

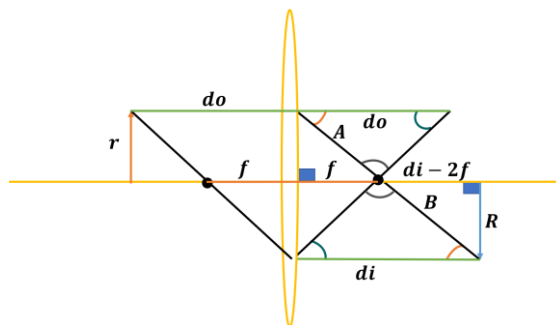
109590015 楊挺煜

關鍵詞：YOLOv7[2]、車載系統、行車安全、PyQt[1]

在這個人口與科技快速進步的時代下，世界各地的用路人與日俱增，尤其在台灣一個人口密度極高的國家，交通意外事件層出不窮，每天都有事故在道路上上演，跟車過近、會車擦撞、未禮讓行人等意外頻繁發生，讓台灣人常常自嘲我國的的駕照猶如「用雞腿換來的」，為了解決此況，我們用YOLOv7 [2]這個物體檢測模型，嘗試從道路上偵測到的物件獲取更多資訊，例如：距離、出現時機等，並加以針對不同情境做出反應，達到警示駕駛、增加道路安全的效果。

本專題使用 PyQt[1] GUI框架來實現一個跨平台的圖形用戶界面，並以YOLOv7[2] 模型為基礎，實現行車安全監測功能。該系統可以對車輛進行偵測和距離估計，並根據設定的安全距離閾值警示駕駛。在實際行車安全應用中，本系統可以提高駕駛員的駕駛安全性和駕駛體驗。

具體而言，設計包括三個主要部分：GUI界面、YOLOv7[2]模型和安全監測模塊。GUI界面通過PyQt[1]框架實現，提供了直觀的用戶操作界面。YOLOv7[2]模型用於車輛偵測和距離估計，並且可以實現警示車輛安全距離的功能。安全監測模塊可以根據設定的安全距離閾值，對行駛中的車輛進行監測，並在距離過近時發出警示。



距離估計由於我們可以透過採用以YOLOv7[2]辨識出物件的Bounding Box之Width與Height，再透過如上圖所示的物理成

像原理可得 (1)，再由 (1) 推估目前的物件與鏡頭的距離。

$$(1) \quad d = f + \frac{R}{r}$$

#### 四、使用技術方法

##### (一) 輸入輸出端

PyQt[1]是一個基於Python的UI 框架，可以用來設計和實現跨平台的圖形用戶界面應用程序。

##### (二) 影像辨識

YOLOv7[2]是一種基於深度學習的物體偵測模型，採用擴展的高效層聚合網路 E-ELAN 以及將設計好之模型進行複合模型縮放用以將模型架構優化，可以實現高效快速地實現對圖片或影片中物體的檢測和識別。

##### (三) 影像處理

###### ● 夜間模式

使用了Laplacian運算子進行邊緣銳化後再傳入YOLOv7模型中進行車輛偵測。

###### ● 車道線偵測

先將畫面中的影像進行降噪和增強對比，再用Canny邊緣偵測[5]後用霍夫轉換[6]將車道線偵測出來，並在畫面中設置roi區域(Region Of Interest)，當偵測到的車道線進入此區域時，發出警告提醒駕駛。

#### 五、工具說明

##### (一) Visual Studio Code

Visual Studio Code是一款由微軟開發且跨平台的免費整合式開發環境。該軟體支援語法突顯、程式碼自動補全（又稱 IntelliSense）、程式碼重構功

能，並且內建了命令列工具和 Git 版本控制系統。

##### (二) Qt Designer

PyQt[1]是Python語言的GUI開發工具之一。可以用來代替Python內建的Tkinter。其它替代者還有PyGTK、wxPython等。與Qt一樣，PyQt[1]是一個自由軟體。

#### 六、結論與未來展望

這次的研究我們發現，跟之前使用過的YOLOv4對比，YOLOv7[2]在物件辨識成效上增強許多，能更精確的辨識物件好讓我們得到想要的資訊。

目前研究進度基本功能已實作完畢，可以偵測並辨識道路上的物件，算出每個物件與攝影機的距離，車道偏移功能也能正常運作，不過實際應用時要根據車上鏡頭拍攝的角度(例如是否會拍到引擎蓋，或是視角不同導致路面成像不一樣)以及行駛路面去對邊緣偵測以及ROI的參數做調整，以達到最好的效果，未來會朝檢測效率以及參數調整的方便性改進，也有考慮要新增其他功能(坑洞位置回報[3])，讓這個專題整體更加完整，達到實質的用路安全效果。

#### 七、參考文獻

[1] HOWARD WENG, "如何建立 PyQt 的系統," 嗡嗡的隨手筆記, <https://www.wongwonggoods.com/all-posts/python/pyqt5/pyqt5-5/>, Aug.2021 (accessed Feb.2023)

[2] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebie

s sets new state-of-the-art for real-time object detectors," Github, <https://github.com/WongKinYiu/yolov7>, arXiv preprint arXiv:2207.02696, 2022.

[3] Sovit Rath, "Fine Tuning YOLOv7 on Custom Dataset" LearnOpenCV, <https://learnopencv.com/fine-tuning-yolov7-on-custom-dataset/>, Aug.2022 (accessed Apr.2023)

**[5] Chin-yu Chien, "Canny edge detector 實作 (OpenCV) "** <https://medium.com/@pomelyu5199/canny-edge-detector-%E5%AF%A6%E4%BD%9C-opencv-f7d1a0a57d19>, Aug 4, 2018

[6] Tomasz Kacmajor, "Hough Lines Transform Explained" <https://medium.com/@tomasz.kacmajor/hough-lines-transform-explained-645feda072ab>, Jun 6, 2017