Multi-Camera Tracklet Matching using Group-IOU

Yun-Lun Li [1] Zhi-Yi Chin [2] Ming-Ching Chang [3] Chen-Kuo Chiang [1]

[1] National Chung Cheng University, Taiwan
[2] National Yang Ming Chiao Tung University
[3] University at Albany – SUNY, NY, USA

Abstract

Multi-camera vehicle tracking at the city scale is an essential task in traffic management for smart cities. The large-scale analytics is a challenge due to large data variabilities, frequent occlusions, and appearance differences caused by large viewing angle variations, etc.*. In this work, we develop an efficient multi-camera vehicle tracking system consisting of four modules: (1) Faster-RCNN vehicle detection, (2) vehicle association and re-identification feature map generation, (3) single-camera vehicle tracking to form basic tracklets, (4) multi-camera vehicle tracklet matching and re-identification that creates longer, consistent tracklets across the city scale. Our main efforts are on the tracklet creation, association, and linking in the single-camera tracking and multi-camera tracking. We propose three single-camera tracking (SCT) filters that can effectively eliminate unreliable tracklets. For multi-camera tracking, we propose a novel Group-IOU metric to evaluate the connectivity of tracklets across views. Our system obtains IDF1 score 0.1343 and are ranked 18-th on the AICity 2021 Challenge Track 3 public leaderboard.*

1. Introduction

Although the expansion of city-scale increases city management's difficulty, the development of computer vision and monitoring networks all over the city provides a new choice for city management. Multi-camera vehicle tracking is one of the crucial tasks in traffic management. Its purpose is to achieve better traffic design and traffic flow optimization by tracking many vehicles in a network of multiple surveillance cameras. As Figure 1 shows, we need to connect the same tracklets in every camera.

In recent years, vehicle tracking has become a hot frontier field, especially single-camera tracking. In computer vision research, recent years have witnessed many successful works after releasing some public data sets and challenges in this field. Compared with single-camera tracking, multi-camera vehicle tracking is a much more complicated task, including detection, re-identification (ReID), tracking, and camera synchronization. Although the former methods have made remarkable achievements in various challenges, it also has some shortcomings. First of all, feature aggregation and large models consume many computing resources, such as GPUS. Besides, most of the previous methods need large-scale annotated data sets to train their models. Finally, it is sometimes challenging to collect the required data sets. In addition to the above, there are several challenges in this field.

1. The vehicle's appearance varies depending on its angle and distance of the camera and the brightness of the light. The above reason brings challenges to the accuracy of the vehicle re-identification.

2. Because of the two-level synchronization, it is very time-consuming to synchronize the tracking id. It is required to match the tracklets in the single-camera view in the first level and later match tracklets from different camera views in the second level to complete multi-camera tracking.

Previous works such as [19, 11] engage in generating a discriminative feature of vehicles and improving the performance on single-camera tracking. But in Multi-camera tracking, they simply use Euclidean distance or cosine similarity to measure across-camera tracklets. Then, they associate the tracklet to the tracklet with largest cosine similarity or smallest Euclidean distance. That means that the strategy only consider the tracklet with top-1 score. In order to making good use of the ranking information, we propose Group-IOU to evaluate the connectivity between across-camera tracklets. Moreover, a matching strategy is proposed to grouping tracklets with Group-IOU metric.

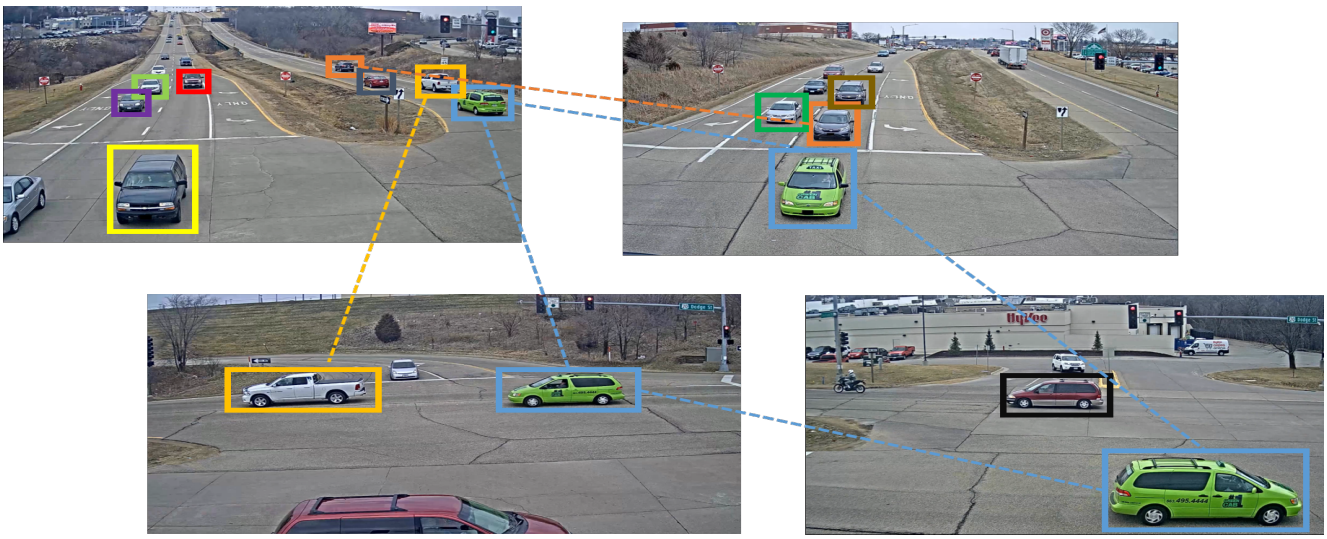

Figure 1. AI City Challenge 2021 – Track 3 Challenge on multi-target multi-camera vehicle detection and tracking.

2. Related Work

2.1. AI City Challenge 2020

[20] proposes an effective multi-camera vehicle tracking system that is accurate and easy to train. In the detection and tracking part, Qian *et al*. use the weighted inter-class non-maximum suppression algorithm to generate more accurate boundaries. In ReID, the aggregate loss is used for training to overcome the appearance differences caused by different perspectives. Finally, given the tracklets and distance matrix, the method uses the fast multi-target cross-camera tracking strategy to generate the result.

[13] proposed a Spatio-temporal consistent hierarchical matching method for tracking vehicles across cameras. It represents the target by combining spatial and temporal features and compares the targets in different cameras using a bottom-up hierarchical matching strategy.

2.2. Vehicle Re-identification

Object re-identification is the task of matching and searching for targets in different scenes. Many papers have focused on person-ReID, such as [6] propose an efficient, end-to-end, fully convolutional Siamese network that computes the similarities at multiple levels to train person ReID. With the increasing attention paid to urban management and intelligent transportation, researchers have paid more attention to vehicle re-identification. To improve accuracy, some methods [14] will provide more information than appearance features, such as license plates and car models, are adopted. However, due to the competition data's limitations, this research can not rely on these attributes. Therefore, this research's model can only rely on the id of the track and the camera's id.

[9] is based on a strong baseline with a bag of tricks (BoT-BS) proposed in person ReID. First, extract the features from the model, and obtain the preliminary results after sorting according to the features. Finally, these preliminary ranking results are post-processed using the weighted feature tracklet-level reranking strategy to obtain the final ReID results.

2.3. Vehicle Detection

Object detector is usually based on the proposal. R-CNN [5] is the first one to use CNN for detection. This method uses selective search to generate proposals. Fast R-CNN [4] introduced sharing the feature map across the entire network. Later Faster R-CNN [24] introduces the concept of Regional Proposal Network (RPN), which can improve the quality of the proposal, which also greatly improves the quality of the detection. Introducing two-stage detection in the front, next to introduce one-stage detection, which can predict the bounding box without generating a proposal, so the speed is much faster than the two-stage method. YOLO [21] and Yolov2 [22] consider target detection as a regression problem. They divide the original images into several network units and predict each network unit's bounding box and the relevant classification probability. SSD [15] predicts different proportions of bounding boxes in different layers, thereby improving the performance of the one-stage detector in complex scenes. All in all, the two-stage (proposal-based) detector focuses on high accuracy, while the one-stage detector focuses on the speed of detection.

2.4. Single-Camera Tracking (SCT)

Most of the recent multi-camera tracking (MOT) methods based on tracking by detection schemes [3], that is,

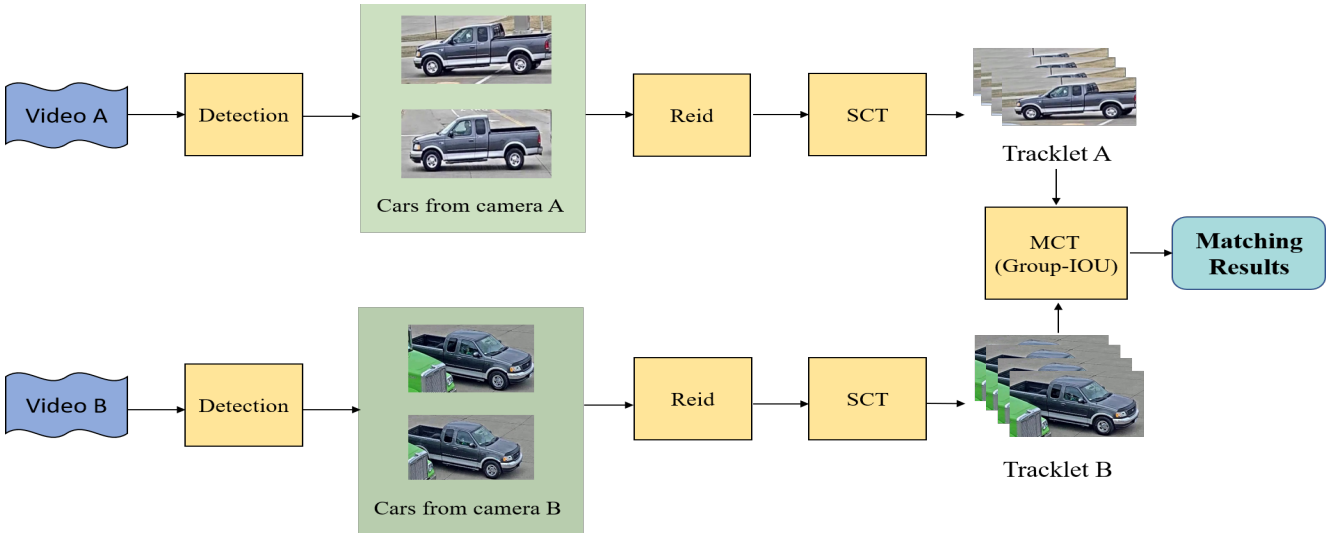

Figure 2. The proposed multi-camera vehicle tracking pipeline. SCT: Single-Camera Tracking. MCT: Multi-Camera Tracking.

given the detection results, we hope to correlate the results across frames and hope to be able to estimate the position of the object even when the detection result is unreliable, or occlusion occurs. [1, 2, 10] follows the tracking by detection schemes. Many tracking methods are based on graph models [25] and solve the problem by minimizing the total cost. In [25], Tang *et al*. regards the detected objects as point (vertices) in the graph model, and in [27], tracklets are the points of the graph. For detection-based models, there are two main disadvantages. First, one of the fundamental assumptions is the independence of each point. However, detection is not conditionally independent from frame to frame. Therefore, if we want to track an object for a longer time, the time information must be used more effectively. Second, the detection-based graph usually has a dimensional affinity matrix, making it challenging to find the global minimum solution. However, the tracklet-based graph model can better use the short trajectory information to measure the relationship between points, but this is under the premise of the careful handling of false associations in the tracklet generation step.

2.5. Multi-Camera Tracking

With the development of smart cities, road camera sensors have greatly improved multi-target multi-camera tracking research. [17] proposed a pipeline for multi-target visual tracking in a multi-camera system. The pipeline also extracts the similarity of appearance and dynamic motion. In order to track the loss of the tracking target, a General Multi-view Tracking (GMT) framework focusing on cross-camera trajectory prediction is proposed in [29]. To reduce the search and matching space of multi-camera tracking (MCT), [12, 26, 27] also considered camera link models with Spatio-temporal constraints. For example, in the estimation process of unsupervised schemes in [12], camera link models use two-way transition time distribution. In [26, 27], Tang *et al*. uses the estimated vehicle speed to establish the transition time distribution for each connected camera pair. Using reliable camera link models, the candidate set for matching will become smaller, significantly improving cross-camera association accuracy.

3. Method

The proposed multi-camera vehicle detection and tracking pipeline consists of four main modules as shown in Figure 2: (1) vehicle detection, (2) vehicle (re-id) feature extraction, (3) single camera tracking and tracklet filtering, and (4) multi-camera tracking with a newly proposed **Group-IOU** metric.

Vehicles are first detected using Mask-RCNN. Re-id appearance features are extracted using ResNet101-ibn-a. The detected vehicles are associated into individual tracklets using TrackletNet Tracker (TNT) [28] provided by the AICity 2021 Challenge organization. We perform SCT tracklet filtering to remove unreliable tracklet predictions as post processing. For multi-camera tracking, We propose a new Group-IOU metric to evaluate the similarity of each tracklets that leads to the grouping and linking of tracklets across camera views. We next describe detailed steps.

3.1. Vehicle Re-identification

We extract vehicle re-identification features using ResNet-101 together with the IBN-Net-a. The Instance-Batch Normalization Network (IBN-Net) [18] is a winning method from the Drivable Area Segmentation (find the road area that the vehicle is driving or it can potentially drive

on) in 2018 WAD Challenge. Unlike ResNet as an independent network, IBN-Net can be combined with other deep learning models to improve performance without increasing the computational cost. Here we combine IBN-Net with ResNet101 for re-id feature extraction.

The starting point of IBN-Net is to improve the model's to changes in the appearance of images, so it combines two normalization layers (instance normalization and batch normalization) to improve on various tasks. To reduce the feature changes introduced by superficial appearance without affecting the recognition of more profound content, this method only adds instance normalization to the superficial level. Moreover, to preserve the superficial level's image content information, half of the batch normalization in the superficial level is replaced by instance normalization instead of all of it.

IBN-Net has instance and batch normalization in shallow layers and only batch normalization is adopted in top layers to get better features. The difference between instance normalization and batch normalization is that instance normalization uses statistics of each sample to localize features; the features it learns do not affect appearance changes such as color, style, and virtuality/reality. Batch normalization uses mini-batch with statistic mean and variance, and training normalizes each channel feature. If we want to retain content and information, batch normalization is needed. At the same time, batch normalization can speed up training and learn more distinguishing features. In summary, IBN-Net can mainly learn the style of the vehicles and learn the content in the top layers.

3.2. Vehicle Detection

The AICITY contest provides the detection results using Mask-RCNN[7], SSD512[16], YOLOv3[23]. The Mask-RCNN consists of Faster R-CNN[24] and the mask module. This research only uses Faster R-CNN part for vehicle detection. Using convolution feature maps of a region-based detector (such as Fast R-CNN) to generate region proposals is what Ren *et al*. found out. So on top of these features, by adding some convolutional layers to build a regional proposal network (RPN), simultaneously output region bounds and objectness score for each location. Therefore, RPN is a full convolution network (FCN), which trains end-to-end to generate high-quality region proposals and then sent to Fast R-CNN for detection. The input of RPN is a road camera frame, and the output is a set of rectangular proposals (vehicles), each including a target score. This method slides a small network on the last shared convolution feature map to generate region proposals. This network is fully connected to an $n \times n$ $(n = 3)$ spatial window of the input feature map. It maps each sliding window to a low-dimensional vector, which is input to two fully connected layers, a regression layer, and a classification layer.

3.3. Single-camera Tracking (SCT)

We adopt the TrackletNet Tracker (TNT) [28], a graph-based tracklet model for SCT. TNT consists of three components: (1) trajectory generation, (2) connectivity measurement, and (3) graph-based clustering.

Given each frame's detection result, based on the camera motion and the appearance similarity between two consecutive frames, tracklets are generated through the IoU (intersection-over-union) with epipolar geometry constraint compensation. Each generated tracklet is regarded as a node in the graph. Between every two tracklets, the edges' weight in the graph model measures the connectivity, where the connectivity represents the possibility of tracklets coming from the same car. The TrackletNet architecture describes as following. First, for each track-lets, enter its 4D position information and 2048D appearance in-formation, and spread it out in a 64D time dimension. To better characterize the duration of the two tracklets, we are adding two binary masks to the entry and exit channels, one for tracklet-1 and the other for tracklet-2. On condition that any frame in 64D time dimension, the entire column is set to 1, and if it does not exist, it is set to 0. Connect all feature channels to 3 Conv/MaxPool layers, and use four types of 1D filers to down-sample all features in the time dimension to calculate the features' continuity. Followed by average pooling, calculate the average value of all appearance features in all the time dimensions; that is, each channel has only five-dimensional features left in each dimension of the time do-main. Finally, after concatenating all the features, two fully connected layers are used to output the similarity score. The network helps us associate the same tracklet in a camera. Then we use the method of [27] to perform clustering to minimize the total cost in the graph. After clustering, tracklets with the same ID will combine into a group.

3.4. Post SCT Tracklet Filtering

Post processing is an important in MCT task. Because the wrong tracklets may make the association become harder, we need to remove the amount of across-camera comparisons. With visualization results of SCT, we found some weird detection such as parked cars, boxes with no vehicles and so on. Therefore, we engage in the post processing to eliminate these low quality detection.

For post processing, we first connect the lost tracklets using appearance re-id features. By observing the single-camera tracking result, we notice the following common problems:

1. Tracklets with vehicle detection in a very high speed
2. Tracklets with vehicle detection stay in a very short/long time, but in a normal speed
3. Parked cars

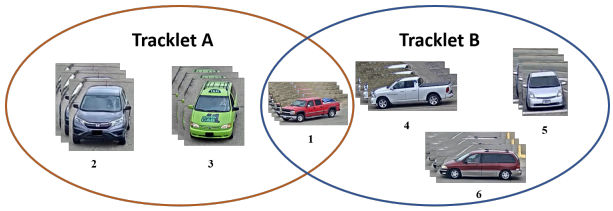

Figure 3. An example of Group-IOU

To address these problems, we propose the foolowing three types of tracklet filtering, that can effectively remove abnormal tracklets.

SCT Filter 1. The first filter is defined by Eq.1

$$\frac{speed - \mu_{speed}}{\sigma_{speed}} > thres_{speed} \tag{1}$$

The speed is calculated by the GPS positions provided by AICITY contest. With this equation, we can easily remove tracklets with abnormal speed. μ_{speed} is the average speed of tracklets in one camera and σ_{speed} is the standard deviation of tracklets. To avoid deleting too many tracklets, $thres_{speed}$ will increase until the ratio of removed tracklets to total amount of tracklets is not over 0.03.

SCT Filter 2. The second filter is defined by Eq.2

$$abs(\frac{staytime - \mu_{staytime}}{\sigma_{staytime}}) > thres_{staytime} \tag{2}$$

The equation is similar to Eq.1. The purpose of this filter is to remove the tracklets which stay too long in the camera. We found that there are some wrong tracklets with normal speed, but they stay too much time or too less time in the camera. Therefore, we applied this filter to remove these tracklets.

SCT Filter 3. The third filter aims to remove no moving tracklets. It defined by Eq.3

$$\frac{box_{first} \cap box_{last}}{box_{first} \cup box_{last}} > thres_{iou} \tag{3}$$

box_{first} and box_{last} represent the first detection box and last detection box in the tracklets. If the IOU of the first detection box and the last detection box is bigger than threshold, we regard this tracklet as no moving tracklet such as parked cars. With these post processing techniques, we can remove the redundant tracklets as more as possible. Also, it avoid pairing invalid tracklets in the Multi-target Multi-camera Tracking stage.

3.5. Multi-target Multi-camera Tracking (MTMC)

The proposed MTMC matching is performed according to the following steps.

First, we measure the GPS position of vehicle detection box by using the provided calibration matrix. Then, we can

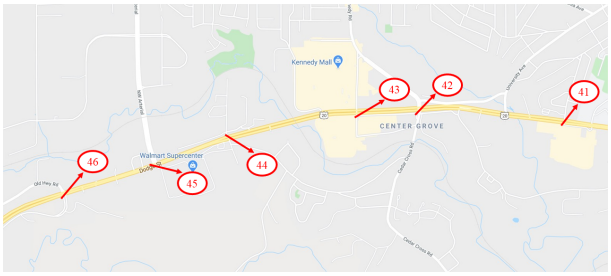

Figure 4. The locations of cameras in test set

obtain the direction of tracklets and use the cosine angle to know if the two tracklets drive on the same direction or not. If the direction of the two tracklets are different, which means the cosine angle is smaller than 0.5, we consider that it's a impossible matching and the similarity would be set to zero. Because the cameras are set on an arterial road(As shown in Figure 4), the direction filter has significant impact. The car would disappear in the road if it makes a turn. Therefore, we set the threshold to 0.5 to remove the tracklets which makes a turn or goes to the opposite direction. Finally, We calculate the cosine similarity between tracklets

We next rank the similarities and select the tracklets that meet the condition below:

$$\frac{sim - \mu_{sim}}{\sigma_{sim}} >= 1 \tag{4}$$

If the ratio of chosen tracklets to the total amount of possible tracklets is bigger than 15%, we consider that the tracklet does not matching any tracklet. That means $sigma_{sim}$ is too small so that the similarity between query tracklet and other tracklets is close.

After the two steps before, we can obtain the possible matching tracklets of every tracklet. Because the tracklet can match only one tracklet in one camera, we only keep one matching tracklet in every camera.

We propose Group-IOU to evaluate the similarity between two tracklets. Group-IOU describes in Eq.5:

$$GroupIOU = \frac{size(MatchList_A \cap MatchList_B)}{min(size(MatchList_A), size(MatchList_B))} \tag{5}$$

As Figure 3 shows, matching list of tracklet A includes 1, 2, 3 and matching list of tracklet B includes 4, 5, 6. The Group-IOU score between tracklet A and tracklet B would be 0.33.

To avoid cycle pairing, we define a pairing group as a tree. The size of parent node should be larger than child node or equal to the size of child node. Figure 5 is an example of the grouping result. We compute Group-IOU between every tracklet and the tracklets whose size are larger or equal to the first one. For example, we want to associate

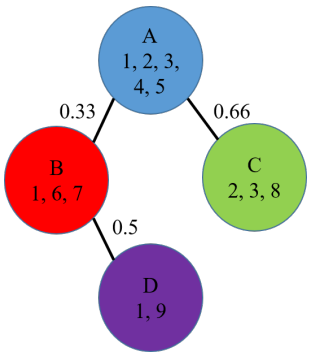

Figure 5. An example of grouping tracklets

tracklet B to others. We find that tracklet B has biggest Group-IOU with tracklet A and the Group-IOU score is higher than threshold. Then, There are two conditions:

1. If size of the matching list of tracklet A is larger than size of the matching list of tracklet B, tracklet B is set to be the child node of tracklet A.

2. If size of the matching list of tracklet A is equal to size of the matching list of tracklet B, we would check tracklet B isn't the parent of A and the grandparent of A and so on. If tracklet B is, tracklet B is set to be the parent of tracklet A. Otherwise, tracklet A is set to be the parent of tracklet B.

Finally, every tracklet set to the same id as its root parent.

With the two constraints in step 5, we can completely prevent from cycle pairing problem. Our proposed Group-IOU and pairing method consider not only top-1 matching tracklets, but all possible matching tracklets.

4. Experiments

Datasets. The organizer of the AI City Challenge provides the training data we used in this research. The data is the short section of road camera videos, in which there are 58 videos recorded by different cameras for training and verification, and the test data have 6 videos. The dataset totally contains 3.58 hours of videos collected from 46 cameras spanning 16 intersections in a mid-sized U.S. city. They also provides the training and verification data include the final results of detection, ReID, and tracking.

Evaluation Metrics. The F1 score of vehicle identity (IDF1) is used to evaluate the performance for multi-camera tracking task. The following equation shows how to calculate IDF1 score:

$$IDF1 = \frac{2TP_{id}}{2TP_{id} + FP_{id} + FN_{id}} \tag{6}$$

where TP_{id} represents true-positive, FP_{id} represents false-positive, FN_{id} represents false-negative. IDF1 measures the ratio of correctly identified detections over the average number of ground-truth and computed detections. Finally, we got IDF1 score of 0.1343. There are 20 submissions on public leaderboard in this track. We rank 18-th in public leaderboard of all the participant teams.

4.1. Vehicle Re-identification

IBN-Net-a changes to ResNet adds instance normalization to the superficial network. As for where to add specifically, we refer to [8]. [8] illustrates the necessity of the identity mapping path, so we believe adding instance normalization to the residual path is the right choice. To get the features aligned with the identity mapping path, instance normalization is added after the convolution in the residual module, rather than added after the last convolution, which can prevent the feature from being misaligned. According to the design principles mentioned in the method, the shallow layer should use batch normalization and instance normalization simultaneously, so half of the channels are calculated through batch normalization, and the other half of the channels are calculated through instance normalization. Because instance normalization will only be added to the superficial network, and ResNet is composed of 4 residual modules, the improvement of IBN-Net will only add instance normalization to the three blocks, which are Conv2_x, Conv3_x, Conv4_x, and Conv5_x will not be changed.

Our model's input is all of size 224×224 images, and the data augmentation methods include random horizontal flip, padding, and random erasing. During training, we use the aggregation of cross-entropy loss and triplet loss. The optimizer we use is SGD, and the initial learning rate is 0.01, the momentum is 0.9, and the weight decay is 0.0005. In addition to the optimizer, we also use WarmupMultiStepLR as our scheduler. In the first ten epochs, our learning rate will linearly increase from 0.001 to 0.01, which can solve initial training instability. In the 40th epoch and the 70th epoch, the learning rate reduces to 0.001 and 0.0001, respectively, which can help the model converge in the later training stages. We train a total of 100 epochs.

Figure 6 shows the example results. It shows that our reid model can generate a discriminative feature for cars, even if they are from different camera and different orientation.

Table 1 is an ablation study for different combination of backbone model. The combination of ResNet101 and IBN-Net-a performs the best on validation set. It gets 30.49% mAP score. The IBN-Net-a actually makes the feature better by the normalization design. It finally improves the performance of mAP score about 10%.

Figure 6. **Vehicle Re-ID results.** The query image in on the left hand side. The top 50 gallery images that match the query image are on the right hand side.

Model	mAP
res50 + center loss	13.09%
res101	14.34%
res101 + center loss	19.11%
res50 + ibn-a	27.2%
res50 + ibn-a + with no data augmentation	27.41%
res101 + ibn-a + with no data augmentation	29.09%
res101 + ibn-a	30.49%

Table 1. Re-identification mAP results on validation set with different backbone models

4.2. Single-camera Tracking (SCT)

The AICity Contest only provides SCT result for TrackletNet on testing set. Therefore, we want to confirm that it is the suitable algorithm.

We use a graph-based model to implement our SCT. In a graph, the tracklet is our vertex. Tracklets generation needs to use the similarity between IoU and appearance, and in order to reduce false negatives, we use an epipolar geometry algorithm to improve IoU. In simple terms, the epipolar algorithm assumes that the detected bounding box is stationary or moving slowly between adjacent frames, and the size remains the same. Next, for the edges of the graph, we will first connect the tracklets at adjacent time points without calculating the edges' weight. Then delete the edges of tracklets with overlapping time because the same object cannot appear in different places simultaneously. Next, the pair of the tracklets connected by the edges are brought to the multi-scale TrackletNet to calculate the similarity. At last, we pass the weighted graph into the clustering method proposed in [27] to cluster our tracklets. Finally, the same ID gives to tracklets clustered in the same subgraph.

Our setting for TrackletNet training is as follows: the initial learning rate is 0.001, we will reduce the learning rate by ten times every 2000 steps until the learning rate becomes 0.00001 and stop training.

Because the AICity contest does not provide TrackletNet results on validation set, we have to train our own model.We compare TrackletNet tracker with Hungarian algorithm and Tracklet Clustering on validation set. The result shows in Table 2. The performance of TNT increases 9% compare to performance of Hungarian algorithm, so we finally decide to use TNT as our SCT module.

Method	IDF1
Hungarian	60.22%
Tracklet Clustering	61.00%
TrackletsNet Tracker	69.31%

Table 2. SCT IDF1 results on validation set with Hungarian (traditional method) and TrackletNet Tracker (deep learning method).

4.3. Post SCT Tracklet Filtering

$thres_{staytime}$	$thres_{speed}$	$thres_{iou}$	IDF1
1	1	0.1	60.72%
1	2	0.1	60.70%
2	1	0.1	60.69%
2	2	0.1	60.65 %
1	1	0.05	61.00%
1	1	0.01	60.72%

Table 3. Results for using different threshold values.

Table 3 shows the IDF1 score for using different threshold values. We first keep the $thres_{iou}$ being 0.1. We notice that when both $thres_{staytime}$ and $thres_{speed}$ set to 1, we can obtain the best IDF1 score. However, there is only sightly difference between each result. Because we limit the ratio of removed tracklets to total amount of tracklets is not over 3%, the limit effectively avoids removing too many valid tracklets. Therefore, there is only little change for using different thresholds. Next, we fix $thres_{staytime}$ and $thres_{speed}$ and adjust $thres_{iou}$. Generally, the IOU of the first box and last box in the tracklet should be 0 if the tracklet is not a parked car. However, this constraint may remove the correct tracklets which are far from camera. Although the IDF1 has only slightly difference, we set $thres_{iou}$ to 0.05. It is the most proper value to keep correct tracklets and remove wrong tracklets as possible.

Table 4 shows the improvement that our filters bring. The baseline denotes SCT with only removing overlapping boxes which does not apply on the testing set because of the heavy traffic. Due to the heavy traffic, the process is not suitable. It would remove too many correct tracklets. With these three filter, we obtain almost 20% increment of IDF1 score. These three filters can deal with different problems

filters	IDF1
baseline	40.99%
speed	44.10%
speed + staytime	51.41%
speed + staytime + IOU	61%

Table 4. Results for using different filters

respectively. All of the three filters, the IOU filter has a hugest impact.

4.4. Multi-target Multi-camera Tracking (MTMC)

Due to a tracklet consists of many detections, we first calculate the standard deviation and mean of the tracklets' features. The two values can represent the distribution of tracklets' feature. Then, we can calculate with cosine similarity between tracklets. In Table 5, top-1 methods denotes that the tracklet would associate to the track with top-1 cosine similarity. The IDF1 score increases almost 2.5% with our Group-IOU method. Moreover, the amount of false positive decreases almost 50%. Because some tracklets is hard to evaluate the similarity with cosine similarity, top-1 method may associate tracklet to wrong group. Our Group-IOU method consider all the possible matching tracklets. It can tolerate the misleading information from cosine similarity to avoid wrong matching. Accordingly, it cause the increment of the IDF1 score.

Method	IDF1	IDTP	IDFP	IDFN
Top-1	32.65%	48718	64305	136709
Group-IOU	35.13%	47094	35610	138333

Table 5. MTMC IDF1 results on validation set with Group-IOU and Top-1.

5. Conclusion

In this work, we present Group-IOU to evaluate the similarity between tracklets. It improves the IDF1 score by considering all tracklets matching. Also, it decreases the amount of wrong matching and keep the amount of correct matching at the same time. We propose three filters including speed filter, stay time filter, IOU filter to conquer detection issues. Compare to baseline, we obtain almost 20% increment of IDF1 score. According to this research, we know that cosine similarity may cause some problems, and so does the Euclidean distance. Our future work is to find a better metric which can replace the cosine similarity and Euclidean distance.

References

[1] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking, 2017. 3

[2] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4310–4318, 2015. 3

[3] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):1012–1025, 2014. 2

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1440–1448, USA, 2015. IEEE Computer Society. 2

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 2

[6] Yiluan Guo and Ngai-Man Cheung. Efficient and deep person re-identification using multi-level similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 4

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[9] Shuting He, Hao Luo, Weihua Chen, Miao Zhang, Yuqi Zhang, Fan Wang, Hao Li, and Wei Jiang. Multi-domain learning and identity mining for vehicle re-identification. In *Proc. CVPR Workshops*, 2020. 2

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015. 3

[11] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[12] Young-Gun Lee, J. Hwang, and Zhijun Fang. Combined estimation of camera link models for human tracking across nonoverlapping cameras. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2254–2258, 2015. 3

[13] Peilun Li, Guozhen Li, Zhangxi Yan, Youzeng Li, Meiqi Lu, Pengfei Xu, Yang Gu, Bing Bai, and Yifei Zhang. Spatio-temporal consistency and hierarchical matching for multi-target multi-camera vehicle tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2

[14] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the

difference between similar vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. 2016. To appear. 2

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. 4

[17] Wenqian Liu, Octavia Camps, and Mario Sznaier. Multi-camera multi-object tracking, 2017. 3

[18] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 484–500, Cham, 2018. Springer International Publishing. 3

[19] Y. Qian, L. Yu, W. Liu, and A. G. Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2511–2519, 2020. 1

[20] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2

[21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[22] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017. 2

[23] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 4

[24] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *NIPS*, pages 91–99, 2015. 2, 4

[25] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[26] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David C. Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8797–8806. Computer Vision Foundation / IEEE, 2019. 3

[27] 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 (CVPR) Workshops*, June 2018. 3, 4, 7

[28] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 482–490, New York, NY, USA, 2019. Association for Computing Machinery. 3, 4

[29] Peng Wang and Qiang Ji. Robust face tracking via collaboration of generic and specific models. *IEEE Trans. Image Process.*, 17(7):1189–1199, 2008. 3