

Vehicle Speed Estimation Based on Monitor Frames

吳柏呈、蔡汶勳

國立臺南大學、臺南、郵遞區號:701、臺灣

摘要

交通事故每天都會發生，對員警而言，處理車禍事故最重要的任務是一事故責任釐清，由於當員警抵達車禍事故現場時已是事後，目前的路口監視攝影機雖能提供事故發生當下的影像來協助員警進行事故鑑定，但也只能提供定性描述，對於車輛是否超速，這種需要定量資訊的部份仍是力有未逮。有鑑於此種缺憾，本專題透過 AI 影像分析技術分析監視攝影機畫面，估測出車輛的移動速度，協助警方判斷肇事車輛是否超速，以明確釐清肇事車輛的法律責任。

關鍵詞：電腦視覺、YOLOv4、SORT、U-Net

Abstract

Traffic accidents happen every day. For the police, the most important task in dealing with traffic accidents is to clarify the responsibility for the accident. Since when the police arrive at the scene of the accident, it is already after the incident, although the current intersection monitor can provide the scene of the accident to assist the police in the accident identification, it can only provide a qualitative description. As to whether the vehicle is speeding, the part that requires quantitative information is still not available. In view of this shortcoming, this research uses AI image analysis technology to analyze the monitor frames to estimate the moving speed of the vehicle, assist the police to determine whether the vehicle is speeding, and clarify the legal responsibility of the vehicle.

Keywords: Computer Vision, YOLOv4, SORT, U-Net

前言

在車禍事故調查中，事故車輛是否超速這項因素在整起事件調查中具有關鍵性角色，目前事故調查人員可藉由觀看路旁監視攝影機的歷史錄影，了解事故發生的過程，但無法得知事故車輛當時的速度，對於事故責任的釐清尚缺乏具體數據。隨著人工智慧技術的蓬勃發展，透過 AI 技術從路旁監視攝影機的錄影畫面估測出車輛速度，已經成為可能。

近幾年已有一些透過單眼監視攝影機估測車輛速度的方法被提出，這些方法可歸納成三大類型：基於輔助線的估測方法[1]、基於透視投影的估測方法[2]以及基於車牌的估測方法[3]。第一種類型方法不需要相機校準，而是測量道路上兩條

或多條輔助線之間的實際距離，然後將距離估計問題轉為檢測問題。由於輔助線位在道路上，準確的距離估計涉及車身與輔助線接觸點的準確定位，且車身的連續接觸點位置應該要相同，以獲得對速度的一致性估算。然而由於透視投影限制和監視攝影機幀速率(frame rate)不足的問題，讓速度估測顯得複雜。第二種類型方法將單應性計算(homography computation)視為從 3D 投影空間平面(道路)轉至 2D 投影空間(相機圖像平面)的線性轉換過程，通過這種方式，影像可以轉換為鳥瞰圖(BEV)，其中的像素位移可以直接轉換為真實世界的位移距離。然而此種方法需要對監視攝影機進行內外參數校準，絕大多數路旁監視攝影機並未進行過校準，限制了此類方法的適用性。第三種類型方法乃基於對車牌真實尺寸知識的掌握，來推估車輛位移距離，再間接計算出車速。然而此種作法非常仰賴車牌擷取的精準度與相機的解析度，不論是擷取車牌的精準度不佳，抑或是相機的解析度不夠，都會嚴重影響車速估測的準確度。

本研究所提出的方法屬於第一種類型方法，針對既有方法[1]無法精準偵測車身與輔助線接觸點，導致速度估測誤差大的問題，本研究透過深度學習技術來提升車身與輔助線接觸點的偵測精準度，間接提升車輛速度估測的準確度，以協助員警在車禍事故發生後，釐清事故車輛是否超速的疑問。

1. 方法與流程

1.1 研究方法

本研究使用深度學習的 YOLOv4[4]模型進行道路上的車輛偵測，並辨識出大型車、小型車以及機車，接著利用 SORT[5]追蹤演算法，進行車輛物件追蹤，再用 U-Net[6]進行語義分割，分割出車輛輪廓以定位出車輛前緣點(Leading Edge Point, LEP)的座標以及分割出道路線，藉以標示出車速估測基準線(Speed Estimation Baseline, SEB)，最後透過 LEP 通過 2 條相鄰 SEB 所花費的時間，換算出車輛的行進速度。

1.2 AI 模型訓練

步驟一：訓練 YOLOv4[4]模型。我們針對四種常見的道旁監視攝影機裝設場景蒐集監視畫面，

其中 706 幀當成訓練集，每幀都標記出我們要辨識的目標車輛的邊界框與車種，如圖 1 所示。

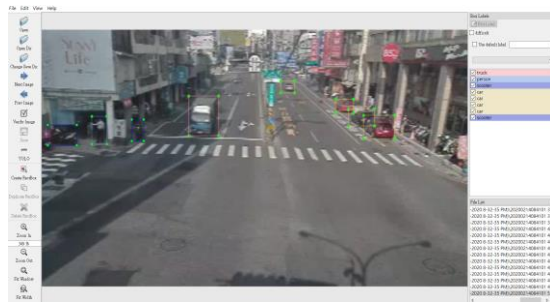


圖 1 訓練資料圖片。

步驟二：將 YOLOv4[4]模型所偵測到的車輛結合 SORT[5]演算法進行車輛追蹤訓練。

步驟三：訓練 U-Net[6]車道白虛線語義分割模型，本研究共使用 550 幀進行訓練，每幀都標記車道白虛線的區域範圍。

步驟四：訓練 U-Net[6]車輛語義分割模型，本研究共訓練 524 幀，每幀都標記車輛所屬像素區域。分割結果如圖 2



圖 2 小型車分割圖片

1.3 車速估測

車輛速度估測需藉助標定車輛位移距離與車行時間來計算，以下針對如何標定車輛位移距離與車行時間做進一步說明，系統流程圖如圖 3 所示。道旁監視攝影機基本上設置完成後鮮少更動，可視為固定式裝置，因此畫面中的車道白虛線位置是固定的，系統管理者可事先設定電子圍籬(如圖 4 中的黃色線)，一旦車輛碰觸電子圍籬，後續幀便會饋入 YOLOv4[4]模型進行車輛偵測與追蹤。

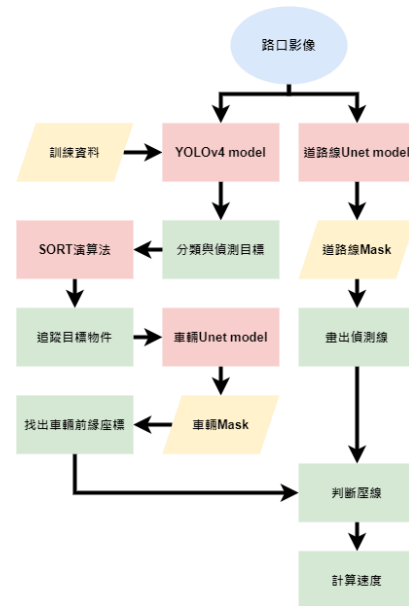


圖 3 系統流程圖



圖 4 系統執行示例畫面

步驟一：透過 U-Net[6]車道白虛線語義分割模型，從初始幀中分割出車道白虛線區域(如圖 5 所示)。從畫面底部往上找到第三塊白線區塊與第五塊白線區塊，分別定出垂直第三塊白線區塊與垂直第五塊白線區塊的二條直線，這二條直線被當作 SEB(如圖 4 中的紅色線)。

步驟二：透過 SORT[5]演算法將 YOLOv4[4]模型所偵測到的車輛進行追蹤。



圖 5 U-Net 模型所切割出的道路白虛線區

步驟三：將 YOLOv4[4]模型所偵測到的車輛框饋入 U-Net[6]車輛語義分割模型，精確地分割出屬於該車輛的所有像素點，並定位出最前緣點 LEP(如圖 4 中的綠色點)。

步驟四：紀錄該車輛的 LEP 分別碰觸二條 SEB 的對應幀編號。

步驟五：將這二個對應幀編號相減，得出經歷幀數(Number of Frames, NOF)。

步驟六：透過 NOF 可換算出標定車行時間，透過二條 SEB 的白線區塊間隔可換算出車輛位移距離，所以透過公式(1)即可求得車輛速度(乘上 3.6 是為了換算成公里/小時)。

$$\text{速度} = \{\text{距離} \div [(1/\text{FPS}) \times \text{NOF}]\} \times 3.6 \quad (1)$$

2. 研究成果

為實際測試所提出方法的效能，本研究分別對汽車及機車各進行數筆測試，實際車速是以 GPS 行車紀錄器測得，而估測的車速是以錄製的路口影片當作系統的輸入所計算出的結果。表 1 是汽車的實測數據，而表 2 則是機車的實測數據，其中誤差率的定義如方程式(2)所示。表 1 的第 4 筆數據誤差率超過 10%，經進一步分析與確認，發現是因為輸入影片掉幀導致，只要影片未掉幀，系統所估測的汽車速度誤差都在正負 3 左右，表 2 的數據顯示系統所估測的機車速度誤差都在正負 3 以內，顯示本研究所提出的方法可獲得相當不錯的準確度。另外，圖 6 和圖 7 分別是台南市小東路口與復興路口的車輛速度估測示例。

Machado 等人[3]於 2021 年提出植基於車牌偵測的車輛速度估測方法，他們使用車牌具有固定長寬尺寸特性並結合事先校準過的相機，從該相機拍攝的每一幀所偵測到的車牌，可藉由車牌真實尺寸與相機參數推算出該車牌的真實世界坐標，透過經歷幀數算出車輛的移動時間，而透過對應的真實世界坐標差異算出車輛的移動距離，即可估測出車輛的速度，其速度估測的平均誤差率為 11.15%，其誤差明顯高於我們所提出的方法，而且我們的方法無需進行相機校準，因為對路旁監視攝影機而言通常並無相關校準參數。相較於同屬基於輔助線的估測方法的[1]，其平均誤差率為 6.10%，顯示本研究透過語意分割網路能提升車身與輔助線接觸點的偵測精準度，進而有效提升車輛速度估測的準確度。

$$\text{誤差率} = (|\text{估測車速} - \text{實際車速}| \div \text{實際車速}) \times 100\% \quad (2)$$

表 1 汽車數據表

汽車				
數據	實際車速	估測速度	誤差	誤差率
1	40	42.4	2.4	6%
2	40	40	0	0%
3	46	42.4	-3.6	7.8%
4	25	27.7	2.7	10.8%
5	40	40	0	0%
6	38	36	-2	5.2%
7	39	36	-3	7.6%
8	28	30	2	7.1%
9	42	42.4	0.4	0.9%
10	41	45	4	9.7%
平均誤差率	5.50%			

備註：表 1 大部分的數據誤差大多在正負 3 以內。

表 2 機車數據表

機車				
數據	實際車速	估測速度	誤差	誤差率
1	34	36	2	5.8%
2	44	42.4	-1.6	3.6%
3	38	40	2	5.2%
4	37	37.9	0.9	2.4%
5	44	45	1	2.2%
6	28	26.7	-1.3	4.6%
7	49	51.4	2.4	4.8%
8	38	40	2	5.2%
平均誤差率	3.38%			

備註：表 2 的數據其誤差都在正負 3 以內。



圖 6 小東路口車速估測示例



圖 7 復興路口車速估測示例

3. 結論與未來展望

3.1 結論

車輛速度估測需藉助標定車輛位移距離與車行時間來計算，本專題透過 AI 影像分析技術分析監視攝影機畫面，先以 YOLOv4[4]模型偵測車輛，再結合 SORT[5]演算法進行車輛追蹤，分別透過 U-Net[6]車道白虛線語義分割模型以及車身語義分割模型，標示出車速估測基準線 SEB，以及定位出車輛前緣點 LEP 的座標，最後透過計算 LEP 通過 2 條相鄰 SEB 所花費的時間，估測出車輛的移動速度，能協助警方判斷肇事車輛是否超速。實測結果顯示本專題估測出的車輛速度平均誤差率為 5.50%，而機車的估測速度平均誤差率為 3.38%。相較於文獻[3]所提方法的平均誤差率 11.15%，更為準確。相較於同屬基於輔助線的估測方法的[1]，本研究的車輛速度估測準確度更高。

3.2 未來展望

本專題的研究成果經初步驗證能透過分析既有的路口監視影片，有效地估測出實際車輛移動速度，然而，現實路口的情況相當多元，本研究成果要能真正運用到所有道路，必須收集更多道路的監視影片，藉由累積大量數據來調整系統，這樣才能讓系統更臻於完善，所估測的汽車、機車移動速度也才能更加準確。

4. 參考文獻

- [1] T.T. Nguyen, X.D. Pham, J.H. Song, S. Jin, D. Kim, J.W. Jeon, "Compensating Background For Noise Due To Camera Vibration In Uncalibrated-Camera-Based Vehicle Speed Measurement System," IEEE Trans Vehicular Technol, Vol. 60, No. 1, pp. 30-43, 2011.
- [2] A.S.G. Alexander, A.T. Deasy, E.G. Fergyanto, "Detection of Vehicle Position and Speed Using Camera Calibration and Image Projection Methods," Procedia Computer Science, Vol. 157, pp. 255-265, 2019.
- [3] Machado, Felipe Leivas. "Vehicle speed estimation based on license plate detection." (2021).
- [4] A. Bochkovskiy, C. Y. Wang, H. Y. Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Computer Vision and Pattern Recognition 2020.
- [5] Bewley, Alex and Ge, Zongyuan and Ott, Lionel and Ramos, Fabio and Upcroft, "Simple online and realtime tracking," Computer Vision and Pattern Recognition 2016.
- [6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Computer Vision and Pattern Recognition 2015.