Multi-Camera Vehicle Tracking System Based on Spatial-Temporal Filtering

Pengfei Ren‡, Kang Lu‡, Yu Yang‡, Yun Yang‡, Guangze Sun, Wei Wang, Gang Wang, Junliang Cao, Zhifeng Zhao, Wei Liu
Nanjing Fiberhome Tiandi CO., LTD
Nanjing, Jiangsu, China
pfren@fiberhome.com klu@fiberhome.com

Abstract

Multi-Camera multi-target tracking is essential in the research field of urban intelligence traffic. It shows that the task becomes challenging due to differences of illumination, angle, and occlusion under different cameras. In this paper, we propose an efficient multi-camera vehicle tracking system, which contains a model trained with multi-loss to extract appearance features, and a filter with spatial-temporal information between cameras. The proposed system includes three parts. Firstly, we generate tracklets in a single-camera with different views by vehicle detection and multi-target tracking. Secondly, we extract the appearance feature of each tracklet through the trained vehicle ReID model. Thirdly, we innovatively propose a matching strategy that calculates several factors, the similarity of appearance features, the time information, and the space information of target ID between adjacent cameras. The proposed system ranks the sixth place in the City-Scale Multi-Camera Vehicle Tracking of AI City 2021 Challenge (Track 3) with a score of 0.5763.

1. Introduction

In recent years, video intelligence analysis is replacing manual supervision step by step, which brings great convenience to intelligent transportation and urban management. However, city-level multi-camera vehicle tracking is a difficult task as it's still challenging in cross-camera target matching. Because of the different camera parameters and various installation positions, images of same vehicle caught by different cameras differ a lot, which caused by degrees of occlusion, distortion and motion blur. Accurate cross-camera target matching has become the key to solving the task of multi-camera vehicle tracking.

Recently, the problem of multi-camera vehicle tracking has seen much academic research in the field of computer vision. There are several public datasets and excellent results have been achieved in many competitions. However, multi-camera vehicle tracking needs to consume many GPU resources, and a large amount of labeled data is required to train the model, which makes the existing approaches difficult to adopt in real application scenarios. In order to solve these problems, we propose an effective multi-camera vehicle tracking system based on spatial-temporal filtering.

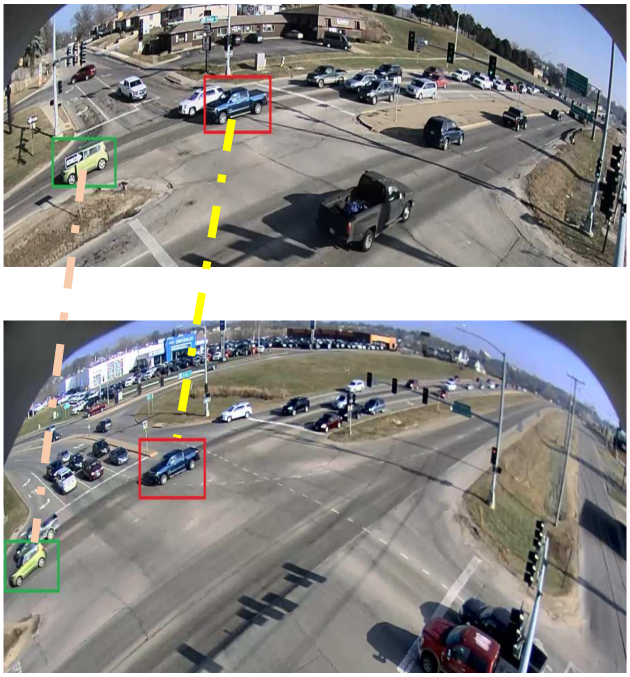

Figure 1: Multi-camera vehicle tracking needs to find out the same vehicles which crosses in multiple cameras. Their appearance and size usually varies a lot due to the difference of viewing perspective and distance to cameras.

Our proposed method works in three steps. Firstly, we perform object detection on the video frame. Secondly, we perform multi-object tracking on a single camera. Finally, we associate the target tracklets under multiple cameras according to feature similarity and spatial-temporal information.

Our main contributions are as follows:

‡ Equal Contribution

1. We propose a powerful vehicle ReID model which is robust against conditions such as occlusion deformation by considering joint loss.
2. A time-GPS-fusing strategy that can efficiently improve the accuracy of cross-camera vehicle matching is proposed in this paper.
3. We propose a technique to filter the optimal frame for each target which can significantly avoid feature mismatching between cross-cameras.

2. Related Research

2.1. Object Detection

Object Detection is a challenging task in computer vision field. In computer vision, Object Detection can usually be divided into single stage methods and two stage methods. two-stage object detection networks, such as Faster RCNN[1], usually join the RPN network to filter candidate boxes, and then send the candidate boxes to the head network for detection, which will generate confidence and coordinate boxes. Single stage has better real-time performance than two-stage, such as YOLO[2,3,4], SSD[5], RetinaNet[6], which directly predicts the confidence score and the position of the regression box for each anchor. Recently, the more popular anchor-free object detection algorithm also belongs to single stage, such as CenterNet[7], CornerNet[8], which can achieve higher accuracy and speed by predicting the width and height of the target from the center point or corner point. Instance segmentation can usually be used as an extension of the object detection task, and each pixel is classified through additional branches to generate a regression box with higher accuracy.

The video object detection algorithm is used to solve the problems of motion blur, partial occlusion, and deformation in the image. It improves the detection speed and accuracy by extracting redundant information between frames. For example, Liu[9] et al. proposed to use slownetwork and fastnetwork to extract different feature, based on convLSTM feature fusion to generate a detection frame.

2.2. Object tracking

Object tracking has been widely used in the fields of security monitoring, drones and autonomous driving in recent years. Object tracking can be subdivided into single object tracking (SOT) and single camera multiple object tracking (MOT), multi-camera vehicle tracking (MTMC). The single object tracking algorithm tracks the position of the target in the next frame based on the position information of the initial single target, such as SiamRPN++[10], DeepSRDCF[11], and combines the features extracted by the neural network with related filtering to obtain more accurate position information of the target. Unlike single-object tracking, multi-object tracking locates multiple target positions that appear in each frame of the video, and then correlates the targets that appear in the previous and next frames. The current implementation of multi-object tracking algorithms is usually tracking by detection, such as SORT[12] and DeepSORT[13], which use object detection and Kalman filtering to detect and track multiple targets. Among them, SORT uses the calculation of the IOU between adjacent frame targets and the Hungarian algorithm as a matching strategy. DeepSORT improves SORT and adds a feature similarity matching strategy to alleviate SORT's mismatch and target loss problems. Compared with MOT, multi-camera vehicle tracking (MTMC) is more challenging, because of the influence of different camera parameters, heights, camera perspectives, etc., the same target varies greatly under different camera perspectives, and only using cross-camera feature matching will appear. A large number of target mismatches, how to solve this problem will be discussed in detail in the following chapters.

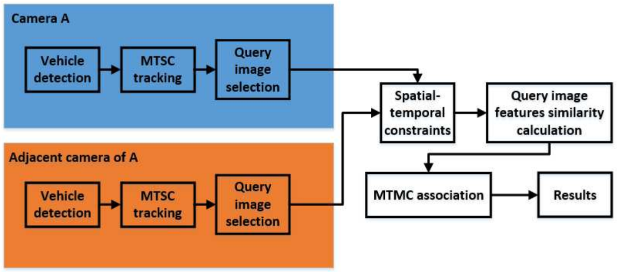

Figure 2: The pipeline of our MTMC tracking.

2.3. Vehicle Re-identification

ReID is usually used in a computer vision search task, which solves the feature matching problem. It can classify different objects by a powerful feature extraction model. In recent years, thanks to the popularization of smart cities and smart transportation, vehicle ReID has received more attention and research. However, unlike pedestrian ReID, vehicles of the same model have the same appearance, which is quite difficult to distinguish the identities between different vehicles using human eyes. Vehicle ReID can add some preliminary information to improve the accuracy of vehicle tracklets matching between different cameras. Vehicle ReID can compare the similarity of feature matching between different cameras and vehicles, and can also add license plate information, vehicle color information, and time and space information and so on. For example, Shen[14] et al. proposed to Siamese-CNN[15] Network and LSTM[16] model use candidate paths and paired queries to generate their similarity scores. He[17] et al. proposed a multi domain learning method(DMT), which combines real data with synthetic data to train the model. Qian[18] et al. proposed to add location information of different vehicles and compare the trackers of adjacent cameras to ensure the accuracy of the vehicle's ReID, Sun[19] et al. proposed batch normalization neck. Synthesize cross-domain image data to improve the generalization ability of the ReID model effectively. Zheng[20] et al. proposed a joint learning framework, which combines ReID learning and data generation end-to-end. Zhou[21] et al. generate multi-view features through single-view features and attention mapping.

3. Methodology

3.1. Vehicle Detection

In traffic scenarios, the focus is mainly on vehicle categories such as car, bus, and truck. However, a pickup truck may be detected as a truck and a car at the same time. So we merge the above categories into a vehicle label. Then NMS[22] is applied to filter out redundant detection boxes. The vehicle detection model is trained based on Microsoft's COCO[23] datasets.

The implementation of single-camera multi-object tracking adopts tracking by detection paradigm. The first step is to detect the vehicles appearing in the video frames. We select the best instance segmentation method Mask-RCNN[24] as our frame level detection model. Mask-RCNN uses strong backbone with feature pyramid network[25]. Many candidate detections can be extracted by applying region proposal network. At the end, the candidate detections are sent to multiple output heads, we can get the object class, confidence score of each detection, bounding box and segmentation mask of each object. Especially, the frame sequence of a single camera can be denoted as $I_i = \{I_1, I_2, \ldots, I_T\}$, where I_t represents the frame corresponding to the t-th moment, and T is the length of the frame sequence. The frames are sent to the Mask-RCNN to generate the vehicle detection results.

3.2. Online Multi-target Single-camera Tracking

To track multiple vehicles within a single view, detection-based-tracking paradigm DeepSORT is used to associate vehicle detections into tracklets. DeepSORT is an online tracking algorithm proposed by Wojke et al. Compared to SORT, it not only incorporates a Kalman filter with a constant velocity model to estimate the location and speed of targets from noisy detections but also introduces deep visual features of targets to ensure the accuracy of association and suppress ID-switches. The deep visual feature model is trained both for DeepSORT and vehicle ReID, which will be introduced in section 3.4.

3.3. Query Images Selection

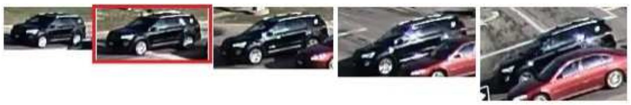

Figure 3: Gallery visualization of vehicle. Red box represents the query image.

After the results of Multi-target Single-camera Tracking are required, it is crucial to select one query image to represent the entire tracklet. To ensure the accuracy of vehicle ReID, query and gallery images should be extracted with larger areas and smaller occlusion. As is shown in figure 3, gallery images of one tracklet are selected by areas larger than threshold and IOU value less than IOU_s with other vehicles in the same frame, while query image is the one that has the least IOU value of gallery images. For gallery images with IOU values equal to 0, the query image is the one with the largest area.

3.4. Vehicle Re-identification

First of all, to obtain the features of vehicle appearance, we built a CNN model, while the backbone uses ResNet[26] series, which has strong generalization ability. Then, we used the aggregation loss to train the network, including cross entropy loss, center loss[27], and triplet loss[28] with id mining. Among them, center loss makes the appearance features of vehicles with the same label closer to the center of the class, reducing the distance between the intra-class features. Meanwhile, triplet loss makes the appearance features of vehicles with different labels more distant. In other words, the distance between the features of the inter-class is greater. At the same time, in each batch, we sampled the data of the same scene but different cameras of each label to enhance the discrimination ability of the appearance feature model.

The triplet loss $loss_{tr}$ are represented as:

$$loss_{tr} = max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2, 0) \quad (1)$$

Where $f(A)$ is the features of anchors, $f(P)$ is the features of positives and $f(N)$ is the features of negatives.

And our aggregation loss $loss_{agg}$ is represented as:

$$loss = \alpha * loss_{ce} + \beta * loss_{tr} + \gamma * loss_{center} \quad (2)$$

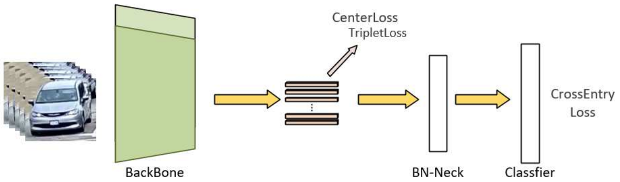

Figure 4: Training pipeline of our vehicle ReID model.

Among them, α is the weight of ID loss, β is the weight of triplet loss, and γ is the weight of center loss.

In the inference stage, we use the results of MTSC as input, extract the features of frames from query set and gallery set.

3.5. Multi-target Multi-camera Tracking

We proposed a low-cost MTMC strategy, which can extremely reduce the number of matching candidates by setting a spatial-temporal filter to vehicles in different cameras.

Considering the location of each camera and their surrounding roads, if the vehicle turns left or right at the intersections, it cannot return to the main street in several minutes, and each of the provided videos only lasts several minutes. So we set two west and east lines for each camera as shown in figure 5, west line drawn in green locates in the west side of intersection and east line drawn in red locates in the east side of intersection. Vehicles run across one or both of the lines are taken into account, while those don't cross any line are considered only to appear in the current camera and are not allowed to match with vehicles in adjacent cameras. We also calculate the directions of vehicles driving across the lines, which can record the driving directions of vehicles. If a vehicle runs across the west line along the heading-west direction, it may appear in the adjacent camera located in the west of the current camera.

Given the projection matrix of each camera, center point GPS coordinates can be calculated by center pixel coordinates and the corresponding projection matrix. As the road is almost straight among cameras, we assume the driving distance between adjacent cameras is equal to the distance between adjacent camera center points. A max driving speed of vehicle v is estimated to calculate time gap between adjacent cameras. If the time interval between vehicles that pass the spatial filter is larger than the corresponding time gap, the vehicles pass the temporal filter. The number of matching candidates can be reduced by wiping off vehicles without passing spatial or temporal filter.

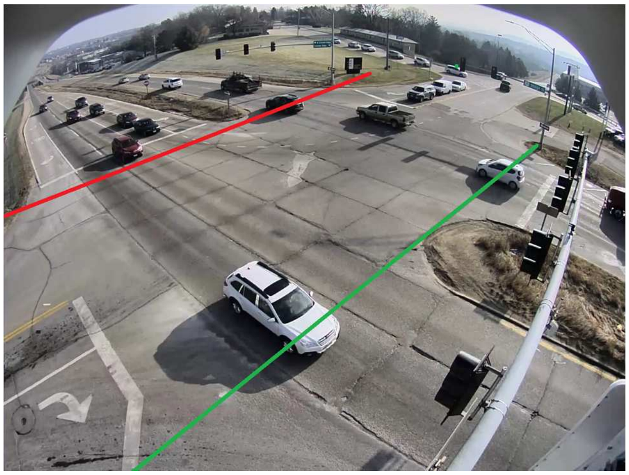

Figure 5: A vehicle crossing the line and entering an intersection needs to be matched with the tracklets of the previous camera, one leaving the intersection and crossing the line needs to be matched with the tracklets of the next camera.

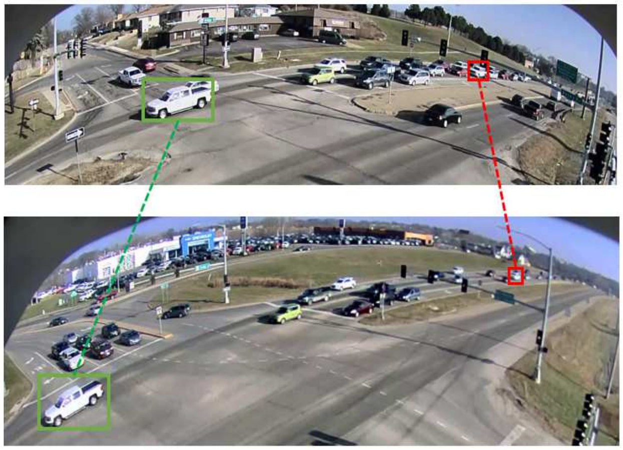

Figure 6: Vehicles with similar appearance are matched in the order in which they appear in the camera.

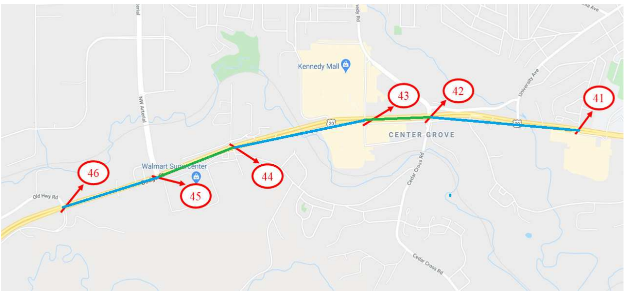

Figure 7: Driving distance between adjacent cameras is equal to the distance between adjacent camera center points.

We first get the top three most matched vehicle pairs of $camera_i$ and $camera_{i+1}$, then we apply a reverse match to get the top three most matched vehicle pairs of $camera_{i+1}$ and $camera_i$. Suppose tracklet A and B (A and B are tracklets in adjacent cameras which pass spatial and temporal filters) are in top3 most matched pairs with each other, A and B are possible to be an exact vehicle appears in adjacent cameras.

Final MTMC results are generated under the following rules:

1. If feature similarity between tracklet A query image and tracklet B query image is larger than 0.99, A and B are strongly matched.

2. If feature similarity between tracklet A query image and tracklet B query image list at the top of tracklet A top three most matched and also list at the top of tracklet B top three most matched, A and B are matched.
3. If tracklet B and tracklet A are matching candidates, at the same time, tracklet C and tracklet A are also matching candidates, matches are picked by the sort of feature similarities.

4. Experiments

4.1. Datasets

Besides COCO[12] and ImageNet[4], which are used to pre-train our backbone networks of detection model and ReID model, we only use the training and validation data of AI City Challenge 2021 Track 3.

4.2. MTSC

DeepSORT is used to perform MTSC. We modify four parameters to get better performance on test set, which are max_iou_distance, max_cosine_distance and max_age. The results are provided in Table 1. For achieving higher IDF1, the final parameters are selected as max_iou_distance 0.1, max_cosine_distance 0.9 and max_age 10.

max_iou_d istance	max_cosine_d istance	max_age	IDF1
0.1	0.9	10	**0.5763**
0.3	0.9	10	0.5650
0.1	0.8	10	0.5591
0.1	0.9	30	0.5028

Table 1. Results on official leaderboard with different MTSC parameters

4.3. Query Image Selection

We test the IOU threshold from 0.2 to 0.4 with step 0.1 to get the best IOU threshold value and the results are provided in Table 2.

IOU_s	IDF1
0.2	**0.5763**
0.3	0.5712
0.4	0.5680

Table 2. Results on official leaderboard with different IOU threshold to select query images

4.4. Vehicle ReID

We generate a series of vehicle images based on the train and validation videos of track3 from the car body frame provided by Mask-RCNN, select S01, S03, S04 and S05 as the training data, S02 as the validation set. The training set has a total of 128659 images. The validation set has 20495 gallery images, and 451 query images. We use ResNet101-ibn-a as the backbone network, and add BN-Neck layers after the backbone network. We chose Adam as the optimizer and combine cross entropy loss, triplet loss and center loss to the loss function. Respectively, the weights α, β, γ are set to 1.0, 1.0 and 0.005. Finally, the model outputs a 2048-dimensional feature vector. The results are provided in Table 3 and Table 4.

Backbone	Rank-1	Rank-5	Rank-10
Res50-ibn-a	54.1	59.4	63.3
Res101-ibn-a	53.3	61.6	67.1
Res101-ibn-a + BN-Neck	**60.2**	**71.4**	**76.3**

Table 3. Comparison of results of Vehicle ReID models with different backbone on S02 validation set

Backbone	Rank-1	Rank-5	Rank-10
DMT	67.3	69.1	72.4
Ours	**78.7**	**85.1**	**87.9**

Table 4. Comparison of results of Vehicle ReID models with diffenent method on inner validation set

Compared to ResNet50, ResNet101 has a more powerful feature extraction capability. BN-Neck can allow triplet loss to constrain features in free European space. Our method can reduce the distance between intra-class features with center loss that DMT don't use. Through the above strategies, the model can learn the differences of different vehicle IDs.

4.5. MTMC

In the spatial-temporal filter, speed parameter v has a great influence on the final matching results. v should be set large enough to avoid the temporal filter from wiping off correct matching candidates. Speed parameter v is set to 28m/s after we analyzed the test videos.

Feature similarity threshold is tested and the results are provided in Table 5.

feature similarity threshold	IDF1	IDP	IDR
0.8	0.5591	0.6627	0.4835
0.9	**0.5763**	**0.7013**	**0.4891**

Table 5. Results on official leaderboard with different feature similarity threshold

Camera	41-42	42-43	43-44	44-45	45-46
Distance	1093.9	476.2	991.0	586.9	675.2
Time interval	39.0	17.0	35.4	21.0	24.1

Table 6. Distance and time interval between adjacent cameras

Rank	TeamID	Score
1	75	0.8095
2	29	0.7787
3	7	0.7651
4	85	0.6910
5	42	0.6238
6	**Ours**	**0.5763**
7	15	0.5654

Table 7. Official results of City-Scale Multi-Camera Vehicle Tracking on the leader board of AI City Challenge 2021

5. Conclusion

In this paper, we propose an efficient multi-camera vehicle tracking system based on spatial-temporal filtering. The system contains three parts. In single-camera vehicle detection and tracking part, parameters are carefully selected to ensure the performance of MTSC result. In vehicle appearance feature extraction part, we use ResNet-101 with the combination of triplet loss and center loss, which can enlarge inter-class distance and reduce intra-class distance. In MTMC vehicle tracking, a spatial-temporal filter is proposed to reduce matching candidates and suppress the wrong matches. Our system ranks sixth place with a score of 0.5763 in the official track3 public leaderboard. In the actual city traffic intelligence system, our system is easy to install and able to adapt to different deployment scenarios.

References

[1] Ren, Shaoqing, *et al*. "Faster r-cnn: Towards real-time object detection with region proposal networks." arXiv preprint arXiv:1506.01497 (2015).

[2] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[3] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[4] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).

[5] Liu, Wei, *et al*. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

[6] Zhou, Xingyi, Dequan Wang, and Philipp Krähenbühl. "Objects as points." arXiv preprint arXiv:1904.07850 (2019).

[7] Law, Hei, and Jia Deng. "Cornernet: Detecting objects as paired keypoints." Proceedings of the European conference on computer vision (ECCV). 2018.

[8] Liu, Mason, *et al*. "Looking fast and slow: Memory-guided mobile video object detection." arXiv preprint arXiv:1903.10172 (2019).

[9] Li, Bo, *et al*. "Siamrpn++: Evolution of siamese visual tracking with very deep networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[10] Danelljan, Martin, *et al*. "Convolutional features for correlation filter based visual tracking." Proceedings of the IEEE international conference on computer vision workshops. 2015.

[11] Bewley, Alex, *et al*. "Simple online and realtime tracking." 2016 IEEE international conference on image processing (ICIP). IEEE, 2016

[12] Wojke, Nicolai, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric." 2017 IEEE international conference on image processing (ICIP). IEEE, 2017.

[13] Qian, Yijun, *et al*. "ELECTRICITY: An efficient multi-camera vehicle tracking system for intelligent city." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.

[14] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In ICCV, 2017. 2.

[15] Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

[16] Staudemeyer, Ralf C., and Eric Rothstein Morris. "Understanding LSTM--a tutorial into Long Short-Term Memory Recurrent Neural Networks." arXiv preprint arXiv:1909.09586 (2019).

[17] He, Shuting, *et al*. "Multidomain learning and identity mini ng for vehicle reidentification." Proceedings of the IEEE/C VF Conference on Computer Vision and Pattern Recognitio n Workshops. 2020.

[18] Sun, Yifan, *et al*. "Svdnet for pedestrian retrieval." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[19] He, Kaiming, *et al*. "Mask r-cnn." Proceedings of the IEEE international conference on computer vision. 2017.

[20] Zheng, Zhedong, *et al*. "Joint discriminative and generative learning for person re-identification." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[21] Zhou, Yi, and Ling Shao. "Aware attentive multi-view inference for vehicle re-identification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[22] Neubeck, Alexander, and Luc Van Gool. "Efficient non-maximum suppression." 18th International Conference on Pattern Recognition (ICPR'06). Vol. 3. IEEE, 2006.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence ´ Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 3, 5.

[24] He, Kaiming, *et al.* "Mask r-cnn." Proceedings of the IEEE international conference on computer vision. 2017.

[25] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[26] He, Kaiming, *et al*. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[27] Wen, Yandong, *et al*. "A discriminative feature learning approach for deep face recognition." European conference on computer vision. Springer, Cham, 2016.

[28] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.