

# 自動駕駛實務 Autonomous-Driving

姓名 / 學號：何宇浩 VK6112026

車道辨識：在行車紀錄器影像中標示出車輛前方的車道線位置。並在受到車道線顏色變化、型態變化及周遭光線明暗和樹影的影響下，仍可清晰標示出前方車道線。

## 一、完成影片連結：

Challenge : <https://youtu.be/WONDwEzAVt4>

solidWhiteRight : <https://youtu.be/SBsczK3lp4Q>

solidYellowLeft : <https://youtu.be/Hrakc0Frtus>

## 二、影像處理程式流程：

以下透過測試影像 'test.jpg' 及 'canny.jpg' 進行程式流程說明及驗證。

### 1. Import Packages

```
[1] from google.colab import drive
drive.mount('/content/drive')
import os
os.chdir('/content/drive/MyDrive/碩一下/自動駕駛/project1')

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
import cv2
import math
from moviepy.editor import VideoFileClip
from IPython.display import HTML
```

### 2. Read in an image

```
[19] #reading in an image
image1 = mpimg.imread('canny.jpg')
image2 = mpimg.imread('test.jpg')

#printing out some stats and plotting
fig = plt.figure(figsize=(15,30))
print('This image is:', type(image1), 'with dimensions:', image1.shape)
plt.subplot(121)
plt.imshow(image1)
print('This image is:', type(image2), 'with dimensions:', image2.shape)
plt.subplot(122)
plt.imshow(image2)

plt.show()
```

This image is: <class 'numpy.ndarray'> with dimensions: (540, 960, 3)  
 This image is: <class 'numpy.ndarray'> with dimensions: (985, 1915, 3)



### 3. Helper Functions

a) 圖片處理流程：

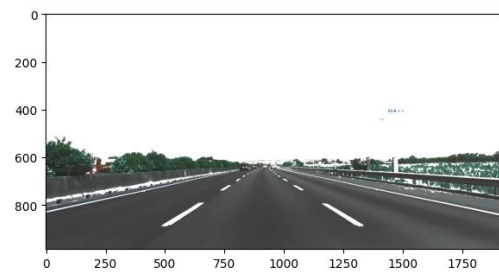
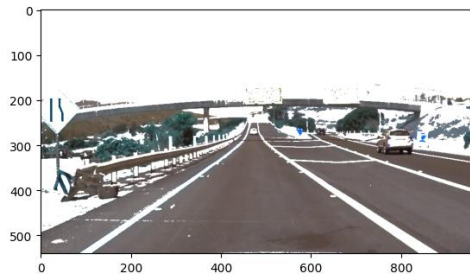
```
def process_image(image):
```

Color change → Gray scaling → Canny Edge Detection and Gaussian smoothing → Region of Interest → Hough Transform line detection → Weighted Image

b) Color Change：將黃色線轉為白線

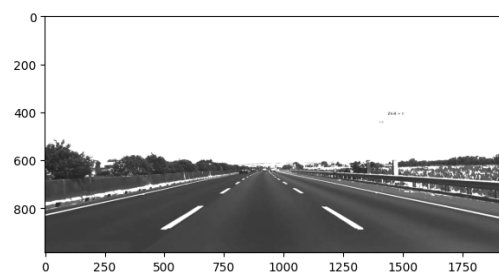
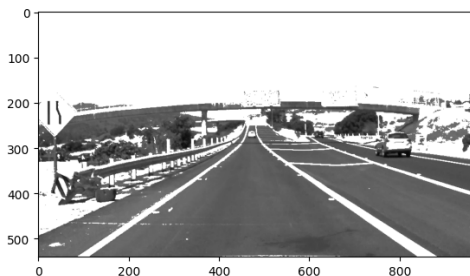
```
def color_change(image):
```

由於黃線在經過灰階及高斯化後，與背景對比度過低將造成辨識不易，因此為了更準確抓到車道線，加入 `color_change` 的函式，使得影片進行車道辨識時較不易在某段消失。



c) Gray scaling：將影像進行灰階處理

```
def grayscale(img):
```

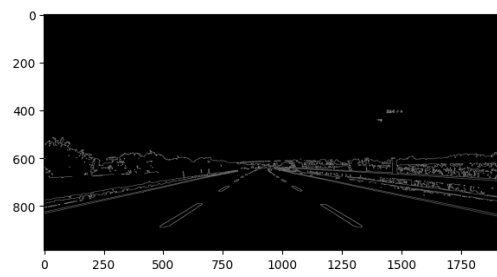
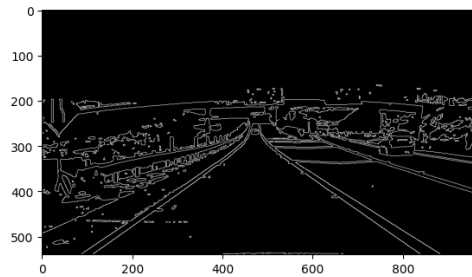


d) Canny Edge Detection and Gaussian smoothing：

對影像做高斯模糊後再進行邊緣偵測，可濾掉一些雜訊，獲得更清楚之

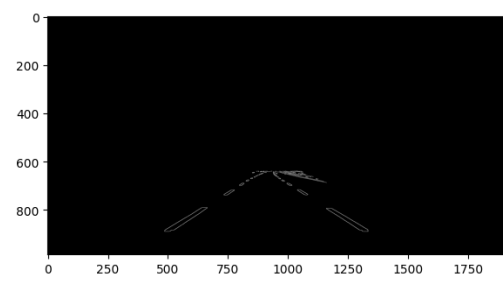
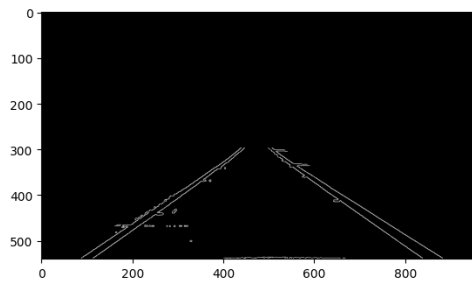
輪廓圖像。

```
def gaussian_blur(img, kernel_size):  
def canny(img, low_threshold, high_threshold):
```



e) Region of Interest : 剔除車道線以外，較不重要之圖像訊息。

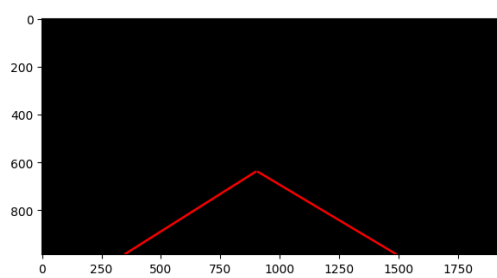
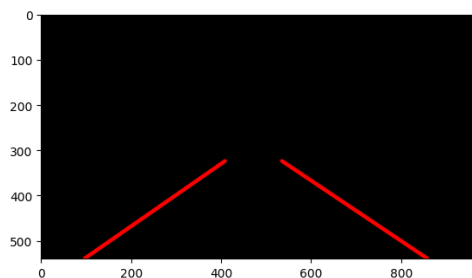
```
def region_of_interest(img, vertices):
```



f) Hough Transform line detection :

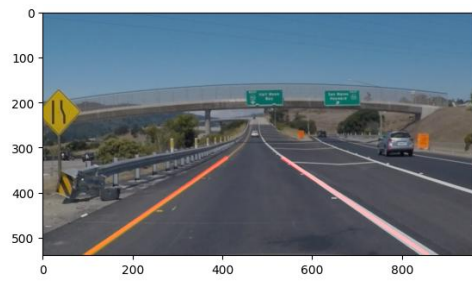
在車道線繪製部分，先將輪廓影像內斜率較奇異的點剔除，再依斜率分為左右車道，此時若直接對每一線段上色，則可能造成不連續的車道線，因此對分邊後的點座標取平均值，獲得中心點，再依斜率繪製車道線，詳細內容可參考完整程式碼。

```
def draw_lines(img, lines, color=[255, 0, 0], thickness=8):  
def get_slope(x1,y1,x2,y2):  
def hough_lines(img, rho, theta, threshold, min_line_len, max_line_gap):
```



g) Weighted Image : 疊圖

```
def weighted_img(img, initial_img,  $\alpha=0.8$ ,  $\beta=1.$ ,  $\gamma=0.$ ):  
    return cv2.addWeighted(initial_img,  $\alpha$ , img,  $\beta$ ,  $\gamma$ )
```



h) 影片輸入、輸出：

```
clip1 = VideoFileClip('/content/drive/MyDrive/碩一下/自動駕駛/project1/solidWhiteRight.mp4')
onlineClip_01 = clip1.fl_image(process_image)
onlineClip_01.write_videofile("solidWhiteRight2.mp4", audio = False)
```