electronics

Article

Online Multiple Object Tracking Using a Novel Discriminative Module for Autonomous Driving

Jia Chen [1], Fan Wang [1], Chunjiang Li [2], Yingjie Zhang [1], Yibo Ai [1,*] and Weidong Zhang [1,*]

[1] National Center for Materials Service Safety, University of Science and Technology Beijing, Beijing 100083, China; chenjia1227@163.com (J.C.); wangfan@xs.ustb.edu.cn (F.W.); zhangyingjie@xs.ustb.edu.cn (Y.Z.)
[2] College of Nuclear Technology and Automation Engineering, Chengdu University of Technology, Chengdu 610000, China; ls_lichunjiang@163.com
* Correspondence: ybai@ustb.edu.cn (Y.A.); zwdpaper@163.com (W.Z.)

Abstract: Multi object tracking (MOT) is a key research technology in the environment sensing system of automatic driving, which is very important to driving safety. Online multi object tracking needs to accurately extend the trajectory of multiple objects without using future frame information, so it will face greater challenges. Most of the existing online MOT methods are anchor-based detectors, which have many misdetections and missed detection problems, and have a poor effect on the trajectory extension of adjacent object objects when they are occluded and overlapped. In this paper, we propose a discrimination learning online tracker that can effectively solve the occlusion problem based on an anchor-free detector. This method uses the different weight characteristics of the object when the occlusion occurs and realizes the extension of the competition trajectory through the discrimination module to prevent the ID-switch problem. In the experimental part, we compared the algorithm with other trackers on two public benchmark datasets, MOT16 and MOT17, and proved that our algorithm has achieved state-of-the-art performance, and conducted a qualitative analysis on the convincing autonomous driving dataset KITTI.

Keywords: multi object tracking (MOT); autonomous driving; discrimination module; anchor-free detector

Citation: Chen, J.; Wang, F.; Li, C.; Zhang, Y.; Ai, Y.; Zhang, W. Online Multiple Object Tracking Using a Novel Discriminative Module for Autonomous Driving. *Electronics* **2021**, *10*, 2479. https://doi.org/10.3390/electronics10202479

Academic Editor: Arturo de la Escalera Hueso

Received: 28 July 2021
Accepted: 5 October 2021
Published: 12 October 2021

1. Introduction

The multi object tracking (MOT) system is an accurate tracking of obstacles moving in front of or in the surrounding environment of an autonomous vehicle, including vehicle path tracking, non-motor vehicle trajectory tracking, pedestrian trajectory tracking, etc. This subsystem helps self-driving cars make decisions and avoid collisions with objects that may move (for example, other vehicles and pedestrians) [1–3]. In the above scenarios, the main task of the multi-object tracking algorithm is to track many objects simultaneously, assign and maintain a corresponding ID for each object, and record the trajectory, which cannot be achieved by only using the object detection algorithm or single object tracking algorithm.

The object tracking task is very important to driving safety and can effectively predict the trajectory of object movement, so that the control layer can make decisions such as collision warning and lane change processing in advance. The application of object tracking can be divided into single object tracking (SOT) [4,5] and multi-object tracking (MOT) in terms of the number of objects. In the actual traffic scene, MOT is more common, and the matching relationship between the previous frame and the next frame of multiple moving objects in the actual movement should be taken into account. An example of an output diagram is shown in Figure 1. As an important task branch of computer vision, the MOT algorithm has also been widely used in the fields of intelligent surveillance systems [6], medical image processing [7] and human–computer interaction [8].

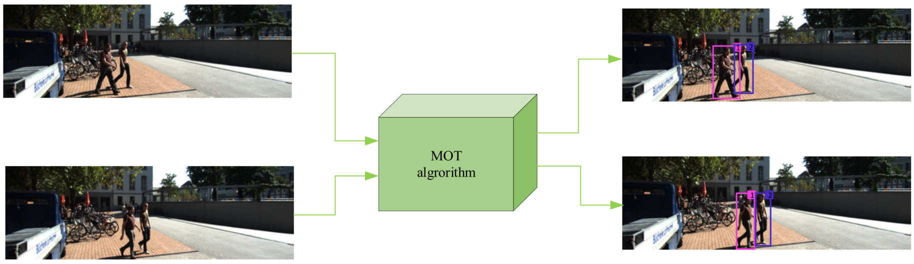


Figure 1. Sample output of the MOT algorithm.

MOT can be divided into offline mode and online mode in terms of processing mode. Offline tracking usually uses global information to track objects, so its accuracy is relatively high. However, due to its high computational cost and huge storage space, it is not suitable for automatic driving which requires high portability and real-time performance. Online tracking, due to its real-time requirements, can only use the information of the current frame and historical frame, which brings more challenges to researchers. Due to the complexity of multiple tracking problems, we need to consider not only the change of shooting angle and illumination, but also the emergence of a new object, the disappearance of an old object, and how to recognize the lost object again. This makes a robust tracking algorithm still a huge challenge.

Recently, deep learning technology based on neural networks has made great progress. Representative detection algorithms include Fast R-CNN [9], SSD [10] and YOLO [11–13] algorithms and so on. With the advancement of object detection technology, detection based tracking-by-detection has taken the lead. The algorithm detects the object in each frame and then matches it with the existing tracking trajectory. For a new object in the current frame, a new trajectory needs to be formed. For an object leaving the field of view in the current frame, the trajectory of the object needs to be terminated. However, whether it is a detector based on Faster-RCNN [14] or a detector based on SDP [15], they are all anchor-based detectors, which are prone to the problems of center point offset and low accuracy of the regression frame. Therefore, in this paper, we use the anchor-free detector algorithm.

In this work, in order to meet the scene requirements of real-time online tracking of autonomous vehicles, we are inspired by the pipeline FairMOT [16] algorithm and propose an online multi-object real-time tracker based on the feature extraction of ROI regions. This algorithm designs a multifunctional discriminant model by differently affecting the driver in the autonomous driving scene by overlapping or adjacent objects and backgrounds. The model determines the type of trajectory by calibrating the ROI of the object detected in the previous frame, and then uses the discriminative model to solve the change in the appearance of the object due to the occlusion of the object or the interaction between the objects, and then obtain the global characteristic trajectory of the object during the movement. At the same time, in order to meet the real-time requirements of autonomous vehicles, historical information and future information are used at the same time to smooth the trajectory of objects on multiple frames. The main contributions of this work are as follows:

i. An online multi-object tracking algorithm suitable for the process of autonomous driving environment perception is proposed.
ii. For the occlusion problem of different objects or overlapping adjacent objects when the object is moving, a discriminative learning model is proposed.
iii. The performance of our proposed MOT tracker has achieved competitive performance on the MOT [16], MOT [17] benchmark and KITTI datasets.

2. Our Proposed Tracker

In this section, we first introduce the FairMOT pipeline and the novel detection strategy, then introduce the proposed online MOT tracking algorithm, and finally, introduce in detail our optimized trajectory extension strategy for different tracking objects during the tracking process.

2.1. Baseline FairMOT

2.1.1. Problem Formulation

Since multi-object tracking is used to predict the position state of multiple objects in the next frame, the tracking method of MOT can be described as a multi-variable optimisation problem. Given an image sequence, suppose that A_t^i and X_t^i are the state value and observation value of the i-th target in frame t respectively, and $A_t = \left(A_t^1, A_t^2, \ldots A_t^{M_t}\right)$ is the track sequence value of all targets M_t in frame t. $A^i_{i_s:i_e} = \left\{A^i_{i_s}, \ldots, A^i_{i_e}\right\}$ is the track sequence value of the i-th target, where i_s and i_e respectively denote the object i for the start and end frame that the object i appears, while $A_{1:t} = \{A_1, A_2, \ldots A_t\}$ represents the track sequence of all objects in the image from the start frame to the t-th frame. $X_t = \left(X_t^1, X_t^2, \ldots X_t^{M_t}\right)$ is used to refer to the observed values of all objects M_t in frame t. $X_{1:t} = \{X_1, X_2, \ldots X_t\}$ represents the observation values of all the object bears from the start frame to the t-th frame in the image.

The research purpose of MOT is to find the best trajectory of all objects. Therefore, under the condition that all object state values are known, the optimization problem of MOT can be modeled by the maximal a posteriori (MAP) probability model as:

$$\hat{A}_{1:t} = \underset{A_{1:t}}{\operatorname{argmax}} P(A_{1:t}|X_{1:t}) \tag{1}$$

The prediction and update process is obtained by the following formula:

$$\text{Predict} : P(A_t|X_{1:t-1}) = \int P(A_t|A_{t-1})P(A_{t-1}|X_{1:t-1})dA_{t-1}$$

$$\text{Update} : P(A_t|X_{1:t}) \propto P(X_t|A_t)P(A_t|X_{1:t-1})\,) \tag{2}$$

2.1.2. FairMOT Pipeline

For the detection-based multi-object tracking algorithm, the detection performance of the detector directly affects the tracking effect. The traditional MOT algorithm basically uses an anchor-based detection algorithm. However, anchor-based detection not only has a large number of hyperparameters, but also has low detection accuracy. The FairMOT algorithm adopts anchorless detection, which improved detection accuracy effectively. The highlight of the FairMOT algorithm is that it combines the anchor-free detection algorithm and the Re-ID feature for end-to-end tracking. The tracking process is shown in Figure 2.

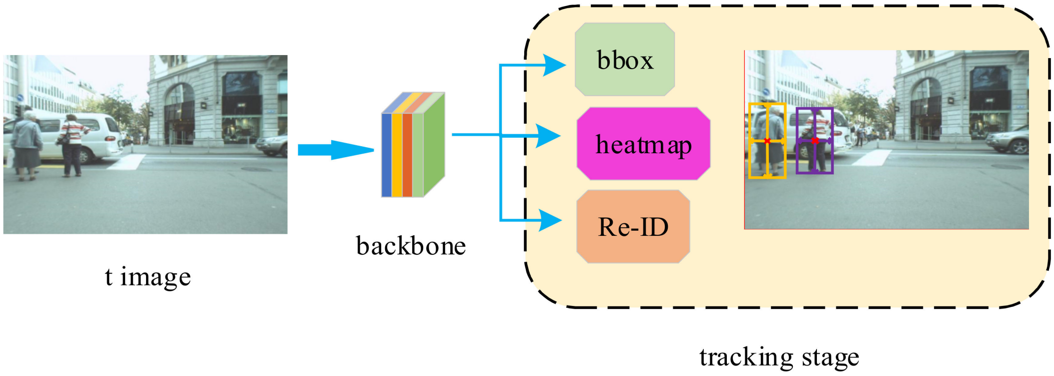


Figure 2. Simplified FairMOT pipeline.

The object detection process is regarded as a center-based bounding box regression task on high-resolution feature maps. Three parallel regression heads are added to the backbone to predict the heatmap, object center offset, and box size. The loss functions of the three processes can be obtained by the following formulas:

$$\mathrm{L_{heatmap}} = -\frac{1}{\mathrm{N}}\sum_{\mathrm{xy}}\begin{cases} \left(1-\hat{\mathrm{M}}_{\mathrm{xy}}\right)^{\alpha}\log\left(\hat{\mathrm{M}}_{\mathrm{xy}}\right) & \text{if } \mathrm{M_{xy}} = 1 \\ \left(1-\mathrm{M_{xy}}\right)^{\beta}\left(\hat{\mathrm{M}}_{\mathrm{xy}}\right)^{\alpha}\log\left(1-\left(\hat{\mathrm{M}}_{\mathrm{xy}}\right)\right. & \text{otherwise} \end{cases} \tag{3}$$

where $\mathrm{M_{xy}}$ denotes the response of (x,y),

$$\mathrm{L_{box}} = \sum_{\mathrm{i}=1}^{\mathrm{N}} ||\mathrm{o^i} - \hat{\mathrm{o}}^{\mathrm{i}}||_1 + ||\mathrm{s^i} - \hat{\mathrm{s}}^{\mathrm{i}}||_1 \tag{4}$$

$$\mathrm{L_{identity}} = -\sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{L^i(k)}\log(\mathrm{P(k)}) \tag{5}$$

Among them, $\hat{S} \in R^{9W\times H\times 2}$ and $\hat{O} \in R^{W\times H\times 2}$ are the output size and offset, respectively. $b^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ is each corresponding ground truth (GT) of the image, and its size can be represented by $S^i = (x_2^i - x_1^i, y_2^i - y_1^i)$. In the same way, the offset of GT can be obtained as $O^i = \left(\frac{c_x^i}{4}, \frac{c_y^i}{4}\right) - \left(\left\lfloor\frac{c_x^i}{4}\right\rfloor, \left\lfloor\frac{c_y^i}{4}\right\rfloor\right)$. Then, $\hat{s}^i$ and $\hat{o}^i$ are the estimated size and offset of the corresponding position, respectively. $\mathrm{L_{box}}$ is the L_1 loss function formula of the two. $P(k)$ is the distribution vector of our identity feature vector mapping at the center of the GT box. $L^i(k)$ represents the one-hot value of the GT label. Embed object recognition as a classification task. All object instances with the same identity in the training set are regarded as one class. For each label box in the picture, obtain the object center $\left(\mathrm{C_{x_i}}, \mathrm{C_{y_i}}\right)$ on the heatmap, extract an identity feature vector $\mathrm{E_{x_i,y_i}}$ to locate and learn to map it to a class distribution vector $\mathrm{P(k)}$, which represents the encoding of the label $\mathrm{L_i(k)}$.

2.2. Discrimination Learning Model

For multi-object tracking, occlusion has always been a difficult problem to overcome, although many scholars have tried to deal with occlusion. For example, Naiyan Wang et al. [17] treats the occlusion problem as a trajectory association problem, which is analogous to the data association of detection. The tracklet is put into the optical flow network for model optimization, thereby ignoring the failed detection object and continuing the tracking. However, this method did not achieve a good anti-occlusion effect because it did not pay attention to the importance of the sample itself. In this article, in order to meet the real-time performance of autonomous vehicles and the frequent occlusion problems in the process of vehicle travel, we introduce the discrimination model to solve the problem of the occlusion of moving objects.

For two known competing trajectories, as shown in Figure 3, suppose there are the previous M historical trajectories and the feature map $\mathrm{Z_1}$. In order to reduce the influence of ambient noise, we use spatial Gaussian weights to denoise each channel. Through 1 × 1 convolution operation and global maximum pooling, we get our abstract invariant features $\mathrm{S} \in \mathbb{R}^{\mathrm{N}\times\mathrm{C}}$. After the S matrix is multiplied by its transposed matrix, the $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ matrix is obtained after the softmat operation. The calculation of the correlation matrix $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ can be obtained as follows:

$$\mathrm{X_{ij}} = \frac{\exp\left(\mathrm{X_i}\cdot\mathrm{X_j^T}\right)}{\sum_{\mathrm{k}=1}^{\mathrm{N}}\exp\left(\mathrm{X_i}\cdot\mathrm{X_k^T}\right)} \tag{6}$$

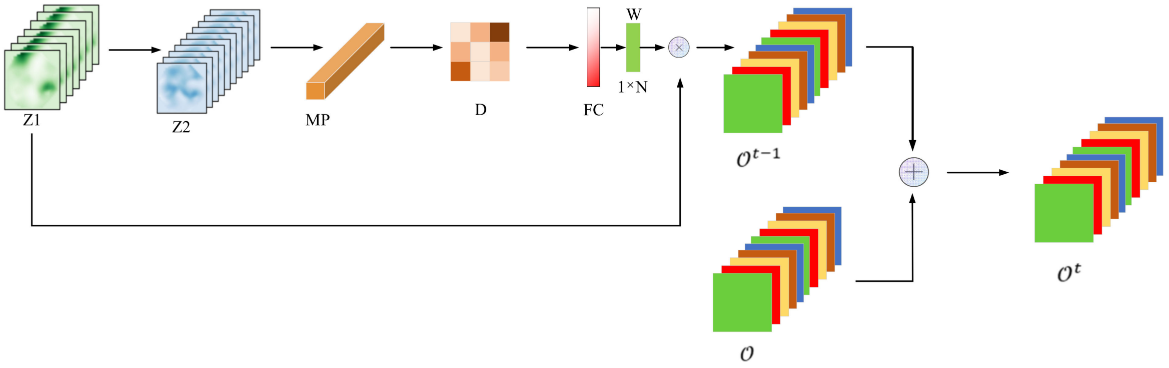


Figure 3. The details of discrimination learning model.

We can draw X_{ij} on behalf the spatial correlation between trajectory j-th and trajectory i-th. Where spatial correlation map $X \in \mathbb{R}^{N \times N}$ is a matrix composed of X_{ij}.

Next, the correlation map X is reshaped and input to the two fully connected layers and the softmaxx layer, and then the attention score $y \in \mathbb{R}^{N}$ of each position is obtained.

Finally, the final output result is obtained by:

$$\mathcal{O} = \sum_{i=1}^{N} y_i Z_{1i} \tag{7}$$

2.3. Trajectory Extension Strategy

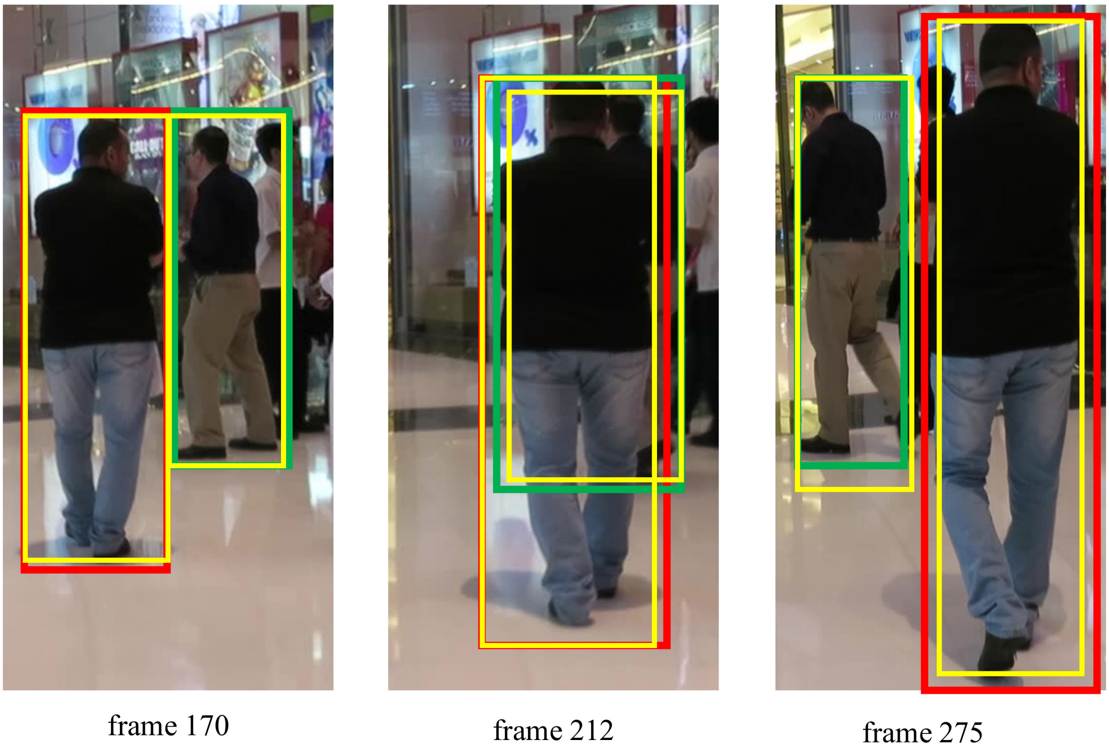
In the tracking phase, trajectory extension in MOT is one of the most challenging tasks. In order to effectively overcome the problems caused by trajectory extension, we propose a position discrimination model, which can effectively separate the object from the background and its surrounding adjacent or overlapping areas. Since the trajectories in the tracking process can be divided into isolated trajectories and competitive trajectories, we have designed different tracking strategies for them, and still adopt the classic two-stage tracking strategy.

First, for each current active trajectory, we extract its region of interest as a candidate region, and use instance segmentation to refine its bounding box. If the trajectory is an isolated trajectory, when its confidence is greater than threshold σ_t (as Equation (7)), it will be stored as a new trajectory.

$$Z_{T_n} = \begin{cases} \frac{\sum_i^n Z_i}{t_p} \cdot \left(2 - \exp\left(\vartheta \sqrt{t_p}\right)\right), & \text{if } t_p > 0 \\ 1, & \text{else} \end{cases} \tag{8}$$

Here, t_p represents the continuous tracking time in the first stage and Z_i refers to the refinement confidence in the ith growth. $\vartheta \approx \log(2)/\sqrt{T_{max}}$ is measured by the maximum number T_{max} of consecutive failures matches, which is a balance parameter. In this experiment, all ϑ values are set to 0.1.

Secondly, for the trajectory with competitive relationship, the detection example is shown in Figure 4, and the overlapping detection area after the ROI candidate area and the instance segmentation refined bounding box is taken as our candidate object. The discrimination model is used to calculate the similarity between the competition trajectory and the candidate region, and the deep Hungary algorithm is used to associate the similarity matrix to carry out the correct extension of the trajectory.



frame 170 frame 212 frame 275

Figure 4. Examples of competitive trajectory tracking results, where yellow is the detection result, and red and green indicate the correct tracking result.

The final stage is the allocation of the trajectory of the untracked object, and the IOU calculation between the detector and the threshold τ_{iou} is tracked, and it is allocated to the remaining detection results. After data association, each untracked trajectory is considered lost in the current frame, and a new trajectory is initialized with high response confidence for each unmatched detection. In order to reduce the influence of false detection, once any new trajectory is lost in any first τi frame, it will be deleted. If the trajectory continuously exceeds τt and is lost or leaves the field of view, the trajectory will terminate.

2.4. Proposed Online MOT Tracking Network

Multi-object tracking based on detection can be divided into online tracking and offline tracking. Online multi-object tracking is a frame-by-frame progressive tracking method, which is similar to the real-time tracking process of human eyes. Firstly, each moving object should be identified and confirmed (object detection), and then its next action should be predicted (trajectory prediction). Finally, the motion direction (motion model), appearance shape (appearance model) and other features of the object are associated with the previous trajectory (data correlation matching).

In this section, we will introduce the main tracking process of our algorithm. Due to anchor-based detectors have many hyperparameters and the shortcomings of features that are not easy to counteract, we employ anchor-free detectors in the detection process. As shown in Figure 5, after the t-th frame image of the current frame passes through the backbone network, the region of interest is extracted and the result of the $t - 1$th frame detection is performed to correct the position to obtain the trajectory of the object in the current frame. If the trajectory of the object in the frame is an isolated trajectory, the trajectory is stored and extended directly, and the tracking is successful. If the trajectory of the object in the t-th frame is in a competitive relationship, that is, there is occlusion, input the discrimination learning model to solve the occlusion problem through position correlation, realize the storage and extension of the trajectory, and track successfully. If the trajectory of the object in the t-th frame belongs to the new object, the trajectory is initialized. If the trajectory of the object in frame t-th does not appear in consecutive frames, the tracking is stopped and the tracking ends.

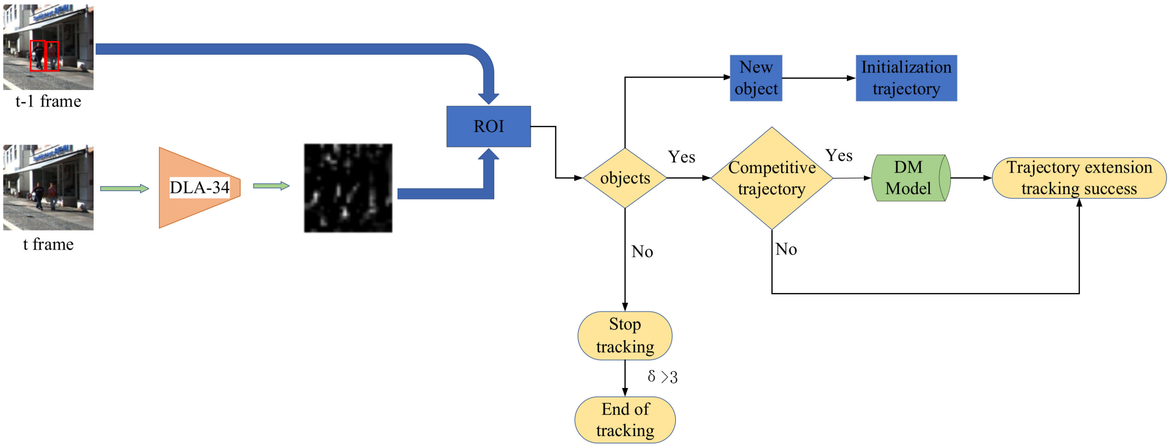


Figure 5. Simplified ours pipeline.

In order to better balance the two performances of speed and accuracy, we use the ResNet-34 backbone network with strong feature extraction capabilities like the FairMOT detection method. As shown in Figure 6, in order to better integrate the semantic and location information of different layers, we use a backbone network of Deep Layer Aggregation [18] to extract image features. At the same time, in order to dynamically adjust the receptive field when the proportion and posture of the object change, we use deformable convolution [19] to complete the up-sampling. The size of the input images are $H_{image} \times W_{image}$, and the output feature map has the shape of $C \times H \times W$ where $H = H_{image}/4$ and $W = W_{image}/4$. The proposed tracking flow is summarized in Algorithm 1.

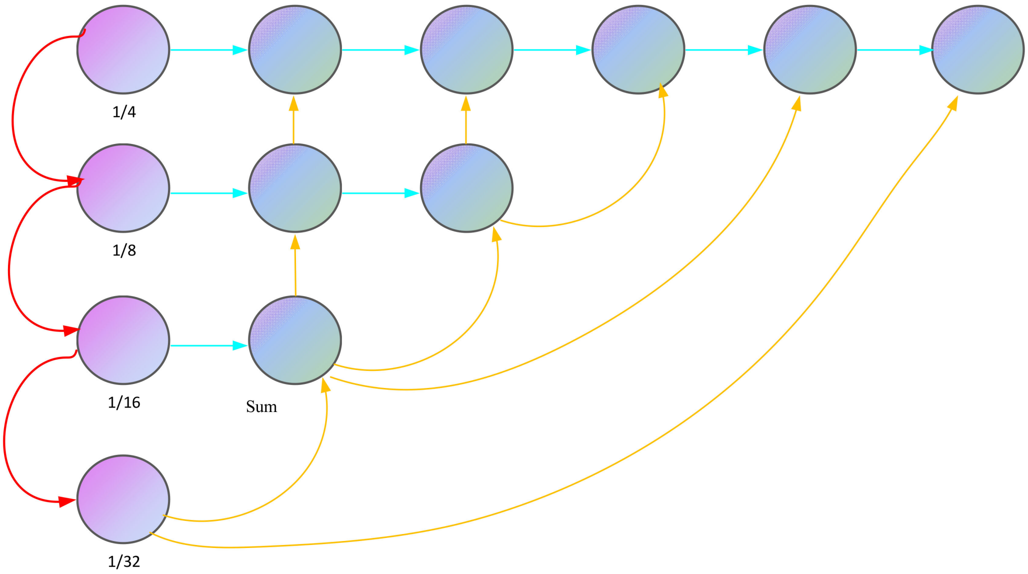


Figure 6. Deep Layer Aggregation (DLA-34) backbone network structure, where the red arrow denotes down-sampling, the yellow arrow denotes up-sampling, the blue arrow denotes the resolution keeping.

Algorithm 1: The proposed Method

```
Input: The pre-trained network model, the first frame, initial obkect location bounding box $b_1$
Output: The object location $b_2, b_3, \ldots b_n$ of the subsequent frames
1. Input the initial frame and initial bounding box
2. for $i = 2 : n$ do
            Get the ROI feature
3. Calculate the correlation matrix using Equation (6)
4. Calculate the maximum response using Equations (4) and (5)
5. Calculate the bounding box
6. end for
```

3. Experiments and Evaluation

In this section, we will introduce the experimental details of our proposed algorithm in detail and compare it with the most representative MOT16 [20] MOT17 [20] public benchmark in the MOT Challenge and an autonomous driving dataset KITTI [21,22].

3.1. Experiment Implementation Details

Our algorithm is implemented based on Pytorch in an Ubuntu 16.04 desktop computer with Intel i7-9700k CPU, 16G RAM and two Nvidia GTX1080Ti GPUs. In this experiment, we use the DLA-34 pre-trained multi-layer feature fusion on the COCO dataset [23] as the backbone network. The ADM optimizer is used for 30 epochs of training on ETH [24], city person [25] and crowd human [26]. During our experiment, the input size of all training set images is 1088 × 608, and the feature map resolution is 272 × 152.

3.2. Results on MOT16

MOT16 mainly detects moving pedestrians and vehicles. It is a dataset based on MOT15 [27] with more detailed annotations and more bounding boxes. MOT16 has a richer picture, different shooting angles and camera movements, as well as different weather condition videos. It is marked by a group of qualified researchers in strict compliance with the corresponding marking guidelines, and finally a double detection method is used to ensure the high accuracy of the marked information. The trajectory marked by MOT16 is 2D. There are 14 video sequences in the MOT16 dataset, of which 7 are training sets with annotation information, and the other 7 are test datasets.

The detector used in the MOT16 data set is DPM [28], which has a good performance in detecting the pedestrian category. The main information of these videos is as follows: including FPS, resolution, video duration, number of tracks, object book, density, static or moving shooting, low, medium and high angle shooting, weather conditions for shooting, etc.

Table 1 shows our comparison with the most state-of-the-art algorithm on the MOT16 public benchmark. The results show that whether we compare with offline trackers or online trackers, the algorithm we proposed obtains the best results on several important indicators such as MOTA, MOTP and IDF1.

Table 1. Comparison of our algorithm with other state-of-the-art algorithms.

Mode	Method	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓
Off-line	FWT [29]	47.8	75.5	44.3	19.1	38.2	8886	85,487
	TPM [30]	51.3	75.2	47.9	18.7	40.8	2701	85,504
	LMP [31]	48.8	79.0	51.3	18.2	40.1	6654	86,245
On-line	DeepMOT [32]	54.8	77.5	53.4	19.1	37.0	2955	78,765
	Tracktor++ [33]	54.4	78.2	52.5	19	36.9	3280	79,149
	DMAN [34]	51.4	76.9	54	16.5	34.9	21,042	251,873
	PV [35]	50.4	77.7	50.8	14.9	38.9	2600	86,780
	Ours	**56.3**	**79.2**	**55.1**	20.4	35.6	3095	79,634

In Table 1, FP represents false positive samples during the tracking process. The lower the value, the better. The number of false positive samples detected in our algorithm is 79,634, which ranks in the middle. FN is the false negative sample, ML is the mostly lost sample; the smaller the value of both the better. The results of our algorithm have achieved good performance in the eight competitive algorithms in 2016. MT is mostly tracking, IDF1 refers to the F value of the pedestrian ID in each pedestrian frame. The larger the value of the two, the better. MOTA and MOTP are the other most important indicators to measure tracking accuracy and position error in multi-object tracking, and can be expressed by Formulas (9) and (10) as:

$$\mathrm{MOTA} = 1 - \frac{\sum_{t}(\mathrm{FN_t} + \mathrm{FP_t} + \mathrm{IDSW_t})}{\sum_{t}\mathrm{GT_t}} \tag{9}$$

$$\mathrm{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_{t} c_t} \tag{10}$$

where t is the index of each frame of image, and GT is the ground truth label, and c_t denotes the number of matches in frame t and $d_{t,i}$ is the bounding box overlap of target i with its assigned ground truth object.

As shown in Table 1, in the three most important indicators of multi-object tracking performance, MOTA, MOTP and IDF1, the algorithm we proposed all ranked first.

3.3. Results on MOT17

3.3.1. Quantitative Analysis

MOT17 are datasets based on MOT15 with more detailed annotations and more bounding boxes, mainly for pedestrians and vehicles. They have a richer picture, different shooting angles and camera movements, as well as different weather condition videos. They are marked by a group of qualified researchers in strict compliance with the corresponding marking guidelines, and finally a double detection method is used to ensure the high accuracy of the marked information. The motion trajectory marked by MOT17 is 2D, which is a brand new data set. Compared with MOT15 of pedestrian density, it is more difficult. Therefore, in this experiment, we will use MOT17 as our verification data set to verify the performance of our algorithm.

As shown in Table 2, the best performance has been bolded in black. Compared with the online tracker or offline tracker, our algorithm has significant advantages. Because the offline tracker can use the global information to track, the overall performance of the tracker is better than the online tracker. However, due to the wide application of deep learning in the field of detection and its obvious advantages, the gap between the two is getting smaller and smaller, and will even surpass some offline trackers.

Table 2. Comparison of our algorithm with other state-of-the-art algorithms.

Mode	Method	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓
Off-line	EDMT [36]	50.9	76.6	52.7	17.5	35.7	24,069	250,768
	TT17 [37]	54.9	77.2	**63.1**	24.4	38.1	20,236	233,295
	TPM [30]	54.2	76.7	52.6	22.8	37.5	13,739	242,730
On-line	FPSN [38]	44.9	76.6	48.4	16.5	35.8	33,757	269,952
	DeepMOT [32]	53.7	77.2	53.8	19.4	36.6	11,731	247,447
	FAMNet [39]	52	76.5	48.7	19.1	33.4	14,138	253,616
	DMAN [34]	48.2	75.7	55.7	19.3	38.3	26,218	263,6083
	Ours	**55.1**	**78.9**	54.1	20.0	35.6	8524	241,795

In Table 2, among the two most indicators MOTA and MOTP to measure multi-object tracking, our algorithm exceeds the tracking performance of offline algorithms and ranks first.

In order to show the performance of our tracker more intuitively, we further compare the performance of different detectors on the test set in Table 3. Overall, the performance of the SDP [15] detector is the best among the three detectors. DPM is a traditional algorithm that uses the sliding window idea, while FRCNN and SDP are both detection methods using convolutional neural networks.

Table 3. Comparison results of different detector algorithms for MOT17.

Sequence	MOTA(↑)	MOTP(↑)	IDF1(↑)	MT(↑)	ML(↓)	FP(↑)	FN(↑)	IDSW(↑)
MOT17-01-DPM	41.7	78.4	40.3	5	11	23	3716	21
MOT17-03-DPM	65.3	79.1	59.7	51	19	1552	34,530	216
MOT17-06-DPM	54.0	80.6	55.9	47	86	120	5227	79
MOT17-07-DPM	41.6	79.3	45.9	5	22	94	9699	74
MOT17-08-DPM	26.6	83.5	32.7	8	39	68	15,375	64
MOT17-12-DPM	45.9	82.8	53.8	16	43	26	4635	27
MOT17-14-DPM	31.7	77.3	39.5	11	81	218	12,263	142
average	43.83	80.1	56.8	20.4	43	300.1	12,206.4	89
MOT17-01-FRCNN	43.6	77.9	41.1	6	10	107	3505	24
MOT17-03-FRCNN	67.7	78.7	60.3	54	18	1578	32,032	198
MOT17-06-FRCNN	57.5	80.0	58.6	55	61	225	4657	125
MOT17-07-FRCNN	41.9	79.1	46.9	6	22	219	9517	83
MOT17-08-FRCNN	26.2	83.5	32.1	8	40	94	15,431	60
MOT17-12-FRCNN	44.8	82.5	54.7	15	44	34	4728	18
MOT17-14-FRCNN	33.0	76.2	39.9	12	78	457	11,734	197
average	45.0	79.7	47.7	22.3	39	359.1	11,657	100.7
MOT17-01-SDP	43.9	77.7	59.7	6	10	104	3488	26
MOT17-03-SDP	71.8	78.1	62.7	62	16	2380	26,774	333
MOT17-06-SDP	58.0	80.0	56.9	58	65	282	4545	127
MOT17-07-SDP	43.9	78.7	45.8	8	19	222	9149	98
MOT17-08-SDP	27.7	82.7	32.4	10	37	146	15,057	74
MOT17-12-SDP	46.3	82.2	54.4	17	44	97	4532	26
MOT17-14-SDP	35.4	76.3	42.3	11	70	476	11,254	208
average	46.7	79.4	50.6	24.6	37.3	529.6	12,114	1513.4

Table 3 shows the results of various indicators in different sequences of different detectors in the MOT2017 video. The performance of our proposed algorithm has achieved good results.

3.3.2. Qualitative Analysis

In order to show the performance of our algorithm more intuitively, we conducted a qualitative analysis of the proposed algorithm as shown in Figure 7. In the first sequence of the MOT17 test dataset, a lady wearing a black skirt on a street corner can still accurately track her with the same ID after crossing and overlapping with a pedestrian next to her. Sequence 3 is a scene with a lot of people and crowded at night, and the tracker we proposed still shows good tracking performance. Sequence 6 uses a mobile camera to shoot in a busy commercial block, and still has a good tracking performance after experiencing a large range of deformation and occlusion. For MOT, in addition to difficulties such as occlusion and illumination deformation, the tracking of small objects is also an extremely challenging task. Since our algorithm uses a feature pyramid network with multi-feature fusion in the feature extraction stage, the tracking of small objects in Sequence 7 shows good performance. False detection, missed detection and occlusion have always been huge challenges faced by MOT. In order to overcome these difficulties, we adopted an anchor-free detector in the detection branch that does not rely on the experience setting, which not only effectively avoids false detections and missed detections, but also in sequence 7 we can see that the man in the white shirt was tracked accurately even after severe occlusion, and Sequence 6 shows that in a complex indoor shopping mall, we also tracked the men in black shirts that appeared midway. In the actual autonomous driving environment in the city, the tracking of pedestrians on both sides of the road and crossing the road is particularly important. Sequence 7 is taken by the in-car dash cam, which not only tracks the pedestrians on both sides of the station, but also in the distance small object pedestrians crossing the road on the zebra crossing have also been accurately tracked, which has played an important role in taking avoidance measures for subsequent vehicles and avoiding traffic accidents.

3.4. Results on the Autonomous Driving Dataset KITTI

The KITTI dataset is a computer vision algorithm evaluation dataset used in autonomous driving scenarios. It was co-founded by the Karlsruhe Institute of Technology (KIT) and Toyota Institute of Technology Chicago (TTIC). The scenes mainly include urban areas, villages and highways. Among them, the data set used for the multi-object tracking algorithm consists of 21 training sequences and 29 test sequences. Here, we have selected KITTI-16 and KITTI-19 for qualitative analysis, as shown in Figure 8 below. Since the pedestrian is a non-rigid object in MOT, it is the most difficult to track, so we only show the tracking effect on pedestrians.

KITTI-16 is a high-traffic intersection shot by a static camera. Intersections, overlaps, and occlusion frequently occur. Because we use the DM module to effectively solve the ID-switch problem caused by occlusion. KITTI-19 is a bustling road scene in the city captured by a mobile camera in the car. Our algorithm can still accurately track the road and pedestrians on both sides.

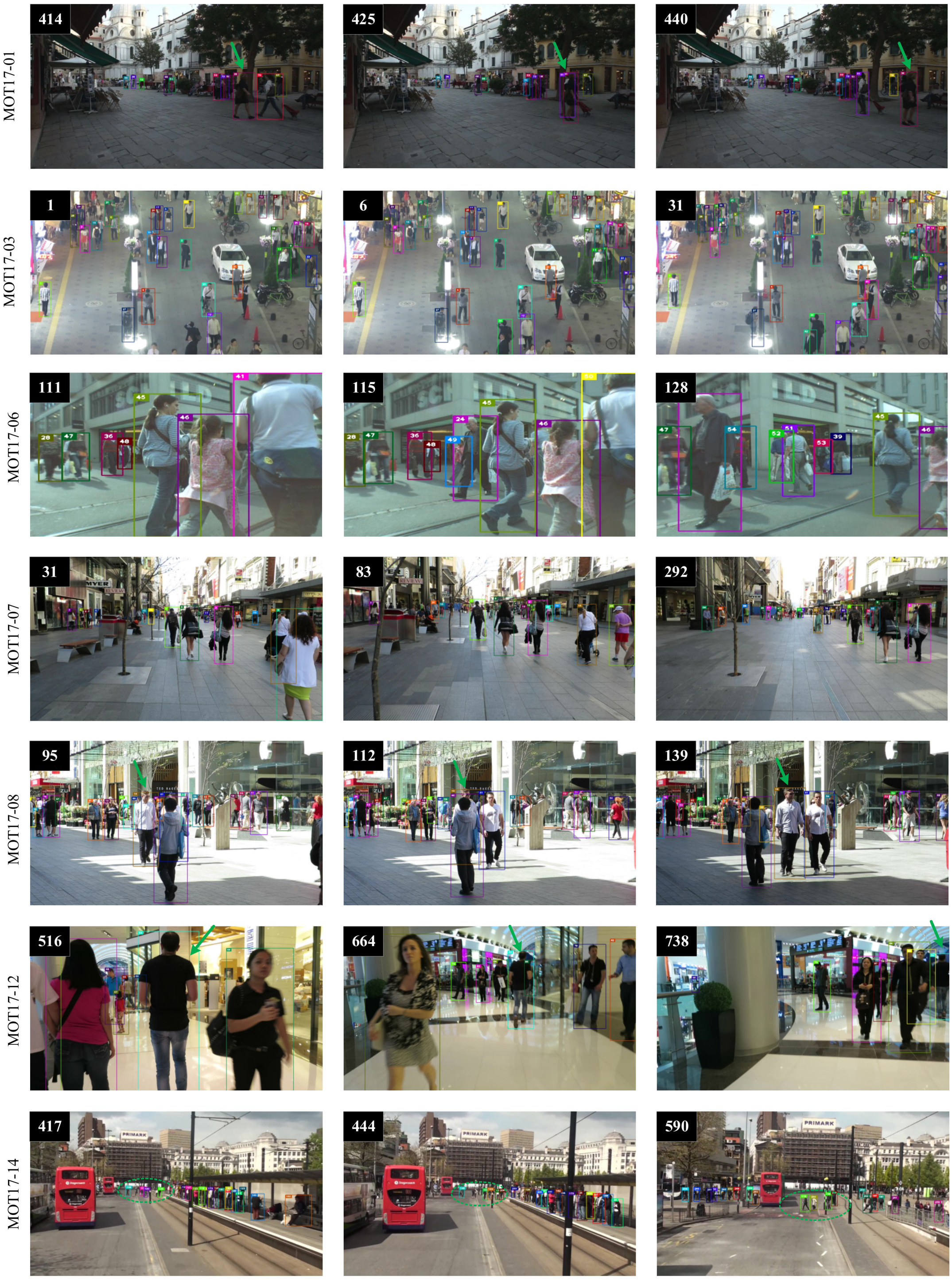


Figure 7. The tracking results of our algorithm on the MOT17 test sequence. In units of rows, from top to bottom are sequence 1, sequence 2, sequence 3, sequence 4, sequence 5, sequence 6, and sequence 7.

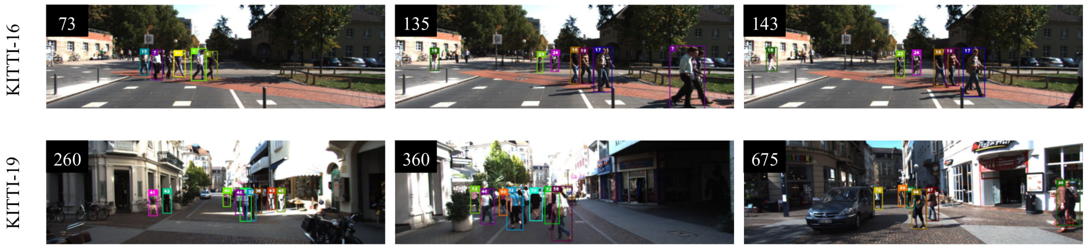


Figure 8. The tracking results of our algorithm on the KITTI dataset.

3.5. Ablation Experiment

The most important process in multi-object tracking is the early detection and the later trajectory extension. The detection accuracy directly affects our later tracking results. The innovation of our algorithm is in the detection and trajectory extension part. In order to show the performance of our algorithm more intuitively, we conducted an ablation experiment analysis on each part of our proposed algorithm on the MOT2016 dataset, as shown in Table 4. In the experiment, we list three indexes which can best reflect the performance of multi-object tracking.

Table 4. The results of ablation experiments of different models of our algorithm on the MOT16 dataset.

Method	MOTA↑	MOTP↑	MT↑
Anchor-based tracking	48.7	67.8	49.2
Anchor-free tracking	52.3	70.1	52.4
Anchor-free tracking + trajectory extension strategy (ours)	56.3	79.2	55.1

4. Conclusions

While self-driving cars bring us a lot of convenience, there are still many difficulties and challenges in real life. To this end, we use a multi-feature fusion pyramid feature extractor and anchor-free detector combined with the DM module to propose a multi-object tracking algorithm that takes into account both accuracy and speed. In particular, the proposal and application of the DM module effectively solve the problem of frequent ID-switch when the object overlaps or occludes the background and surrounding objects, and extends the competitive trajectory well. Compared with the most advanced trackers in the two benchmarks of MOT16 and MOT17, it is more competitive. In the future, we will continue to study the problems existing in the two-stage tracking, realize end-to-end multi-object tracking, and further improve the accuracy and speed of the tracker.

Author Contributions: J.C. and C.L. conceived and designed the experiments; J.C. performed the experiments; Y.A. and F.W. analyzed the data; W.Z. and Y.Z. contributed reagents/materials/analysis tools; J.C. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the financial support provided by the Fundamental Research Funds for the Central Universities of China (Grant No. FRF-GF-20-24B, FRF-MP-19-014), the project supported by Innovation Group Project of Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai) (No. 311021013) the 111 Project (Grant No. B12012).

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

References

1. Ding, S.; Liu, L.; Park, J.H. A novel adaptive nonsingular terminal sliding mode controller design and its application to active front steering system. *Int. J. Robust Nonlinear Control* **2019**, *29*, 4250–4269. [CrossRef]
2. Norouzi, A.; Masoumi, M.; Barari, A.; Farrokhpour Sani, S. Lateral control of an autonomous vehicle using integrated backstepping and sliding mode controller. *Proc. Inst. Mech. Eng. Part K J. Multi-Body Dyn.* **2019**, *233*, 141–151. [CrossRef]
3. Formentin, S.; Garatti, S.; Rallo, G.; Savaresi, S.M. Robust direct data-driven controller tuning with an application to vehicle stability control. *Int. J. Robust Nonlinear Control* **2018**, *28*, 3752–3765. [CrossRef]
4. Chen, J.; Ai, Y.; Qian, Y.; Zhang, W. A novel Siamese Attention Network for visual object tracking of autonomous vehicles. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2021**. [CrossRef]
5. Gao, M.; Jin, L.; Jiang, Y.; Guo, B. Manifold Siamese Network: A Novel Visual Tracking ConvNet for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1612–1623. [CrossRef]
6. Zhang, Q.N.; Sun, Y.D.; Yang, J.; Liu, H.B. Real-time multi-class moving target tracking and recognition. *IET Intell. Transp. Syst.* **2016**, *10*, 308–317. [CrossRef]
7. Türetken, E.; Wang, X.; Becker, C.J.; Haubold, C.; Fua, P. Network flow integer programming to track elliptical cells in time-lapse sequences. *IEEE Trans. Med. Imaging* **2017**, *36*, 942–951. [CrossRef] [PubMed]
8. Yan, X.; Kakadiaris, I.; Shah, A. Modeling local behavior for predicting social interactions towards human tracking. *Pattern Recognit.* **2014**, *47*, 1626–1641. [CrossRef]
9. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multi box detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer Press: New York, NY, USA, 2016; pp. 21–37.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE Computer Society Press: Washington, DC, USA, 2015; pp. 779–788.
12. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society Press: Washington, DC, USA, 2017; pp. 6517–6525.
13. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; IEEE Computer Society Press: Los Alamices, CA, USA, 2018; pp. 1–6.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
15. Yang, F.; Choi, W.; Lin, Y. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2129–2137.
16. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking. *Int. J. Comput. Vis.* **2020**. [CrossRef]
17. Hu, Y.; Song, R.; Li, Y. Efficient coarse-to-fine patchmatch for large displacement optical flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5704–5712.
18. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
19. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
20. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
21. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
22. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
23. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755.
24. Ess, A.; Leibe, B.; Schindler, K.; van Gool, L. A mobile vision system for robust multi-person tracking. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
25. Zhang, S.; Benenson, R.; Schiele, B. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3213–3221.
26. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv* **2018**, arXiv:1805.00123.
27. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv* **2015**, arXiv:1504.01942.
28. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [CrossRef] [PubMed]

29. Henschel, R.; Leal-Taixé, L.; Cremers, D.; Rosenhahn, B. Fusion of Head and Full-Body Detectors for Multi-Object Tracking. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018.
30. Peng, J.; Wang, T.; Lin, W.; Wang, J.; See, J.; Wen, S.; Ding, E. TPM: Multiple Object Tracking with Tracklet-Plane Matching. *Pattern Recognit.* **2020**, *107*, 107480. [CrossRef]
31. Tang, S.; Andriluka, M.; Andres, B.; Schiele, B. Multiple People Tracking by Lifted Multicut and Person Re-identification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
32. Xu, Y.; Osep, A.; Ban, Y.; Horaud, R.; Leal-Taixé, L.; Alameda-Pineda, X. How to train your deep multi-object tracker. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6787–6796.
33. Bergmann, P.; Meinhardt, T.; Leal-Taixe, L. Tracking without bells and whistles. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
34. Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; Yang, M.H. Online multi-object tracking with dual matching attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 379–396.
35. Li, X.; Liu, Y.; Wang, K.; Yan, Y.; Wang, F.Y. Multi-Target Tracking with Trajectory Prediction and Re-Identification. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019.
36. Chen, J.; Sheng, H.; Zhang, Y.; Xiong, Z. Enhancing Detection Model for Multiple Hypothesis Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 2143–2152. [CrossRef]
37. Zhang, Y.; Sheng, H.; Wu, Y.; Wang, S.; Lyu, W.; Ke, W.; Xiong, Z. Long-Term Tracking With Deep Tracklet Association. *IEEE Trans. Image Process.* **2020**, *29*, 6694–6706. [CrossRef]
38. Lee, S.; Kim, E. Multiple object tracking via feature pyramid Siamese networks. *IEEE Access* **2019**, *7*, 8181–8194. [CrossRef]
39. Chu, P.; Ling, H. FAMNet: Joint Learning of Feature, Affinity and Multi-dimensional Assignment for Online Multiple Object Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.