mathematics

MDPI

Article

Multi-Camera Multi-Vehicle Tracking Guided by Highway Overlapping FoVs

Hongkai Zhang [1,2,†], Ruidi Fang [1], Suqiang Li [1], Qiqi Miao [1], Xinggang Fan [3,*], Jie Hu [4,*] and Sixian Chan [5,6,†]

[1] School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei 230601, China; zhk@ahjzu.edu.cn (H.Z.); frd@stu.ahjzu.edu.cn (R.F.); sqli228@stu.ahjzu.edu.cn (S.L.); mqq@stu.ahjzu.edu.cn (Q.M.)
[2] Key Laboratory of Architectural Cold Climate Energy Management, Ministry of Education, Jilin Jianzhu University, Changchun 130119, China
[3] Zhijiang College, Zhejiang University of Technology, Hangzhou 312030, China
[4] Key Laboratory of Intelligent Informatics for Safety & Emergency of Zhejiang Province, Wenzhou University, Wenzhou 325035, China
[5] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China; sxchan@zjut.edu.cn
[6] Hangzhou Xsuan Technology Co., Ltd., Hangzhou 310000, China
* Correspondence: xgfan@zjut.edu.cn (X.F.); 20160204@wzu.edu.cn (J.H.)
† These authors contributed equally to this work.

Abstract: Multi-Camera Multi-Vehicle Tracking (MCMVT) is a critical task in Intelligent Transportation Systems (ITS). Differently to in urban environments, challenges in highway tunnel MCMVT arise from the changing target scales as vehicles traverse the narrow tunnels, intense light exposure within the tunnels, high similarity in vehicle appearances, and overlapping camera fields of view, making highway MCMVT more challenging. This paper presents an MCMVT system tailored for highway tunnel roads incorporating road topology structures and the overlapping camera fields of view. The system integrates a Cascade Multi-Level Multi-Target Tracking strategy (CMLM), a trajectory refinement method (HTCF) based on road topology structures, and a spatio-temporal constraint module (HSTC) considering highway entry–exit flow in overlapping fields of view. The CMLM strategy exploits phased vehicle movements within the camera's fields of view, addressing such challenges as those presented by fast-moving vehicles and appearance variations in long tunnels. The HTCF method filters static traffic signs in the tunnel, compensating for detector imperfections and mitigating the strong lighting effects caused by the tunnel lighting. The HSTC module incorporates spatio-temporal constraints designed for accurate inter-camera trajectory matching within overlapping fields of view. Experiments on the proposed Highway Surveillance Traffic (HST) dataset and CityFlow dataset validate the system's effectiveness and robustness, achieving an IDF1 score of 81.20% for the HST dataset.

Keywords: multi-camera multi-vehicle tracking; ReID; camera overlapping fields of view; highway tunnel scenarios; topology modeling

MSC: 68T45

Citation: Zhang, H.; Fang, R.; Li, S.; Miao, Q.; Fan, X.; Hu, J.; Chan, S. Multi-Camera Multi-Vehicle Tracking Guided by Highway Overlapping FoVs. *Mathematics* **2024**, *12*, 1467. https://doi.org/10.3390/math12101467

Academic Editor: Radu Tudor Ionescu

Received: 1 April 2024
Revised: 27 April 2024
Accepted: 7 May 2024
Published: 9 May 2024

1. Introduction

With the rapid development of Intelligent Transportation Systems (ITS) and the increasing demand for digital traffic management platforms, Multi-Camera Multi-Vehicle Tracking (MCMVT) [1–8] has attracted widespread attention in recent years. Applications similar to MCMVT can assist in modern traffic flow prediction and analysis. The MCMVT task in highway scenarios aims to track the trajectory of each vehicle target across multiple

cameras. Figure 1 illustrates the complete tracking chain of a vehicle's identity under different cameras in a highway tunnel scenario.

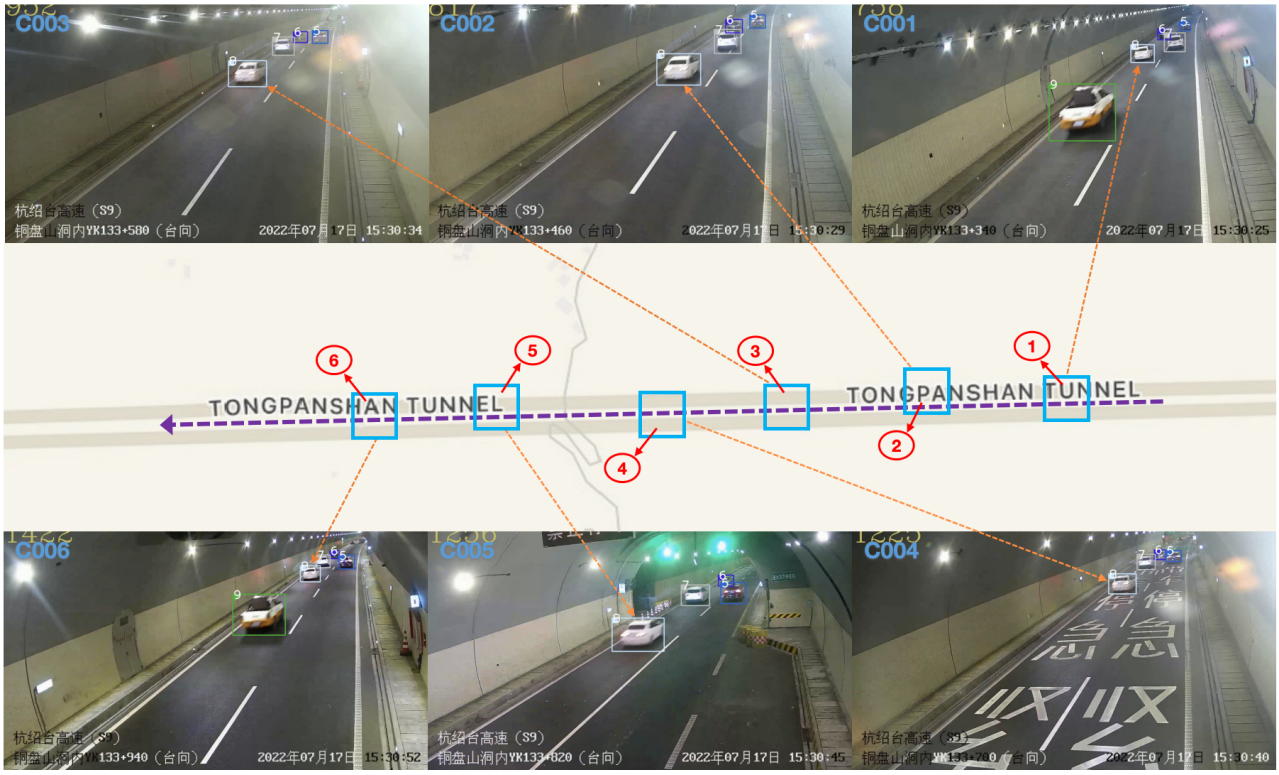

Figure 1. Illustration of Multi-Camera Multi-Vehicle Tracking (MCMVT) task. The second row is the camera position distribution map in the scene of the Tongpan Mountain Highway Tunnel. Vehicles with IDs 5, 6, 7, 8, and 9 are correctly tracked by multiple cameras, and a visualization tracking chain is presented for the vehicle with ID 8. "C00X" represents the camera ID.

The Multi-Camera Multi-Vehicle Tracking task typically consists of four sub-tasks: vehicle detection, vehicle ReID, single-camera multi-object tracking, and cross-camera trajectory association. The general workflow can be summarized as follows: First, the vehicle detection module outputs the coordinates and categories of vehicles in each frame, and the ReID module extracts the appearance features of the vehicles. Then, based on the results of vehicle detection and ReID, the Single-Camera Multi-Target Tracking (SCMT) module generates candidate trajectories for the vehicles under each camera. At last, the cross-camera matching module utilizes information such as road topology structures, adjacent camera field types, and vehicle transition times to match these candidate trajectories across different cameras. This process associates vehicle targets with global IDs, obtaining the final cross-camera trajectories for vehicles.

The primary challenges [9–11] of the MCMVT task in highway tunnel scenarios, as visualized in Figure 2, include:

1. Due to the large camera field of view, the scale of vehicle targets changes as they travel through narrow tunnels. In the near field of view, vehicle confidence is high and there is no occlusion; in the middle field of view, confidence is relatively high and vehicles begin to occlude each other; in the far field of view, confidence is low and occlusion is severe. During long-term tracking, there is a risk of fragmentation and loss of vehicle trajectories, especially for large trucks;
2. The appearance of flares on both sides of the tunnel walls may cause glare in the camera field of view. Due to intense lighting within the highway tunnel, the camera faces difficulty accurately capturing the features of various vehicles, especially when their appearances are similar. This situation may result in false positives, missing, and

other issues, affecting the effectiveness of target detection and tracking. Consequently, it impacts the accuracy and robustness of the overall MCMVT;

3. According to the road topology structures of each camera field of view, within the narrow and lengthy highway tunnel, the camera field of view is larger near the camera and smaller in the distance. Additionally, due to the highway movement of vehicles, they appear to move rapidly in the near field of each camera view and have a longer transition time in the distance. The lower resolution in the distant field is not conducive to effective vehicle detection and tracking. Furthermore, in order to provide more field of view coverage and obtain more continuous target trajectories in high-speed tunnels, the distance between adjacent cameras is usually shortened to provide higher-resolution data. This results in the same vehicle identity existing within the field of view of a single camera. It takes a long time, but it takes a relatively short time to appear in the next camera field of view.

(**a**) (**b**) (**c**) (**d**)

Figure 2. Illustration of challenges in MCMVT for highways. (**a**) represents the target scale variation, in which the yellow, orange and red bboxes respectively represent no occlusion at high scores, slight occlusion at high scores and severe occlusion at low scores; (**b**,**d**) represents intense lighting in the tunnel; (**c**) illustrates vehicles within green boxes being captured by adjacent cameras at the same time.

In this paper, we propose a Multi-Camera Multi-Vehicle Tracking (MCMVT) system designed for highway tunnel scenarios, utilizing road topology structures and spatio-temporal prior information to alleviate these challenges. The workflow of our proposed tracking system is illustrated in Figure 3. Given a sequence of videos from cameras, we first employ Swin Transformer Detection to detect all vehicles. Subsequently, the detected vehicle images are batched and input into the ReID module to extract 2048-dimensional appearance features. Based on the detected vehicle bounding boxes (BBoxs) and vehicle appearance features, we adopt a Cascade Multi-Level Multi-Target Tracking strategy in the Single-Camera Multi-Target Tracking module to generate candidate trajectories within each sub-camera and cross-match them. Finally, due to the presence of static traffic signs on both sides of the highway tunnel and a small number of false-positive detections from the detector, before the cross-camera association stage, we perform scene modeling based on the road topology structures. We filter the original vehicle tracking trajectories for each camera based on highway entry–exit flow, obtaining refined cross-camera candidate vehicle matching trajectories. These refined trajectories are then input into the spatio-temporal constraint module based on the overlapping field of view for highway entry–exit flow,

enabling cross-camera matching in overlapping regions. A similarity distance matrix is constructed, and k-reciprocal nearest-neighbors reranking is also applied to optimize the similarity matrix based on the size of similarities for vehicle matching pairs. Finally, local trajectory clustering is performed for trajectories within connected regions of adjacent camera entry–exit flow, followed by hierarchical clustering at the global trajectory ID level for the camera group. Notably, the cross-camera trajectory matching module takes into account road topology structures, vehicle travel time, vehicle entry–exit flow, and camera field of view constraints to reduce the ReID search space and accelerate the matching speed.

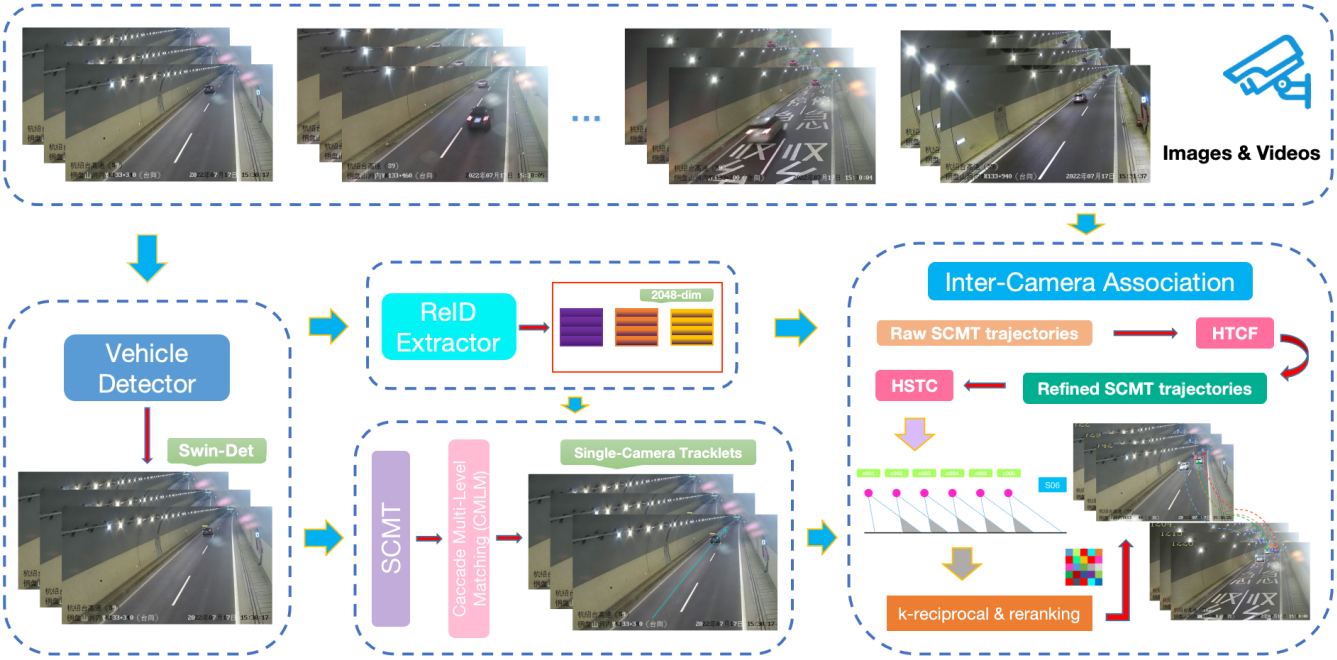

Figure 3. The pipeline of our MCMVT system. All vehicle objects are first detected using the detector to obtain the BBox of the vehicle from each frame of each camera video and then the ReID module extracts the corresponding appearance features of the vehicle BBox. The SCMT module uses the BBox and ReID feature for each tracking target to generate all trajectory candidates for every single camera. Finally, the ICA module comprehensively considers road topology structures and spatio-temporal information to match all candidate trajectories across all cameras.

In summary, we make the following contributions in this paper:

- Introduce a Cascade Multi-Level Multi-Target Tracking strategy, grouping vehicles based on the scores of detection boxes. Prioritizing cascade matching helps minimize uncertainties introduced by occlusion. The multi-level association process, considering the changing scales of targets as vehicles traverse narrow tunnels, maximizes the advantages of high-quality appearance features from high-scoring boxes in the near field. This strategy effectively associates and tracks both small target vehicles in the distant field and large target vehicles in the tunnel;
- Implement a scene modeling and candidate trajectory filtering module based on road topology structures. This module defines predefined sub-regions for each field of view based on highway entry–exit flow, allowing vehicles to transition through ordered regions. This helps alleviate motion blur and detector false positives caused by static traffic signs on both sides of the tunnel. The module aims to filter and obtain reliable cross-camera candidate matching trajectories and improve the performance of Multi-Camera Multi-Vehicle Tracking;
- Design a spatio-temporal constraint module tailored for highway overlapping fields of view. Building upon high-quality cross-camera candidate vehicle tracking trajectories, this module reduces the spatial matching domain, identifying effective matching

segments within the long, narrow tunnel for each camera field of view. It combines highway entry–exit flow spatio-temporal masks to rationalize the ordered transition time sequence of vehicles. This significantly reduces the ReID search space during cross-matching and accelerates the matching process. The result is the generation of complete cross-camera vehicle tracking chains.

The rest of the paper is organized as follows: The Related Work section provides an overview of related work. The Method section details the tracking system proposed in this paper. The Experiment section presents comparative and ablation experiments on both our self-created Highway Surveillance Traffic (HST) dataset and the CityFlow dataset to showcase the effectiveness of our approach. Finally, the Conclusions section concludes the paper.

2. Related Work

2.1. Vehicle Detection

Object detection, as a fundamental task in computer vision, utilizes bounding boxes to locate the presence of objects in an image and classify the identified objects. Vehicle detection is a specialized branch of object detection that focuses specifically on identifying vehicles in images or videos. Based on different backbone networks, existing object detectors can be categorized into two branches: CNN-based detectors [12–21] and transformer-based detectors [22–25]. Due to the success of Convolutional Neural Networks, detectors based on CNNs have significantly progressed and are generally divided into two main types. One type is the one-stage detector, which balances speed and accuracy for real-time operation. The YOLO series [12–16,26–28] is the representative; YOLO attempts to build a fast real-time object detector by treating the detection task as a regression problem. The other is the two-stage method, with the R-CNN series [17–20] being a representative work that balances prediction accuracy and inference performance. However, these are time consuming. Additionally, detection methods can be further classified into anchor-based and anchor-free approaches. Another branch involves detectors based on transformers, such as DETR [25] and the Swin Transformer [22]. The transformer structure, utilizing self-attention mechanisms to learn sequences, introduces visual transformers. By treating an image as a sequence of patches, transformer-based detectors achieve competitive performance in terms of object detection benchmarks.

2.2. Re-Identification

As an essential component of vehicle Multi-Camera Multi-Target Tracking in smart city applications, ReID aims to retrieve the same vehicle captured by different cameras. Vehicle ReID not only assists in the intra-camera tracking process but also plays an irreplaceable role in inter-camera tracking. CNN-based ReID methods have received extensive attention and demonstrate powerful feature representation capabilities. Much research has focused on designing effective loss functions, with representative ones including cross-entropy loss [29], triplet loss [30], and cyclic loss [31]. Triplet loss [30] treats the ReID training process as a retrieval ranking problem, aiming to minimize the distance between positive pairs and maximize the distance between negative pairs. Sampling strategies and data generation methods for learning discriminative feature representations, as well as weakly supervised detection and synthetic data [32], have proven beneficial, expanding the training data and reducing the receptive field. Additionally, some post-processing techniques have been proposed to further optimize recognition results, such as model ensemble [33], reranking [34], and image-to-trajectory retrieval [35].

With the development of transformer-based visual tasks, vehicle re-identification has seen significant improvements. However, due to the high cost of data annotation and limited labeled data, CNN-based ReID methods face performance limitations. To address this issue, Generative Adversarial Networks (GAN) [36] are employed in the ReID training process to synthesize more vehicle images for robust vehicle representation. GAN transfer styles from the source domain to the target domain and generate samples

conditioned on specific attributes (e.g., color, orientation, or occlusion). In addition, a vehicle re-identification task based on a transformer [37] has achieved good performance.

2.3. Single-Camera Multi-Target Tracking

SCMT aims to associate the same targets in video sequences and connect each target from an individual camera into a tracking chain with a unique identity. Classical tracking methods are mainly based on probability theory, with Kalman filtering and particle filters laying a solid mathematical foundation for tracking [38] problems. Modern SCMT trackers can be divided into the tracking-by-detection [39–45] paradigm and joint-detection-tracking [46,47] paradigm, with the majority of SCMT algorithms following the tracking-by-detection paradigm. Tracking-by-detection methods, prevalent in SCMT tasks, involve obtaining multiple detection boxes in each frame using an effective detector and then associating them based on appearance and motion cues. With the gradual improvement of object detection technology, tracking-by-detection methods have dominated SCMT tasks over the years. SORT [39] employs the Kalman filter algorithm for motion-based multi-object tracking based on observations from a deep detection model. DeepSORT [40] introduces deep visual features into the SORT framework for object association, using an independent ReID model to reduce identity switches. BoT-SORT [48] adopts a strategy of IoU–ReID fusion, which further improves IDF1 and MOTA by combining IoU and cosine distance matrices with motion and appearance information. Recently, several joint-detection-tracking methods have been designed to integrate appearance embeddings or motion predictions into the detection framework and train both tasks within the same framework. JDE [46] uses weight-sharing CNNs for object detection and ReID feature extraction. FairMOT [45] delves into the differences between detection and ReID tasks. While joint trackers achieve comparable performance at a lower computational cost, they face the challenge of reduced tracking performance due to competition between different components. In addition, recent applications of transformer-based tracking, along with ECO [49], have shown success. TransMOT [50] utilizes graph transformers for spatio-temporal modeling. Xu et al. [11] partially address the issue of object detection and tracking loss caused by specific environmental conditions by employing high recall and static object filtering methods. Rios et al. [10] demonstrate that integrating motion information as supplementary data alongside appearance-based cues extracted in Haar feature form relative to given detection windows can enhance the overall performance of multi-camera tracking systems. Wang et al. [8] contributed to tracking by leveraging MedianFlow for robust vehicle localization under occlusion and integrating ECO for improved predictions during fast or sharp movements. Furthermore, leveraging stage-wise differences in vehicle motion and employing hierarchical matching have been found to optimize tracking performance. The success of the latest SORT-like [41–43,48] frameworks suggests that, in terms of tracking accuracy, the tracking-by-detection paradigm remains the optimal solution.

2.4. Inter-Camera Association

The task of Multi-Camera Multi-Vehicle Tracking is to link tracking chains from different cameras. Based on the results obtained from the three modules mentioned above, inter-camera association can be considered as a trajectory matching or trajectory retrieval problem. Many previous works [1–4,6–8] have treated cross-camera matching as a trajectory clustering problem. Recently, many studies have found that traffic rules and spatio-temporal constraints can serve as prior knowledge for filtering candidate trajectories, significantly reducing the search space and thereby greatly improving the accuracy of vehicle ReID. Ye et al. [2] associated all candidate trajectories between two successive cameras using the Box-Grained Reranking Matching algorithm. Yao et al. [3] predefined entry/exit zones and proposed a zone-gate- and time-decay-based matching mechanism to adjust the original appearance matrix. Liu et al. [1] introduced inter-camera sub-clustering to collect potential trajectory pairs from two cameras and adopted partitioning for intersections to exclude vehicles leaving the main road and to resolve conflicts in

the direction of travel. He et al. [51] performed matching of candidate trajectories from adjacent cameras based on the overlap of camera fields of view. In this paper, we impose spatio-temporal constraints on candidate trajectory pairs for overlapping regions using a time mask based on the flow of vehicles entering and exiting the highway tunnel. Finally, hierarchical agglomerative clustering is applied to assign global trajectory IDs.

3. Method

This section introduces detailed information about our Multi-Camera Multi-Vehicle Tracking system. As shown in Figure 2, our system comprises four stages, vehicle detection, vehicle re-identification, Single-Camera Multi-Vehicle Tracking, and inter-camera association. The detailed process will be described below.

3.1. Vehicle Detection

Vehicle detection is a fundamental and crucial module in Multi-Camera Multi-Vehicle Tracking, and reliable vehicle detection is a prerequisite for such tracking. In a highway tunnel scenario, we only need to focus on two categories related to vehicles: cars and trucks. The challenges in the scene arise from dim lighting inside the highway tunnel and motion blur caused by the highway movement of vehicles. To mitigate the impact of these issues and achieve optimal performance in obtaining bounding boxes for each frame, we employ a state-of-the-art object detection framework, Swin Transformer Detection [22]. This framework can generate multi-scale feature maps to search for fine-grained features, better concentrating on the regions of interest for detecting vehicles. Simultaneously, to avoid the same target being detected multiple times in different categories, we apply Non-Maximum Suppression (NMS) to all detected targets. All detected BBoxes in the same video frame are filtered based on Intersection over Union (IoU) and confidence scores to prevent duplicate detection boxes. Finally, the cropped images of the detected vehicle BBoxes for each frame are sent to the next stage to extract ReID features.

3.2. Vehicle Re-Identification

Vehicle re-identification is a fundamental task in Multi-Camera Multi-Vehicle Tracking. To extract robust and discriminative vehicle appearance features, common networks such as ResNet-101-IBN, ResNeXt-101-IBN, ResNest101, and Se-ResNet101-IBN are used as backbone networks for ReID training. Given the high similarity in brand, model, and color among vehicles in cross-camera vehicle tracking tasks, as well as the significant visual appearance variations caused by different cameras, lighting conditions, and occlusions inside the tunnel, it is crucial to reduce the intra-class distance to keep vehicles with similar appearance features close in the feature space. Simultaneously, expanding the inter-class distance ensures that vehicles with significantly different appearances remain at a considerable distance. To achieve this, a combination of cross-entropy loss [29] and triplet loss [30] functions is employed.

Specifically, we use ResNet-101-IBN [52,53], ResNext-101-IBN [53], ResNest101 [54], and Se-ResNet101-IBN [53] as backbone networks for training ReID. These networks are pre-trained on 384 $\times$ 384 input images, and the stride of the last pooling layer is set to 1 to retain more details of the vehicles, obtaining fine-grained features. The final result is a 2048-dimensional average feature vector representing the vehicle. Additionally, extensive data augmentation is applied during training. The ReID network is trained using a combination of cross-entropy loss and triplet loss. The loss function can be expressed as:

$$L_{reid} = L_{cls} + \alpha L_{trp} \tag{1}$$

where L_{cls} represents softmax cross-entropy loss, L_{trp} represents triplet loss, and α represents the balance weight.

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log\left(\hat{y}_{ij}\right) \tag{2}$$

where N is the number of training samples per batch, C is the number of vehicle identities, y is the label of the input image, $\hat{y}$ is the predicted probability.

$$L_{trp} = \sum_{i=1}^{N} max(m + dist(f_i^a, f_i^n) - dist(f_i^a, f_i^p), 0) \quad (3)$$

where f_a is the feature representation of the anchor sample, f_p is the feature representation of the positive sample, f_n is the feature representation of the negative sample, m is the margin, and dist represents the distance between two features.

3.3. Single-Camera Multi-Target Tracking

The multi-object tracking still focuses on following the tracking-by-detection paradigm, which inputs vehicle detection boxes and 2048-dimensional appearance features into the Single-Camera Multi-Object Tracking stage. Target objects are associated in each camera's video frames so that single-camera vehicle tracking trajectories can be obtained. The scale variation of targets caused by the near-large far-small field of view in high-speed scene cameras results in challenges. When vehicles just enter the camera's field of view and are within the near field of view of the camera, the quality of vehicle appearance features is high and occlusion is not severe, facilitating tracking. However, as vehicles drive away to the far end of narrow tunnels, the size of vehicle targets gradually decreases, the detection confidence decreases, and there are track fragmentations caused by occlusions by large vehicles. The scale variation of targets during the vehicle driving process is shown in Figure 4.

(**a**) (**b**) (**c**)

Figure 4. Illustration of target scale variation of vehicles in near, medium, and far field of view. In (**a**–**c**), the vehicles highlighted in red boxes represent high confidence with no occlusion, high confidence with slight occlusion, and low confidence with severe occlusion, respectively.

Based on the scale variation of vehicles during their driving process, we divide the road segment into near, medium, and far field of view regions, as shown in Figure 5, taking C004 as an example.

(**a**) Original image. (**b**) CMLM region map.

Figure 5. Multi-level field of view illustration. The CMLM region map indicates different intervals based on the confidence of the detection boxes, with red, blue, and green representing near, medium, and far field of view regions, respectively, and their confidence thresholds are 0.5, 0.3, and 0.1.

Therefore, we introduce a Cascade Multi-Level Multi-Object Tracking strategy, fully exploiting the advantages of high- and low-confidence vehicle detection boxes in the near-medium field of view to associate and explore potential targets of low-confidence detection boxes in the distant field of view. The matching process is illustrated in Figure 6.

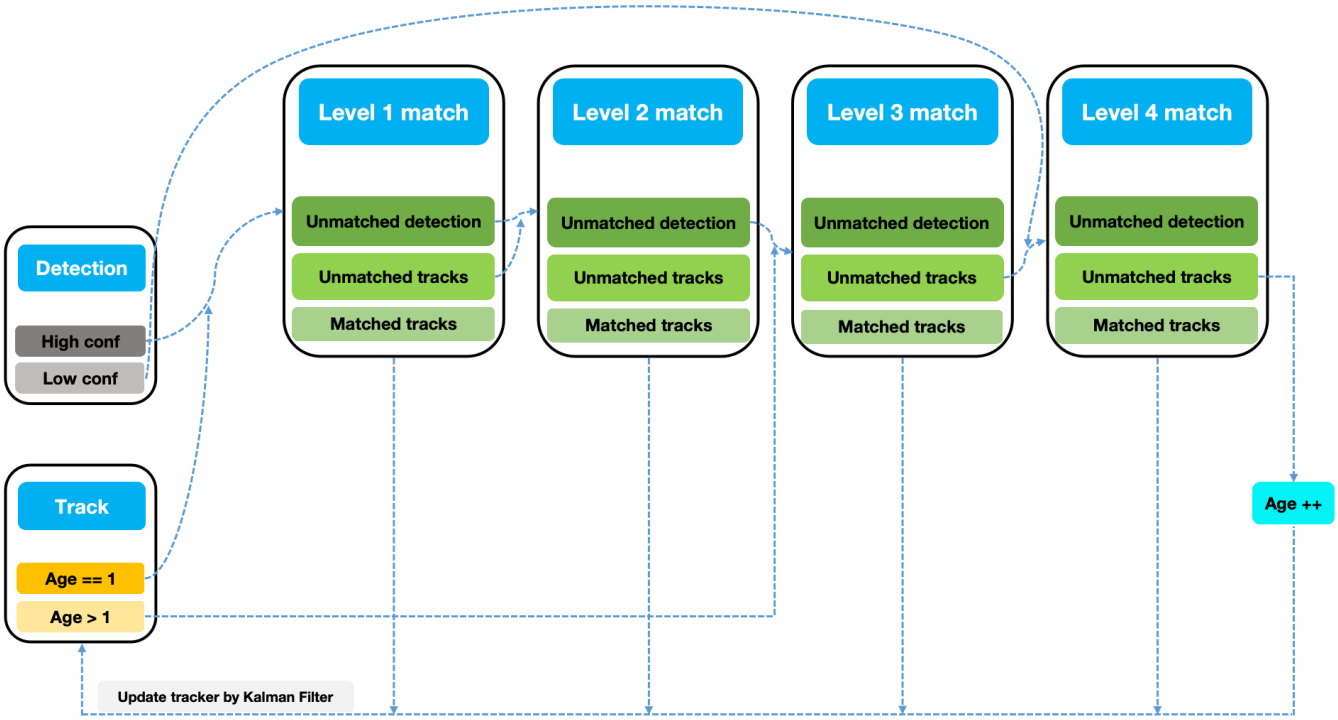

Figure 6. Cascade Multi-Level Multi-Target Tracking association.

Based on the confidence threshold of vehicle detection boxes in the field of view, the vehicle detection boxes are divided into two groups: a high-confidence group with a threshold of 0.3 and a low-confidence group with a threshold of 0.1. Similarity matching is performed on vehicles in the high-confidence group based on their high-quality appearance features, while IoU matching is applied to the low-confidence group. The age of the vehicle tracker is influenced by various factors such as the type of field of view and vehicle occlusion. When a vehicle enters the current camera's field of view, the vehicle tracker is initialized with an age of 1, indicating the initiation of a new vehicle trajectory. As the vehicle target travels to the segment under the field of view of the camera, where the confidence of vehicle detection remains high, occasional mutual occlusion of vehicles may occur. During this period, the age of the tracker gradually increases. An age greater than 1 indicates that the tracker corresponds to an existing vehicle trajectory, while a smaller age reflects the occurrence of occlusion compared to initialization and tracking. Due to increased uncertainty caused by occlusion during long-term tracking, priority is given to trackers with smaller ages when comparing detection boxes. When the vehicle leaves the camera's field of view and moves to a distant section of the road, the tracker's age is larger, and the confidence of vehicle detection is lower. During this period, the dynamic changes in tracker age are accompanied by the entry of large vehicles and occlusion scenarios.

First, the trackers are initialized, and, based on the appearance features of high-confidence vehicle detection boxes in the near-medium field of view, i.e., Figure 4a,b, matching is performed. Successful matching updates the tracker via Kalman filtering. Next, for the medium field of view, due to occlusion between vehicles and the glare effect in the central field of view, first-level unmatched detection boxes and trajectories still enter the second-level association based on high-quality appearance features. Then, for unmatched

detection boxes at this level (e.g., due to occlusion or low appearance similarity caused by lighting changes), they are matched with trackers with ages greater than 1 and smaller ages in the third-level matching. Successful matching updates the tracker. At this point, the advantage of high-quality detection boxes in the near-medium field of view has been fully utilized. Finally, unmatched trajectories in the third-level association (e.g., missed detections, long-term occlusion, or leaving the field of view) are sent to the fourth-level association with low-confidence detection boxes in the distant field of view, i.e., Figure 4c, based on the IoU to enhance the stability of the tracker.

Subsequently, we obtain the trajectory for each vehicle under a single camera, and, for each tracklet, generate a set of vectors:

$$T^{id} = [\mathrm{t_i}, \mathrm{b_i}, \mathrm{f_i}] \tag{4}$$

where T^{id} is the corresponding tracklet's id, t_i is the timestamp, b_i is the corresponding BBox information, and f_i is the associated 2048-dimensional ReID feature.

Figure 7 illustrates the visual tracking results of introducing the Cascade Multi-Level Multi-Target Tracking strategy under a single camera. It is evident that this strategy plays a crucial role in maintaining the coherence of tracking trajectories for large vehicles and improves recognition recall rates.

(**a**) (**b**)

Figure 7. Illustration of vehicle tracking visualization before and after introduction of CMLM under the C004 camera. (**a**) represents before introducing CMLM, there was a situation of trajectory loss for large vehicles; (**b**) represents vehicle with id 19 is tracked correctly after introducing CMLM.

3.4. Inter-Camera Association

The association between adjacent cameras is the final but crucial module in the Multi-Camera Multi-Vehicle Tracking task. After obtaining trajectories generated in the Single-Camera Multi-Vehicle Tracking stage, inter-camera association links all vehicle trajectories with the same ID through vehicle appearance features, road topology structures, and vehicle spatio-temporal information. Considering the characteristics of the highway tunnel section, where overtaking is not considered and vehicles transition in a first-in, first-out manner, various factors are taken into account in this section. We model the scene for the Multi-Camera Multi-Vehicle Tracking task for the highway tunnel road, taking into consideration road topology structures, highway vehicle inflow and outflow, and camera field of view types to guide the cross-camera vehicle trajectory association matching. The process of inter-camera association is illustrated in Figure 8: (1) filtering and refining the raw trajectories generated in the Single-Camera Multi-Vehicle Tracking stage to obtain genuine candidate cross-matching trajectories; (2) sending the candidate cross-matching trajectories to the overlapped field of view Highway Spatio-Temporal Constraint module, calculating similarity distance matrices for trajectories in overlapping views of adjacent cameras, and combining the spatio-temporal mask constraints of highway entry–exit flow to reduce the search space and accelerate matching; (3) performing k-reciprocal reranking [34] for fine-grained updating of the similarity distance matrix; (4) hierarchically clustering based on the similarity matrix to assign global IDs, obtaining cross-camera vehicle tracking chain trajectories.

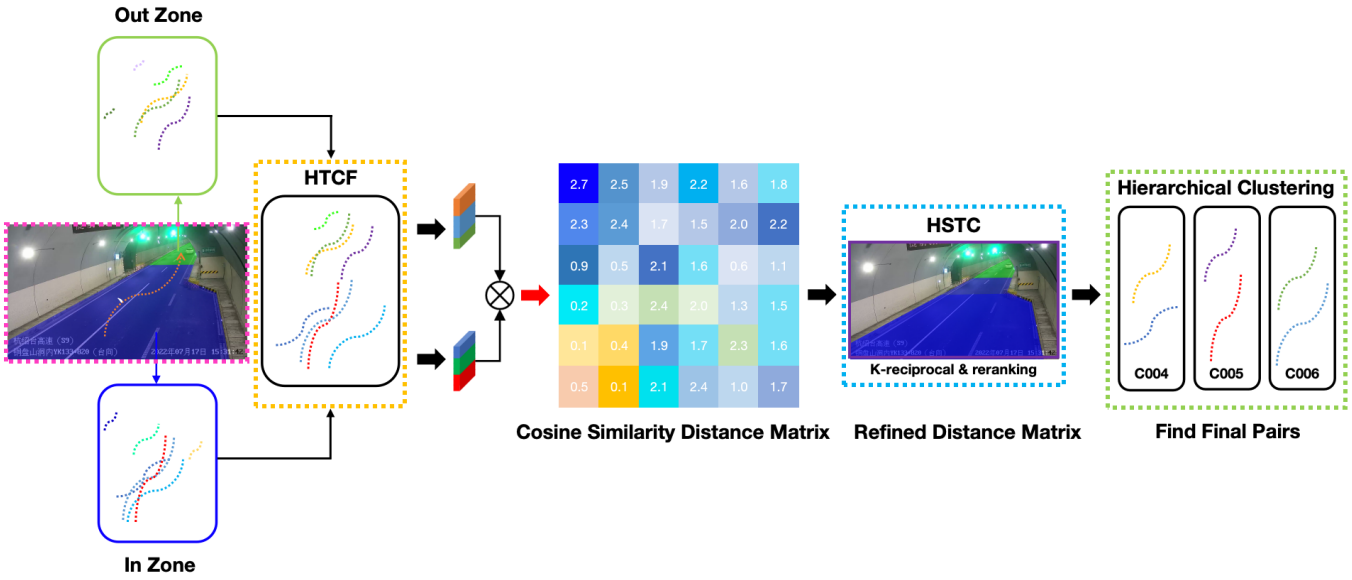

Figure 8. Cross-camera candidate trajectories matching process. The red-boxed image represents single-camera entry and exit trajectory processing, the yellow-boxed module represents the candidate trajectory filtering, while the blue-boxed image and green-boxed stage respectively represent the spatio-temporal constraint module for overlapping fields of view and trajectory clustering across adjacent cameras.

3.4.1. Highway Entry–Exit Flow

This system models the scene based on the road topology structures of each camera's road segment and obtains the highway vehicle inflow and outflow according to the predefined road entry area (in blue) and exit area (in green), as shown in Figure 9, using C005 as an example. For adjacent cameras, the current entry area connects to the exit area of the previous camera (C004), where vehicles leave the current exit area and enter the entry area of the next camera (C006). For each vehicle trajectory under a single camera, we can obtain the transition region $[Z_{in}, Z_{out}]$ for the current camera and their respective entry–exit times $[t_s, t_e]$.

(**a**) C004 (**b**) C005 (**c**) C006

Figure 9. Illustration of zones in highway entry–exit flow. Blue represents driving into the area; green represents driving out of the field of view.

3.4.2. Highway Tracklets Candidate Filter

Due to the strong light caused by static traffic signs on the roadside of the highway and the headlights in the closed tunnel, as well as the imperfections of the vehicle detectors, single-camera vehicle original tracking trajectories may contain a few false-positive detections and errors. These incorrect trajectories will ultimately affect the matching of cross-camera candidate trajectories. Therefore, it is necessary to filter and refine the original trajectories from a single camera to obtain refined candidate cross-matching tracking trajectories. Building upon Figure 9, further distinguishing the candidate trajectory filtering area (white and red) from the highway entry–exit flow area (blue and green) is shown in Figure 10. Additionally, there is a situation where the presence of a tunnel height limit causes the trajectories of generally captured vehicles to be fragmented due to intermittent

obstruction by nearby large vehicles during their complete journey in the tunnel. We need to appropriately split these vehicle trajectories based on the inflow and outflow. When the time not capturing the current vehicle exceeds a predefined threshold, assuming an obstruction has occurred, the camera captures the driving area information of the vehicle from the previous frame, starts splitting the trajectory from the vehicle's entry to the onset of the obstruction, and continues until the obstruction ends. The camera then captures the current vehicle's driving area information, restoring the tracking trajectory and obtaining the transition region [Z_{in}, Z_{out}] and transition time [t_s, t_e] for this vehicle during the obstruction phase. Finally, we obtain the cross-camera candidate matching trajectories for cross-camera matching.

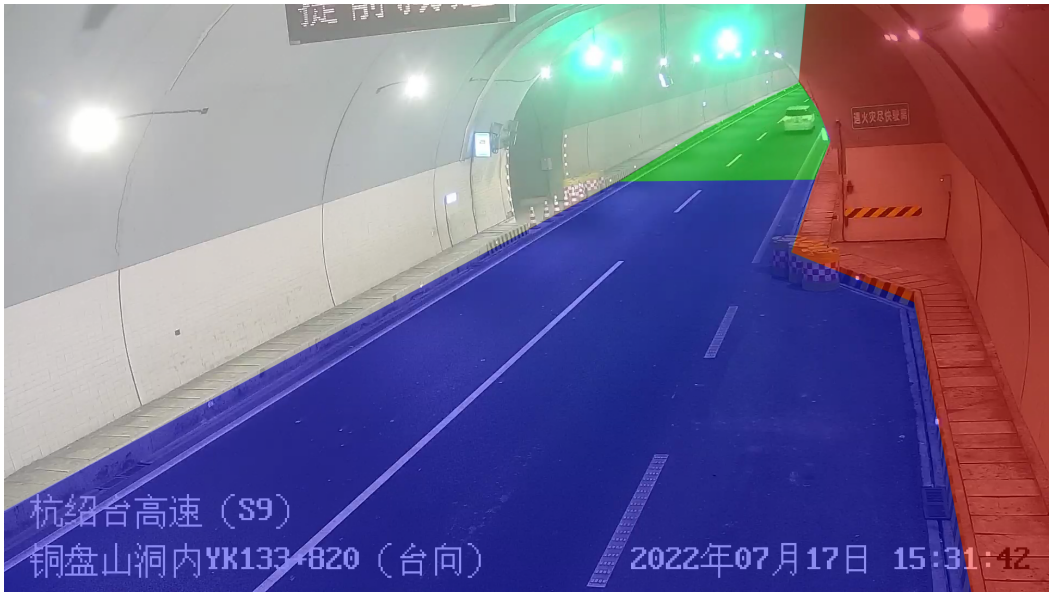

Figure 10. Highway tunnel for C005. Blue and green represent the high-speed incoming and outgoing flows in Figure 9b, respectively, while red and white represent the filtering area (static traffic signs and strong light false detection).

3.4.3. Highway Spatio-Temporal Constraint

There is a large overlap between adjacent cameras, as shown in Figure 11. After obtaining the candidate matching trajectories for cross-camera vehicles, cross-camera trajectory matching is performed based on spatio-temporal prior information constraints within the predefined exit and entry areas of adjacent sub-cameras. Matching pairs are successively searched between adjacent cameras, and, finally, all matching pairs are consolidated across all cameras. The similarity distance matrix for cross-camera candidate matching trajectories is calculated based on the flow of vehicles entering and leaving. The formula for the similarity distance matrix is as follows:

$$S = \begin{bmatrix} \cos(f_{t1}^{N}, f_{t1}^{N+1}) \cdots \cos(f_{t1}^{N}, f_{tj}^{N+1}) \cdots \cos(f_{t1}^{N}, f_{tJ}^{N+1}) \\ \cdots \quad \cdots \quad \cdots \\ \cos(f_{ti}^{N}, f_{t1}^{N+1}) \cdots \cos(f_{ti}^{N}, f_{tj}^{N+1}) \cdots \cos(f_{ti}^{N}, f_{tJ}^{N+1}) \\ \cdots \quad \cdots \quad \cdots \\ \cos(f_{tI}^{N}, f_{t1}^{N+1}) \cdots \cos(f_{tI}^{N}, f_{tj}^{N+1}) \cdots \cos(f_{tI}^{N}, f_{tJ}^{N+1}) \end{bmatrix} \tag{5}$$

where f_{ti}^{N} represents the trajectory features for cross-camera matching using the average features of all frames. I represents the total number of tracklets in camera N, J represents the total number of tracklets in camera $N+1$, N is the ID of the camera, with sequence numbers ranging from C001 to C005, and $\cos(f_{ti}^{N}, f_{tj}^{N+1})$ indicates the cosine similarity between trajectory t_i^N and trajectory t_j^{N+1}.

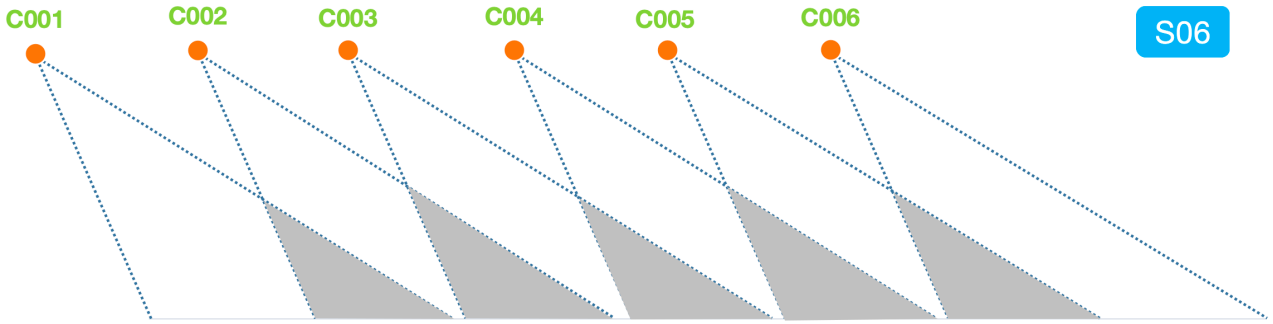

Figure 11. The analysis diagram of overlapping fields of view of adjacent cameras in HST dataset scene S06.

To reduce the search space, we use spatio-temporal masks based on the highway entry–exit flow, ensuring that the travel time of vehicles under adjacent cameras falls within a reasonable interval. Specifically, as shown in Table 1, according to the spatio-temporal mask table for highway flow, the time when a vehicle from the previous camera appears in the exit area should be earlier than the time when the vehicle is in the entry area of the next camera. For example, if the time a vehicle exits C005 is greater than the time it enters C006, it is considered a conflict and cannot be matched. By using spatio-temporal masks based on highway flow, the search space is significantly reduced, accelerating the matching process. Additionally, k-reciprocal reranking is used for fine-grained optimization of the similarity distance matrix. The matrix is reranked based on the magnitude of similarity, placing vehicles with higher similarity to the target vehicle in the front positions and keeping those with lower similarity at a greater distance, further reducing the ReID search space.

Table 1. Overlapping field of view highway entry–exit flow mask table. When T^i and T^j meet the three conditions of camera and time at the same time, T^i and T^j fail to match.

Camera	Time	Conflict
$c^i < c^j$	$t_s^i > [t_s + \frac{di}{\bar{v}}]^j$ $t_s^i > t_e^j$ $t_e^i > t_e^j$	True (T^i, T^j)

c represents the camera ID, t_s^i and t_e^i are the times when the trajectory enters and exits the current camera i, t_e^j is the time when the trajectory leaves the current camera j, d_i (i = 1,2,3,4,5) is the distance between adjacent cameras, and $[t_s + \frac{di}{\bar{v}}]^j$ represents the time when the trajectory enters camera j, specifically, the time when it enters the overlapping field of view of cameras i and j.

Finally, based on the fine-grained similarity distance matrix, hierarchical clustering is performed on the candidate trajectories under adjacent cameras. The results are further integrated into a group clustering across all cameras to assign the same global trajectory ID, resulting in cross-camera vehicle tracking trajectories. The matching algorithm for cross-camera tracking is described in Algorithm 1.

Algorithm 1 Inter-Camera Association

Require: The distance d between each camera $d_i(i$ = 1,2,3,4,5), vehicle transition time under each camera $[t_s, t_e]$, the average speed $\bar{v}$, the average time $\bar{t}$ for the vehicle to enter the current camera N and appear in the next camera $N+1$, tracking trajectory $t_x^N \& t_y^{N+1}$. Average characteristics of trajectory $f_{ti}^N(N$ = 1,2,3,4,5).

Ensure matching trajectory across cameras

CAM_DIST = A, $A = A^T$;
for each camera N, N belongs to [1, 5] **do**
  $\Delta t = t_e - \bar{t}$, $Di = \bar{v} \cdot \Delta t$. (Overlapping-Fovs Dist);
end for
for each trajectory transition time $[t_s, t_e]$ **do**
  Get overlapping-fovs transition time$[t_s + \frac{d_i}{\bar{v}}, t_e]$;
end for
for each trajectory t_x^N, t_y^{N+1} **do**
  Judge whether t_x^N, t_y^{N+1}, conflict based on the spatio-temporal mask of highway entry-exit flow;
end for
for rest trajectory t_x^N, t_y^{N+1} **do**
  According on the mean feature f_{ti}^N of the trajectory to calculate the cosine similarity $\cos(f_{ti}^N, f_{tj}^{N+1})$ between a and b;
  Calculate similarity distance matrix $sim(t_x^N, t_y^{N+1})$;
end for
K-reciprocal nearest reranking to refine S_{sim};
Hierarchical clustering:
  (1) Set a distance threshold of 0.5 within a single camera for tracklets clustering;
  (2) Set a distance threshold of 0.9 between adjacent cameras and select the tracklets pairs with the smallest distance for clustering;
  Assign same global ID of trajectory;
  return Visual tracking chain

4. Experiment

4.1. Datasets

Our cross-camera multi-vehicle tracking system is evaluated using the S06 scene of the Highway Surveillance Traffic (HST) dataset and the S02 scene of the CityFlow dataset. The HST dataset was meticulously curated and gathered by our team in the Tongpanshan Tunnel section of the Hangzhou–Shaoxing–Taizhou Highway in Shaoxing, Zhejiang Province, China. The videos, captured by six cameras (C001, C002, C003, C004, C005, C006), feature a resolution of 1920 × 1080 pixels. They encompass a highway tunnel road divided into six segments (340, 460, 580, 700, 820, 940), providing approximately 3.03 h (181.8 min) of traffic videos. The CityFlow [55] dataset, on the other hand, is the largest and most representative Multi-Camera Multi-Vehicle Tracking dataset captured in a real urban scenario. It comprises videos from 46 cameras at 16 intersections in a medium-sized U.S. city: 3.58 h (215.03 min) of traffic videos covering six different road scenes. For our evaluation, we select scenario S02 from the CityFlow dataset. The experimental results for these datasets demonstrate the effectiveness and robustness of the proposed method.

4.2. Evaluation Metrics

For the Multi-Camera Multi-Vehicle Tracking task, we utilize IDF1, IDP, IDR, and MOTA as evaluation metrics. IDF1 measures the ratio of correctly identified detections to the ground truth and considers the average number of computed detections. MOTA comprehensively takes into account factors such as false positives, false negatives, trajectory switches, miss rate, and false alarm rate, providing a holistic assessment of the system's overall accuracy and reliability in the Multi-Camera Multi-Vehicle Tracking (MCMVT) task. The specific formulas for IDF1, IDP, IDR, and MOTA are as follows:

$$IDP = \frac{IDTP}{IDTP + IDFP} \tag{6}$$

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{7}$$

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{8}$$

$$MOTA = 1 - \frac{FN + FP + IDSW}{GT} \tag{9}$$

where $IDTP$ represents the number of true-positive identities, $IDFP$ is the number of false-positive identities, $IDFN$ denotes the number of false-negative identities, FP is the number of false positives (incorrectly detected as targets), FN is the number of false negatives (targets not detected), $IDSW$ is the number of identity switches (i.e., the times different targets are incorrectly linked), and GT stands for the ground truth. The formulas for calculating $IDTP$, $IDFP$, and $IDFN$ are as follows:

$$IDFN = \sum_{\tau} \sum_{t \in T_{\tau}} m(\tau, \gamma_m(\tau), t, \Delta) \tag{10}$$

$$IDFP = \sum_{\gamma} \sum_{t \in T_{\gamma}} m(\tau_m(\gamma), \gamma, t, \Delta) \tag{11}$$

$$IDTP = \sum_{\tau} len(\tau) - IDFN = \sum_{\gamma} len(\gamma) - IDFP \tag{12}$$

where τ represents the ground truth trajectory, and $\gamma_m(\tau)$ denotes the best match for the computed trajectory τ. Here, γ represents the computed trajectory, $\tau_m(\gamma)$ represents the best match for the ground truth trajectory γ, t stands for the frame index, and Δ is used as the IoU threshold to determine whether the computed bounding box matches the ground truth bounding box (where Δ is set to 0.5). Additionally, $m(\cdot)$ denotes the mismatch function, set to 1 if a mismatch exists, otherwise unmatched at t and 0.

4.3. Implementation Details

This Multi-Camera Multi-Vehicle Tracking system operates on a single Tesla V100 (32G) GPU, employing the PyTorch 1.13.1 deep learning framework in a CUDA 11.6 and Python 3.10 environment. For vehicle detection, the sliding window self-attention mechanism has been integrated into YOLOv5, along with the addition of the CBAM [56] attention module. A model pre-trained on the UA-DETRAC [57] dataset is used for vehicle detection, with an image size set to 1280, confidence threshold at 0.1, and NMS and IoU algorithms applied to reduce redundant detection boxes. In vehicle ReID, training data include CityFlow-V2, CityFlow-V2-CROP (an extension of CityFlow-V2 with weakly supervised detection to augment training data to overcome a lack of real-world data), VehicleX, and VehicleX-SPGAN (using SPGAN to transfer synthetic images from VehicleX to real-world images to minimize domain bias).

ResNet101-IBN, ResNext101-IBN, Se-ResNet101-IBN, and ResNest101 are used as backbone networks for training and inference. The cropped image size is adjusted to (384, 384) for ReID feature extraction with a batch size of 64 and a feature dimension of 2048. Additionally, a combination of cross-entropy loss and triplet loss functions is used to enable the ReID model to simultaneously increase inter-class distance and decrease intra-class distance. Finally, the three backbone networks are fused into a unified model to enhance the robustness of vehicle ReID.

For the Single-Camera Multi-Vehicle Tracking stage, the high confidence threshold is set at 0.3, the low confidence threshold at 0.1, and the IoU confidence for associating low-confidence and high-confidence vehicle detection boxes is set to 0.5 and 0.85. The maximum cosine distance is 0.5, the frame threshold is 50, and the average transition time for vehicles across cameras is shown in Table 2, with a video frame rate of 25.

During the selection of thresholds in hierarchical clustering, vehicles with relatively high similarity are considered likely to belong to the same category in the first clustering

step. Therefore, we choose a relatively low threshold of 0.5 for the initial clustering. In the second clustering step, a threshold of 0.9 is selected to define vehicle similarity more strictly, ensuring that only vehicles with very high similarity are grouped into the same category.

Table 2. The average transition time of different adjacent camera pairs.

Camera Pairs	(C001, C002)	(C002, C003)	(C003, C004)	(C004, C005)	(C005, C006)
Average Transition time (s)	4.2	5.6	5.3	5.3	5.2

4.4. Comparative Experiment

(1) Comparison of different detectors in the performance of the MCMVT system, as shown in Table 3.

- YOLOv8 [26]: the YOLOv8 model enhances its detection capabilities by combining state-of-the-art backbone and neck architectures and introducing an anchor-free split Ultralytics head. This approach eliminates the need for anchor boxes and balances optimization accuracy and speed, further enhancing the model's detection abilities;
- Swin Transformer [22]: the Swin Transformer leverages a sliding window self-attention mechanism to effectively handle both global and local information through hierarchical partitioning, achieving outstanding performance in object detection tasks with strong scalability and computational efficiency;
- YOLOv9 [27]: YOLOv9 introduces the concept of Programmable Gradient Information (PGI) to address information loss during data transmission in deep networks. Additionally, it designs a lightweight network architecture called the General Efficient Layer Aggregation Network (GELAN) based on gradient path planning, achieving better parameter utilization using traditional convolution operators alone.

Table 3. Comparison of MCMVT results and parameter quantities with different detectors.

Model	Param	IDF1	IDP	IDR	MOTA
YOLOv8-x	64.97 M	73.53	74.67	72.41	61.32
YOLOv9-e	68.66 M	74.11	74.04	74.18	61.69
Ours	92.65 M	**81.20**	**81.79**	**80.62**	**62.67**

(2) Comparison with other methods for HST database: As shown in Table 4, our proposed method is compared with several state-of-the-art Multi-Target Multi-Camera Tracking (MTMCT) systems using the HST database. The visual tracking trajectories of the HST database are represented in Figure 12.

- NCCU: NCCU focuses on optimizing the matching of vehicle image features and geometric factors, such as trajectory continuity, vehicle movement direction, and the continuous travel duration under different camera views;
- ELECTRICITY: ELECTRICITY is an efficient and accurate multi-camera vehicle tracking system that combines aggregation loss with a rapid multi-target cross-camera tracking strategy, enabling real-time and robust vehicle tracking across diverse surveillance scenarios;
- DyGlip: DyGlip introduces attention mechanisms into dynamic graphs. It includes a structured attention layer that considers embedded feature information and camera-specific information. Additionally, it incorporates a time attention layer containing temporal information. These components are then encoded and decoded;
- BUPT: BUPT employs clustering loss and trajectory consistency loss to extract robust visual vehicle features optimized for the MTMCT task. By leveraging clustering techniques and trajectory consistency measures, the goal is to enhance the robustness and accuracy of vehicle tracking across multiple cameras, facilitating effective multi-target tracking in complex surveillance scenarios.

Table 4. Comparison of MCMVT results for HST database.

Method	IDF1	IDP	IDR	MOTA
NCCU	43.38	62.05	33.35	12.99
ELECTRICITY	56.23	73.24	45.63	35.16
DyGlip	66.75	73.69	61.01	55.75
BUPT	70.94	71.24	70.64	54.62
Ours	**81.20**	**81.79**	**80.62**	**62.67**

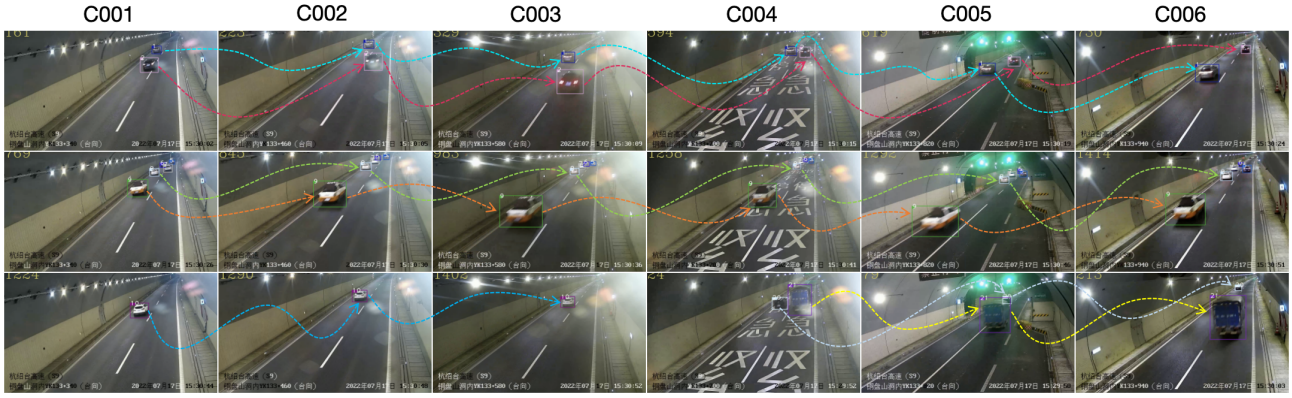

Figure 12. Visualization of Multi-Camera Multi-Vehicle Tracking results for the HST dataset. Each column represents a camera ID. Each row represents vehicles with the same ID correctly tracked by the camera group. For example, in the first row, vehicles in dark-blue and pink boxes are correctly tracked by cameras C001 to C006; the second row represents tracking chains for vehicles with IDs 5, 6, 7, 8, and 9, respectively; the third row represents tracking trajectories for cars with IDs 10 and 20 and a truck with ID 21, respectively.

(3) Comparison with other methods using CityFlow: As shown in Table 5, our method is evaluated on the validation set S02 of CityFlow and compared with other methods. The qualitative results for CityFlow are represented in Figure 13.

Table 5. Comparison of MCMVT results for CityFlow.

Method	IDF1	IDP	IDR
NCCU	45.97	48.91	43.35
ELECTRICITY	53.80	-	-
DyGlip	**64.90**	59.56	**71.25**
Ours	64.43	**67.56**	61.59

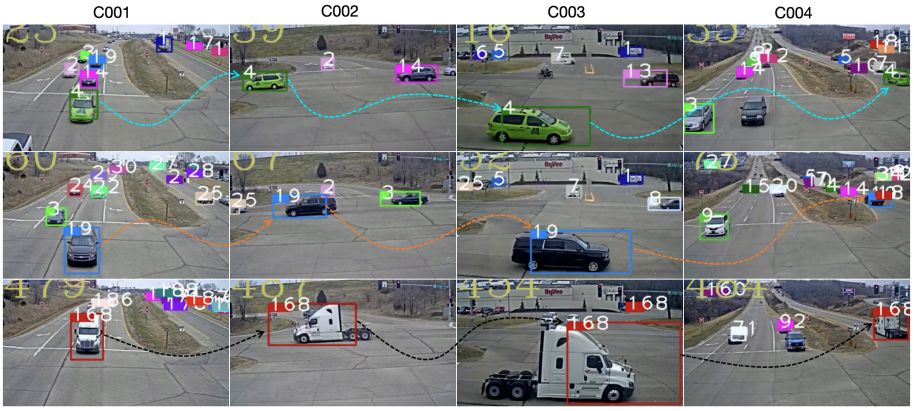

Figure 13. Qualitative results of MCMVT for CityFlow. From left to right, the blue, orange, and red lines with arrows represent the tracking trajectories of IDs 4, 19, and 168 under each camera.

4.5. Ablation Experiment

To evaluate the effectiveness of our proposed method, we conduct ablation experiments on each increment of the HST dataset. Table 6 demonstrates the impact of different modules on performance (after k-reciprocal reranking). Our baseline uses Swin-Det as the detector, and the ReID model integrates ResNet101-IBN* and Se-ResNet101-IBN. It can be observed that the CMLM module increases the IDF1 score from 64.03% to 69.71%. This improvement is attributed to leveraging the advantages of associating high-confidence detection boxes with high-quality appearance features and exploring potential information from low-confidence detection boxes. The HTCF module effectively enhances the IDF1 value by filtering and refining the original tracking trajectories, which contain noise from roadside static signs and false detections. The HSTC module, combining field of view types with vehicle spatio-temporal priors, significantly boosts the IDF1 score to 81.20%. The experimental results demonstrate the effectiveness of the proposed modules for the Multi-Camera Multi-Vehicle Tracking task for highways.

Table 6. Ablation experiments for each incremental module.

Swin-Det	CMLM	HTCF	HSTC	IDF1	IDP	IDR	MOTA
✓				64.03	58.54	70.64	50.45
✓	✓			69.71	67.68	71.87	62.15
✓		✓		65.77	72.47	60.21	50.30
✓	✓	✓		77.52	78.98	76.11	61.31
✓	✓	✓	✓	**81.20**	**81.79**	**80.62**	**62.67**

Table 6 shows the incremental module strategy comparison. "CMLM" indicates the Cascade Multi-Level Multi-Target Tracking strategy, "HTCF" indicates the Highway Trajectory Candidate Filtering module, and "HSTC" indicates the Highway Spatio-Temporal Constraints module. Experimental results for each incremental module are obtained after applying k-reciprocal reranking.

As shown in Table 7, the impact on system performance is demonstrated by solely utilizing motion information and employing different vehicle appearance feature thresholds under the Cascade Multi-Level Multi-Object association strategy, where setting the ReID threshold to 0.3 achieves an IDF1 of 81.20%.

Table 7. Performance comparison between utilizing solely motion information and employing different ReID thresholds (thr).

Method	IDF1	IDP	IDR	MOTA
IoU	79.12	**86.11**	73.17	61.37
CMLM/thr=0.1	80.38	81.64	79.17	61.36
CMLM/thr=0.2	80.43	81.48	79.40	61.35
CMLM/thr=0.3	**81.20**	81.79	**80.62**	**62.67**
CMLM/thr=0.4	80.68	80.99	80.38	61.52
CMLM/thr=0.5	80.43	81.31	79.57	61.28

Additionally, we examine the impact of different backbone networks (ResNet101-IBN, ResNet101-IBN*, ResNext101-IBN, ResNext101-IBN*, Resnest101, and Se-ResNet101-IBN) on the system, as shown in Table 8. Subsequently, we conduct combination experiments to assess the influence of different backbone networks on system performance individually, as presented in Table 9. Finally, we choose to merge ResNet101-IBN* and Se-ResNet101-IBN to obtain robust features for vehicle tracking.

Table 8. The performances of each backbone.

Backbone	IDF1	IDP	IDR	MOTA
IBN-ResNet101	80.05	81.53	78.62	61.09
IBN-ResNet101*	80.30	81.81	78.84	61.31
ResNest101	77.19	78.56	75.86	60.85
IBN-ResNext101	76.75	78.19	75.36	61.25
IBN-ResNext101*	77.82	79.70	76.03	59.98
IBN-Se-ResNet101	80.26	81.80	78.78	61.26

Table 9. The performances of different backbone combinations.

IBN-ResNet101	IBN-ResNet101*	ResNest101	IBN-ResNext101	IBN-ResNext101*	IBN-Se-ResNet101	IDF1	IDP	IDR	MOTA
✓	✓				✓	80.50	80.74	80.27	61.38
		✓	✓	✓		79.46	79.85	79.06	60.41
	✓		✓	✓		80.55	80.72	80.38	61.18
	✓		✓		✓	80.55	80.73	80.38	61.19
	✓				✓	**81.20**	**81.79**	**80.62**	**62.67**
		✓		✓		80.37	80.38	80.36	60.75
			✓	✓		80.36	80.43	80.29	60.76
✓					✓	80.18	80.09	80.27	60.58
✓	✓	✓	✓	✓	✓	80.50	80.63	80.36	61.06

5. Discussion

Despite our proposed method and modules showing promising results for the HST and CityFlow datasets, it is important to acknowledge certain limitations. Firstly, our MCMVT system lacks real-time capability and cannot accurately track vehicle targets in real time online. Secondly, our system is designed for specific scenario patterns; thus, the robustness of the system in different scenarios needs improvement, and it cannot simultaneously accommodate scenes such as large-scale intersections and highways.

6. Conclusions

In this paper, for the Multi-Camera Multi-Vehicle Tracking task and highway tunnel scenarios, we introduced a Cascade Multi-Level Multi-Target Tracking strategy to address the issue of losing tracking trajectories of large vehicles in the field of view, thereby improving the tracking performance for large vehicles. The Candidate Trajectory Filtering module, based on scene modeling, aims to refine the original trajectories of Single-Camera Multi-Vehicle Tracking based on highway entry–exit flow, cutting off garbage trajectories in the cross-matching process at the source. The Highway Spatio-Temporal Constraints module leverages the topological structure types and spatio-temporal prior information of the camera field of view, combined with the highway entry–exit-flow-based spatio-temporal mask, significantly reducing the search space, thereby achieving fast and high-quality cross-camera matching. Additionally, ablation experiments validated the effectiveness of each module. Our proposed method was applied to a Multi-Camera Multi-Vehicle Tracking system on the HST dataset, achieving an IDF1 of 81.20%. Moreover, the versatility of our system was corroborated through validation on the CityFlow dataset, highlighting its effectiveness and robustness.

In future work, we will focus on two aspects. Firstly, we will collect datasets from diverse highway scenarios, including dual carriageways, nighttime conditions, and various weather conditions such as snowy and foggy. Additionally, we will focus on the construction of a real-time online MCMVT system while enhancing scene robustness to further advance research on intelligent highway traffic systems.

Author Contributions: Conceptualization, H.Z., R.F., X.F. and S.C.; investigation, R.F., S.L., Q.M., J.H. and S.C.; methodology, H.Z., R.F., S.C. and X.F.; software and validation, H.Z., R.F., Q.M., J.H. and X.F.; writing—original draft preparation, H.Z., R.F. and S.C.; data Curation, H.Z., R.F., X.F. and S.C.; writing—review and editing, S.L., Q.M., J.H. and X.F.; formal analysis, R.F., S.L., Q.M., J.H. and H.Z.; visualization, R.F., S.L., Q.M., X.F. and H.Z.; supervision, H.Z., X.F. and S.C.; funding acquisition, H.Z., X.F., J.H. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Zhejiang Provincial Natural Science Foundation of China (grant nos. LY23F020023, LZ23F020001); the National Natural Science Foundation of China (grant nos. 61906168, 62201400, and 62272267); the Construction of Hubei Provincial Key Laboratory for Intelligent Visual Monitoring of Hydropower Projects (grant no. 2022SDSJ01), the Hangzhou AI major scientific and technological innovation project (grant no. 2022AIZD0061), and the Foundation of Key Laboratory of Architectural Cold Climate Energy Management, Ministry of Education (no. JLJZHDKF022023003).

Data Availability Statement: The MCMVT system proposed by us was evaluated on the HST dataset, created by our team, and CityFlow, a widely used public dataset for Multi-Camera Multi-Vehicle Tracking. The HST dataset, presented in this article, is not readily available due to several reasons. First, the data acquisition process was challenging, involving multiple cameras and vehicles in complex environments. Then, annotating the dataset was labor intensive and time consuming, requiring meticulous labeling of vehicle trajectories across different camera views. At last, the HST dataset is part of an ongoing study, and the data are currently being used for further analysis and experimentation. However, the CityFlow dataset can be accessed on 31 May 2023 https://www.aicitychallenge.org/2022-data-and-evaluation/, providing researchers with a valuable resource for Multi-Camera Multi-Vehicle Tracking research.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MCMVT	Multi-Camera Multi-Vehicle Tracking
CMLM	Cascade Multi-Level Multi-Target Tracking
HTCF	Highway Trajectory Candidate Filtering
HSTC	Highway Spatio-Temporal Constraints
HST	Highway Surveillance Traffic
Swin-Det	Swin Transformer Detection

References

1. Liu, C.; Zhang, Y.; Luo, H.; Tang, J.; Chen, W.; Xu, X.; Wang, F.; Li, H.; Shen, Y.D. City-scale multi-camera vehicle tracking guided by crossroad zones. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4129–4137.
2. Yang, X.; Ye, J.; Lu, J.; Gong, C.; Jiang, M.; Lin, X.; Zhang, W.; Tan, X.; Li, Y.; Ye, X.; et al. Box-grained reranking matching for multi-camera multi-target tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 3096–3106.
3. Yao, H.; Duan, Z.; Xie, Z.; Chen, J.; Wu, X.; Xu, D.; Gao, Y. City-scale multi-camera vehicle tracking based on space-time-appearance features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 3310–3318.
4. Huang, H.W.; Yang, C.Y.; Hwang, J.N. Multi-target multi-camera vehicle tracking using transformer-based camera link model and spatial-temporal information. *arXiv* **2023**, arXiv:2301.07805.
5. Hsu, H.M.; Cai, J.; Wang, Y.; Hwang, J.N.; Kim, K.J. Multi-target multi-camera tracking of vehicles using metadata-aided re-id and trajectory-based camera link model. *IEEE Trans. Image Process.* **2021**, *30*, 5198–5210. [CrossRef] [PubMed]
6. Ye, J.; Yang, X.; Kang, S.; He, Y.; Zhang, W.; Huang, L.; Jiang, M.; Zhang, W.; Shi, Y.; Xia, M.; et al. A robust mtmc tracking system for ai-city challenge 2021. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4044–4053.
7. Hsu, H.M.; Huang, T.W.; Wang, G.; Cai, J.; Lei, Z.; Hwang, J.N. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 15–20 June 2019; pp. 416–424.

8. Li, F.; Wang, Z.; Nie, D.; Zhang, S.; Jiang, X.; Zhao, X.; Hu, P. Multi-camera vehicle tracking system for AI City Challenge 2022. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 3265–3273.
9. Castañeda, J.N.; Jelaca, V.; Frías, A.; Pizurica, A.; Philips, W.; Cabrera, R.R.; Tuytelaars, T. Non-overlapping multi-camera detection and tracking of vehicles in tunnel surveillance. In Proceedings of the 2011 International Conference on Digital Image Computing: Techniques and Applications, Noosa, QLD, Australia, 6–8 December 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 591–596.
10. Rios-Cabrera, R.; Tuytelaars, T.; Van Gool, L. Efficient multi-camera vehicle detection, tracking, and identification in a tunnel surveillance application. *Comput. Vis. Image Underst.* **2012**, *116*, 742–753. [CrossRef]
11. Xu, D.; Jiang, Q.; Gu, Y.; Chen, Y.; Wang, Y.; Li, Y.; Gao, M. A Kind of Cross-Camera Tracking Strategy in Tunnel Environment. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 5390–5395.
12. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
14. Jocher, G. *YOLOv5 by Ultralytics*; Zenodo: Geneva, Switzerland, 2020. [CrossRef]
15. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
16. Lin, J.; Yang, C.; Lu, Y.; Cai, Y.; Zhan, H.; Zhang, Z. An improved soft-YOLOX for garbage quantity identification. *Mathematics* **2022**, *10*, 2650. [CrossRef]
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
18. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
19. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
22. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
23. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
24. Zong, Z.; Song, G.; Liu, Y. Detrs with collaborative hybrid assignments training. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 6748–6758.
25. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
26. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO, 2023.
27. Wang, C.Y.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
28. Chang, H.S.; Wang, C.Y.; Wang, R.R.; Chou, G.; Liao, H.Y.M. YOLOR-Based Multi-Task Learning. *arXiv* **2023**, arXiv:2309.16921.
29. Zhang, Z.; Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–11.
30. Hermans, A.; Beyer, L.; Leibe, B. In defense of the triplet loss for person re-identification. *arXiv* **2017**, arXiv:1703.07737.
31. Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; Wei, Y. Circle loss: A unified perspective of pair similarity optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6398–6407.
32. Zheng, Z.; Jiang, M.; Wang, Z.; Wang, J.; Bai, Z.; Zhang, X.; Yu, X.; Tan, X.; Yang, Y.; Wen, S.; et al. Going beyond real data: A robust visual representation for vehicle re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 598–599.
33. Luo, H.; Chen, W.; Xu, X.; Gu, J.; Zhang, Y.; Liu, C.; Jiang, Y.; He, S.; Wang, F.; Li, H. An empirical study of vehicle re-identification on the AI City Challenge. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4095–4102.
34. Zhong, Z.; Zheng, L.; Cao, D.; Li, S. Re-ranking person re-identification with k-reciprocal encoding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1318–1327.

35. Liu, X.; Liu, W.; Mei, T.; Ma, H. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part II 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 869–884.
36. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
37. He, S.; Luo, H.; Wang, P.; Wang, F.; Li, H.; Jiang, W. Transreid: Transformer-based object re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15013–15022.
38. Bishop, G.; Welch, G. An introduction to the kalman filter. *Proc. SIGGRAPH Course* **2001**, *8*, 41.
39. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 3464–3468.
40. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3645–3649.
41. Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; Kitani, K. Observation-centric sort: Rethinking sort for robust multi-object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 9686–9696.
42. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. Strongsort: Make deepsort great again. *IEEE Trans. Multimed.* **2023**, *25*, 8725–8737. [CrossRef]
43. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-object tracking by associating every detection box. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–21.
44. Yu, F.; Li, W.; Li, Q.; Liu, Y.; Shi, X.; Yan, J. Poi: Multiple object tracking with high performance detection and appearance feature. In Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–16 October 2016; Proceedings, Part II 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 36–42.
45. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
46. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards real-time multi-object tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 107–122.
47. Jin, J.; Wang, L.; You, Q.; Sun, J. Multi-Object Tracking Algorithm of Fusing Trajectory Compensation. *Mathematics* **2022**, *10*, 2606. [CrossRef]
48. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv* **2022**, arXiv:2206.14651.
49. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
50. Chu, P.; Wang, J.; You, Q.; Ling, H.; Liu, Z. Transmot: Spatial-temporal graph transformer for multiple object tracking. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 4870–4880.
51. He, Y.; Wei, X.; Hong, X.; Shi, W.; Gong, Y. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Trans. Image Process.* **2020**, *29*, 5191–5205. [CrossRef]
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
53. Pan, X.; Luo, P.; Shi, J.; Tang, X. Two at once: Enhancing learning and generalization capacities via ibn-net. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 464–479.
54. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. Resnest: Split-attention networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 2736–2746.
55. Tang, Z.; Naphade, M.; Liu, M.Y.; Yang, X.; Birchfield, S.; Wang, S.; Kumar, R.; Anastasiu, D.; Hwang, J.N. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8797–8806.
56. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
57. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.C.; Qi, H.; Lim, J.; Yang, M.H.; Lyu, S. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [CrossRef]