



多目標車輛追蹤

SMPT-Tracker: Similarity Model with Point Tracking Tracker

RE6124027 數據所 曾文海

RE6124019 數據所 吳明軒

H24095316 統計系 陳子揚

報告組別: 第6組





Table of contents

01

Problem Define

02

Relate Work

03

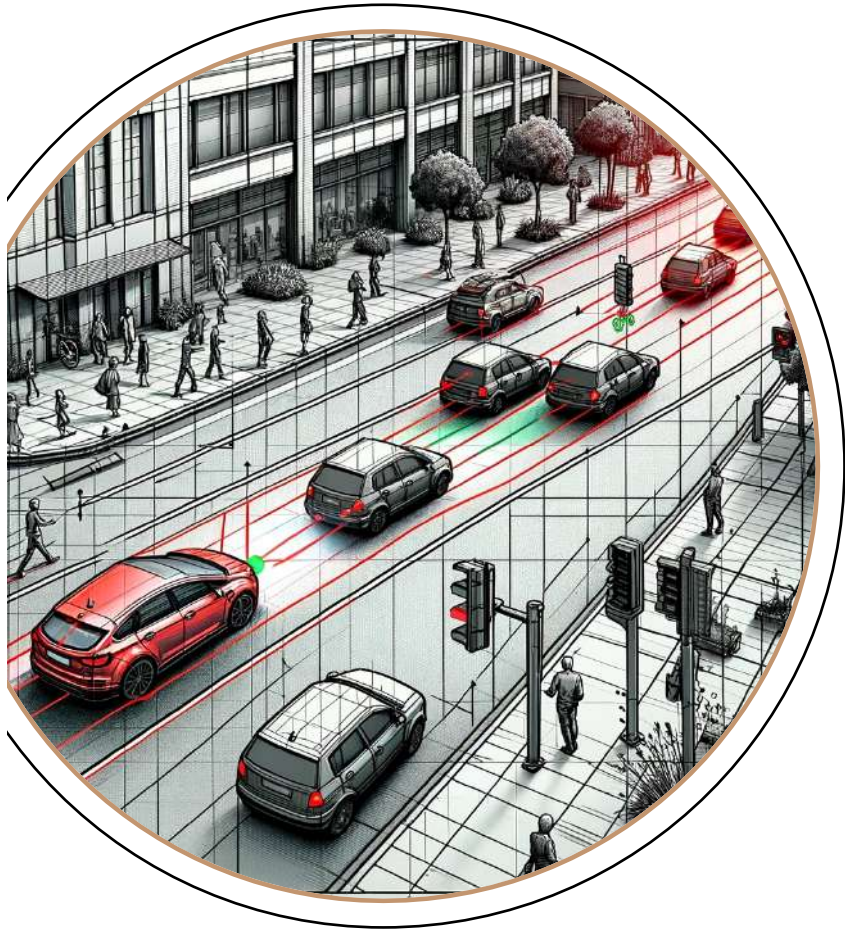
Method

04

Experiment

05

Conclusion



01

Problem Define

Problem Define





Problem Define

近年來，為了滿足居家安全和犯罪偵防的需求，攝影機系統在道路上的應用越來越廣泛。目前大多數的監視系統基於單相機錄影，每個攝影機獨立運作，因此當移動物件離開拍攝範圍後，便無法繼續追蹤。此外，當發生車禍或犯罪事件時，由於各台相機獨立錄影且無協同作業機制，警政單位需耗費大量人力資源，人工搜尋監視系統錄製的影片，以追蹤可疑車輛或行人的路徑與軌跡。

為了解決上述問題，本務研究多目標追蹤技術。資料集提供了道路車輛行駛影片，我們需發展多目標追蹤的AI模型，目標是能夠偵測並辨識出不同相機中的相同車輛。能深化台灣在智慧交通方面的AI技術，促進技術的多元發展。

$$IDP = \frac{IDTP}{IDTP + IDFP}, \quad IDR = \frac{IDTP}{IDTP + IDTF}$$
$$IDF_1 = 2 \frac{IDP * IDR}{IDP + IDR} = \frac{2IDTP}{2IDTP + IDFP + IDFN}$$

用於衡量追蹤的ID準確程度

$$MOTA = 1 - \frac{FN + FP + IDS}{GT}$$

衡量多目標跟蹤準確度

資料集描述:

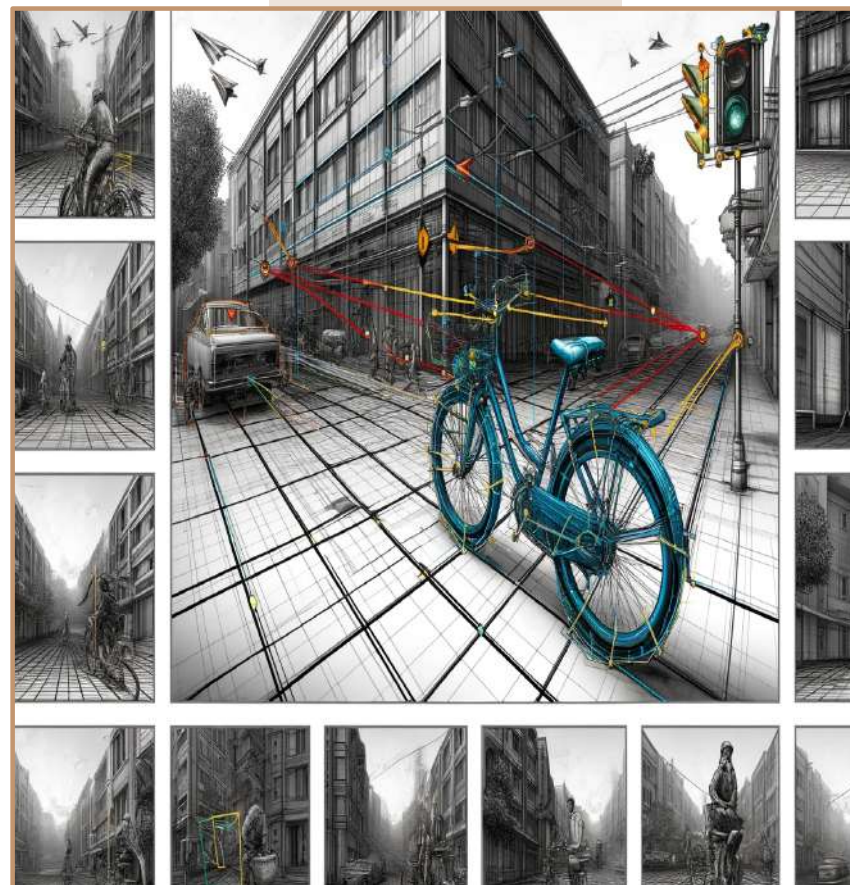
- 部分每秒多張
- 共7個鏡頭
- 7個鏡頭時間不同步
- 部分鏡頭提供影像品質不佳

Problem Define



資料集的追蹤效果不佳的原因?

- 大幅度的幀數變動
- 物體的變形
- 物體的相似性





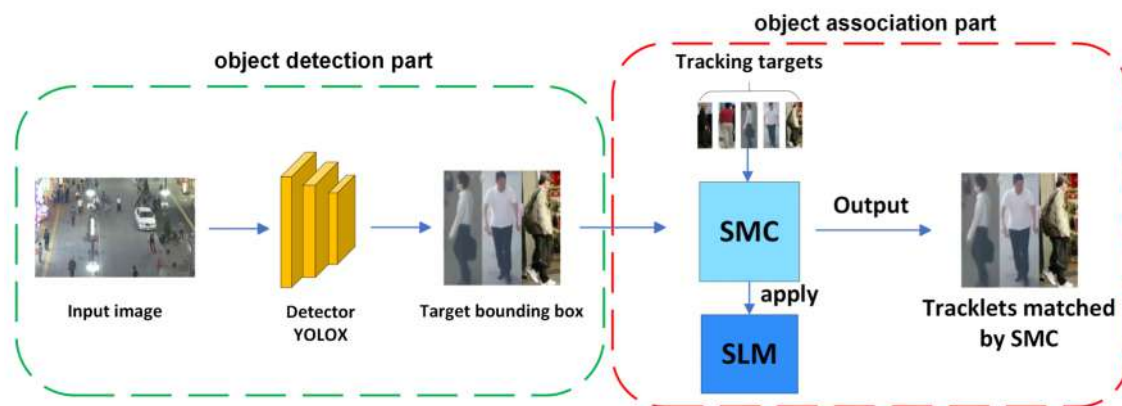
02

Relate Work

Relate Work

SMILETtack:

- 提出了一個外觀特徵提取器，即相似性學習模組（SLM），以明確區分使用注意力檢測的個體
- 提出了相似性匹配級聯（SMC），以實現加入穩健性外觀資訊的關聯

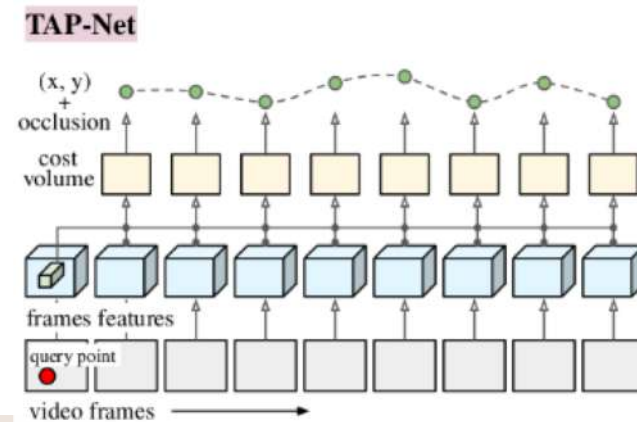


Relate Work



TAPNET:

- 追蹤任意點。給定一個視訊和一組查詢點，即視訊任何幀上的 2D 位置
- 匹配階段，為每隔一幀上的查詢點獨立定位合適的候選點匹配

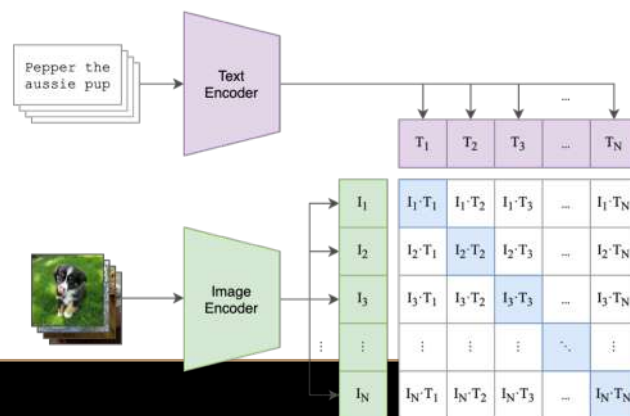


Relate Work

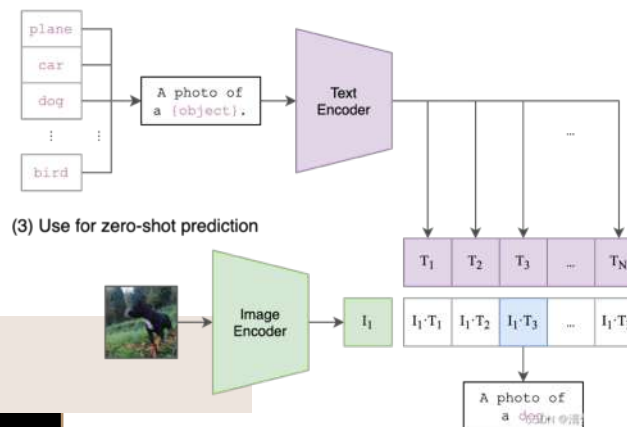
CLIP:

- 核心思想：用大量圖像和文字配對資料預先訓練，學習圖像與文字的對齊關係。
- 模型結構：Text Encoder：將文字轉換為低維向量表示。Image Encoder：將影像轉換為類似的向量表示。
- 任務能力：預訓練期間學習OCR、地理定位、動作辨識等任務，運算效率高，優於公開的最佳ImageNet模型。
- 預測機制：透過計算文字和圖像向量的餘弦相似度產生預測，適用於零樣本學習任務。應用領域：影像文字檢索、圖文產生等。

(1) Contrastive pre-training



(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



03

Method

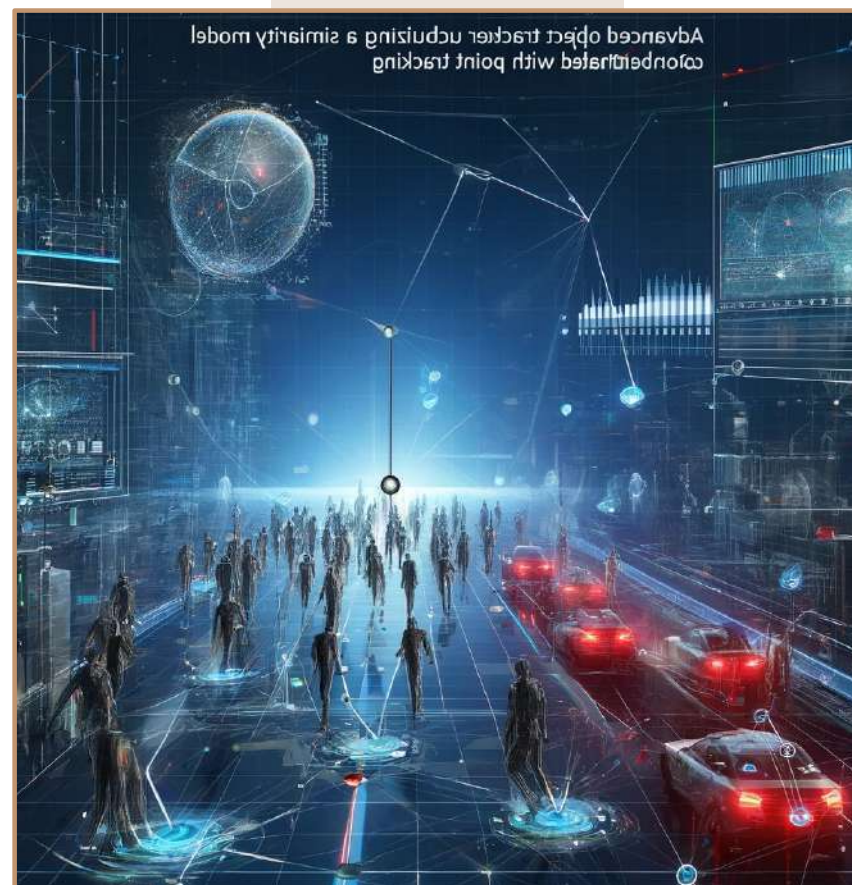
Method

資料集的追蹤效果不佳的原因?

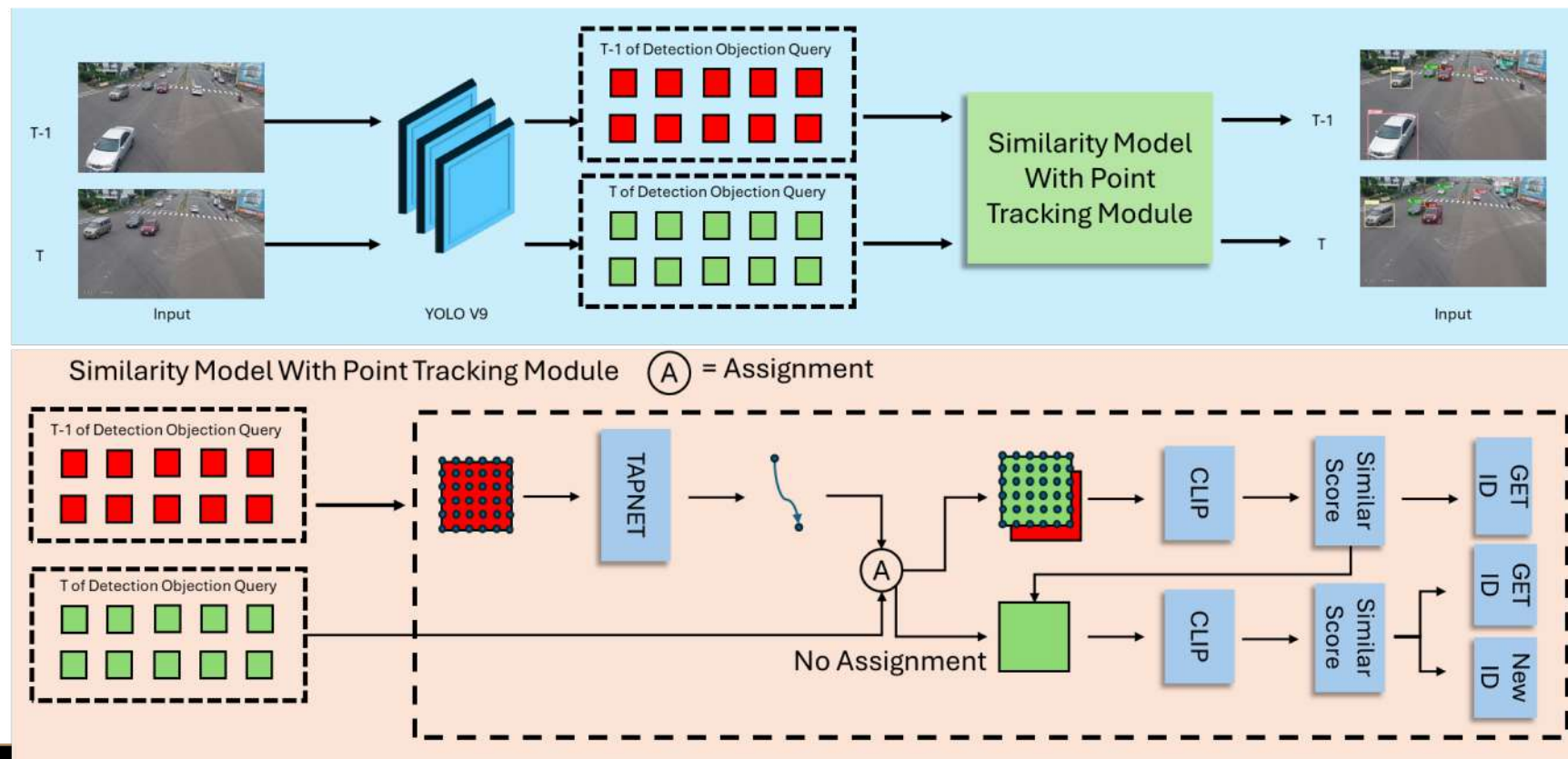
- CLIP Score計算物體前後幀的相似度
- TAP-NET讓追蹤器更加準確

SMPT-Tracker優點:

- CLIP 與TAP-NET不需要做Fine tune
- 解決Smilarity Model做Tracker的不穩定性



Method(SMPT-Tracker)





04

Experiment

Experiment



Tracker	Detector	IDF1	IDP	IDR	Rell	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
SMILETrack	YOLO V7	48.70%	47.40%	50.00%	91.50%	86.80%	4967	4113	816	38	8549	5182	20405	1012	44.20%	0.117	3038	17656	499
BOT_SORT	YOLOV7	59.70%	64.70%	55.40%	77.40%	90.60%	4967	2456	1997	514	4918	13836	11646	1458	50.30%	0.115	442	11305	110
BOT_SORT	YOLOV9_C	59.30%	64.70%	54.70%	79.10%	93.70%	4967	2924	1779	264	3261	12772	13174	1478	52.30%	0.109	452	12815	114
Ours(SMPT-Tracker)	YOLOV9_C	64.70%	67.30%	62.30%	85.10%	92.10%	4967	3445	1333	189	4460	9099	4358	1198	70.70%	0.114	2413	2979	993
Ours(SM-Tracker)	YOLOV9_E	63.50%	67.30%	60.00%	83.70%	93.90%	4967	3309	1425	233	3309	9999	5218	1270	69.70%	0.107	1833	4108	698
Ours(SMPT-Tracker)	YOLOV9_E	64.30%	68.20%	60.80%	83.70%	93.90%	4967	3309	1425	233	3309	9999	4197	1270	71.40%	0.107	2279	2914	960

Validation set

Experiment



Tracker	Detector	Point	IDF1	IDP	IDR	Rcll	Prcn	MOTA	MOTP
Ours(SMPT-Tracker)	YOLOV9_C	3x3	64.00%	66.60%	61.60%	85.10%	92.10%	70.10%	0.114
		4x4	64.70%	67.30%	62.20%	85.10%	92.10%	70.70%	0.114
		6x6	64.70%	67.30%	62.30%	85.10%	92.10%	70.70%	0.114
		9x9	64.70%	67.30%	62.30%	85.10%	92.10%	70.70%	0.114

Use different numbers of points in the validation set

Experiment



Tracker	Detector	Public分數
BOT_SORT	YOLOV7	0.639211
BOT_SORT	YOLOV9_C	0.685378
Ours(PT-Tracker)	YOLOV9_C	0.78818
Ours(SMPT-Tracker)	YOLOV9_C	0.888778
Ours(SMPT-Tracker)	YOLOV9_E	0.891143

Testing Set:

1. Baseline Model
2. Baseline Model change detector model use YOLOV9_C
3. Using PT-Tracker , Our performance improved by 10% compared to BOTSORT
4. Using SMPT-Tracker , Our performance improved by 10% compared to PT-Tracker
5. Finally, we changed the detector to YOLOV9_E
6. , but we did not finish the run due to lack of time, but we can find that the accuracy has improved by 0.3% .

Experiment



SMPT-Tracker



BoTSort



SM-Tracker



SMILEtrack



05

Conclusion



Conclusion

我們將本來不穩定的Similarity Model物件追蹤方法加上點追蹤模型TAP-NET，因此提出 SMPT-Tracker方法，我們的方法讓Similarity Base物件追蹤方法有更好的追蹤效果，並且我們不需要對CLIP與TAP-NET做fine tune，在上面實驗的章節也有充分展示效果。



Thanks!

Do you have any questions?

Our Github:

https://github.com/w5535586/2024-AICUP_Cross-Camera-Multi-Target-Vehicle-Tracking-Competition

Reference:

<https://github.com/google-deepmind/tapnet>

<https://github.com/openai/CLIP>

<https://github.com/WWangYuHsiang/SMILEtrack>

