Box-Grained Reranking Matching for Multi-Camera Multi-Target Tracking

Xipeng Yang* Jin Ye*† Jincheng Lu* Chenting Gong Minyue Jiang
Xiangru Lin Wei Zhang Xiao Tan Yingying Li Xiaoqing Ye Errui Ding
Department of Computer Vision Technology (VIS), Baidu Inc., China
{yangxipeng01,yejin03,lujincheng01,gongchenting,jiangminyue}@baidu.com
{linxiangru,zhangwei99,tanxiao01,liyingying05,yexiaoqing,dingerrui}@baidu.com

Abstract

*Multi-Camera Multi-Target tracking (MCMT) is an essential task in intelligent transportation systems. It is highly challenging due to several problems such as heavy occlusion and appearance variance caused by various camera perspectives and congested vehicles. In this paper, we propose a practical framework for dealing with the city-scale MCMT task, consisting of four modules. The vehicles detection and ReID feature extraction are the first two modules, which locate all vehicles and extract the appearance features for all cameras. The third module is Single-Camera Multi-Target tracking (SCMT), which tracks multiple vehicles to generate candidate trajectories within each camera on the basis of the detected boxes and appearance features. The last module is Inter-Camera Association (ICA), which associates all candidate trajectories between two successive cameras using the K-reciprocal nearest neighbors algorithm, and combines all successively matched trajectories for final results. The ICA module takes the constraints of traveling time, road topology structures, and traffic rules into consideration to reduce the searching space and accelerate the matching speed. Experiments results on the public test set of 2022 AI CITY CHALLENGE Track1 demonstrate the effectiveness of our method, which achieves IDF1 of 84.86%, ranking **1**st on the leaderboard.*

1. Introduction

With the rapid development of intelligent transportation systems, the demand for Multi-Camera Multi-Target tracking (MCMT) has attracted extensive attention in recent years. The purpose of the MCMT task is to track various vehicle targets across multiple cameras as shown in Figure 1, which helps to analyze the traffic flow and travel times along entire corridors. A system designed to tackle the MCMT task typically consists of four sub-modules, namely vehicle detection, Re-Identification (ReID), Single-Camera Multi-Target tracking (SCMT), and Inter-Camera Association (ICA). The general pipeline can be summarized as follows: First, the module of vehicle detection outputs vehicle coordinates and categories in frames and extracts vehicle features by ReID. Then, based on the vehicle location and learned features, the SCMT module generates candidate trajectories for every single camera. At last, the ICA module matches these candidate trajectories across different cameras to associate targets with global identities.

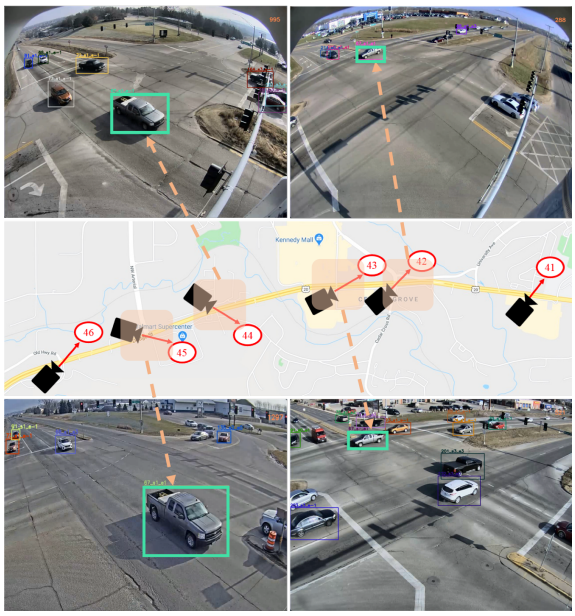


Figure 1. Illustration of Multi-camera Multi-target tracking (MCMT) task. The vehicles with the same identity that appear in multiple cameras will be matched by our proposed system.

In the last few years, there are an increasing number of research efforts dedicated to solving the MCMT task [16, 18, 19, 21, 33, 38, 40, 42, 43]. Although the performance of the current state-of-the-art MCMT model is competitive, there are still several challenges that remain

*Equally-contributed authors. †Corresponding author.

unsolved: the occluded vehicles are hard to detect and vehicles in different cameras have high intra-class variation. For occluded vehicles detection problem, some vehicles can be severely occluded by front ones in heavy traffic scenarios, and cause difficulties for the SCMT module. For high intra-class variation problems, one vehicle in different cameras may suffer from appearance changes caused by light and various camera perspectives. Furthermore, even in the same camera, the extracted features are still hard to distinguish between the two vehicles with similar colors or types.

In this paper, we propose a new MCMT tracking system utilizing the priors of basic traffic rules to alleviate these problems. The pipeline of our proposed MCMT tracking system is shown in Figure 2. Given a set of videos under different cameras, our proposed system first detects all vehicles via a detector and then extracts the appearance features with ReID module. On the basis of the detected boxes and appearance features, our SCMT module generates candidate trajectories within each camera. Finally, the ICA module associates all candidate trajectories between two successive cameras using k-reciprocal nearest neighbors, which are based on the box-grained appearance distance and then combines all matched targets for multi-camera results. Particularly, the ICA module considers the constraints of traveling time, road topology structures, and traffic rules to reduce the searching space as well as accelerate the matching speed.

The rest of the paper is organized as follows: an overview of related work is described in Section 2. Section 3 introduces our proposed framework in detail. In Section 4, we demonstrate sufficient experiments of our method on the track1 of CVPR 2022 AI City Challenge. Finally, we present the conclusion in Section 5.

2. Related work

2.1. Vehicle Detection

Object detection is one of the most popular tasks in the field of computer vision and image processing, and it locates the existence of objects in an image by predicting the bounding boxes and categories. The vehicle detection task is a special object detection branch, which pays more attention to vehicles in images or videos. Based on different backbones, existing object detectors can be divided into two branches: CNN-based object detectors [3, 14, 26, 34, 35] and transformer-based detectors [5, 27, 62].

Owing to the success of convolution networks, the CNN-based detectors have achieved tremendous progress, such as SSD [26], Yolo [34], Faster-RCNN [35], Cascade-RCNN [3]. SSD and Yolo are one-stage detectors, which trade off the speed and accuracy to run in a real-time manner. Faster-RCNN and Cascade-RCNN are two-stage detectors, which are usually more accurate and flexible but time-consuming. The other branch is transformer-based detectors, which are inspired by the success in natural language processing. Transformer structure can learn sequences via self-attention mechanism. The recently-proposed object detectors, such as DETR [5], Swin Transformer [27], introduced vision transformers that achieve competitive performances on object detection benchmarks by treating an image as a sequence of patches. In general, CNN-based detector can capture spatial information inside each patch, which means it can well handle the spatially-local patches, and transformer-based detectors are better at capturing a long-distance pixel relation.

2.2. Re-identification

As one of the most important components in the multi-camera traffic flow, re-identification (ReID) aims to retrieve the same vehicle captured by different cameras [58]. CNN-based ReID methods have received extensive attention and shown strong feature representation ability. In these methods, several loss functions, sampling strategies, and data generation methods are proposed to learn discriminative feature representation.

There are three commonly used loss functions for ReID, including identity loss, verification loss, and triplet loss [53]. By using identity loss such as cross-entropy loss [57], the training process of ReID is treated as an image classification problem. Verification loss such as contrastive loss optimizes the pairwise relationship [44] by treating the training process as an image matching problem. Triplet loss treats the ReID training process as a retrieval ranking problem [17], aiming to make the distance between positive pairs smaller than negative pairs.

Because of the imbalance between positive and negative pairs, sampling strategies play a constructive role in the training process of ReID model. Hermans *et al.* [17] and Chen *et al.* [6] adopt identity sampling to mine informative samples. The basic idea is to sample a certain number of identities in each training batch. Moreover, several adaptive sampling strategies are proposed to better adjust the contribution of positive and negative samples, such as Sample Rate Learning (SRL) [47] and curriculum sampling [45].

CNN-based ReID methods have performance limitations due to the limited amount of labeled data because data annotation is costly. To solve this problem, Generative Adversarial Networks (GAN) are used to synthesize more vehicle images in the ReID training process. Zhou *et al.* [61] extract view-invariant features by transforming single-view features into multi-view features. Chen *et al.* [9] use synthetic data and real data to improve the performance of the foggy vehicle ReID task.

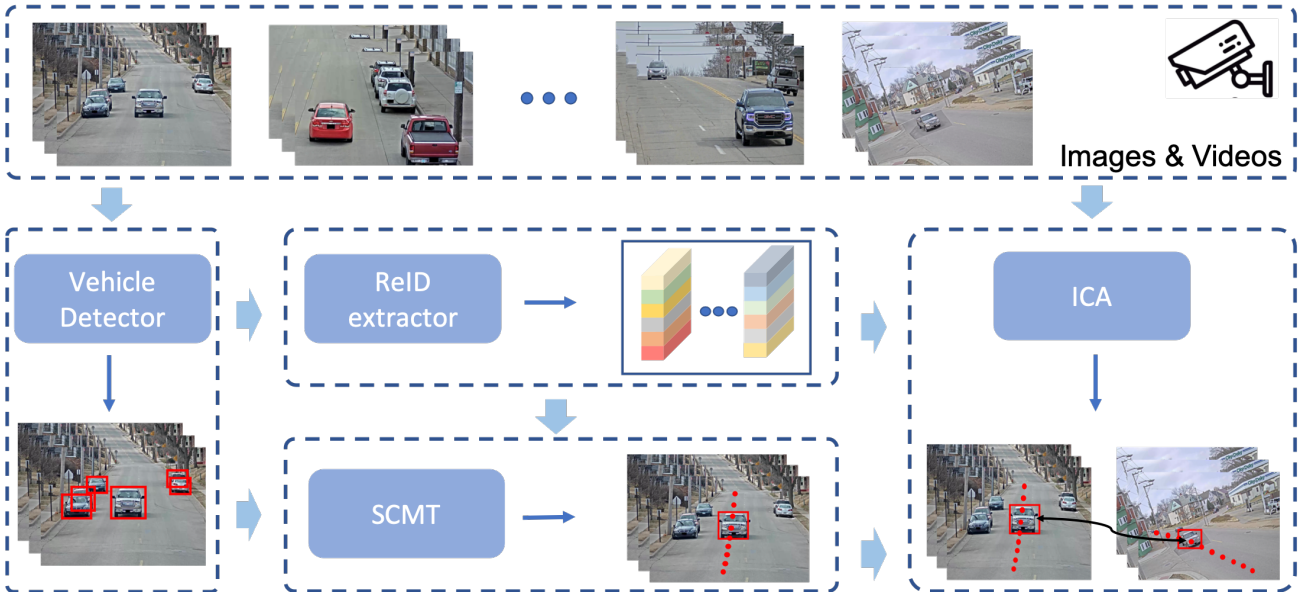


Figure 2. The pipeline of our MCMT tracking system. All vehicle objects are first detected using the detectors and then the ReID module extracts the corresponding appearance features. Then, all the detected boxes and their features are fed into the SCMT Module, which generates all trajectory candidates for every single camera. Finally, our proposed ICA module matches all trajectory candidates across all cameras as the final result.

2.3. Single-Camera Multi-Target Tracking

Modern SCMT trackers can be classified as tracking-by-detection methods and joint-detection-tracking methods. Tracking-by-detection methods [2, 4, 10, 32, 49, 54, 55] obtain detection boxes first and then associate them based on appearance and motion clues. With the improvement of object detection techniques [13, 14, 27, 34, 35], tracking-by-detection methods have dominated SCMT task for years. SORT [2] adopts the Kalman filter algorithm for motion-based multi-target tracking given observations from deep detection models. DeepSORT [49] introduces deep visual features into object association in the framework of SORT. Recently, several joint-detection-tracking methods incorporate appearance embedding or motion prediction into detection frameworks [23, 37, 48, 56, 60]. The joint trackers achieve comparable performance with low computational costs. However, the joint trackers are facing the problem that the competition between different components lowers the upper bound of tracking performance. The success of the latest SORT-like frameworks [4, 10, 55] indicates that the tracking-by-detection paradigm is still the optimal solution in terms of tracking accuracy.

2.4. Inter-Camera Association

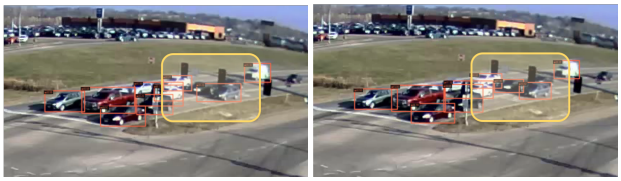
After obtaining all results from the above three modules, the inter-camera association can be treated as trajectories matching or tracklets retrieval problem. Many previous works attempt to tackle this problem from different aspects. Chen *et al*. [7, 8] establish a global graph for multiple tracklets in different cameras and optimize for an MCMT solution. Recently, many works [19, 25, 43, 52] find that traffic rules and spatial-temporal constraints can be regarded as the prior knowledge to filter out the tracklets candidates, which reduces the searching space significantly. After the preprocessing, Hsu *et al*. [19] use the greedy algorithm to search the valid tracklet pairs. Ye *et al*. [52] adopts the Hungarian matching algorithm to find the global optimization results with the distance matrix of all tracklets candidates between two successive cameras. Liu *et al*. [25] introduces hierarchical clustering to gather potential trajectory pairs within two cameras. Different from the existing works, on the one hand, we first construct the distance matrix with box-grained features. On the other hand, we find tracklet pairs in a novel reranking-based way.

3. Method

This section presents the details of our framework for Multi-Camera Multi-Target tracking (MCMT). As shown in Figure 2, there are four modules in our system, including Detection, ReID, SCMT, and ICA.

3.1. Detection

Vehicle detection is a basic and important module in MCMT. To obtain the best performance of bounding boxes



(a) (b)

Figure 3. Examples of detected small and occluded vehicles. The left image shows the vehicle lost if without large input size and image patches strategies. The right image demonstrate our detection model can detect the small and occluded vehicles.

in each frames, we select the state-of-art object detection framework Cascade-RCNN [3]. As most two stage detection network, Feature Pyramid Network(FPN) [24] is followed by backbone to increase semantic features information at each level in the extracted features. We train this vehicle detection model with COCO pretrained weights, and use train and validation data in track1 of AI City Challenge 2022 for final detection model. In the traffic scene, as shown in Figure 3, we find small and occluded vehicles hard to detection out. To solve the above problems, firstly, the larger resolution, data flipping and data cropping are also exploited as data augmentation for facilitating training. Secondly, in test phase, we use two methods to process input images, increasing the maximum resolution to 2666*1600 of input images, and splitting an input RGB image into four patches with overlapping. Then, we feed the processed images into the vehicle detection model, and merge the two output results. From Figure 3, we can see that some small objects and occluded objects can be detected.

3.2. Re-identification

Re-identification is a fundamental task in MCMT. In order to extract robust and discriminative appearance feature representations for vehicles, several common networks such as HRNet [46], ResNeXt101 [51], ResNet [15], Res2Net [12] and ConvNeXt [28] are used as backbone for ReID training. Due to the large visual appearance changes caused by different cameras, vehicle orientation, illuminations and occlusions in the MCMT task, both cross-entropy loss and triplet loss are used for optimization. Given an input image x with label y, the predicted probability of x can be represented as $\hat{y}$. The cross-entropy loss is formulated as follow:

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log(\hat{y}_{ij}) \quad (1)$$

where N represents the number of training samples within each batch, and C represents the number of vehicle identities. Note that $y_{ij} = 1$ if x_i belongs to j th ID, else $y_{ij} = 0$.

Triplet loss focuses on optimizing the distance from a triplet which contains one anchor sample x^a, one positive sample x^p and one negative sample x^n. Given a pre-defined margin m, triplet loss aims to make the distance between positive pairs smaller than negative pairs by m:

$$L_{tri} = \sum_{i=1}^{N} max(m + d(f_i^a, f_i^n) - d(f_i^a, f_i^p), 0) \quad (2)$$

where f^a, f^p, f^n are the feature representations of anchor sample, positive sample and negative sample respectively, and $d(\cdot)$ represents the distance between two features.

In order to improve the performance of ReID features in Track1 of AICity Challenge 2022, two types of model ensemble methods are tried in the ReID module, namely the model soups [50] and feature concatenation. Finally, we simply concatenate ReID features extracted from five models to obtain the ensembled ReID features.

3.3. Single-Camera Multi-Target Tracking

Provided with the high-quality detection results and ReID features, our Single-Camera Multi-Target Tracking focuses on associating targets throughout the video frames following the tracking-by-detection paradigm. We adopt the classic tracker DeepSORT [49] as our baseline method and improve it with various advanced techniques. DeepSORT uses the Kalman filter [2] to predict motion of the tracked targets and adopts Hungarian algorithm [20] to associate detection results to tracklets according to appearance and motion similarity. The unmatched detections are used to initialize new tracklets.

In order not to miss potential targets with low detection confidence (e.g. occluded targets and small targets), we set a high score threshold and a low score threshold to filter detection results as BYTETrack [55] does. We first match the high score detection boxes to the tracklets based on appearance similarity. Then we perform IoU matching between the unmatched tracklets and low score detections with a strict minimum IoU threshold. New tracklets are initiated only for unmatched high score detections.

For the appearance matching, the ensembled ReID features are employed to discriminate targets. Similar to [10, 48] the tracklets appearance states are updated in an exponential moving average (EMA) manner as follows:

$$e_i^t = \alpha e_i^{t-1} + (1 - \alpha) f_i^t \quad (3)$$

where we denote e_i^t as the appearance state of the i-th tracklet at frame t and f_i^t as the ReID feature of the current matched detection, α = 0.9 is a momentum term. For the motion prediction, we upgrade the Kalman filter from two aspects. The vanilla Kalman filter is based on constant velocity motion and a linear observation model, which is not suitable for all situations in reality. In order to reduce the impact of detection noise, we borrow NSA

(a) (b)

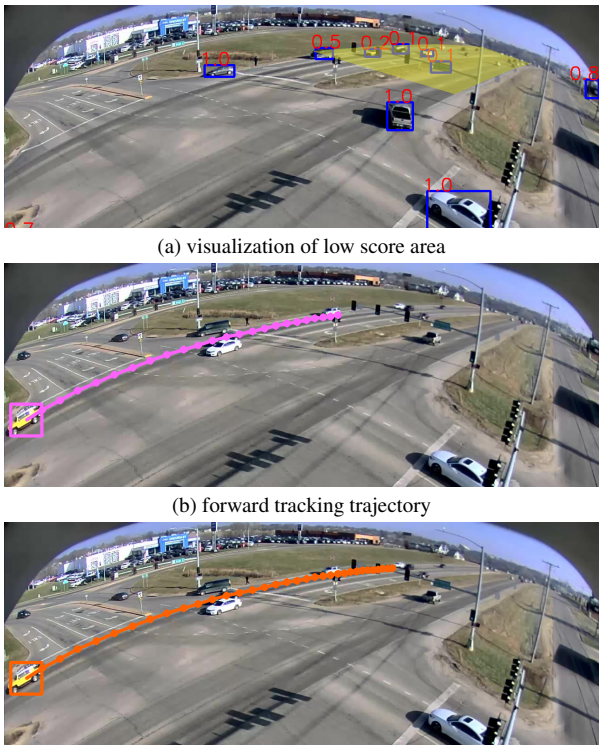
Figure 4. Comparison of tracking results before and after using offline re-link strategy. (a) is the results that the ID of the occluded white car switched from 5 to 94. (b) is the results of using offline re-link.

kalman filter from [10, 11], which incorporates the confidence of detection into covariance calculation. To further improve the robustness of nonlinear motion, we replace the standard Mahalanobis distance [31] with smoothed Mahalanobis distance when measuring motion similarities, similar to [29, 30]. The final similarity distance D is a weighted sum of appearance feature cosine distance d_a and smoothed Mahalanobis distance d_m as follows:

$$D = \lambda d_a + (1 - \lambda) d_m \tag{4}$$

It is difficult to detect objects under severe occlusion in heavy traffic scenarios. Lots of ID switches will occur and affect multi-camera tracking results seriously. As shown in Figure 4a, the track ID of the white car switches from 5 to 94 after being occluded. Because the velocity of the white car changes sharply when it starts, the Kalman filter is unable to predict correct states. To solve this problem, we refine the tracking results with offline re-link. We first screen out trajectories end or start in the middle of the scene. Then we adopt the greedy algorithm to merge the broken tracklets based on the appearance similarities. An interval threshold and a maximum cosine distance threshold are preset to disregard infeasible matching. Figure 4b shows the tracking results of using offline re-link strategy, the track identity of the white car maintains 5 after being occluded.

Although we have adopted various techniques to reduce ID switches, the trajectories can still be incomplete. As shown in Figure 5, the targets entering the scene from the area away from the camera tend to be ignored at the beginning frames, because the detection confidences of small targets are too low to initiate tracklets. Tracking backwards can solve the problem by initiating tracklets in the area close



(a) visualization of low score area

(b) forward tracking trajectory

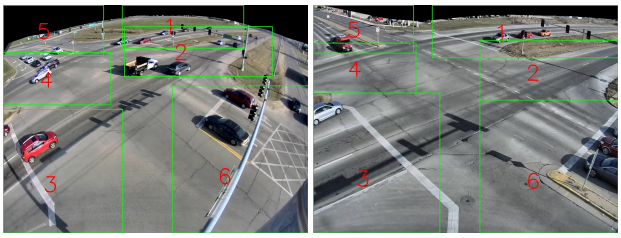
(c) forward and backward tracking trajectory

Figure 5. Example of our tracking method in both temporal directions. (a) shows all the detection boxes with their scores. The detection scores in the yellow area are too low to initialize new tracklets due to occlusion and small size. (b) shows the trajectory of forward tracking results. The beginning part of the trajectory is missing. (c) shows the trajectory after merging forward and backward tracking results, which is more complete.

to the camera. For further improvement, we perform tracking in both temporal directions similar to [39]. By running our tracker on the video frames one time in the forward direction, one time in the backward direction, and merging the tracked targets to generate complete trajectories, the recall can be further improved.

3.4. Inter-Camera Association

Inter-Camera Association (ICA) is the last but important module of the MCMT. Using the trajectories generated by the former three modules, ICA associate all tracklets with same identities by appearance features and spatial-temporal information. It uses two consecutive cameras to match tracklets according to the entry and exit of the road. However, ICA also faces some challenges need to be tackled. For example, there are several vehicles with similar appearance in the matching pool of tracklet candidates, which makes the model prone to match with those noisy samples. Besides, due to the different location of cameras, some objective factors like illumination, perspective also make this



(a) zones in camera c042 (b) zones in camera c043

Figure 6. Examples of predefined zones to describe trajectories. According to the traffic rules, each trajectory must be valid. For the two cameras, valid trajectory is [(1, 4), (1, 5), (1, 6), (3, 2), (3, 5), (3, 6), (5, 4), (5, 2), (5, 6), (6, 2), (6, 4)]. For those vehicles from c042 to c043, they must drive out of zone 4 in c042 and must drive in zone 1 in c043. Only a small number of trajectories will be retained with this strict constraint. For c042 to c043, Only trajectories through [(1, 4), (5, 4), (6, 4)] in c042 and [(1, 5), (1, 4), (1, 6)] in c043 will be filtered out as possible matching candidates.

part more difficult. To tackle these challenges, as shown in Figure 7, different to previous tracklet-grained matching strategy, we propose a novel box-grained matching module to find the same identities in box-level in a successive and sequential way, and the solution of details are as follows.

3.4.1 Zone-based Tracklet Candidates Filter

Since vehicles must obey the traffic rules and can only pass along special routes due to road topology structures, we use the same strategy as [52] to filter almost all vehicles that are impossible to cross other cameras. First, we predefine zones for every enter/exit area of all camera. Figure 6 shows an example of camera C042 and C043. Zone 1, 3, 5, 6 are termed as "in zone" in camera C043, which allow a car to enter this camera. Zone 2, 4, 5, 6 are termed as "out zone" in camera C042, which permit a car to leave this camera. Once zones are allocated, we assign every tracklet as a certain trajectory with an "in zone"-"out zone" and corresponding start-end time,

$$Traj_i = \{[z_{in}, z_{out}], [t_{in}, t_{out}]\} \tag{5}$$

where $[t_{in}, t_{out}]$ is the start-end time. $[z_{in}, z_{out}]$ is the "in zone" id and "out zone" id of trajectory i, respectively. For z_{in}, the id should be assigned when a vehicle is just entering the camera and its center point first touches corresponding zone. For z_{out}, the id should be determined with the zone where the last frame of a trajectory last appears.

After all trajectories are generated, the raw tracklets may contain false positives, such as the first batch of "in zone" tracklets and the last batch of "out zone" tracklets. In addition, the tracklets only pass through sub-paths without entering the main road are impossible to find their other parts from other cameras. So we roughly filter out tracklet candidates with traffic rules, road structures, and traveling time. Specifically, take an connected road (zone 4 of C042 as "out zone" and zone 1 of C043 as "in zone") into account. On the one hand, we drop tracklets with zone id not equal to 4 and tracklets with zone id equal to 4 that show up late in camera C042. On the other hand, we drop tracklets with zone id not equal to 1 and tracklets with zone id equal to 1 that show up early in camera C043, which can be formulated as follows,

$$Traj_{out} = [Traj^i_{out}(t_{out}) < T_{out} \ \& \ Traj^i_{out}(z_{out}) = 4] \tag{6}$$

$$Traj_{in} = [Traj^i_{in}(t_{in}) > T_{in} \ \& \ Traj^i_{in}(z_{in}) = 1] \tag{7}$$

Where T_{out} and T_{in} are the thresholds of frame id for "out zone" and "in zone". After the filter, the search space is significantly reduced.

3.4.2 Box-grained Distance matrix Construction and Optimization

Once $Traj_{out}$ set and $Traj_{in}$ set are obtained, previous methods [25, 52] calculate the tracklet-grained distance matrix for the final matching. This way only can get limited performance due to some noisy appearance features within tracklets may dominate their representations. To solve this problem, we calculate the box-grained distance matrix instead. Take two connected zones (e.g. zone 4 of C042 as "out zone" and zone 1 of C043 as "in zone") into account, before starting to match, we need to calculate the distance between each box for the two zones. "Out zone" and "In zone" can be termed as $Z_{out} = [T_1, ...T_n]$ and $Z_{in} = [\bar{T}_1, ..\bar{T}_m]$, where $T_i = [B^1_i, ..., B^{h_i}_i]$ and $\bar{T}_j = [\bar{B}^1_j, ..., \bar{B}^{h_j}_j]$ are the tracklets of "Out zone" and "In zone"', respectively. B^h_i is the hth box feature of tracklet i. From this we can get the similarity matrix S between the two zones:

$$S = \begin{bmatrix} cos(B^1_1, \bar{B}^1_1) & \dots & cos(B^1_1, \bar{B}^{h_m}_m) \\ \vdots & \ddots & \vdots \\ cos(B^{h_n}_n, \bar{B}^1_1) & \dots & cos(B^{h_n}_n, \bar{B}^{h_m}_m) \end{bmatrix}_{\sum_{i=1}^{n} h_i \times \sum_{j=1}^{m} h_j} \tag{8}$$

where cos represents cosine distance, B^h_i is the hth ($h \in [0, h_i]$) box feature of tracklet i of "out zone", $\bar{B}^h_j$ is the hth ($h \in [0, h_j]$) box feature of tracklet j of "in zone". The similarity matrix still needs to be optimized because of severe occlusion, illumination or different view perspective. To make the similarity matrix more convincing, we focus on adjusting the weights among boxes in three steps. First, we introduce the reranking method [59] to reconstruct the similarity matrix S of distance matrix D. Second, we refine

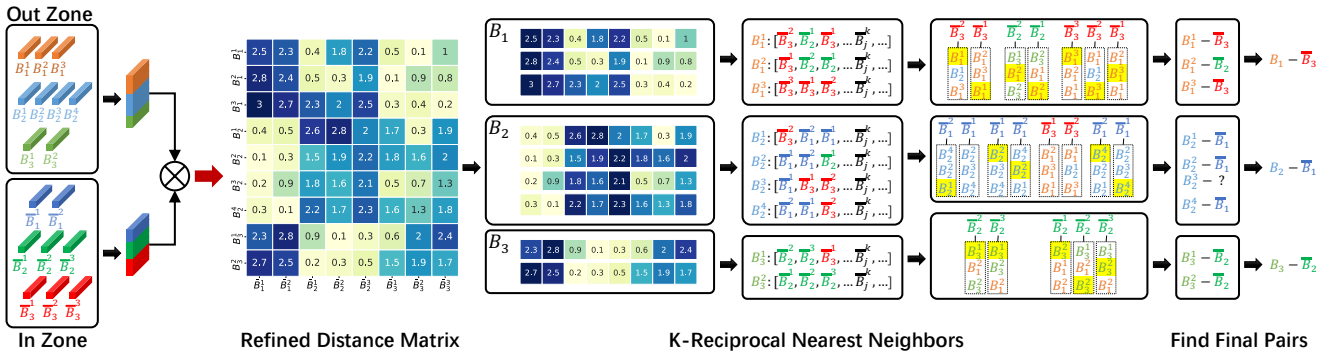


Figure 7. Matching process. Once the tracklet candidates between two connected zones are provided, we calculate and refine the distance matrix first ($*$ The dark read bold arrow indicates the refine operations). Then we treat every box B_i^h in "out zone" as a probe to find their associated tracklet by the principle of k-reciprocal nearest neighbors. The final pairs of tracklets are generated by simple count calculation.

it with interval prior,

$$D_{i,j} = \begin{cases} e^{\frac{\alpha_t \times (t_{low} - t_{i,j})}{\beta_t}} \times D_{i,j}, & t_{i,j} < t_{low} \\ e^{\frac{\alpha_t \times (t_{i,j} - t_{upp})}{\beta_t}} \times D_{i,j}, & t_{i,j} > t_{upp} \end{cases} \quad (9)$$

where $D_{i,j}$ is the distance between i and j in the distance matrix, α_t and β_t are the hyper parameters, t_{low} and t_{upp} are the lower threshold and upper threshold of traveling time window, respectively. Finally, we refine the distance matrix with occlusion rate to generate a convincing distance matrix D for final matching,

$$D_{i,j} = \begin{cases} e^{\alpha_o \times (1 + r_o)} \times D_{i,j}, & r_o > r_{thre} \\ D_{i,j} \end{cases} \quad (10)$$

Where α_o is the hyper parameter, r_o and r_{thre} are the occlusion rate for box i and occlusion threshold, respectively.

3.4.3 Tracklet Association with k-reciprocal Nearest Neighbors

For associating tracklets between two connected zones with the distance matrix D, we propose a novel and effective matching strategy to find all the convincing pairs. Inspired by [59], all tracklets are associated with the principle of k-reciprocal nearest neighbors. First, we define $N(B_i^h, k)$ as the k nearest neighbors of a probe B_i^h,

$$N(B_i^h, k) = (\bar{B}_1, \bar{B}_2, ..., \bar{B}_k), |N(B_i^h, k)| = k \quad (11)$$

Where $|\cdot|$ is the number of top-k candidates. According to [59], the k-reciprocal nearest neighbors are more related to probe B_i^h than k-nearest neighbors. Then we find a matched tracklet for every probe B_i^h of "out zone" by counting the most frequent k-reciprocal nearest frames of a tracklet among all tracklet boxes from "in zone",

$$M(B_i^h, \bar{T}_j) = MaxCount\{N(B_i^h, B_j) \wedge N(\bar{B}_j, B_i^h)\} \quad (12)$$

Where $\bar{B}_j$ is the box in tracklet $\bar{T}_j$ of "in zone". Once every box in "out zone" is assigned one matched tracklet from "in zone", two tracklets are the same vehicle if the in-tracklets with the most matching times in every out-tracklets.

3.4.4 Post-processing after matching

Following the tracklet association process, we first check the validity of all matched pairs, it it a invalid pair, if the time of 'out zone' is behind of 'in zone', and then we remove these invalid pairs. Secondly, we assign a global id if the two pairs share the same tracklet, meanwhile, if two matched pairs have the common trajectory, the two global id would be merged to a unique one, we process all cameras pairs and output the final matched result for submission.

4. Experiments

4.1. Datasets

The CityFlowV2 [1] dataset are collected from 46 cameras spanning 16 intersections in a mid-sized U.S. city. The dataset covers a diverse set of location types, including intersections, stretches of roadways, and highways. For city-scale multi-camera vehicle tracking track, it is divided into 6 scenarios. 3 scenarios are used for training, 2 scenarios are for validation, and the remaining scenario is for testing.

In order to improve the performance of ReID features, we use not only real data, *i.e.*, train and validation data from Track1 of AICity Challenge 2022, but also synthetic data for re-ID model training. Synthetic data is generated by VehicleX [41], which is a publicly aviablable 3D engine. There are 2028 vehicles (666 real vehicles and 1362 synthetic vehicles) and 229345 images (27195 real images and 192150 synthetic images) used for ReID training.

[1] https://www.aicitychallenge.org/2022-data-and-evaluation/

4.2. Implementation Details

Re-identification. We mainly use PaddlePaddle framework to train our models, and the model is trained using SGD with momentum 0.9. During training, the cos-decay learning rate scheduler is adopted. Moreover, several data augmentations such as random crop, random flip and auto augment are used as in training.

Single-Camera Multi-Target Tracking. For filtering the detection results, the high score threshold is 0.6 and low score threshold is 0.1. For data association, the similarity distance threshold is 0.45, the momentum term α in EMA is 0.9 and the weight factor for appearance feature distance λ is 0.98.

4.3. Metrics of Evaluation

For MCMT tracking, the IDF1 score [36] is used to rank the performance in the final leader board. IDF1 measures the ratio of correctly identified detections over the average number of ground-truth and computed detections. The evaluation tool provided by the challenge, which are adopted by the MOTChallenge [1, 22], which are computed in the evaluation system, are IDF1, IDP, IDR, Precision and Recall.

4.4. Experiments Results

ReID ensemble. In order to evaluate the performance of ReID features, we split a query set and a gallery set from train and validation data of Track1 of AICity Challenge 2022, and use the mean Average Precision (mAP) of the top-K (K=100) matches as the metric. As shown in the Table 1, HRNet48 is the best individual model to extract ReID features, and ensembled ReID features obtained by concatenating the features of five models achieve the best results. Therefore, ensembled ReID features are employed to SCMT module for appearance matching.

Feature	mAP
HRNet48	48.76
ResNeXt101	47.15
ResNet50	46.69
ConvNeXt-tiny	45.77
Res2Net200	41.44
Ensembled	**49.23**

Table 1. Comparison of different ReID features.

Inter-Camera Association. Table 2 shows the ablation of different proposed strategies. Compared with the baseline of tracklet-grained matching, we can get 2.14% IDF1 gain with box-grained k-reciprocal nearest neighbors, which demonstrate the effectiveness of proposed matching method. After that, IDF1 achieves 81.58% when we replace the distance matrix with reranking method. We introduce post-process to throw away invalid pairs that have impossible traveling time. This process further improves IDF1 with 1.52%. Besides, refining the distance matrix with interval prior and occlusion rate makes our model achieve the SOTA results, which obains 84.86% in final leaderboard.

Method	Dist	Re	Ass	PP	IDF1	IDP	IDR
tracklet	L2		H		78.26	80.61	76.05
box	L2		R		80.40	82.66	78.25
box	Rr		R		81.58	84.52	78.84
box	Rr		R	✓	83.10	87.96	78.75
box	Rr	✓	R	✓	**84.86**	**91.37**	**79.21**

Table 2. Comparison of strategies in matching stage. "Dist" indicates how to build distance matrix, which has two options: L2 and Rr (Reranking). "Re" indicates whether refine the distance matrix with interval prior and occlusion rate. "Ass" indicates association methods, which contains H (Hungarian) and R (k-Reciprocal). "PP" is the post-process after matching.

Rank	Team ID	Team Name	IDF1
1	**28**	**matcher (ours)**	**84.86**
2	59	BOE	84.37
3	37	TAG	83.71
4	50	FraunhoferIOSB	83.48
5	70	appolo	82.51
6	36	Li-Chen-Yi	82.18
7	10	Terminus-AI	81.71
8	118	FourBeauties	81.66
9	110	Orange Peel	81.40
10	94	SKKU Automation Lab	81.29

Table 3. Leaderboard of Track1 in the AI City Challenge 2022.

Comparison with other teams. The proposed system is submitted to the Track1 of AICity Challenge 2022 for evaluation. As shown in Table 3, our system scores 84.86% IDF1 and ranks first place among over 20+ teams from all over the world.

5. Conclusion

In this paper, we propose an effective Multi-Camera Multi-Target Tracking framework. The proposed framework obtains MCMT results by performing vehicle detection, re-identification, single-camera multi-target tracking and inter-camera association. Experiments results on the public test set of 2022 AI CITY CHALLENGE Track1 demonstrate the effectiveness of our method, which achieves IDF1 of 84.86%, ranking first on the leaderboard.

References

[1] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 8

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 3, 4

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2, 4

[4] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*, 2022. 3

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[7] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. A novel solution for multi-camera object tracking. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2329–2333. IEEE, 2014. 3

[8] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. An equalized global graph model-based approach for multi-camera object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2367–2381, 2016. 3

[9] Wei-Ting Chen, I-Hsiang Chen, Chih-Yuan Yeh, Hao-Hsiang Yang, Jian-Jiun Ding, and Sy-Yen Kuo. Sjdl-vehicle: Semi-supervised joint defogging learning for foggy vehicle re-identification. 2022. 2

[10] Yunhao Du, Yang Song, Bo Yang, and Yanyun Zhao. Strongsort: Make deepsort great again. *arXiv preprint arXiv:2202.13514*, 2022. 3, 4, 5

[11] Yunhao Du, Junfeng Wan, Yanyun Zhao, Binyu Zhang, Zhihang Tong, and Junhao Dong. Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2809–2819, 2021. 5

[12] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019. 4

[13] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3

[14] 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, 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[16] Zhiqun He, Yu Lei, Shuai Bai, and Wei Wu. Multi-camera vehicle tracking with powerful visual features and spatial-temporal cue. In *CVPR Workshops*, pages 203–212, 2019. 1

[17] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 2

[18] Yunzhong Hou, Heming Du, and Liang Zheng. A locality aware city-scale multi-camera vehicle tracking system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 167–174, 2019. 1

[19] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *CVPR Workshops*, pages 416–424, 2019. 1, 3

[20] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4

[21] Young-Gun Lee, Jenq-Neng Hwang, and Zhijun Fang. Combined estimation of camera link models for human tracking across nonoverlapping cameras. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2254–2258. IEEE, 2015. 1

[22] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *2009 IEEE conference on computer vision and pattern recognition*, pages 2953–2960. IEEE, 2009. 8

[23] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *arXiv preprint arXiv:2010.12138*, 2020. 3

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[25] Chong Liu, Yuqi Zhang, Hao Luo, Jiasheng Tang, Weihua Chen, Xianzhe Xu, Fan Wang, Hao Li, and Yi-Dong Shen. City-scale multi-camera vehicle tracking guided by crossroad zones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4129–4137, 2021. 3, 6

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021. 2, 3

[28] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 4

[29] Zhongji Liu, Wei Zhang, Xu Gao, Hao Meng, Xiao Tan, Xiaoxing Zhu, Zhan Xue, Xiaoqing Ye, Hongwu Zhang, Shilei Wen, et al. Robust movement-specific vehicle counting at crowded intersections. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2617–2625. IEEE, 2020. 5

[30] Jincheng Lu, Meng Xia, Xu Gao, Xipeng Yang, Tianran Tao, Hao Meng, Wei Zhang, Xiao Tan, Yifeng Shi, Guanbin Li, et al. Robust and online vehicle counting at crowded intersections. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3997–4003. IEEE, 2021. 5

[31] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936. 5

[32] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 164–173, 2021. 3

[33] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 1

[34] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2, 3

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2, 3

[36] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016. 8

[37] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. Siammot: Siamese multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12372–12382, 2021. 3

[38] Jakub Španhel, Vojtech Bartl, Roman Juránek, and Adam Herout. Vehicle re-identification and multi-camera tracking in challenging city-scale environment. In *Proc. CVPR Workshops*, volume 2, 2019. 1

[39] Daniel Stadler and Jurgen Beyerer. Improving multiple pedestrian tracking by track management and occlusion handling. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10953–10962. IEEE Computer Society, 2021. 5

[40] Xiao Tan, Zhigang Wang, Minyue Jiang, Xipeng Yang, Jian Wang, Yuan Gao, Xiangbo Su, Xiaoqing Ye, Yuchen Yuan, Dongliang He, et al. Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features. In *CVPR Workshops*, pages 275–284, 2019. 1

[41] Zheng Tang, Milind Naphade, Stan Birchfield, Jonathan Tremblay, William Hodge, Ratnesh Kumar, Shuo Wang, and Xiaodong Yang. Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 211–220, 2019. 7

[42] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. 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*, pages 8797–8806, 2019. 1

[43] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 108–115, 2018. 1, 3

[44] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *European conference on computer vision*, pages 135–153. Springer, 2016. 2

[45] Cheng Wang, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 365–381, 2018. 2

[46] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 4

[47] Yicheng Wang, Zhenzhong Chen, Feng Wu, and Gang Wang. Person re-identification with cascaded pairwise convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1470–1478, 2018. 2

[48] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, pages 107–122. Springer, 2020. 3, 4

[49] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 3, 4

[50] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *arXiv preprint arXiv:2203.05482*, 2022. 4

[51] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[52] Jin Ye, Xipeng Yang, Shuai Kang, Yue He, Weiming Zhang, Leping Huang, Minyue Jiang, Wei Zhang, Yifeng Shi, Meng Xia, 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*, pages 4044–4053, 2021. 3, 6

[53] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[54] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *Computer Vision – ECCV 2016 Workshops*, pages 36–42, Cham, 2016. Springer International Publishing. 3

[55] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*, 2021. 3, 4

[56] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021. 3

[57] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 2

[58] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, and Tao Mei. Vehiclenet: Learning robust visual representation for vehicle re-identification. *IEEE Transactions on Multimedia*, 23:2683–2693, 2020. 2

[59] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1318–1327, 2017. 6, 7

[60] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020. 3

[61] Yi Zhou and Ling Shao. Aware attentive multi-view inference for vehicle re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6489–6498, 2018. 2

[62] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2