

透過攝影機擷取即時影像，並透過 **影像處理技術 (Canny + Hough 變換)** 找出車道線後，系統計算 **車道中心與攝影機中心的偏移量**，並透過 **GPIO 控制馬達**，使小車能夠自動修正行駛方向，完成 **左轉、右轉與直行**，確保小車能夠穩定行駛於車道內。

本研究的技術可應用於：

- **智慧交通系統**：發展低成本的自駕技術，提高行車安全性。
- **機器人導航**：應用於智慧機器人，使其能夠在特定路徑上自主行動。
- **自動化運輸**：例如 **倉儲物流** 或 **無人載具**，可在固定軌道內運行，提高工作效率。

(二)未來發展與改進

目前本系統僅依靠單一攝影機進行車道偵測，未來可透過以下方式提升性能：

- 加入**超音波感測器**，偵測障礙物，提升避障能力。
- 使用**深度學習 (如 CNN)** 來強化影像辨識，提高車道偵測的精準度。
- 結合 **GPS** 與 **LiDAR**，進一步提升導航能力，使其適應更多環境。

本研究證明了 **影像處理技術** 可有效應用於自駕導航，為未來智慧交通與機器人技術發展提供了基礎技術參考，並可進一步結合 **AI 技術**，使其具備更強的環境適應能力與自主行駛能力。

參考資料

(一)陳宥任 (2018)。

以樹莓派嵌入式系統及攝像頭進行 CNN 深度學習之模型自駕車。

國立臺北科技大學車輛工程系碩士論文。

<https://hdl.handle.net/11296/nt6s73>

(二)陳立昂 (2019)。

以樹莓派嵌入式系統搭配攝像頭及陀螺儀進行深度學習之模型自駕車。

國立臺北科技大學車輛工程系碩士論文。

[https://ndltd.ncl.edu.tw/cgi-](https://ndltd.ncl.edu.tw/cgi-bin/gs32/gsweb.cgi/login?o=dnclcdr&s=id=%22108TIT00703009%22.&searchmode=basic)

[bin/gs32/gsweb.cgi/login?o=dnclcdr&s=id=%22108TIT00703009%22.&searchmode=basic](https://ndltd.ncl.edu.tw/cgi-bin/gs32/gsweb.cgi/login?o=dnclcdr&s=id=%22108TIT00703009%22.&searchmode=basic)