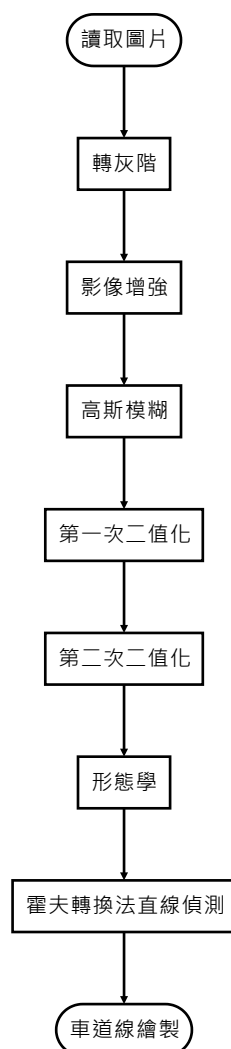


Homework 1

專案實作環境

- 作業系統
 - windows 11
- 程式語言與版本
 - Python 3.7
- OpenCV 版本
 - 4.5.5.62

實作方法流程



引用函式所採用之演算法與參數值

- 轉灰階

```
1 | photoGray = cv2.cvtColor(photo, cv2.COLOR_BGR2GRAY)
```

- 影像增強

```
1 photoEnhance = cv2.equalizeHist(photoGray)
```

- 高斯模糊

```
1 photoBlur = cv2.GaussianBlur(photoEnhance, (3, 3), 0)
```

- 第一次二值化

```
1 ret, photo2 = cv2.threshold(photoBlur, 197, 255, cv2.THRESH_TOZERO_INV)
```

- 第二次二值化

```
1 ret2, photo3 = cv2.threshold(photo2, 193, 255, cv2.THRESH_BINARY)
```

- 形態學

```
1 kernel = np.ones((3, 3), np.uint8)
2 picture0 = cv2.morphologyEx(photo3, cv2.MORPH_CLOSE, kernel)
```

- 霍夫轉換法直線偵測

```
1 lines = cv2.HoughLinesP(picture0, 1, np.pi / 180, 100, 100, 90, 4)
```

遇到困難及解決辦法

由於天空的亮度有可能比車道線亮，二值化時會將天空也會是白色，導致車道線會畫到天空上，因此我選擇了使用兩次二值化，第一次先將大於門檻值的顏色轉呈黑色，天空就會是黑色，第二次再將小於門檻值的顏色轉成黑色這樣。就可以只讓在這兩個門檻值的車道線變白色，讓後面的轉換更便利。

本次專案中個人所學

對於在上課中講到的演算法更有實際應用的感覺。

各階段處理後的結果影片

1. 讀入照片



2. 轉灰階



3. 影像增強



4. 高斯模糊



5. 第一次二值化



6. 第二次二值化



7. 形態學



8. 霍夫轉換法直線偵測

