NM6101072 張育嘉

Finding Lane Lines in a Video Stream

- 車道白線辨識： https://youtu.be/Oy-7XuHtI9Y
- 車道黃線辨識： https://youtu.be/NDc7QMZUJgs

使用工具

- 開發平台：Jupyter Notebook
- 程式語言：Python 3.8
- Packages：
  - NumPy
  - OpenCV
  - Matplotlib
  - MoviePy

實作步驟

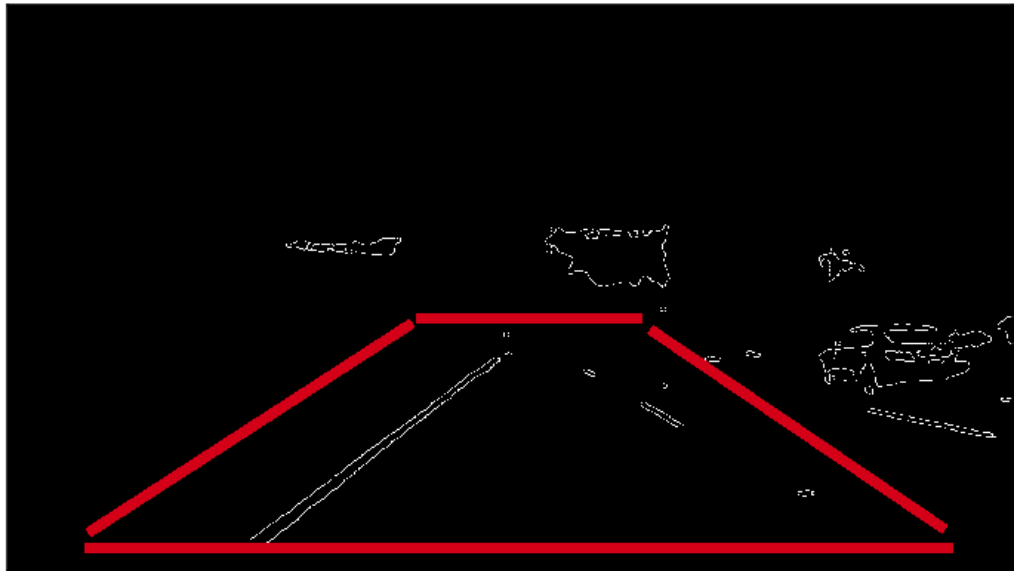
- 定義出白線及黃線範圍
- 找出圖上所有物體邊緣
- 框出ROI(Region of Interest)範圍
- 選好的區域做Hough Transform，把細碎的邊緣連成線
- 最後，將每一幀圖像合成一段完整影片

參數設定

- Gaussian kernel：3
- Edge Detection使用Canny Edge Detector
  - Low threshold：50
  - High threshold：150
  - Canny算法：尋找圖像中的亮度梯度
    - 圖像邊緣亮度梯度比較大
    - 利用上述特性找出圖像邊緣
    - 需要給定梯度閾值

參數設定

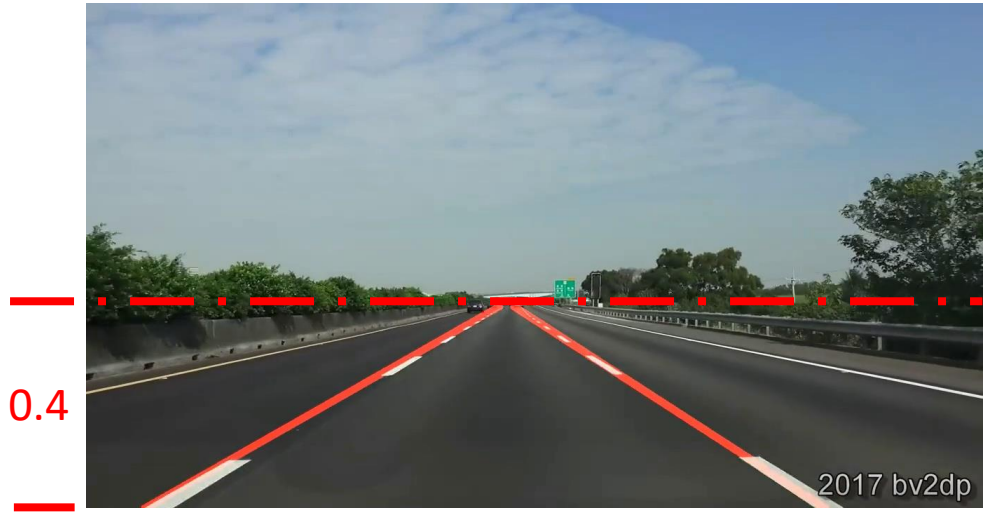
- 框出ROI
  - 我們需要的是車道線的部分，而每段影片的車道線比例略有差異，需選定合適的ROI以框出我們要的車道線。
  - 白車道線影片設定
    - bottom_left = [cols*0.1, rows*0.95]
    - top_left   = [cols*0.4, rows*0.7]
    - bottom_right = [cols*0.8, rows*0.95]
    - top_right   = [cols*0.5, rows*0.7]
  - 黃車道線影片設定
    - bottom_left = [cols*0.1, rows*0.95]
    - top_left   = [cols*0.5, rows*0.7]
    - bottom_right = [cols*0.9, rows*0.95]
    - top_right   = [cols*0.6, rows*0.7]



圖片來源：https://github.com/naokishibuya/car-finding-lane-lines/blob/master/images/region-of-interest.png

參數設定

- Hough Transform
  - threshold：20
  - minLineLength：20
  - maxLineGap：300
- 畫線 (lane_lines)
  - 白車線影片：從畫面0.4倍高開始畫線
  - 黃車線影片：從畫面0.3倍高開始畫線



心得

- 在定義ROI範圍的時候，發現參數必須依照影片情景手動調整，因車道線比例不會是固定的ROI，尤其在變換車道時該現象就更明顯。

右側白線偏向畫面右下角，需調整bottom_right 參數，車道線才會被框在合理的ROI範圍內。

心得

- 另外，Canny算法跟Hough Transform也都需手動給定閾值，以找出我們要的直線，但邊緣很容易被遮蔽或因光線不足，常常是不連續或不容易被偵測出來。

右側白線疑似因右前方車輛阻擋而偏移

心得

- 上述需手動調整參數的部分，也許可以透過機器學習中的監督式學習改善，透過類神經網路學習大量特徵來取代人類手動調整參數。