電資二 111820006 陳羿錦 HW2 個人報告

1. 專案實作環境：
    1. 作業系統：Windows 10
    2. 程式語言與版本：Python 3.9.5
    3. OpenCV 版本：4.9.0.80
2. 實作方法流程與參數使用說明：
    1. 定義函式 addDIr() 建立 out_pic 資料夾。
    2. 定義函式 readImage() 進行讀取圖片，利用 cv2.imread() 讀入圖片。
    3. 定義函式 toGray() 進行轉成灰階圖片，利用 cv2.cvtColor(img , cv2.COLOR_BGR2GRAY)。

    - 第一張照片

    1. 使用 blur() 進行濾波，參數調 (20 , 15)。
    2. 二值化圖片，使用 type = cv2.THRESH_BINARY，門檻與最大值調 140 與 240。
    3. 使用 Scherr() 進行邊緣檢測，使用 cv2.CV_64F，方向為 (0 , 1)。
    4. 使用閉運算進行形態學，kernal 調 5*5，迭代 2 次。
    5. 使用 cv2.line() 與 cv2.HoughLinesP() 偵測與繪製車道線。

    - 第二張照片

    1. 使用 medianblur() 進行濾波，參數調 39。
    2. 二值化圖片，使用 type = cv2.THRESH_BINARY，門檻與最大值調 105 與 250。
    3. 使用 Scherr() 進行邊緣檢測，使用 cv2.CV_32F，方向為 (0 , 1)。
    4. 使用閉運算進行形態學，kernal 調 11*11，迭代 15 次。
    5. 使用 cv2.line() 與 cv2.HoughLinesP() 偵測與繪製車道線，取車道線斜率絕對值 > 0.3 者。

    - 第三張照片

    1. 使用 cv2.bilateralFilter() 進行濾波，參數調 50，50，1000。
    2. 二值化圖片，使用 type = cv2.THRESH_BINARY，門檻與最大值調 130 與 250。
    3. 使用 Canny() 進行邊緣檢測，參數選擇 200，100。
    4. 使用閉運算進行形態學，kernal 調 3*3，迭代 15 次。
    5. 使用 cv2.line() 與 cv2.HoughLinesP() 偵測與繪製車道線。

3. 遇到困難與解決方法：
    1. 使用各種濾波模式與邊緣偵測對於圖片清晰化效果不佳。
        解決：努力 5 個小時慢慢測試出最佳結果。

4. 個人所學與心得：

本次作業為邊緣偵測應用的作業，目的為熟悉邊緣偵測處裡的工具與函式，我認為比較大的困難是判定處理後效果是否有更好，以及需要不斷重複改變參數與使用之濾波模式使之能有更佳的影像結果，需要重複調整，對於本次作業，使得我再次體會到訓練資料預處理的困難。

5. 各項結果圖片：