

國立台北大學資訊工程學系專題報告

基於深度學習與電腦視覺技術之道路異常偵測系統

專題組員: 李昇謙、沈銘浩、李俊賢、戴安廣

專題編號: PRJ-NTPUCSIE-110-007

執行期間: 2021 年 8 月 至 2022 年 5 月

1. 摘要

交通工具的普及，使現今社會的交通更為順暢。但在步入交通發達的同時，各種交通問題也隨之而生，各式各樣的交通問題層出不窮。造成交通異常的狀況眾多，此研究將道路上的交通異常歸類為兩種，分別為車輛偏離正常道路，以及發生於道路中的車輛碰撞與停止，藉此針對不同的異常情況進行偵測。結合上述兩種的交通異常分類，此研究涵蓋了絕大多數的交通問題，藉此達到讀取影像自動辨識交通異常的目的。

2. 簡介

隨著時代的演進，交通工具早已融入了我們日常生活的一部分，但這也同樣意謂著將會衍伸出許多的交通問題。因此，如何去防範並減少交通相關問題，也成為我們現代人無可避免的課題。對於迎面而來的交通問題，紀錄下發生車禍的時間與地點，可說是最有效的做法之一。藉由尋找出較頻繁發生事故的路段與原因，協助我們對於交通問題的事後分析與改善。同時，記錄交通事故的發生時間，也便於我們事後調查車禍相關事宜，無須花費額外的人力與資源，能夠進行更有效率的搜索。因此，我們對於

改善交通異常的發生提出了此研究。此研究將透過讀取道路監視器的影像，偵測出道路異常情況的發生，並記錄車禍相關的時間資訊。

此研究針對紀錄交通異常提出了兩種解決方案，第一種採取檢測車輛是否偏離正常道路的做法，利用刮選出所需範圍的道路 mask，藉此檢測當前車輛所行使的地方是否不合理。此作法能夠準確的判斷車輛的行駛軌跡，然而對於道路間的異常與車輛碰撞卻無法進行有效的辨識。為此，我們提出了第二種解決方案，利用車輛對應的座標與像素值，檢測道路中的車輛是否發生碰撞產生停頓，用此方式來補足的前者的檢測盲區。

3. 專題進行方式

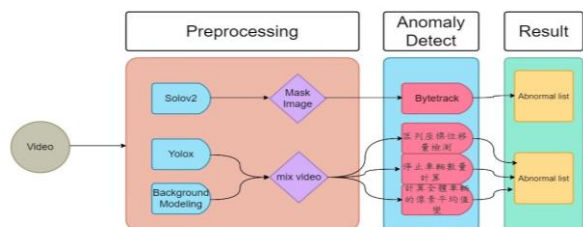


Figure 1 交通異常檢測架構圖

3.1 交通異常判別架構

此研究的交通異常檢測架構圖如 Figure. 1. 所示。我們對於檢測交通異常

提出了兩種檢測方法，分別是 Figure. 1. 上半部的車輛軌跡判定與下半部的車輛碰撞靜止判別。並且我們將每個檢測方法，各別分成前處理、偵測異常與輸出結果三個階段。

車輛軌跡判定的部分，前處理我們採取建置道路的 mask，並選用 SOLOV2 來刮選出車輛正常行駛的道路。接著在偵測異常階段，我們使用 ByteTrack 來追蹤車輛的行駛路線，檢測行駛路線異常的車輛。最後，輸出結果階段則是將超出正常道路的車輛時間做紀錄。

車輛碰撞靜止判別的部分，模型使用 YOLOX 與 BackgroundSubtractor。前處理階段，我們利用 YOLOX 與 BackgroundSubtractor 產生兩部影片。接著，偵測異常階段則是利用產生的兩部影片去進行三種不同的判別方式，並將三種不同的方式進行交叉比對，藉此排除單一方法的錯誤偵測，提升辨識異常的可信度。依序分別是匡列座標位移量(YOLOX)、停止車輛數量計算(YOLOX+BackgroundSubtractor)以及計算全體車輛的像素平均值變化(YOLOX+BackgroundSubtractor)，檢測道路內車輛的異常狀況。最後是輸出結果階段，將同時符合三種辨識方式的異常車輛時間做紀錄。

最後，由於兩種檢測方法能夠補足彼此之間的檢測盲區，引此我們將上述兩種檢測方法的結果進行整合，統整出最終的交通異常資料。

3.2 車輛碰撞靜止判別

3.2.1 模型與權重檔選擇

我們所選擇的模型為 YOLOX_s，其原因是它的速度最快，再加上模組與其他的相比較為輕量化，能夠更容易的使用，較符合此研究未來預期在即時功能上的需求。雖然在精確性不如其他模型(如 YOLOX_m 及 YOLOX_x)，不過經過我們的測試後發現其依然符合我們的需求。另外我們也對例如 YOLOX_m 及 YOLOX_x 等模型進行測試後，發現在精確度上的進步遠比不上在速度上的犧牲、因此最終選擇使用 YOLOX_s。

權重檔的部分，我們選擇了官方所提供的預設 YOLOX_s 權重檔，預設權重檔是基於 coco dataset 所訓練出來的權重。我們的偵測目標是一般車輛以及卡車、在 coco dataset 中有非常充足的樣本、因此可以提供相對精確的偵測，我們也嘗試過自行訓練出權重檔發現遠不如官方的效果、決定直接使用官方權重。

3.2.2 前處理

首先要分成兩部影片來處理、因為一邊需要 YOLOX 去獲得車輛座標位置資訊、另一邊需要 backgroundsubtract 去取得像素質的數值，順序是先透過 YOLOX 找出車輛的位置再針對框選的座標來計算需要的像素質、最後會把資訊集中在經過 YOLOX 處理後的影片上、透過兩部影片的資料整合來達到我們偵測的目的、也可以避免不必要的影響。

3.2.3 匡列座標位移量檢測(YOLOX)

使用 YOLOX 模型所產生的影片來

偵測交通異常。利用 YOLOX 框選車輛的四個點取平均，藉此找到位於 box 中心的車輛中心點，以作為後續的位移判斷基準。接著，我們將每一幀內所檢測到的車輛中心點紀錄至陣列 recordxy 中，藉由計算中心點的位移量來判斷當前車輛是移動車輛還是靜止車輛。

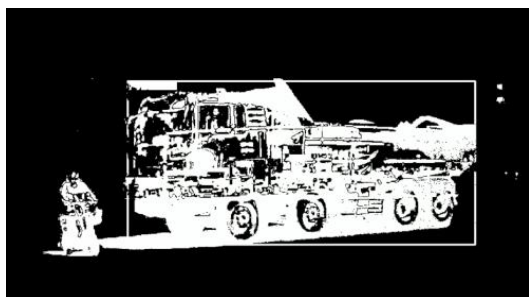


Figure 2 車輛移動

由於信心指數的浮動會使框選座標產生晃動，導致中心點移動而產生位移，因此我們需要將位移量的基準調大。而當設定的位移量基準調大時，又會導致移動較慢的車輛被視為靜止車輛。於是我們提出與前幾幀進行位移量計算，來解決前面提到的兩種問題。我們最後選擇的是，每一幀與其對應的前十幀進行位移量計算，並且當位移量大於 5 時，才將其視為移動車輛，反之視為靜止車輛。

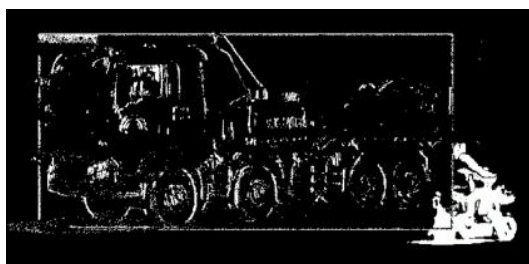


Figure 3 車輛靜止

在正常行駛的交通影片中(此研究撇除等待紅綠燈與路邊停車的狀況)，靜止的車輛數只會維持或減少，並不會增加，因此當靜止車輛數增加

時，則代表當前可能發生交通異常。因此我們記錄每一幀的靜止車輛數，當靜止的車輛數增加則視為發生交通問題並記錄時間。

3.2.4 停止車輛數量計算 (YOLOX+BackgroundSubtractor)

利用模型 YOLOX 與 BackgroundSubtractor 所產生的影片來偵測交通異常。如 Figure. 2.與 Figure. 3.所示，當車輛移動時，其像素值會變亮，而當車輛停止時，其像素值則會變暗。

我們將每輛車對應的框選範圍(box)進行像素值的平均值計算，假如單一車輛對應的平均像素值減少至 100 以下時，便將該車輛視為靜止車輛。接著同樣利用匡列座標位移量的停止車輛數變化方式來檢測是否當前發生交通異常。

為了避免因為 YOLOX 信心指數而導致框選區域晃動，進而影響判斷，於是我們設定條件需要連續五幀皆發生停止車輛增加才會視為發生異常。而又因為 BackgroundSubtractor 處理的影像，其像素值變化需要時間，所以停止的車輛需要數秒的時間來使像素值降低，因此我們將與前幾幀進行比較。最終我們選擇讓每一幀與其對應的前 30~34 幀進行檢測，並且當連續五幀皆發生停止車輛增加才視為發生異常並記錄時間。

3.2.5 計算全體車輛的像素平均值變化 (YOLOX+BackgroundSubtractor)

利用模型 YOLOX 與

BackgroundSub-tractor 所產生的影片來偵測交通異常。此方法利用停止車輛數量計算的每台車像素平均值來作運用，將每台車的像素平均值相加除去總車輛數，則會得到當前這一幀的像素平均值。

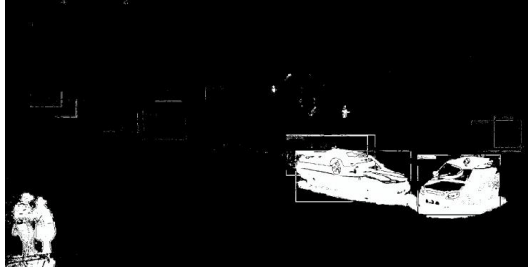


Figure 4 發生交通異常前

$$\text{difnumber} = \frac{255}{\text{caramount} * 2.5} \quad (1)$$

而發生交通異常時，車輛會發生靜止，進而導致平均像素值的降低。如 Figure. 4.與 Figure. 5.所示，發生交通異常前後的像素值會產生變化，原本的像素值來源為移動車輛。而發生交通異常後，該幀會失去像素值的提供來源。而使得 difnumber 數值下降。此方法利用到 BackgroundSubtractor，同樣的需要時間讓像素值做遞減，因此我們將當前幀與其對應的前 45 幀去做比對，同樣當連續五幀符合條件便被視為發生交通異常。



Figure 5 發生交通異常後

3.2.6 紀錄交通異常時間

由於車輛從移動到靜止時，像素值的變化屬於遞減的狀態，所以需要

數秒的時間像素值才能停止變化。因此，停止車輛數量計算及計算全體車輛的像素平均值變化的異常時間偵測會比實際還來的慢。而匡列座標位移量檢測僅根據車輛的座標位移進行檢測，所以異常時間較接近真實狀況。

因此我們選擇匡列座標位移量檢測作為第一層篩選，停止車輛數量計算及計算全體車輛的像素平均值變化為第二、第三層。並針對異常時間的誤差，將判定條件設定為以第一層的時間為基準，往後 0.5~3.5 秒的時間去與二、三層的結果取交集。當三層的時間接互相吻合才將其判定為車輛碰撞靜止的交通異常時間。

3.3 車輛偏移異常判定

3.3.1 模型選擇

在前處理階段，起初是以 YOLOX 來當做生成 mask 的基礎模型，但由於 YOLOX 所偵測出的 bounding box 為矩形，除了車子本身在畫面中所占的面積，還會多框選其餘非車子的部分，進而影響生成的 Road-Mask，因此為了可以生成更精準的 Road-Mask，我們選擇在分割目標物有著更佳表現的 SOLOV2 作為我們前處理的模型。



Figure 6 SOLOV2

在物體追蹤的階段，由於有許多經典演算法都能夠處理 MOT(Multiple Object Tracking)挑戰，因此我們最

初是選擇 DeepSORT 演算法作為我們在物體追蹤的基礎模型，而該演算法是在 2021 年 9 月之前表現最佳的演算法。但經過我們實測該演算法在交通場景的應用上，我們發現了幾個問題：首先是物體較小或模糊，DeepSORT 無法順利追蹤。再來是若車子之間有遮擋的狀況發生，DeepSORT 會在遮擋發生之後，建立新的追蹤，導致同一台車有著兩次的追蹤，且 DeepSORT 會將之視為兩個相異的物體。

由於上述的問題，我們選擇了 ByteTrack 做為新的追蹤模型。ByteTrack 是於 2021 年 10 月發布的新物體追蹤演算法，其特性包含了將高低分物體分開處理、設立多道門檻針對低分物體做處理、利用低分物體與軌跡之間的相似性，挖掘真正的物體、對於沒有匹配上的軌跡會保留 30 幀，直到再次出現時進行匹配。上述的特性不僅可以解決 DeepSORT 追蹤上的誤差，在框選物體的範圍上，精準度更勝於 DeepSORT，綜合上述的討論，我們決定將 ByteTrack 作為我們追蹤任務上的最終模型。

3.3.2 前處理(建立 Road-mask)

在 SOLOV2 的模型中，若物體順利被偵測，其在畫面中所占面積下的布林值會設為 1，我們則透過此布林值做以下操作：

- (1)生成一個與原影片尺寸大小一致的陣列 M，初始值皆設為 0。
- (2)將物體兩幀之間的布林值做差集。
- (3)將差集過後布林值仍為 1 的點於陣列 M 的對應位置加 1。
- (4)陣列 M 中值加到 80 的點之像素值

設為 1。

- (5)將陣列 M 的最終狀態輸出成圖片，生成 Road-Mask。

透過以上的方式，我們可以將像素值為 1 的點定義為主要道路，並同時可以將長久停留的車輛排除在主要道路之外，避免後續處理上的誤差。

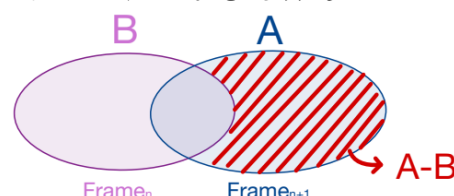


Figure 7 布林值差集

3.3.3 ByteTrack(偵測偏移異常)

將前處理所生成的 Road-Mask 圖片，疊合原影片，並偵測其車輛的 bounding box 是否與 Road-Mask 像素值為 0 之區域疊合超過 bounding box 的 70%且累計時間超過 60 幀(大約 2 秒)，若有則代表此車輛已超出主要道路，並記錄其車輛偏移異常之時間。



Figure 8 Road-Mask

4. 主要成果與評估

我們選擇 NVIDIA AI CITY CHALLENGE Track4 之評估標準，以 RMSE 的值作為在時間上的評估依據，從 RMSE 之定義來看，當值越小，代表越即時偵測到異常，表現較優異。

$$RMSE = \min \left(\sqrt{\frac{1}{TP} * \sum_{i=1}^{TP} (t_i^p - t_i^{gt})^2}, 300 \right)$$

TP: 偵測到的異常個數

t_i^p : 偵測到異常的時間

t_i^{gt} : 實際異常發生時間

表格 1 Comparison of teams under the RMSE standard

能夠觀察到我們在異常發生後，平均可於 2.2 秒內偵測到異常，在異常偵測的敏感度上，表現是優於 2021 AI CITY CHALLENGE 的其他參賽隊伍。

	Average RMSE	Rank
Our Result	2.2759	1
2021 ai city challenge rank 1 team	5.3080	4
2021 ai city challenge rank 2 team	3.1623	2
2021 ai city challenge rank 3 team	3.4039	3
2021 ai city challenge rank 4 team	8.4027	5

表格 2 RMSE evaluation standard



Figure 9 normal

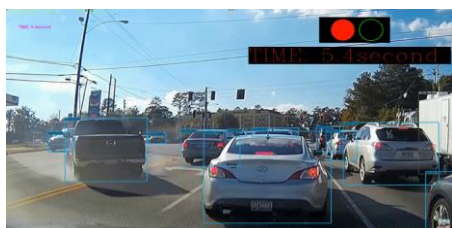


Figure 10 abnormal

5. 結語與展望

以一個交通異常偵測的工具而言，我們認為此研究能夠有效地去達成偵測的目標，並以即時性為目標進行改良。如此一來，便能夠搭配監視器達到即時通報車禍的功能，使得車禍的現場能迅速得到支援。或是加入資料庫整合的系統以分析各種車禍現象的資訊、讓專業人員做研究找出合

適的解決方案以減少臺灣的車禍現象，為臺灣的交通盡上微薄的一份心力。

6. 銘謝

感謝指導教授在我們有任何問題時提供需要的幫助和指導，也提供了相關設備還有實驗室的資源讓我們使用、以及許多專題相關的資訊來讓我們參考，尤其是 AI CITY 競賽的相關資訊，不只給了我們主題上的想法、還有很多相關的論文可以研究，非常感謝一路上幫助過我們的所有人。

7. 參考文獻

- [1] Zhao, Yuxiang, et al. "Good practices and a strong baseline for traffic anomaly detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- [2] Wu, Jie, et al. "Box-level tube tracking and refinement for vehicles anomaly detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- [3] Wang, Xinlong, et al. "SOLOV2: Dynamic and fast instance segmentation." *Advances in Neural information processing systems* 33 (2020): 17721-17732.
- [4] Zhang, Yifu, et al. "ByteTrack: Multi-Object Tracking by Associating Every Detection Box." *arXiv preprint arXiv:2110.06864* (2021).
- [5] Ge, Zheng, et al. "YOLOX: Exceeding yolo series in 2021." *arXiv preprint arXiv:2107.08430* (2021).