



中原大學資訊工程學系

專題實驗 成果報告

## 基於行車影像之危險駕駛行為偵測

指導教授：吳宜鴻 教授

資訊四乙 11027203 葉雨沁

資訊四乙 11027213 李怡萱

資訊四乙 11027221 楊貽婷

# 目錄

摘要 .....	3
動機與目的 .....	4
使用工具 .....	6
系統架構圖 .....	7
系統設計與方法 .....	9
專題成果 .....	12
討論 .....	16
未來展望 .....	19
參考資料 .....	21

# 摘要

本專題開發的系統透過行車影像自動偵測前方車輛的危險駕駛行為，目的是在日益攀升的交通事故中為駕駛提供輔助警示，特別針對新手駕駛在遇到周遭車輛未打方向燈的狀態下，突然變換車道容易出現緊張反應不足的情況。此系統的核心功能包含能夠**判斷車道線座標、偵測車輛改道行為與檢測方向燈閃爍情形**，並在偵測到車輛有變換車道的行為發生時自動**擷取影像片段**，作為後續分析和風險管理的依據。系統設計包括多種模組化功能，讓用戶能輕鬆上傳影像並查看偵測結果。

# 動機與目的

## 動機

台灣的交通事故率近年來持續攀升，尤其是對新手駕駛帶來了嚴重的安全隱患。許多事故源於駕駛對周遭車輛的行為反應不及，特別是在未打方向燈的情況下，周圍車輛突然變換車道，此行為容易讓新手駕駛感到驚慌而降低反應速度，這種情形增加了發生追撞以及其他交通事故的風險。為了降低因駕駛操作不當而導致的意外，本專題旨在開發一套基於電腦視覺與影像辨識結合的「危險駕駛行為偵測系統」，以降低事故發生率。

## 目的

此系統的核心目的是辨識前方車輛的危險行為，並透過行車影像支持多項行車輔助功能。具體而言，本專題的研究目標包括以下幾點：

- 1. 準確辨識車道線位置：**通過影像處理技術，融合相異時間點之車道線檢測結果，並利用特徵提取技術生成連貫車道線影像，確保車道位置準確標記。
- 2. 偵測車輛改道行為：**系統通過 YOLO v8 及物件追蹤技術，實時監控車輛行進路徑，並根據車輛與車道線的交集位置判斷是否存在跨越車道線的行為。當出現跨越車道的行為，系統將視為潛在的危險行為並進行記錄。

3. **偵測方向燈閃爍狀態**：針對車輛方向燈的亮度變化進行分析，判斷方向燈是否有閃爍情形，以便判斷車輛的改道行為是否存在危險性。此功能在前方車輛無方向燈提示而突然改道的情況下極為重要。
4. **擷取危險行為片段**：當偵測到潛在危險行為時，系統將自動擷取並儲存該時刻的影像片段，供後續分析或作為行車證據，便於駕駛者檢視危險行為片段。
5. **網頁操作界面**：設計直觀的前端界面，方便駕駛者或相關人員上傳行車影像進行分析，並在偵測到的危險行為發生後能即時查看結果。V

# 使用工具

本專題的系統開發技術與工具如下，涵蓋前端、後端、伺服器等，以確保系統的準確度與可用性：

## 前端

**HTML、CSS、JavaScript**：用於建構直觀的網頁操作界面，方便使用者上傳行車影像並查看分析結果。

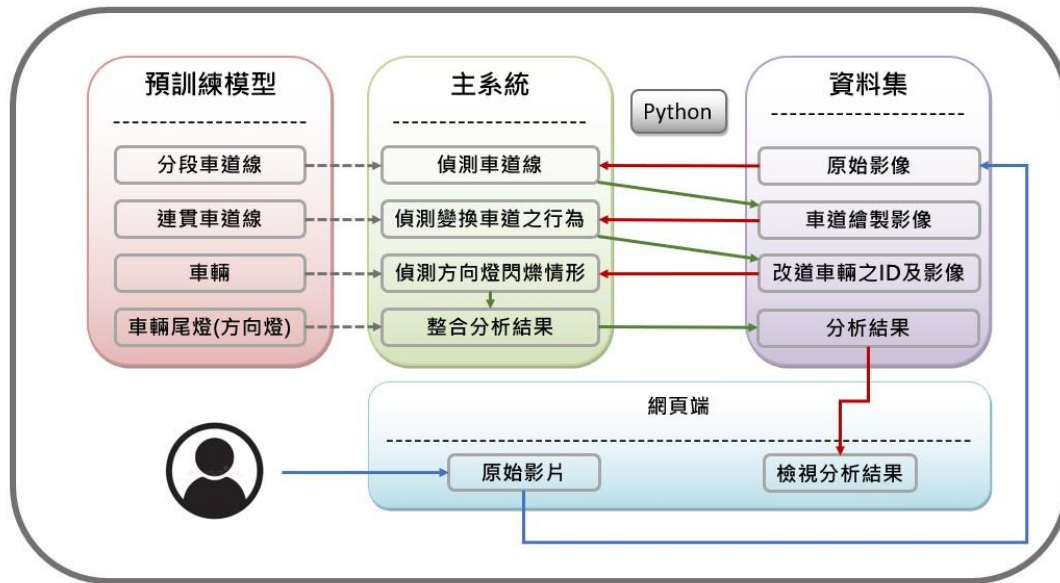
## 後端

1. **Python**：作為後端程式主要使用的語言，負責處理影像數據並進行危險行為的偵測分析。
2. **YOLO v8**：基於深度學習的物件偵測技術，用於車輛、車道線和方向燈的精確定位與物件追蹤。
3. **OpenCV**：用於影像處理和特徵提取，支援影像融合、車道線標記、方向燈檢測等多項操作。

## 伺服器

1. **VS Code**：主要的程式開發環境，用於編寫與維護程式碼。
2. **Anaconda**：管理環境和相關套件，確保深度學習與影像處理的兼容性。
3. **Labellmg**：標記工具，用於建立 YOLO 訓練所需的資料集，精確標記車道線和車輛位置。

# 系統架構圖



圖一、系統架構圖

1. **預訓練模型：**這部分包含多種模型，用於識別不同的車輛和道路特徵，包括分段車道線、連貫車道線、車輛、以及方向燈。
2. **主系統：**主系統負責執行多項偵測任務，包括：
  - ◆ **偵測車道線：**使用預訓練模型的輸出來偵測道路的車道線。
  - ◆ **偵測變換車道：**根據車輛對應之車道線的位置變化以及觸及車道線之先後順序，判斷是否發生變換車道之行為。
  - ◆ **偵測方向燈閃爍情形：**辨識車輛方向燈的狀態，以確認方向燈狀態。
  - ◆ **整合分析結果：**將各項偵測結果整合，產生最終的分析結論，並傳遞至資料集模組。

3. **資料集：**用資料集儲存原始影像、繪製車道線影像、變換車道的車輛 ID 及影像、以及最終的分析結果。主系統的各项偵測結果會儲存到資料集中，便於後續的檢視和使用。
4. **網頁端：**提供使用者介面，允許使用者查看系統的輸出結果，包括原始影像和檢視分析結果的影像。使用者可以透過網頁介面輕鬆瀏覽系統偵測到的變換車道與方向燈使用情況等分析結果。

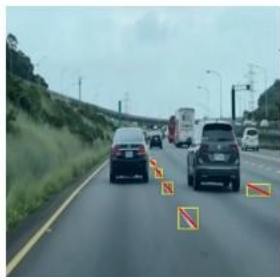


# 系統設計與方法

## 1. 車道線 影像融合

現階段多數辨識車道線的方法為利用遮罩過濾色階後設定 ROI，再使用 OpenCV 識別車道線，然而此作法有諸多缺陷，如：受限於固定點位之車道(左右各一條)、環境光線變化影響遮罩過濾能力、車道線不連貫等，因此我們設計一套辨識車道線的演算法，有效提高辨識車道線之準確度。

首先，我們蒐集相異時間、地點、天氣的行車影像素材，標記車道線精確位置同時賦予物件類別，並利用優於 CNN 架構之 Yolo v8 神經網路訓練物件模型，模型檢測結果如圖二所示。接著我們融合相異時間點之檢測結果於每幀影像中，形成類連貫車道線，如圖三所示。最後利用特徵提取技術辨別類連貫車道線之特徵，從而取得完整的連貫車道線，如圖四所示。



圖二



圖三



圖四

## 2. 方向燈 偵測閃爍情形

道路上行駛的車輛種類繁多，車燈類型極多，加上車尾燈的範圍面積較小，分析時的解析度過於低下，致使提取方向燈範圍的難度大幅提升。因此，我們採用類似於 R-CNN 之技術，首先提取車輛範圍區域以減少周遭雜訊干擾，並利用卷積神經網路從此區域提取車燈特徵，從而確認車燈的精確範圍，接著利用方向燈閃爍之特性偵測擷取範圍之亮度變化，Status\_1 表示亮度較前一幀影像亮，Status\_0 表示亮度較前一幀影像相差無幾，Status\_-1 表示亮度較前一幀影像暗，透過觀察數值變化以判定方向燈閃爍狀態。(如圖五~圖七所示)



圖五



圖六



圖七

### 3. 變換車道 行為偵測演算法

為偵測影片中之車輛是否變換車道，我們利用 Yolo v8 及 dot sort 物件追蹤系統訓練連貫車道之模型，並結合車輛模型設計一套演算法偵測此行為。首先，我們使用 Yolo v8 的 track 模式，賦予各物體獨立 id，並記錄每輛車的判定點是否在相異時間點與同一車道相交，若此情形成立，表示此行為發生，反之則否。  
(如圖八~圖九所示)



圖八



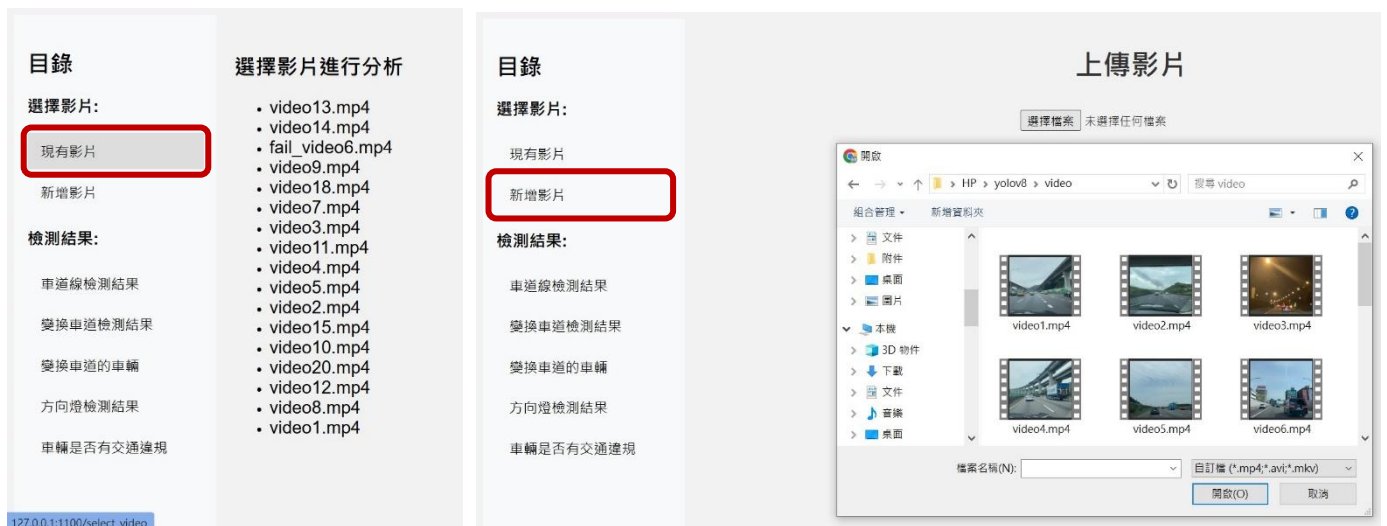
圖九

# 專題成果

## ● 首頁



## ● 選擇範本影片/上傳影片



## ● 預覽影片並進行分析

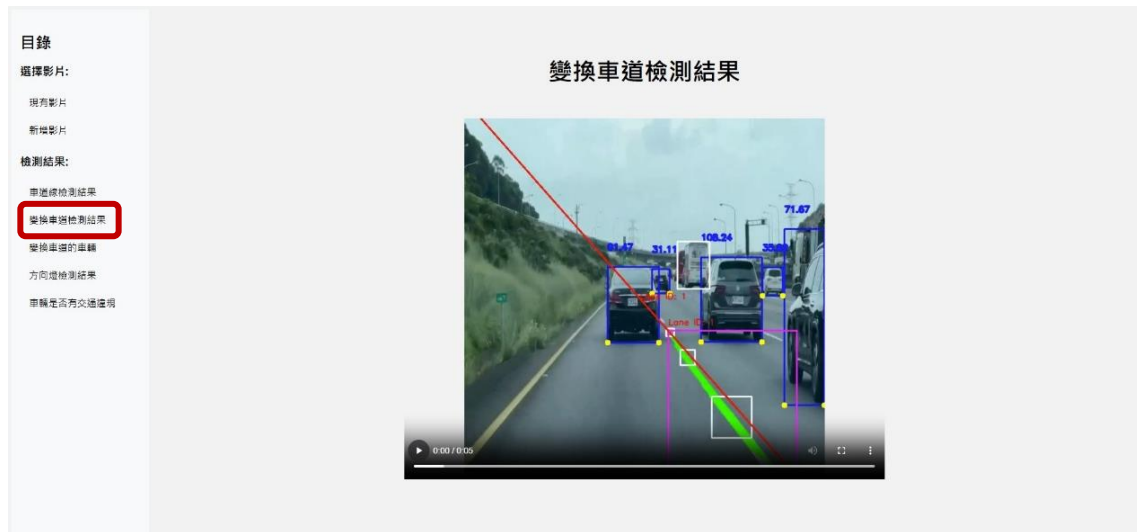


- **車道線檢測結果：**頁面顯示的是系統偵測出的車道線影片，透過可視化標示的綠色線條表示車道邊界。該綠色線條作為後續判定車輛是否發生變換車道行為的基準線，一旦車輛跨越此基準，系統即會偵測並記錄該變換車道行為。



- **變換車道檢測結果：**頁面顯示的是系統檢測車輛是否進行變換車道的影片。

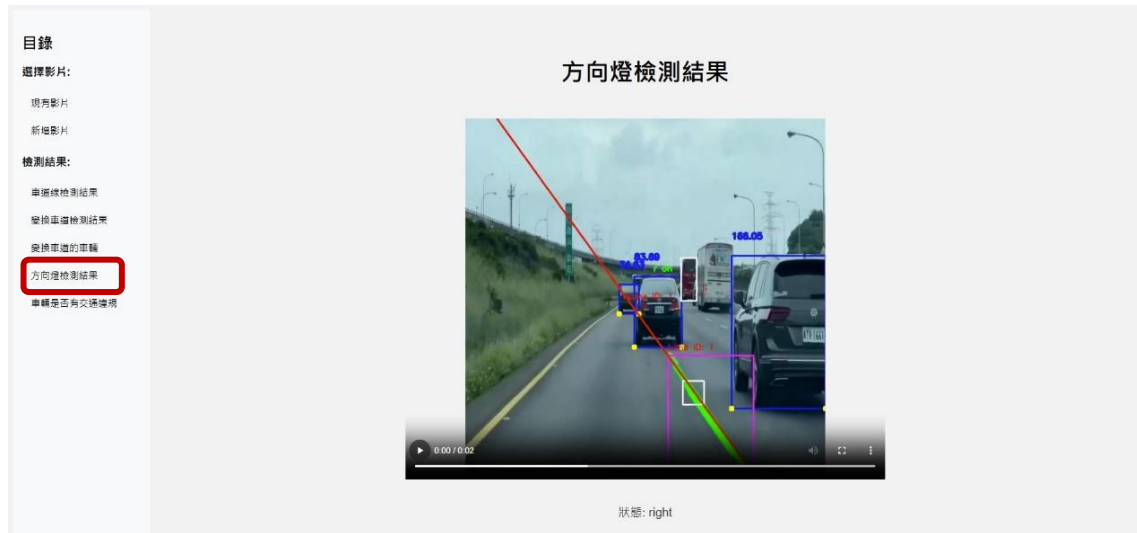
在畫面中，車輛被藍色方框標記，方框下方的兩個黃色點代表車輛的輪子位置。當這兩個黃色點同時跨越紅色基準線時，系統即判定該車輛已進行變換車道的行為。



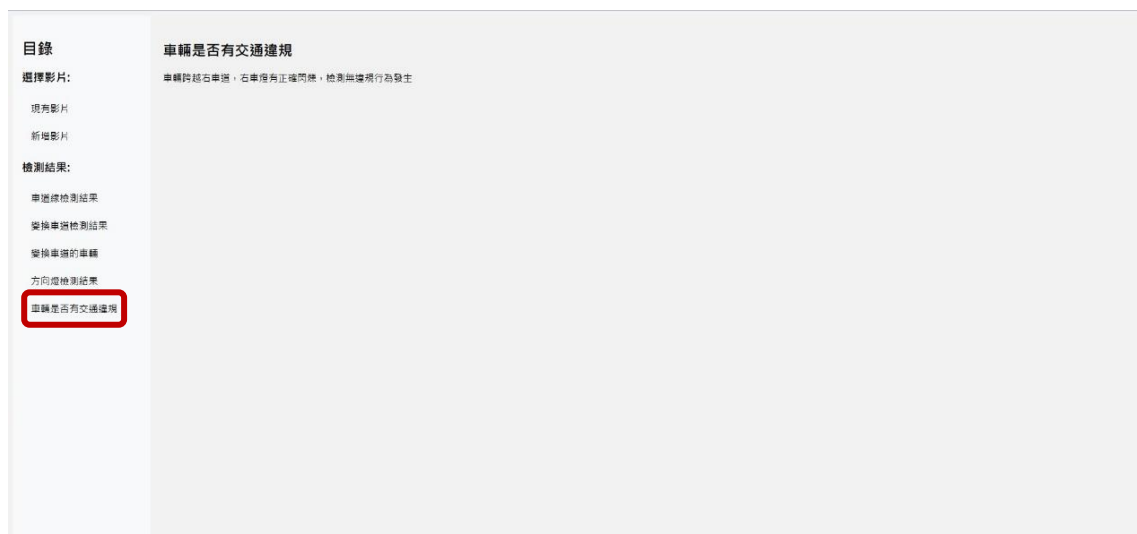
- **變換車道的車輛：**這個頁面顯示的是那台變換車道的車輛。



- **方向燈檢測結果：**頁面顯示系統在偵測到車輛變換車道後，進一步檢測該車輛是否啟用方向燈。影片中，若系統判定右側方向燈亮起，則會在車輛右上方顯示「ron」標記；若為左側方向燈亮起，則顯示「lon」，以此即時標示方向燈的使用狀況。



- **車輛是否有交通違規：**頁面顯示已發生變換車道的原始影片片段，並在旁側展示該違規車輛的照片。如未偵測到違規行為，則僅顯示「未偵測到違規行為」的提示文字。



# 討論

## 1. 方向燈偵測準確度不如預期：

最初的方向燈偵測方法是基於兩張連續影像中的亮度變化。具體來說，對每一組連續影像，計算方向燈區域的平均亮度，並比較前後兩幀的亮度比值。

如果後一幀亮度相對前一幀有明顯增加，則認定方向燈已亮起。

這種方法設置了一個固定的亮度增量閾值 ( `brightness_increase_ratio` )，

例如 1.03。當亮度的變化比值超過這個閾值，就會判斷方向燈亮起。然而，

固定閾值在不同情況下會有局限性：

- 各組影像的環境光線、相機曝光度可能不同，影響亮度基準。
- 在某些環境光強烈的情況下，細微的亮度變化可能被放大，導致誤判。
- 反之，在陰暗環境中，即便亮度變化明顯也可能不足以超過閾值，造成漏判。

為了克服環境和亮度差異的影響，後續設計了一個**動態亮度增量比率**

( `dynamic_brightness_ratio` )。此方法考量整組影像中方向燈亮度的變化

範圍，並自動調整閾值。具體來說，根據影像組中的亮度數據，計算一個動

態的基準增量比率，使得該比率可以適應不同環境的光照變化，達到更靈活

的判斷標準。



這種動態比率有助於提高準確度，特別是在光線不穩定或變化劇烈的情況下，能更準確地識別方向燈的亮起。但此方法也有其限制，主要表現在以下幾個方面：

1. **環境光源影響**：如果光源或外部環境亮度突然改變，動態比率可能過於敏感或不敏感，從而導致誤判。例如，在有陽光照射的路面，車體反光可能使亮度突然增加，動態比率無法區分是方向燈亮起還是環境光源變化。
2. **亮度增量差異不足**：如果連續影像中亮度變化不明顯，無論是固定或動態比率都難以判斷方向燈的狀態。特別是在陰暗情境下，方向燈的亮度增量可能不足以超過基準比率。
3. **過度調整**：動態比率基於整組影像計算，若影像中大部分的亮度變化都較大或較小，動態比率可能因此失去彈性，無法正確地檢測到較小幅度的亮度變化。

動態比率的方法有效改善了固定比率方法的適應性，但由於外部光源及環境亮度的快速變化仍可能導致判斷不準確。未來可考慮進一步引入影像對比度、自適應亮度區間等方式，或使用更複雜的模型來辨識方向燈的亮起，以提升準確度。這些進一步改進可能需要更多的計算資源與調校，並根據不同車輛的方向燈亮度特徵進行優化。

## 2. 車道線準確度

我們判定車道線的第一種方法是利用過濾色階結合霍夫轉換，並描繪出車道線的位置，但過濾色階時容易將與車道線顏色相近的物體一併納入，因此霍夫轉換時會連同非車道線之物體進行特徵提取，造成車道線判斷錯誤，因此我們更改成 yolo v8 結合特徵提取技術以確認車道線的位置。

## 3. 車牌辨識

專題開始時，我們希望能辨識變換車道的車輛之車牌號碼，但由於影像畫質有限，加上距離變化過大，因此無法成功辨識車牌號碼，希望未來有機會能增強影像處理技術，再加入此功能。

# 未來展望

我們認為本專題的系統具備相當大的發展潛力，隨著科技的進步，車聯網、人工智慧等技術不斷成熟，我們的系統未來將能夠與這些技術進一步結合，不僅提升系統的偵測準確性與反應速度，還有機會應用於更多元的駕駛場景。基於此，本專題的未來發展方向涵蓋了技術整合、功能拓展以及商業化應用，旨在提高駕駛安全性並為智慧交通發展做出貢獻。

## 1. 與車聯網技術 ( V2V、V2X ) 深度整合：

結合車輛之間的即時通訊，讓車輛自動傳遞危險行為資訊，達到更高的交通安全效果。

## 2. 提升偵測準確性與反應速度：

不斷優化系統的偵測演算法，提升對各種危險行為的準確辨識與即時反應能力。

## 3. 擴展到其他車輛行為偵測：

除了變換車道行為，未來可以拓展系統功能，例如偵測急煞、過速行駛、未保持安全車距等危險行為。

## 4. 支援多種車型與場景：

調整系統適應不同種類的車型及交通場景 ( 如高速公路、城市道路 )，增強系統的通用性。

## 5. 導入人工智慧與深度學習技術：

利用 AI 技術來學習並預測更多複雜的駕駛行為，讓系統能夠進一步自動學習和優化。

#### 6. 與行車記錄器和自駕技術整合：

系統可以與行車記錄器和自動駕駛技術相結合，提升自駕車的感知能力，確保自動駕駛時的安全性。

#### 7. 商業化與大規模應用：

未來有機會將此技術商業化，應用於汽車製造業者或保險公司，作為提高車輛安全性的賣點。

#### 8. 與智慧城市發展結合：

系統有潛力成為智慧城市交通管理的一部分，協助城市即時監控和分析道路交通狀況。

## 參考資料

[1] 以散射為基礎之方向燈偵測及以反射為基礎之方向燈識別技術：

<https://hdl.handle.net/11296/z4fvyk>

[2] 車道辨識之卷積神經網路架構設計：

<https://hdl.handle.net/11296/xu2uqm>

[3] 基於天氣變化適應閾值之車道辨識及前車距離偵測：

<https://hdl.handle.net/11296/rjh36v>

[4] PyTorch 自行訓練 YOLOv5 物件偵測模型教學與範例：

<https://officeguide.cc/pytorch-yolo-v5-object-egg-detection-models-tutorial-examples/>

[5] PyTorch 使用 YOLOv5 與 DeepSORT 進行車輛物件偵測、計算車流量教學與範例：

<https://officeguide.cc/pytorch-yolo-v5-object-car-detection-calculate-traffic-flow-tutorial-examples/>

[6] 網頁資料：

<https://developer.mozilla.org/zh->

[TW/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://ithelp.ithome.com.tw/docs/Learn/Getting_started_with_the_web/CSS_basics)

<https://ithelp.ithome.com.tw/users/20152424/ironman/5659?page=3>

[7] 影像辨識的處理流程：

<https://ithelp.ithome.com.tw/articles/10262249>

[8] yolo 是什麼？

[https://ai4kids.ai/blogs/blog/what-is-](https://ai4kids.ai/blogs/blog/what-is-yolo)

[yolo#YOLOYou\\_Only\\_Look\\_Once%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC%E6%98%AF%E4%BB%80%E9%BA%BC%EF%BC%9F](https://ai4kids.ai/blogs/blog/what-is-yolo#YOLOYou_Only_Look_Once%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC%E6%98%AF%E4%BB%80%E9%BA%BC%EF%BC%9F)

[9] YOLOv8 介紹與手把手訓練自訂義模型

[https://medium.com/@EricChou711/yolov8-](https://medium.com/@EricChou711/yolov8-%E4%BB%8B%E7%B4%B9%E5%92%8C%E6%89%8B%E6%8A%8A%E6%89%8B%E8%A8%93%E7%B7%B4%E8%87%AA%E8%A8%82%E7%BE%A9%E6%A8%A1%E5%9E%8B-752d8d32cb73)

[%E4%BB%8B%E7%B4%B9%E5%92%8C%E6%89%8B%E6%8A%8A%E6%89%8B%E8%A8%93%E7%B7%B4%E8%87%AA%E8%A8%82%E7%BE%A9%E6%A8%A1%E5%9E%8B-752d8d32cb73](https://medium.com/@EricChou711/yolov8-%E4%BB%8B%E7%B4%B9%E5%92%8C%E6%89%8B%E6%8A%8A%E6%89%8B%E8%A8%93%E7%B7%B4%E8%87%AA%E8%A8%82%E7%BE%A9%E6%A8%A1%E5%9E%8B-752d8d32cb73)

[10] DeepSort:

[https://github.com/mikel-brostrom/yolo\\_tracking](https://github.com/mikel-brostrom/yolo_tracking)

[11] 詳細說明 yolov8 實現目標追蹤：

[https://blog.csdn.net/qq\\_42452134/article/details/135241059](https://blog.csdn.net/qq_42452134/article/details/135241059)