

# Homework1\_OpenCV 車道辨識練習與應用

## 專案實作環境

- 作業系統：macOS
- 程式語言與版本：Python 3.9.10
- OpenCV 版本：4.5.2

## 實作方法流程

- 首先用 `imread` 讀取圖片，之後用 `cvtColor` 轉成灰階影像，以 `equalizeHist` 進行影像強化，用 `GaussianBlur` 影像去雜訊，`threshold` 二值化，`morphologyEx` 進行形態學運算，`Canny` 邊緣檢測，最後用 `HoughLinesP` 直線偵測配合 `line` 在圖上畫線

## 引用函式所採用之演算法與參數值

```
import numpy as np
import cv2

image = cv2.imread('2.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

gray = cv2.equalizeHist(gray)

kernel_size = 5
blur_gray = cv2.GaussianBlur(gray, (kernel_size, kernel_size), 0)
cv2.imshow('blur gray', blur_gray)

_, threshold_image = cv2.threshold(blur_gray, 200, 255,
cv2.THRESH_BINARY)
```

```
cv2.imshow('threshold', threshold_image)

kernel = np.ones((3, 3), np.uint8)
morphology_image = cv2.morphologyEx(threshold_image, cv2.MORPH_OPEN,
kernel)
cv2.imshow('mor', morphology_image)

low_threshold = 100
high_threshold = 200
masked_edges = cv2.Canny(morphology_image, low_threshold,
high_threshold)
cv2.imshow('masked', masked_edges)

rho = 1
theta = np.pi / 180
threshold = 50
min_line_length = 100
max_line_gap = 20

line_image = np.copy(image)

lines = cv2.HoughLinesP(masked_edges, rho, theta, threshold,
min_line_length, max_line_gap)

for line in lines:
    x1,y1,x2,y2 = line[0]
    cv2.line(line_image,(x1,y1),(x2,y2),(0,0,255),10)

# color_edges = np.dstack((masked_edges, masked_edges, masked_edges))

# combo = cv2.addWeighted(color_edges, 0.8, line_image, 1, 0)

cv2.imshow('result', line_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

## 遇到困難及解決方法

- 這次的作業是我第一次接觸到影像辨識方面的程式，必須說參數調整上面遇到了非常大的問題，即使這次勉強有把線畫出來，可是跟目標有著想當大的一個差距，詢問過朋友結論都是我的參數調整有問題，可是不管怎麼調整，畫出來的結果還是不是很滿意，最後實在沒有辦法，借了朋友寫好的程式碼跑跑看，發現即使是同一張圖，用相同的參數，畫出來的線跟朋友最後跑出來的結果差了很多，我們討論了很久也是找不到原因，畢竟是一模一樣的東西，怎麼在 Windows 上面跑起來是正常的，結果到 macOS 上面就不正確了呢？這是一個大問題，到了最後我也沒有找到比較好的解法，網路上也沒有比較好的解釋，下個禮拜會去找教授討論一下這個狀況，看是版本問題還是怎麼樣，不然這個問題實在有點困擾我。

## 本次專案中的個人所學

- 這次的專案中，初次學習到了很多影像辨識相關的函式，也在網路上查了很多資料，發現影像辨識所需要的數理基礎是相當深厚的，即使到了現在，對影像辨識也是有點一知半解的狀態，對於調整參數的部分有點心得，雖然最後調整完的樣子不是很滿意，不過至少有畫出來，也大概知道為什麼前面要對影像做這麼多的前置處理。

## 各階段處理後的結果圖片

- 輸入影像 灰階轉換





- 影像強化



- 影像去雜訊



- 二值化



- 形態學



- 邊緣檢測



- 直線偵測 繪製車道線



