

# 期中進度簡報

行車紀錄器之道路違規行為偵測

組長：11027221 楊貽婷

組員：11027203 葉雨沁

11027213 李怡萱

# 目錄

**1~2**

動機與目的

**3~13**

當前進度

**14**

未來展望

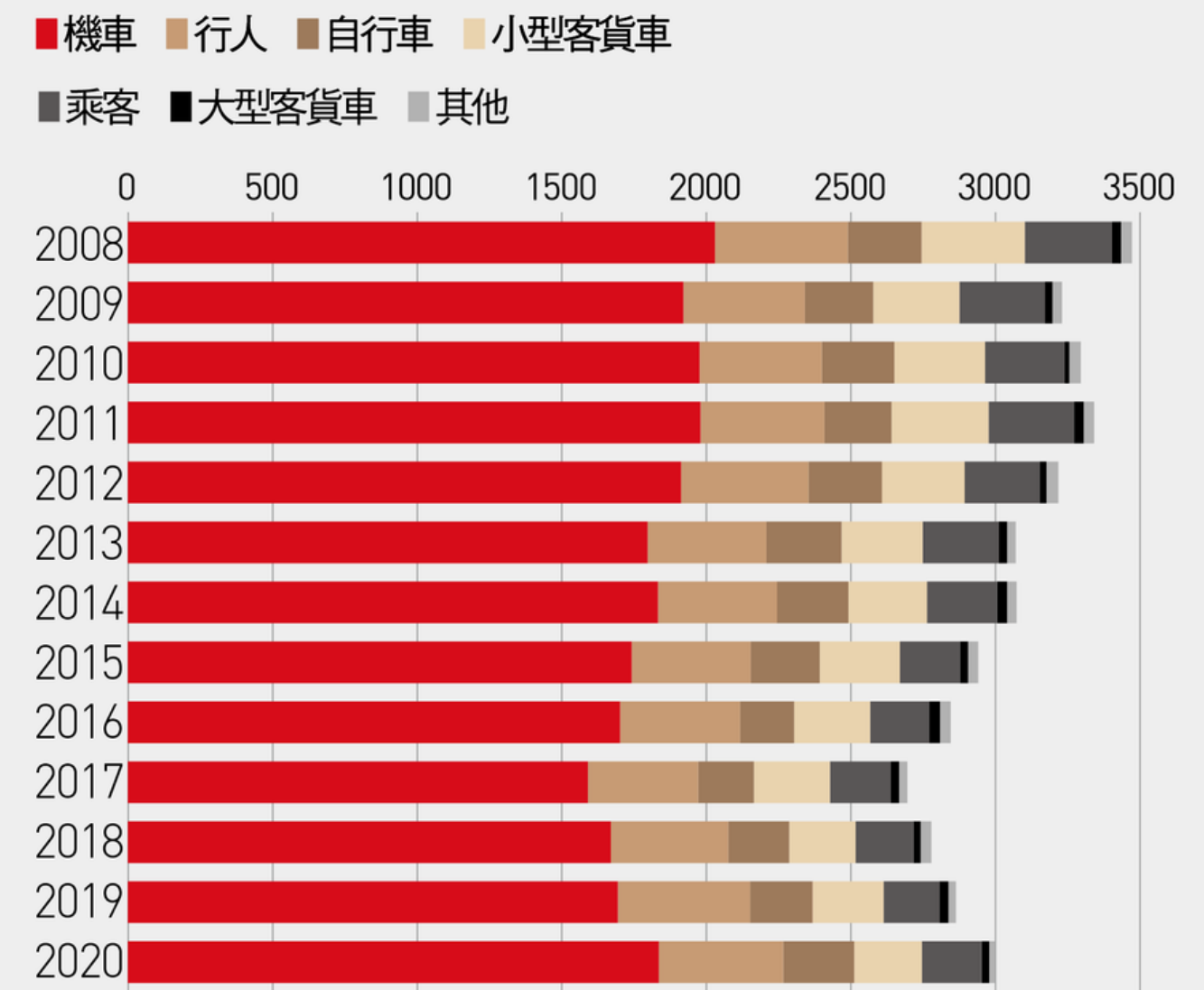
**15**

心得

# 動機與目的

台灣每年有將近三千人因為車禍意外導致失去寶貴的生命，造成家庭破碎，其中，造成車禍發生最大的原因就是道路違規，例如雙黃線違規變換車道、變換車道沒有打方向燈等等，種種的違規行為導致車禍的發生率節節攀升，我們希望透過辨識行車紀錄器自動監測違規行為，並且自動提供駕駛人其行車紀錄器所紀錄之違法影像以及違規法則，希望能透過此種方式提醒違規駕駛應遵守道路規範，不可抱有僥倖心態。

## 致死人數攀5年新高，機車最嚴重 死亡人數(人)



資料來源：交通部道安會

# 道路違規事項 網路平台檢舉流程圖



# 影片辨識--初版

## 處理方式

透過cv2的videocapture讀取影片原始檔案並設定影片大小，將影片透過高斯模糊化、灰階處理、色彩過濾、邊緣檢測，最後利用霍夫轉換顯示影片中車道線的範圍

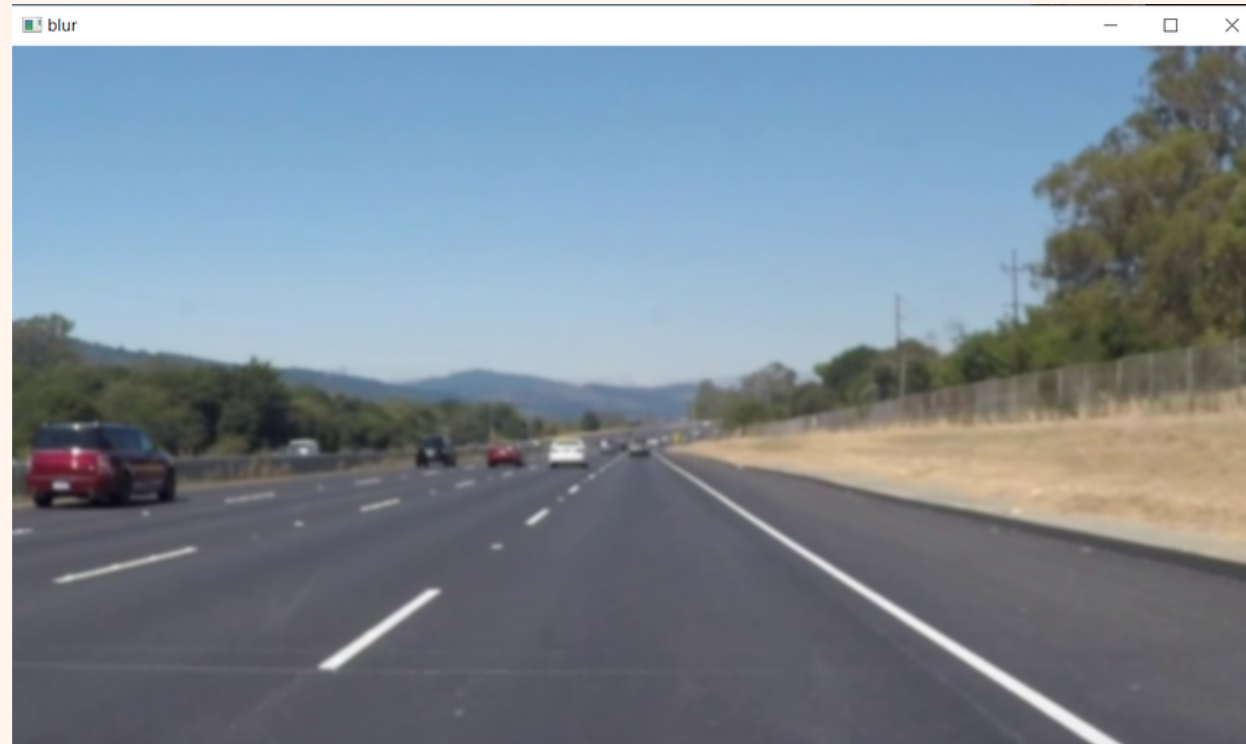
## 優點

- 程式編寫流程迅速
- 執行速度快
- 輸出影片撥放流暢

## 缺點

- 對於影片畫質要求高
- 影片天氣要求高
- 準確度不高
- 僅能檢測白色的車道線
- 不適用於每個影片

step.1 高斯模糊



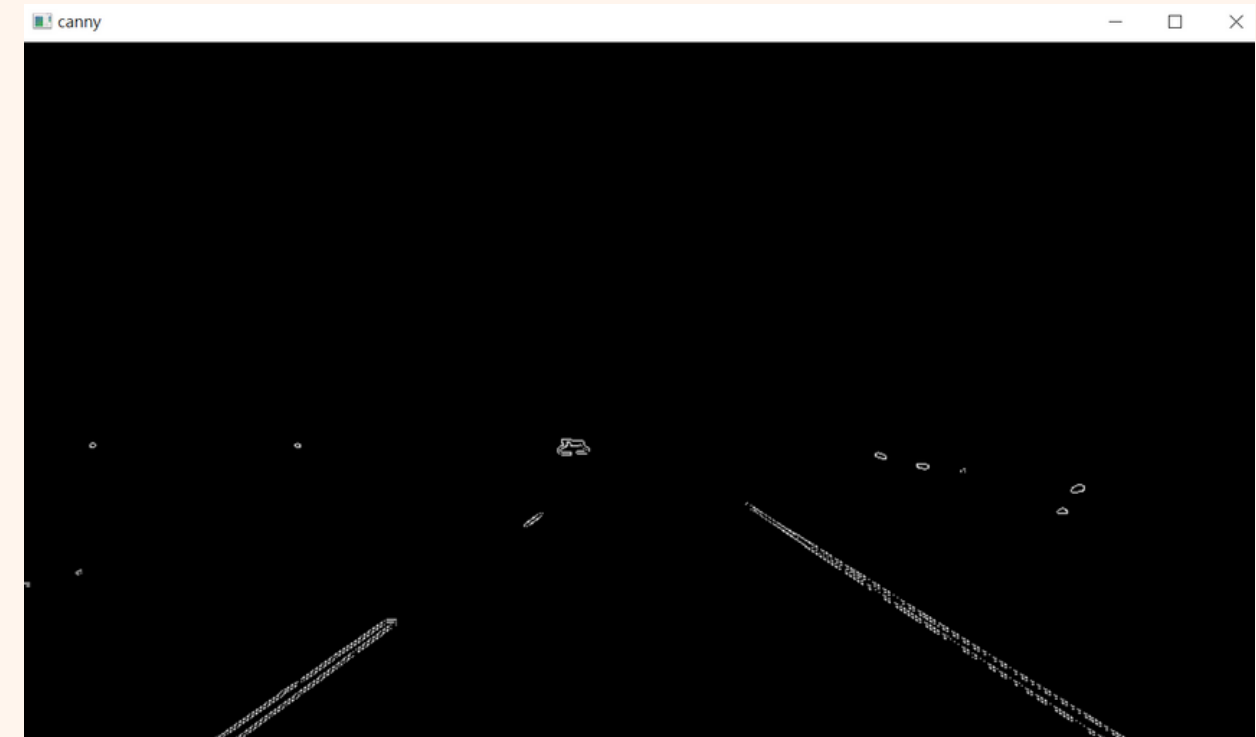
step2. 灰階處理



step3. 色彩過濾



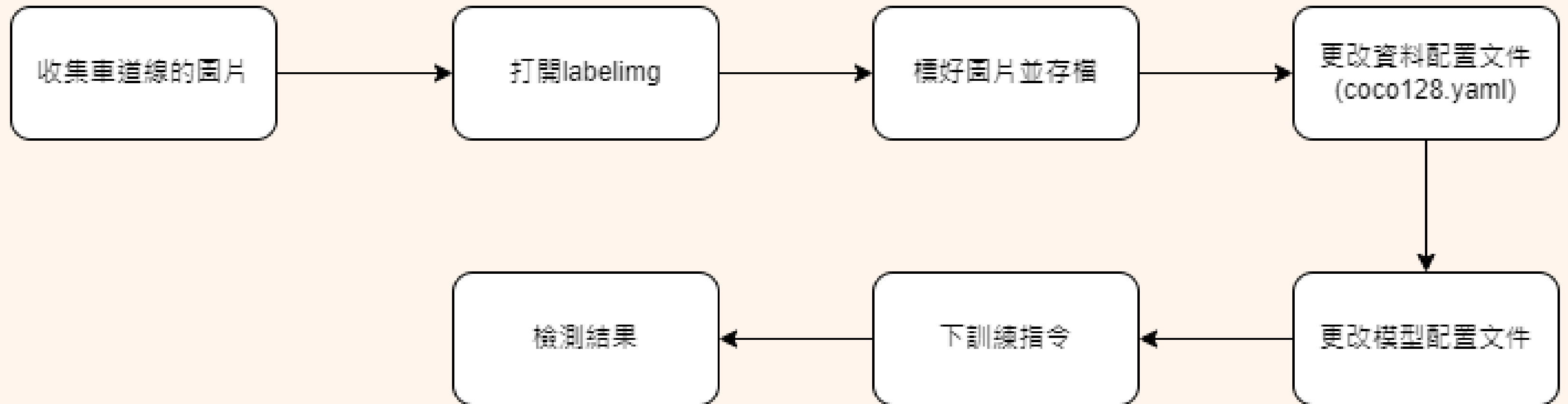
step4. canny邊緣檢測



## step.5 Hough Transform

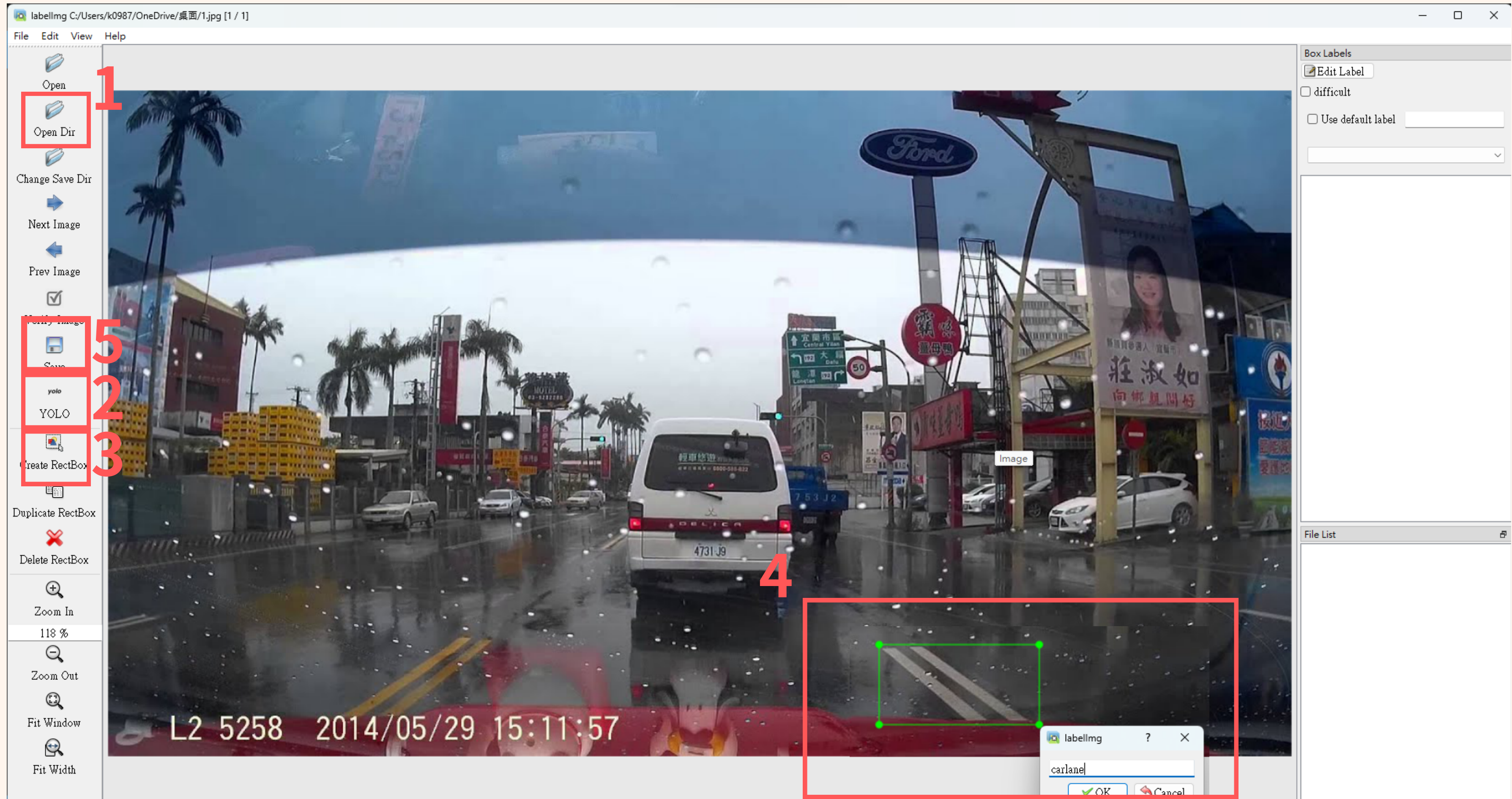


# 訓練辨識車道線模型之步驟





# 使用labelimg步驟



在資料配置文件中的更改：

```
95 76: scissors
96 77: teddy bear
97 78: hair drier
98 79: toothbrush
99 80: carlane
```

增加carlane類別

```
# 用絕對路徑比較好讓程式抓對資料
path: C:\Users\k0987\yolov5\datasets\carlane # dataset root
train: C:\Users\k0987\yolov5\datasets\carlane\images\train #
val: C:\Users\k0987\yolov5\datasets\carlane\images\train # va
test: # test images (optional)

nc: 81 # 總共有81個類別
```

將資料集路徑更改為絕對路徑

將number of classes的值加一(carlane)

在模型配置文件中的更改：

```
3 # Parameters
4 nc: 81 # number of classes
```

將number of classes的值加一

# 訓練指令

```
python train.py --img 640 --batch 16 --data ./data/myTest.yaml --epochs 100 --cfg ./models/myModel.yaml --weights Mybest.pt
```

輸入圖像的尺寸

批次大小

資料配置文件

遍歷整個數據集的次數

模型配置文件

預訓練權重

# 使用yolov5訓練模型後的結果



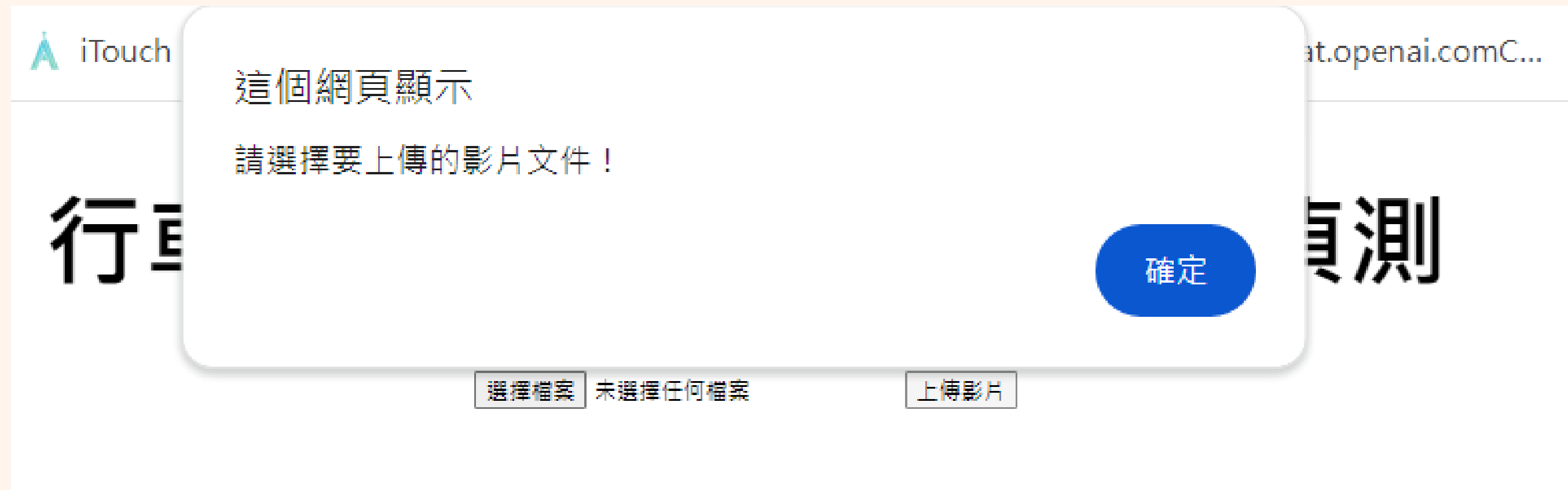


# 程式

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>影片上傳系統</title> <!--顯示在瀏覽器的標題欄-->
7   <h1>行車紀錄器之道路違規行為偵測</h1> <!--顯示在網頁上-->
8   <link rel="stylesheet" href="styles/style.css"> <!--引入.css -->
9 </head>
10 <body>
11   <div id="upload-container">
12     <input type="file" id="video-file" accept="video/*"> <!--accept 接受文件類型為視頻-->
13     <button onclick="uploadVideo()">上傳影片</button> <!--執行上傳影片的操作-->
14     <div id="video-preview"></div> <!--顯示上傳的影片預覽-->
15   </div>
16   <script src="scripts/main.js"></script> <!--引入外部.js-->
17 </body>
18 </html>
```

```
1 // main.js
2
3 function uploadVideo() {
4   const fileInput = document.getElementById('video-file');
5   const videoPreview = document.getElementById('video-preview');
6
7   if (fileInput.files.length > 0) { // 檢查是否有選擇文件
8     const file = fileInput.files[0]; // 第一個文件
9
10    if (file.type.startsWith('video/')) { // 檢查文件類型是否是影片
11      const videoURL = URL.createObjectURL(file);
12      const videoElement = document.createElement('video'); // 在網頁中顯示影片
13      videoElement.src = videoURL; // 設置先前創建的影片URL，將所選影片顯示在網頁上
14      videoElement.controls = true; // 在影片上顯示控制面板 例如播放、暫停...
15      videoPreview.innerHTML = '';
16      videoPreview.appendChild(videoElement);
17    } // if
18    else {
19      alert('請選擇一個有效的影片文件！');
20    } // else
21  } // if
22  else {
23    alert('請選擇要上傳的影片文件！');
24  } // else
25 }
```

## 未選擇任何檔案

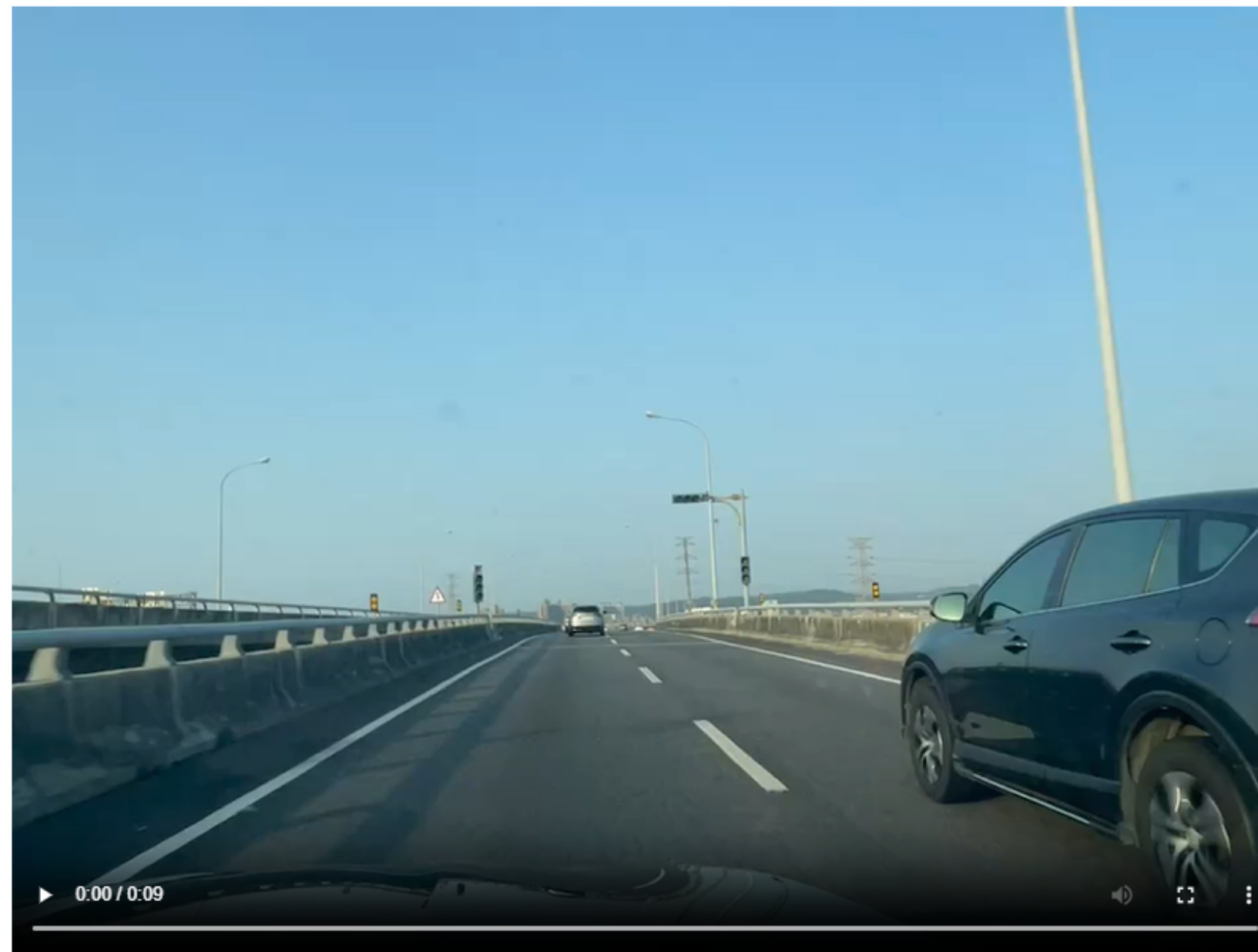


## 上傳影片

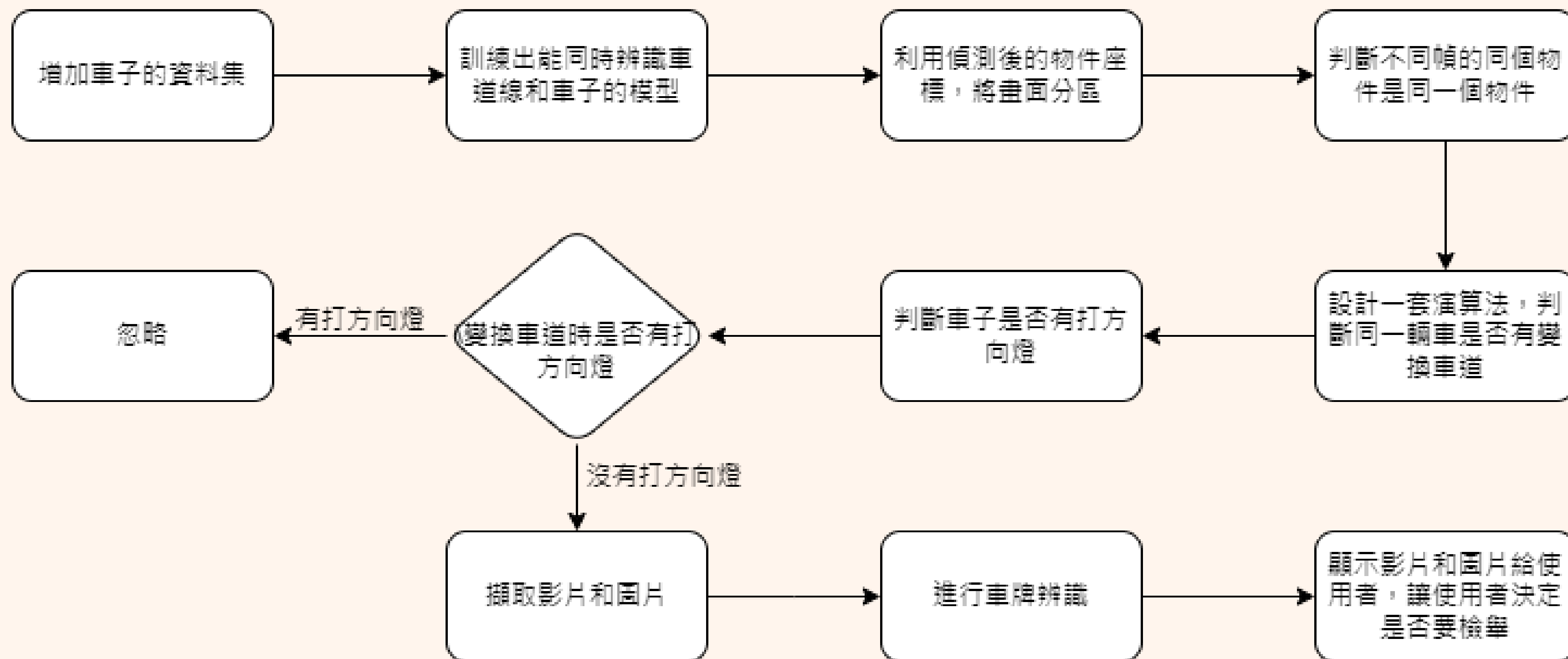
### 行車紀錄器之道路違規行為偵測

選擇檔案 car.mp4

上傳影片



# 未來規劃





# 心得

楊貽婷：一開始處理影片的方式是透opencv的套件，雖然有看到結果，卻不近人意；因此小組共同討論後決定要自行訓練模型，網路上關於訓練模型的資料還是比較少的，因此我們摸索了一陣子，也蒐集許多資料，最後才做出我們想要的東西，這個過程是非常值得挑戰而且有成就感的一件事。

葉雨沁：近期學習到如何自己訓練一個模型，發現其實訓練模型本身並不困難，困難的是資料集的收集過程，不太能夠自動化，因此我們只能一個一個手工標上去，困難的點還有要去猜測模型沒有辨識正確的原因是甚麼，我覺得訓練模型是之前到現在最好玩也最有成就感的一個部分。

李怡萱：寫網頁不會有除錯功能，必須一行一行去找問題，而檔案和檔案之間都彼此相互影響，所以要確認是否呈現的網頁是符合預期的。