Video Surveillance for Road Traffic Monitoring

Iñigo Auzmendi Iriarte, Kyryl Dubovetskyi, Michell Vargas Signoret, Razvan Apatean
Ayan Banerjee
Universitat Autònoma de Barcelona
Plaça Cívica, 08193 Bellaterra, Barcelona
inigoauz.27@gmail.com, krupartea@gmail.com,
michellvsigno21@gmail.com, apatean.razvan@gmail.com, ab2141@cse.jgec.ac.in

Abstract

As of late, Street Traffic Observing has been acquiring consideration as it has turned into an exceptionally requested video surveillance task. For multi-target single-camera (MTSC) tracking and multi-target multi-camera (MTMC) tracking issues, we compare and contrast traditional and cutting-edge methods in this report. Particularly, we concentrate on achieving the highest possible scores in the AI City Challenge 2022. We find that by filtering tracks appropriately and fine-tuning every parameter in the tracking pipeline, significant advancements can be made compared to a standard tracker baseline. We use standard video analysis metrics like IDF1/HOTA for single-camera tracking and IDF1/IDP/IDR for multi-camera tracking to evaluate how well the proposed methods work. The code for this project is publicly available at: https://github.com/mcv-m6-video/mcv-m6-2023-team5

1. Introduction

Road traffic monitoring and Advanced Driver Assistance Systems (ADAS) aim to increase road transportation safety, efficiency, and comfort through state-of-the-art computer vision techniques. In this report, we are going to analyze the benchmark tracking techniques on the AI City challenge dataset [24] which provides single and multi-camera evaluation tracks. This task aims to identify vehicles and track them along large traffic areas and different road intersections through several video cameras. We analyze all the state-of-the-art approaches based on the IDF1/HOTA score [19] and choose the best one to improve road design and traffic flow.

1.1. Multi-Target Single-Camera

The objective of *Multi-Target Single-Camera Tracking (MTSC)* is to identify and assign a unique ID to the same object throughout a video sequence. This involves detecting and predicting the object's position in consecutive frames and possibly re-establishing its trajectory by using identification templates if it becomes fragmented. The overall goal is to accurately track the object's movement over time and provide a reliable identification for it.

1.2. Multi-Target Multi-Camera Tracking

The aim of Multi-Target Multi-Camera Tracking (MTMC) is to monitor and track vehicles as they move through multiple sensors' fields of view. This is a more complex task compared to MTSC because it requires synchronizing camera views and identifying vehicle tracks that appear in different cameras at various time intervals. The process involves re-identifying vehicles as they move across different cameras and dealing with situations where multiple cameras capture them simultaneously. Overall, MTMC is a challenging task that requires sophisticated algorithms and techniques to accurately track vehicles across multiple cameras.

1.3. Organization of the report

The rest of the report is organized as follows. All the recently proposed benchmark object detection and tracking approaches are discussed in *Section 2*. The proposed methodology for MTSC and MTMC has been proposed in *Section 3*. The quantitative and qualitative analysis of the proposed methodology has been depicted in *Section 4* and the report has been concluded in *Section 5*.

2. Related Work

The multi-target single-camera and multi-camera tracking is basically a combination of three downstream tasks: object detection, object tracking, and object identification. We are trying to discuss a short analysis of each of those techniques.

2.1. Object detection

Object detection is a well-known and well-defined problem in the field of computer vision that has been tackled using conventional computer vision techniques before the advent of deep learning methods. These traditional approaches include Viola-Jones Detectors [6], HOG Detectors [20], or a combination of descriptors (such as SIFT, SURF, ORB) and classifiers (such as KNN, SVM). However, the breakthrough came with the invention of Convolutional Neural Networks (CNNs), which led to the development of two-stage object detectors such as RCNNs [5, 11, 16], and single-stage object detectors like SSD [18] or YOLO [8]. These models significantly improved object detection in natural scenes.

Since then, several improvements have been made, including the Faster-RCNN [5] and YOLOv8 [1] networks. Recently, the development of Transformers has led to the development of DETR [7], which further improves object detection performance by utilizing patches, positional embeddings, and spatial features.

2.2. Object Tracking

Various filters can be applied to describe the motion model of objects in video footage, such as Kalman filters [10] or the SORT method [14]. However, these filters may have limitations in certain scenarios due to assumptions of linearity. To address these limitations, particle filters [2] or optical flow analysis [22] can be used for better object tracking.

Alternatively, deep learning methods can also be considered for object tracking, such as DeepSORT [15]. However, this approach may be computationally heavy and must be carefully selected based on the specific situation. DeepSORT employs deep learning to extract features of the appearance of objects, improving its accuracy in tracking.

2.3. Object identification

In order to track an object across multiple cameras, it is necessary to have a way of representing the object that can be used to identify it in different views. Traditional methods like color histograms and feature descriptors can be easy to compute, but they may not be accurate enough in complex scenarios. In these cases, a combination of methods like SIFT, ORB, histograms, and gradients can be used. However, deep learning approaches like embedding encoders have proven to be the most effective. These methods use convolutional neural networks to learn a compact and discriminative representation of the object, which can then be used to match it across different views. The Re-ID network architecture [21] is an example of this, which uses ImageNet pre-trained networks [23] as a starting point. Overall, deep learning approaches have significantly improved the accuracy and robustness of multi-camera tracking.

3. Methodology

Here, we are going to discuss the methodology applied to accomplish the objectives. The section is divided into two parts: MTSC tracking and its extension to perform MTMC tracking.

3.1. Multi-target single camera tracking (MTSC)

MTSC has been conducted in four consecutive steps as depicted in Fig. 1. First, we detect the vehicles with existing object detectors (e.g. YOLOv8 [1], DETR [7] etc.). However, sometimes the pre-trained object detectors are not sufficient, so we need to fine-tune them. Next, we track

Figure 1. General pipeline of MTSC strategy.

the detected vehicles with popular tracking algorithms like Kalman filters with different variants (i.e. maximum overlapping, SORT, etc.) and do some post-processing for further evaluation.

3.1.1 Object detection

Three distinct approaches were tested for object detection, all of which employed object detection networks. The first approach involved utilizing a YOLOv8 model pre-trained on the COCO [17] dataset to infer object detections. Additionally, a Faster R-CNN model, also pre-trained on COCO, was used for inference. Subsequently, a DETR [4] model, which is a transformer-based object detection algorithm that eliminates the need for anchor boxes, was employed.

3.1.2 Object Tracking

In our object tracking system, we have employed both the maximum overlap and Kalman filter methods to establish tracks. The SORT variant [3] of the Kalman filter was used, and a permissive IoU threshold of 0.3 was applied to both methods.

Additionally, we tested the use of Optical Flow for object tracking, which demonstrated better performance than the maximum overlap method, but not as good as SORT. As a result, we decided to evaluate the DeepSORT algorithm [12], which combines deep appearance information with a Kalman filter to improve object tracking, to determine if it could outperform the other methods.

3.1.3 Post-processing

Various measures have been implemented to improve the track quality by filtering out segments that do not meet the criteria established by the ground truth. Firstly, detections

with a width or height smaller than 0.7 of the minimum detection boxes specified in the ground truth, or those outside the region of interest (ROI), have been removed. Additionally, tracks containing stationary vehicles and with a duration of less than five frames have been discarded.

3.2. Multi-target multi-camera tracking (MTMC)

The core task is to establish identity correspondences across different cameras. The pipeline of MTMC has been depicted in Fig. 2. Our implementation of MTMC extends the MTSC pipeline with a matching algorithm for comparing tracked cars across different cameras and a final relabeling stage before evaluation.

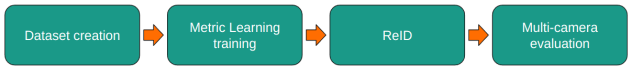

Figure 2. General pipeline of MTMC strategy.

3.2.1 Dataset Creation

A bespoke dataset was created for training that aims to differentiate between various types of vehicles and match identical ones. This dataset was constructed by cropping the video frames utilizing ground truth detections, with each unique track ID serving as a class label. The S01 and S04 sequences from the AI City 2022 challenge were utilized as the training videos.

3.2.2 Metric learning

To develop an effective recognition system that can recognize different objects across multiple cameras, the study experimented with several custom CNN networks as Siamese networks. However, the team ultimately decided to use a triplet network [9] with ResNet18 as its backbone, which was found to be more effective. To determine the optimal model parameters with the lowest triplet loss on the validation set, a grid search technique was applied.

The preferred training setup involves using a batch size of 128, a learning rate of 1e-4, a decay of 0.1 every 3 steps, 50 epochs, Adam optimizer, and a triple margin loss function with a margin of 1. In addition, the triplet generation without mining is utilized, as mining techniques resulted in inferior outcomes.

The trained model generates embeddings for each object detected in a frame, and the L2 distance is used to compute the similarity score between them. If two objects are the same but are captured by different cameras, they will have similar embeddings, leading to a low distance score.

3.2.3 ReID

Upon obtaining results from MTSC, a cross-camera matching process is required. The process begins with creating a new global ID list based on the data collected from the first camera in the sequence. Two matching algorithms are used to determine the similarity between image representations from the first two cameras. These algorithms are:

- Centroids: This algorithm computes the average embedding vector of a track. Next, new camera tracks and global tracks average embeddings are compared, and the same ID is assigned to the new camera track if its lower distance is below a specific threshold.
- Voting: In this algorithm, each frame object embedding in the new camera track casts a vote for a track ID of the global tracks (based on the minimum distance) only if the distance is smaller than a predetermined threshold. The assigned ID to the camera track will be the one that has received the most votes.

This process is iterated for every camera in the sequence until all the camera tracks are incorporated as global tracks.

3.3. Evaluation metrics

To determine the accuracy of the tracks in both MTSC and MTMC tracking, we employed the identification precision (IDP), identification recall (IDR), and IDF1 metrics from TrackEval [13]. IDP and IDR provide information regarding tracking trade-offs, while IDF1 presents a balanced value between the other two through their harmonic mean. Furthermore, detection precision and recall were utilized to measure the quality of the detections used for tracking.

In addition, we utilized the HOTA [19] score to evaluate the performance of MTSC tracking. This score takes into account both the accuracy and completeness of the tracking results, making it a more comprehensive and reliable measure of tracking performance, particularly in complex scenarios with multiple objects and occlusions.

4. Results

The following section summarizes the primary outcomes of MTSC and MTMC tracking for the conducted tests. This report only includes the results of MTSC for sequence 3, as they are representative of the insights that can be gained from other sequences. The results for all sequences are presented for MTMC tracking.

4.1. MTSC Tracking

The MTSC tracking performance has depended on object detection and tracking algorithms. A comparative study of pre-trained object detection frameworks has been shown in Fig. 3

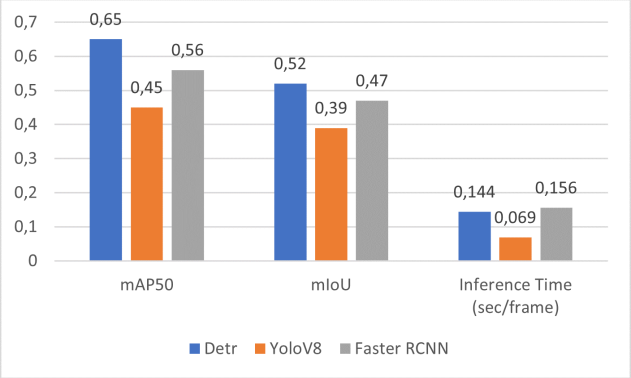

Figure 3. Comparative study of the object detection frameworks.

Although YOLOv8 had the fastest inference time (as shown in Figure 3) due to being a one-stage CNN, it produced the worst results. Similarly, Faster R-CNN produced inferior results compared to DETR, despite having a longer inference time. As a result, DETR was chosen as the object detector for the remaining work. Subsequently, we fine-tuned the DETR on the training video frames and observed a 48.3% improvement in mAP. Therefore, fine-tuned DETR was chosen as the optimal object detector for the remainder of the experiments.

Now that our object detector has been selected, we move on to selecting the tracking algorithms. A similar comparative study of the tracking algorithms is presented in Table 1.

Based on the observations from Table 1, it has been noticed that camera c015 experienced poor results because most of the tracks that were identified belonged to vehicles that were only visible in that camera and were not accounted for in the ground truth. Nevertheless, after conducting several experiments, we selected DeepSORT as the tracking algorithm that yielded the best results.

4.2. MTMC Tracking

In order to determine the optimal matching method and threshold value for each network, a grid search was conducted. The results of this ablation study are presented in Table 2.

Based on the results presented in Table 2, it was observed that the ResNet18 network performed the best with triplet-based fine-tuning. Consequently, the centroids-based method with a threshold value of 17 was chosen for the subsequent experiments since this parameter combination yielded the best results for this network.

In Table 3, the results of our best MTMC method applied to different sequences are presented. It is worth noting that the best parameter tuning was performed on sequence 3, which is reflected in the superior results obtained in that sequence compared to the others.

4.3. Faliure Cases of MTMC Tracking

Due to the complexity of the MTMC tracking task, it is challenging to train the model to have a similar representation without overfitting, even with data augmentation, as all training images are quite similar. This results in some failure cases, as shown in Figure 4.

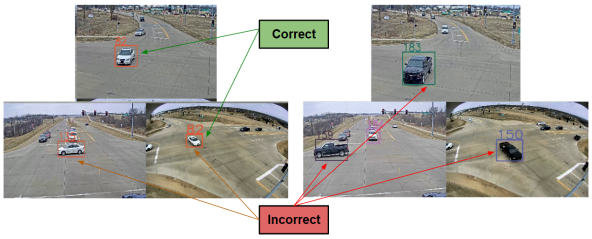

Figure 4. Failure case of MTMC tracking.

The figure illustrates an example where vehicle ID 82 was tracked accurately in one camera but misidentified in another. Conversely, vehicle ID 183 could not be identified correctly across the cameras due to variations in its shape and differences in camera perspectives, leading to dissimilar object embeddings across the cameras.

5. Conclusion

This report has examined various techniques for detecting vehicles in video footage. The use of the pretrained DETR object detection neural network has demonstrated high accuracy, but the results were significantly improved when it was fine-tuned with objects specific to the training videos.

The effectiveness of DeepSORT in establishing robust tracks has been observed, surpassing maximum overlapping or optical flow methods. Additionally, the importance of post-processing has been emphasized, particularly for non-learning-based approaches, as it is crucial to adapt tracks to the unique characteristics of the ground truth.

In this study, a novel approach has been introduced for matching tracks across multiple cameras. The experimental results have indicated that the metric learning algorithm performed well in this task. However, finding the optimal threshold for the matching algorithm has been challenging, which ultimately affected the performance of the proposed method. Thus, further research is needed to develop more robust techniques to determine the matching algorithm threshold and enhance the overall performance of the system.

To make the road monitoring algorithm more practical, it should be tested in real-time and ”online” conditions, where only previous frames are available. Finally, to improve the MTMC tracking system, future work should consider object trajectories and camera perspectives to enhance the results.

Table 1. Comparative study of the MTSC tracking algorithms based on IDF1/HOTA score

Camera	c10	c11	c012	c013	c014	c015	Average
Overlap	46/62	53/56	33/44	42/45	45/54	2/11	36/45
SORT	90/75	65/49	**79/57**	75/52	70/60	11/**19**	65/52
OF	81/65	**82/61**	50/36	76/59	75/60	10/16	62/49
DeepSORT	**91/75**	78/57	75/48	**83/59**	**79/61**	**11**/15	**70/53**

Table 2. Ablation Study of MTMC tracking

Network	Best ReID method		IDF1	IDR	IDP	Det. P	Det R.
	Method	Th. val.					
Siamese Simple Finetuned	Votes	0.6	45.5	44.2	46.9	74.2	71.1
Resnet18 pretrained	Votes	15	44	42.8	45.4	74.2	71.1
Resnet18 siamese finetuned	Votes	13	42	40.8	43.3	74.2	71.1
Resnet18 triplet finetuned	**Centroids**	**17**	**54.3**	**53.1**	**55.5**	**74.2**	**71.1**

Table 3. MTMC tracking best method results in different sequences

Metric	IDF1	IDP	IDR	Det. P	Det. R
SEQ 1	42.8	**57.6**	34	68.5	40.4
SEQ 3	**54.3**	55.5	**56.7**	**74.2**	**71.1**
SEQ 4	40.2	51.9	32.8	75.6	47.8
AVG.	45.8	55	41.2	72.8	53.1

References

[1] Armstrong Aboah, Bin Wang, Ulas Bagci, and Yaw Adu-Gyamfi. Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. *arXiv preprint arXiv:2304.08256*, 2023. 2

[2] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002. 2

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 2

[5] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Feris, Jinjun Xiong, and Thomas Huang. Revisiting rcnn: On awakening the classification power of faster rcnn. In *Proceedings of the European conference on computer vision (ECCV)*, pages 453–468, 2018. 2

[6] Mehul K Dabhi and Bhavna K Pancholi. Face detection system based on viola-jones algorithm. *International Journal of Science and Research (IJSR)*, 5(4):62–64, 2016. 2

[7] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2988–2997, 2021. 2

[8] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. Object detection using yolo: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, 82(6):9243–9275, 2023. 2

[9] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 459–474, 2018. 3

[10] Simon Haykin. Kalman filters. *Kalman filtering and neural networks*, pages 1–21, 2001. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[12] Xinyu Hou, Yi Wang, and Lap-Pui Chau. Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2019. 2

[13] Arne Hoffhues Jonathon Luiten. Trackeval. https://github.com/JonathonLuiten/TrackEval, 2020. 3

[14] Shivani Kapania, Dharmender Saini, Sachin Goyal, Narina Thakur, Rachna Jain, and Preeti Nagrath. Multi object tracking with uavs using deep sort and yolov3 retinanet detection framework. In *Proceedings of the 1st ACM Workshop on Autonomous and Intelligent Mobile Systems*, pages 1–6, 2020. 2

[15] Shailender Kumar, Pranav Sharma, Nitin Pal, et al. Object tracking and counting in a zone using yolov4, deepsort and tensorflow. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 1017–1022. IEEE, 2021. 2

[16] Shaoqi Li, Wenfeng Song, Shuai Li, Aimin Hao, and Hong Qin. Meta-retinanet for few-shot object detection. In *BMVC*, 2020. 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 2

[19] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129:548–578, 2021. 1, 3

[20] Yanwei Pang, Yuan Yuan, Xuelong Li, and Jing Pan. Efficient hog human detection. *Signal processing*, 91(4):773–781, 2011. 2

[21] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 1900–1909, 2017. 2

[22] Deqing Sun, Stefan Roth, and Michael J Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106:115–137, 2014. 2

[23] Abhijit Suprem and Calton Pu. Looking glamorous: Vehicle re-id in heterogeneous cameras networks with global and local attention. *arXiv preprint arXiv:2002.02256*, 2020. 2

[24] Zheng Tang, Milind R. Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, D. Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8798, 2019. 1