

Finding Lane Lines on the Road

N96121147_林璟任

車道辨識結果(solidYellowLeft)

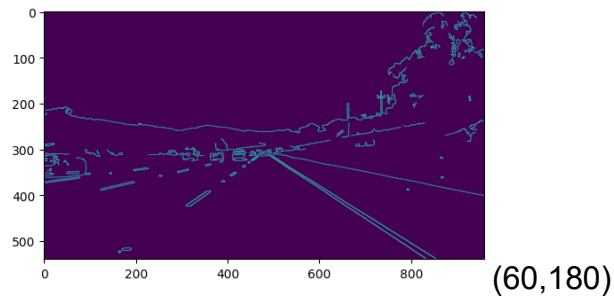
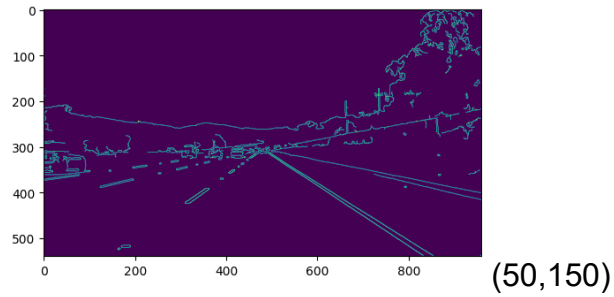
車道辨識結果(solidWhiteRight)

車道辨識結果(challenge)

1. **Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.**

車道辨識的流程如下：

1. 讀圖/影片
2. 灰階處理
3. 高斯模糊
4. **Canny 邊緣檢測：**
 - 在 low_threshold 和 high_threshold 分別測試了(50,150)和講義上的(60,180)，測試後發現對於周遭環境比較有影響，道路線都還是很清楚，因此這部分我還是使用(50,150)的參數



5. 定義 ROI :

- vertices 我設定的是(0,imshape[0]),(450, 350), (490, 350), (imshape[1],imshape[0])，第 1 和 4 的座標分別是圖片的左下和右下，主要調整的是第 2 和第 3 的座標，主要就是想要將這兩個座標定在車道線遠方相較的位置，y 座標的部分取 350 是因為太小會涵蓋到遠方的車子，太大則範圍太小

6. 定義 Hough 轉換參數，並畫出線圖：

- Hough 的參數主要調整的是 threshold = 22、min_line_length = 18 和 max_line_gap = 1，這樣的配置可以減少後續在 challenge 影片中偵測到錯誤的線段

- `Draw_line()`函數主要是參考網路上的作法，首先先計算斜率排除掉不合理線段，計算線段端點的 `mean` 值，算出平均的斜率，最後根據平均斜率計算出 `x1` 和 `x2` 的座標，`y1` 和 `y2` 的座標則是用測試的方式訂出比較適合的座標，最後再利用這四個點繪製線圖

7. 將原圖與線圖疊合：



2. Identify potential shortcomings with your current pipeline

- ROI 的 `vertice` 和 `Draw_line` 有寫死的部分，測試過後發現對於像 `challenge` 這種有轉彎的道路線比較難以處理
- 光線也會影響車道線的判斷(影子落在車道上會被辨識成線段)

3. Suggest possible improvements to your pipeline

- 計算道路線的部分，不要使用平均，使用如 Sobel Edge

Detection 等方式計算道路線

- 再不同的 Color Space 上面尋找適合的 Color Channel，避免光影造成錯誤的判斷

參考資料：

- [自駕車學習之路\(二\). 用影像處理做車道偵測 | by Hsin-Cheng Chao | Medium](#)
- [Self-Driving Car Project1- Lane Finding 實作 | by 天道酬勤 | Medium](#)
- [Day5-當自動駕駛遇見 AI-Project1: Finding Lane Lines\(1\) - iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天 \(ithome.com.tw\)](#)