



# 多媒體技術與應用

## Spring, 2025.

Instructor : Yen-Lin Chen(陳彥霖), PH.D.  
Distinguished Professor  
Dept. Computer Science and Information Engineering  
National Taipei University of Technology





# Project 1

OpenCV車道辨識練習與應用  
個人報告繳交期限:2025/03/12 23:59





# 學習目標

- 利用邊緣檢測、直線偵測等基礎影像處理技術，提取出影像中感興趣的物件及其位置資訊等，做為未來自駕車技術中對於道路影像分析的前置準備。
- 本次作業將使用上述影像處理技術，使用OpenCV進行提取車道邊線實作練習。





# 車道線偵測實作目的

- 在車道線偵測的過程中，車用相機不一定能保有穩定的圖像訊號，容易造成影像中留有許多雜訊，以至於難以取出車道線的特徵。
- 在車道線偵測的過程中，因車道線影像不一定能夠完整保留，所以可以透過形態學對影像進行處理，得到較佳的結果



# OpenCV環境建置

本機端實作需建置的環境



# 安裝Python

- 開啟命令提示字元視窗，輸入「python」以查看是否安裝成功。
- Python模式下輸入「exit()」可離開Python模式。

```
命令提示字元 - python
Microsoft Windows [版本 10.0.19041.746]
(c) 2020 Microsoft Corporation。著作權所有，並保留一切權利。

C:\Users\Chen-Pu>python
Python 3.6.5 (tags/v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```





# 安裝OpenCV函式庫

- 開啟命令提示字元視窗，輸入「pip install opencv-python==4.1.0.25」以安裝Python專用的OpenCV 4.1.0版函式庫。

```
命令提示字元
Microsoft Windows [版本 10.0.19041.746]
(c) 2020 Microsoft Corporation。著作權所有，並保留一切權利。
C:\Users\Chen-Pu>pip install opencv-python==4.1.0.25
```



# 安裝Numpy套件

- 輸入「pip install numpy」以安裝NumPy套件

```
命令提示字元
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\Chen-Pu>pip install numpy
```





# 檢視套件版本

- 進入Python模式檢視OpenCV、NumPy版本：
- 做到這一步，表示已完成Windows上的Python和OpenCV環境建置，後續會再對OpenCV的各種功能函式做說明。

```
命令提示字元 - python
C:\Users\Chen-Pu>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> import numpy as np
>>> cv2.__version__
'4.1.0'
>>> np.__version__
'1.19.5'
>>>
```

# OpenCV環境建置

colab端實作需建置的環境





# 掛載自己的雲端硬碟

- 使用colab新增程式碼，輸入輸入下方兩行程式

```
from google.colab import drive  
drive.mount('/content/drive')
```





# 掛載自己的雲端硬碟

- 選擇{連線至Google雲端硬碟}

要允許這個筆記本存取你的 Google 雲端硬碟檔案嗎？

連線至 Google 雲端硬碟時，這個筆記本中執行的程式碼將可修改 Google 雲端硬碟的檔案，直到存取權遭到撤銷為止。

不用了，謝謝

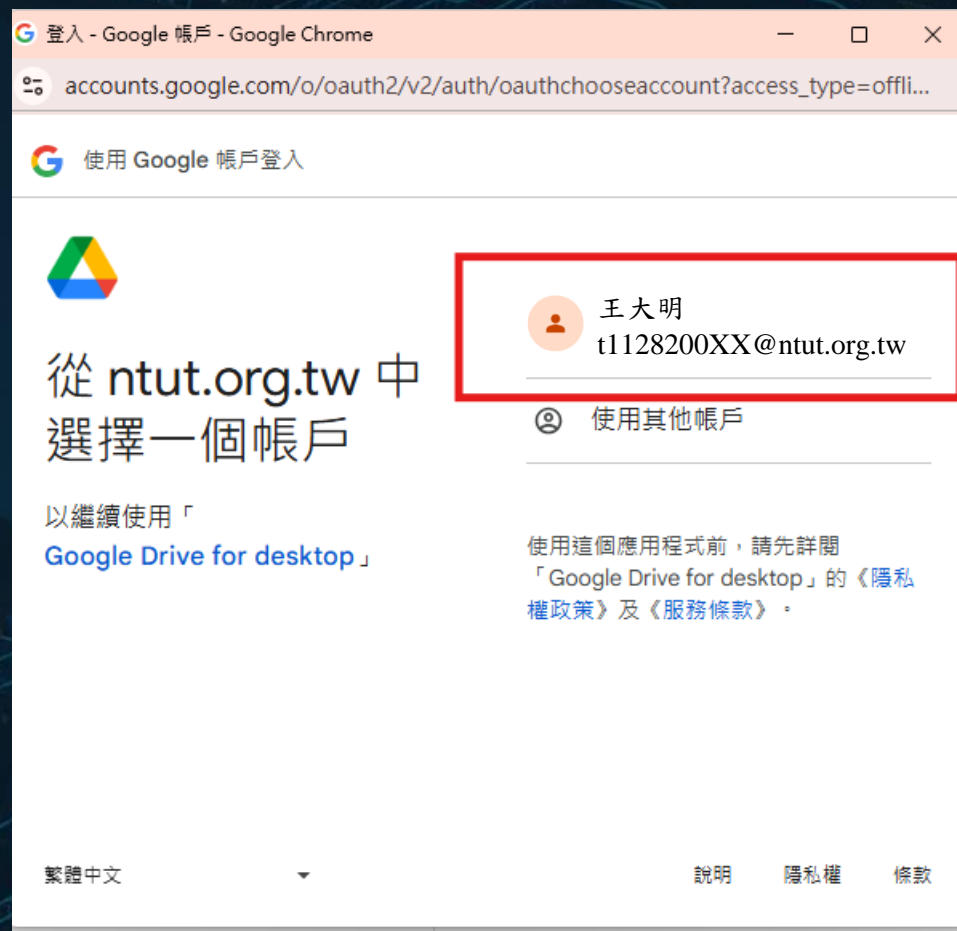
連線至 Google 雲端硬碟





# 掛載自己的雲端硬碟

- 選擇{自己的google帳戶}





# 掛載自己的雲端硬碟

- 選擇{繼續}

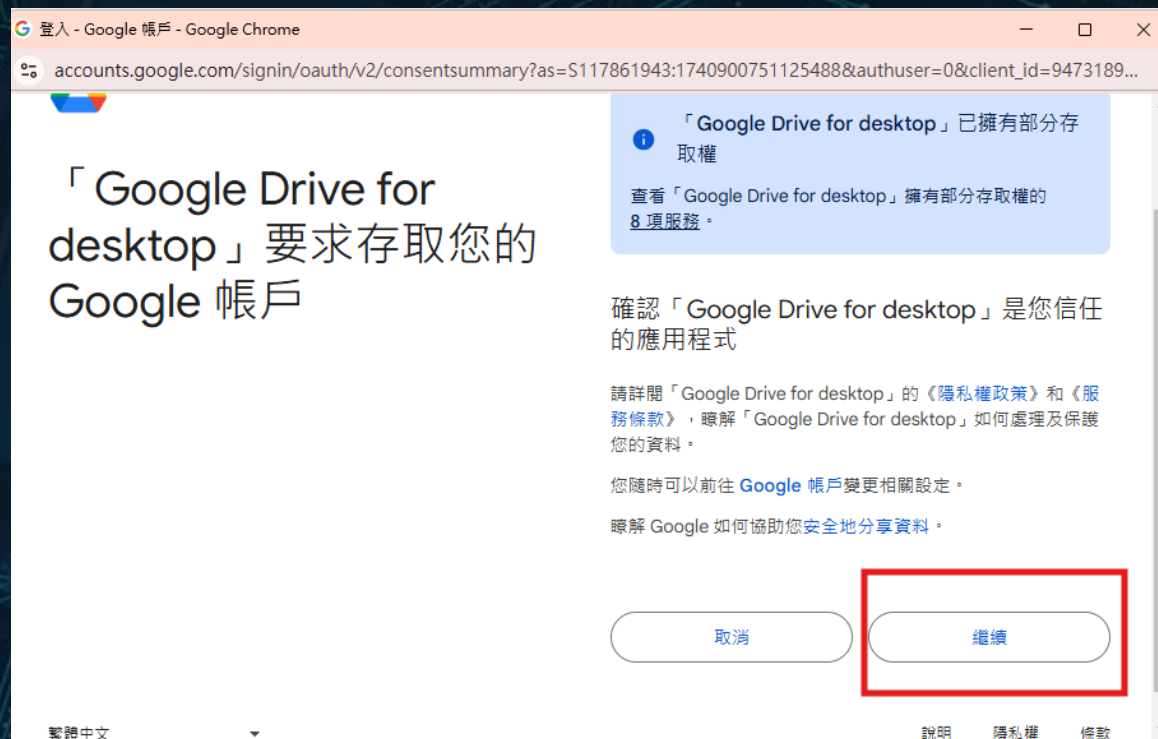






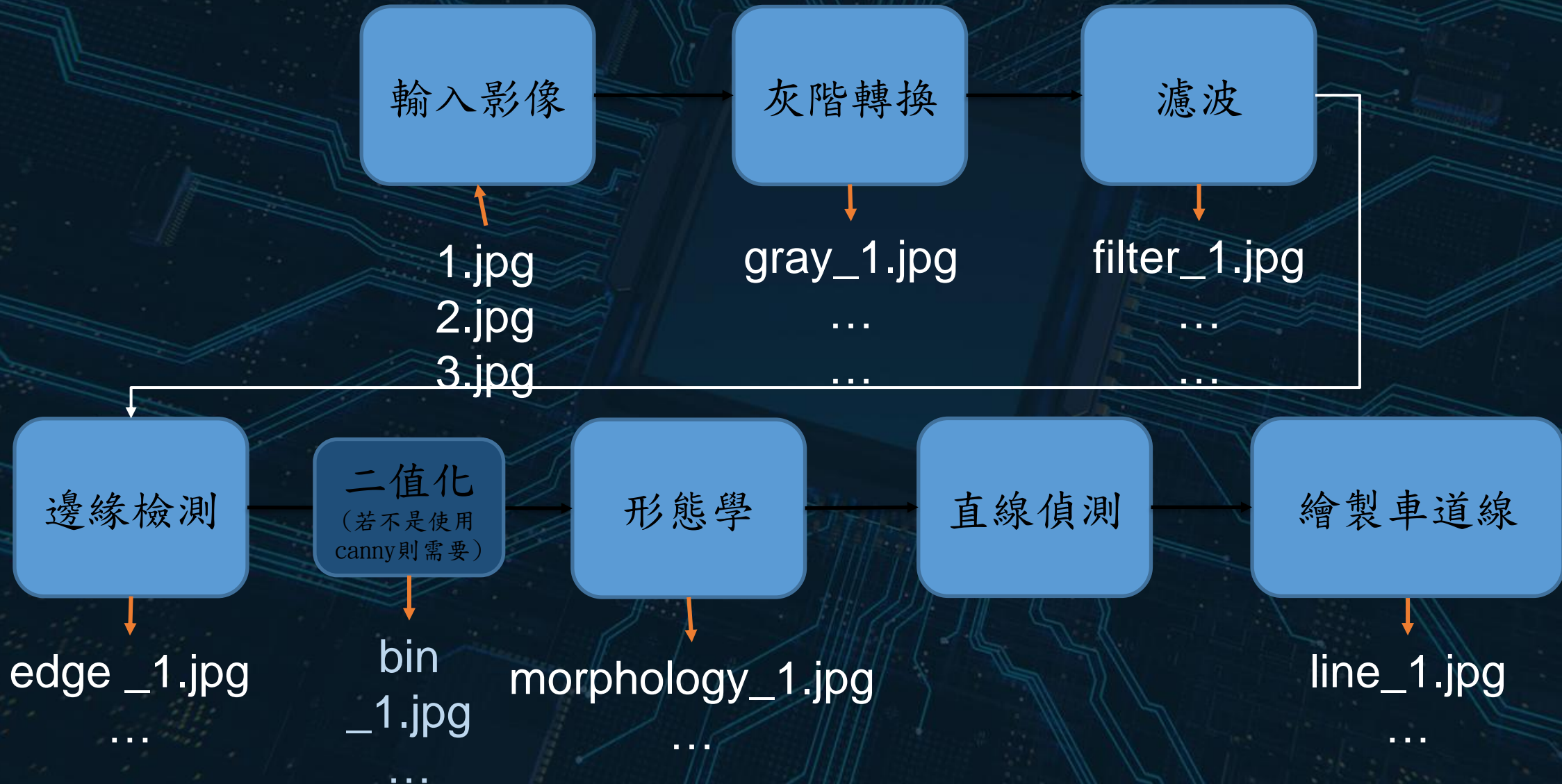
# 掛載自己的雲端硬碟

- 選擇{允許存取}





# 車道線偵測實作流程







# 車道線偵測實作流程概要(作業一部份)

輸入影像



灰階轉換



濾波

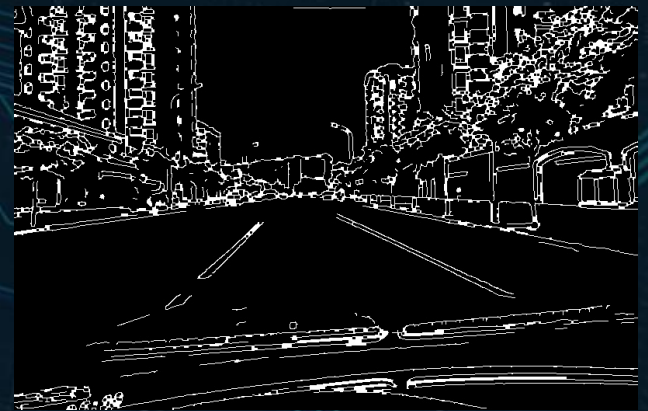






# 車道線偵測實作流程概要(以Python為例)

- 邊緣檢測
  - 常見OpenCv邊緣檢測演算法: `cv2.Sobel()`、`cv2.Scharr()`、`cv2.Laplacian()`、`cv2.Canny()`
  - 透過觀察邊緣檢測率、定位準確度、響應程度驗證效果
  - 更改演算法與相關參數等方法取得最佳結果
- 二值化(若不是使用Canny則需要)
  - 使用`cv2.threshold()`進行灰階影像二值化
  - 可使用自訂二值化門檻值或OTSU自適應門檻值
- 形態學
  - 使用`cv2.morphologyEx()`進行形態學開運算/閉運算
  - 更改開運算與閉運算順序、迭代次數等方法取得最佳結果

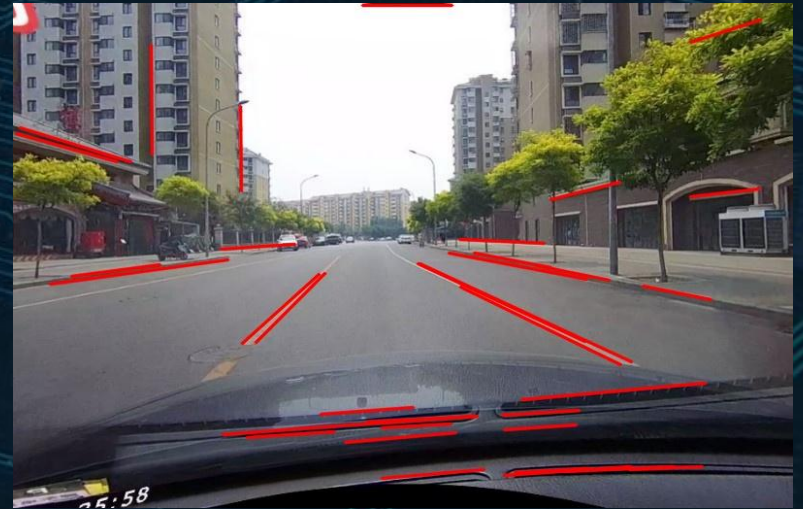






# 車道線偵測實作流程概要(以Python為例)

- 直線偵測
  - 常見OpenCv直線偵測演算法:  
`cv2.HoughLines()`、`cv2.HoughLinesP()`
  - 透過觀察直線偵測率、定位準確度、連續程度驗證效果
  - 更改演算法與相關參數等方法取得最佳結果
- 繪製車道線
  - 霍夫轉換法直線偵測會將所有線段資料(一線段包含x1、y1、x2、y2資訊)儲存至  
`numpy.ndarray`中
  - 使用`np.copy()`複製原圖像至一新畫布
  - 使用`cv2.line()`將偵測車道線段繪至新畫布上

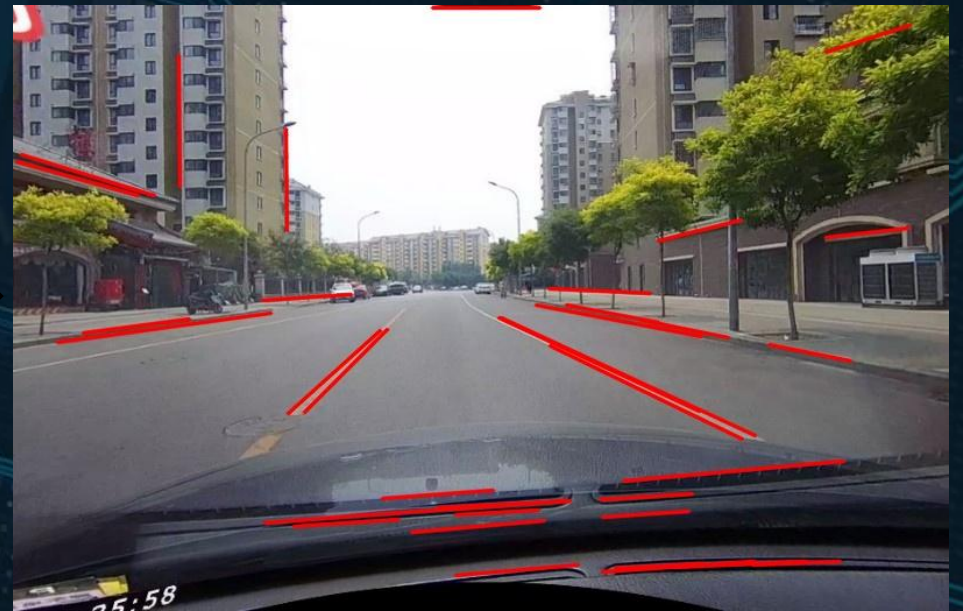
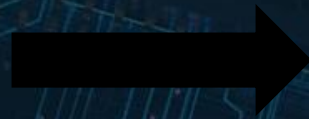






# 實作

- 輸入的影像於北科i學園提供下載(1.jpg/2.jpg/3.jpg)
- 引入OpenCV函式庫，利用函式庫中所提供的函式進行實作。
- 使用第一點的圖像，依上述流程或上課內容所提供之方法進行影像處理
- 將萃取後的車道線繪製在原圖像上







# 作業繳交要求

- 本次作業為個人作業
- 作業繳交項目：
  - 個人報告(pdf)
  - 可執行程式檔(ex: .py、.ipynb)
  - 結果圖片(jpg,15 or 18張)
- 檔名格式：**Project1\_學號\_姓名.zip**
- Project2繳交期限至**2025/03/12(二)23:59:59**
- 超過正常繳交期限一周成績**打9折**，每再遲交一週成績再打9折  
(最多遲交兩週)





# 作業繳交要求

- 個人報告需包含以下內容：
  - 實作方法流程
  - 引用函式所採用之**演算法**與**參數值**(可比較不同參數與圖片的效果)
  - 遇到困難及解決方法
  - 本次專案中個人所學
  - 各階段處理後的結果圖片