


Review

Multi-Object Multi-Camera Tracking Based on Deep Learning for Intelligent Transportation: A Review

Lunlin Fei ^{1,2,*} and Bing Han ³ ¹ School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China² Jiangxi Provincial Transportation Investment Group Co., Ltd., Nanchang 330029, China³ School of Civil Engineering, Beijing Jiaotong University, Beijing 100044, China; bhan@bjtu.edu.cn

* Correspondence: 20114085@bjtu.edu.cn

Abstract: Multi-Objective Multi-Camera Tracking (MOMCT) is aimed at locating and identifying multiple objects from video captured by multiple cameras. With the advancement of technology in recent years, it has received a lot of attention from researchers in applications such as intelligent transportation, public safety and self-driving driving technology. As a result, a large number of excellent research results have emerged in the field of MOMCT. To facilitate the rapid development of intelligent transportation, researchers need to keep abreast of the latest research and current challenges in related field. Therefore, this paper provide a comprehensive review of multi-object multi-camera tracking based on deep learning for intelligent transportation. Specifically, we first introduce the main object detectors for MOMCT in detail. Secondly, we give an in-depth analysis of deep learning based MOMCT and evaluate advanced methods through visualisation. Thirdly, we summarize the popular benchmark data sets and metrics to provide quantitative and comprehensive comparisons. Finally, we point out the challenges faced by MOMCT in intelligent transportation and present practical suggestions for the future direction.

Keywords: multi-object multi-camera tracking; deep neural network; object detector; intelligent transportation



Citation: Fei, L.; Han, B. Multi-Object Multi-Camera Tracking Based on Deep Learning for Intelligent Transportation: A Review. *Sensors* **2023**, *23*, 3852. <https://doi.org/10.3390/s23083852>

Academic Editor: Gwanggil Jeon

Received: 14 March 2023

Revised: 29 March 2023

Accepted: 5 April 2023

Published: 10 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

MOMCT is a crucial problem in computer vision, and it is very useful in public safety. MOMCT aims to track multiple vehicles or other objects in traffic scenes through multiple cameras, which is different from MOT (MOT) in a single camera [1]. A camera network consisting of several cameras has a wider field of view than a single camera and offers more practical application prospects. The main application scenarios include vehicle cross-regional tracking on smart highway, traffic management in smart cities [2], autonomous driving [3], and crowd analysis [4]. Especially in the process of vehicle cross-regional tracking on the highway, the MOMCT task can be used to track multiple vehicles simultaneously, which plays a key role in traffic management and analysis. Therefore, most algorithms used in MOMCT are based on the MOT algorithm, such as feature extraction algorithm, object modeling, and motion detection. The MOMCT system consists of two components: firstly, all the objects in each video frame are tracked and located by a single camera, and the detection output results are connected into a continuous trajectory across time; secondly, the network composed of different cameras matches the accurate vehicle trajectory detected across different cameras through the correlation module. The vehicle cross-regional tracking process of MOMCT system is shown in Figure 1.

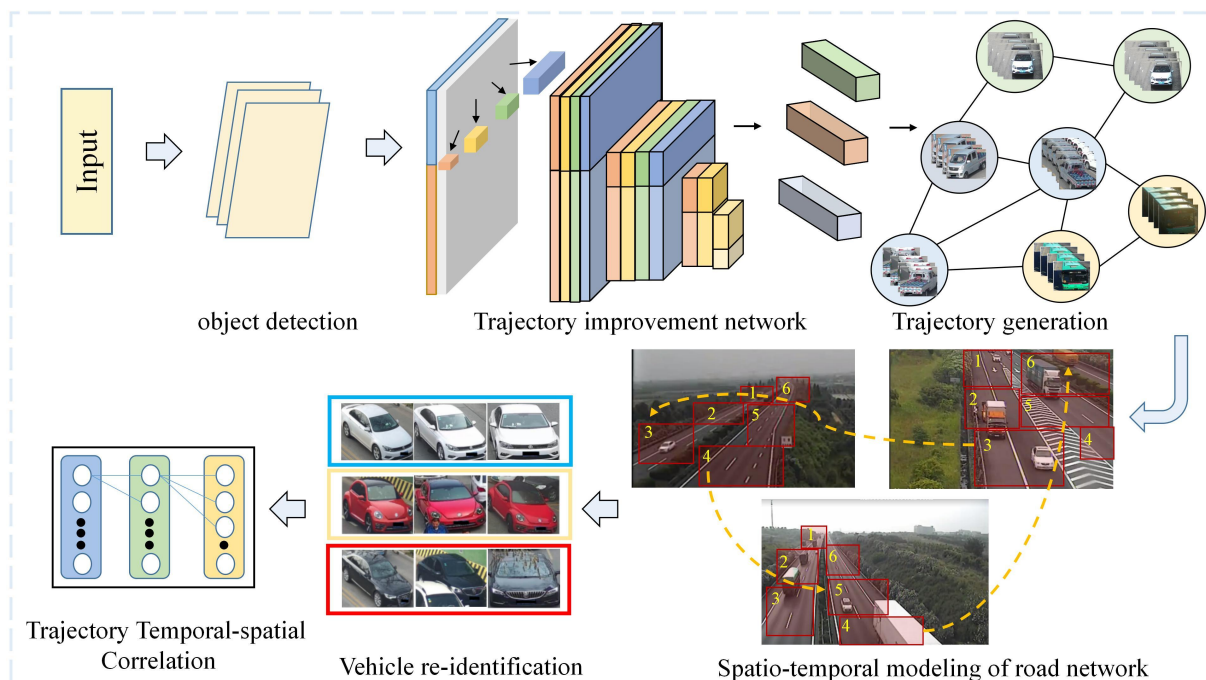


Figure 1. Cross-area tracking of vehicles with multiple objects and cameras.

In the real scenes, besides background confusion [5], posture change [6] and occlusion [7], there are still many difficulties in tracking and detecting objects because of the different video quality, lighting and viewing angle of each camera. In addition, multiple cameras will not share the overlapping area. It means that the appearance features of the same object will be very different in cameras with different perspectives. In the stage of vehicle tracking, there are often problems such as large intra-class variability and inter-class similarity [8]. To solve these problems, most MOMCT methods follow the detection and tracking paradigm: firstly, a set of detection is generated independently for each video frame shot by the camera; secondly, these detections are linked together by similarity measurement to generate a continuous trajectory. Usually, this similarity measure considers the location information and visual characteristics of the object. Visual features are very important for keeping the side of the tracking object.

The vehicle tracking problem in multi-camera system is an extension of the problem in single-camera system. Therefore, the algorithms used in multi-camera tracking are mostly based on the well-known algorithms in single-camera tracking, such as motion detection [9], object modeling [10], and feature extraction [11]. Using multiple cameras has many advantages over using a single camera. It can mainly reduce errors caused by occlusion or other sensors. However, vehicle tracking in multi-camera systems is very challenging, because the tracking process must ensure the integration of information from different sensors. The system designed to deal with MOMCT tasks usually consists of five sub-modules, namely, object detection [12], multi-object single-camera tracking [13], vehicle re-identification (re-ID) [14], and multi-object multi-camera tracking [15,16]. The general process can be summarized as follows: Firstly, the vehicle detection module outputs vehicle coordinates and categories in units of frames. Then, based on the vehicle position and the learned features, the single-camera tracking and detection module generates candidate trajectories for every single camera. Finally, these candidate trajectories are matched on different cameras by associating the object with the global identity.

In the existing reviews [17–20], their work is more focused on reviewing multi-object multi-camera tracking methods using a coarse classification, involving a wide range of application scenarios lacking relevance. In addition, the current reviews lack references to advanced results from the last three years, thus failing to provide a comprehensive overview of the latest developments in MOMCT. Therefore, as shown in Figure 2, this

paper provides an overview of recent advances in MOMCT-related technologies from different aspects for intelligent transportation applications, including four major issues such as object detection, object tracking, vehicle re-identification, and multi-target cross-camera. In addition, this paper further details their technical challenges and compares different solutions. Last but not least, we examine the performance of relevant MOMCT algorithms on various datasets by focusing on data comparisons and explore the potential value of these methods in smart cities. The main contributions of this paper are as follows:

- We provide a comprehensive overview of the application based on deep learning technology in multi-object multi-camera tracking tasks. We have classified and summarised the different stages of deep learning-based MOMCT algorithms, including object detection, object tracking, vehicle re-identification and multi-object cross-camera tracking.
- We have aggregated the most commonly benchmark datasets and standard metrics for MOMCT. We have combined various data for experimental visualisation and comprehensive metrics evaluation of the main algorithms in MOMCT.
- We discuss the challenges MOMCT has faced in recent years from several perspectives, as well as the main application scenarios in practice, and explore potential future directions for MOMCT.

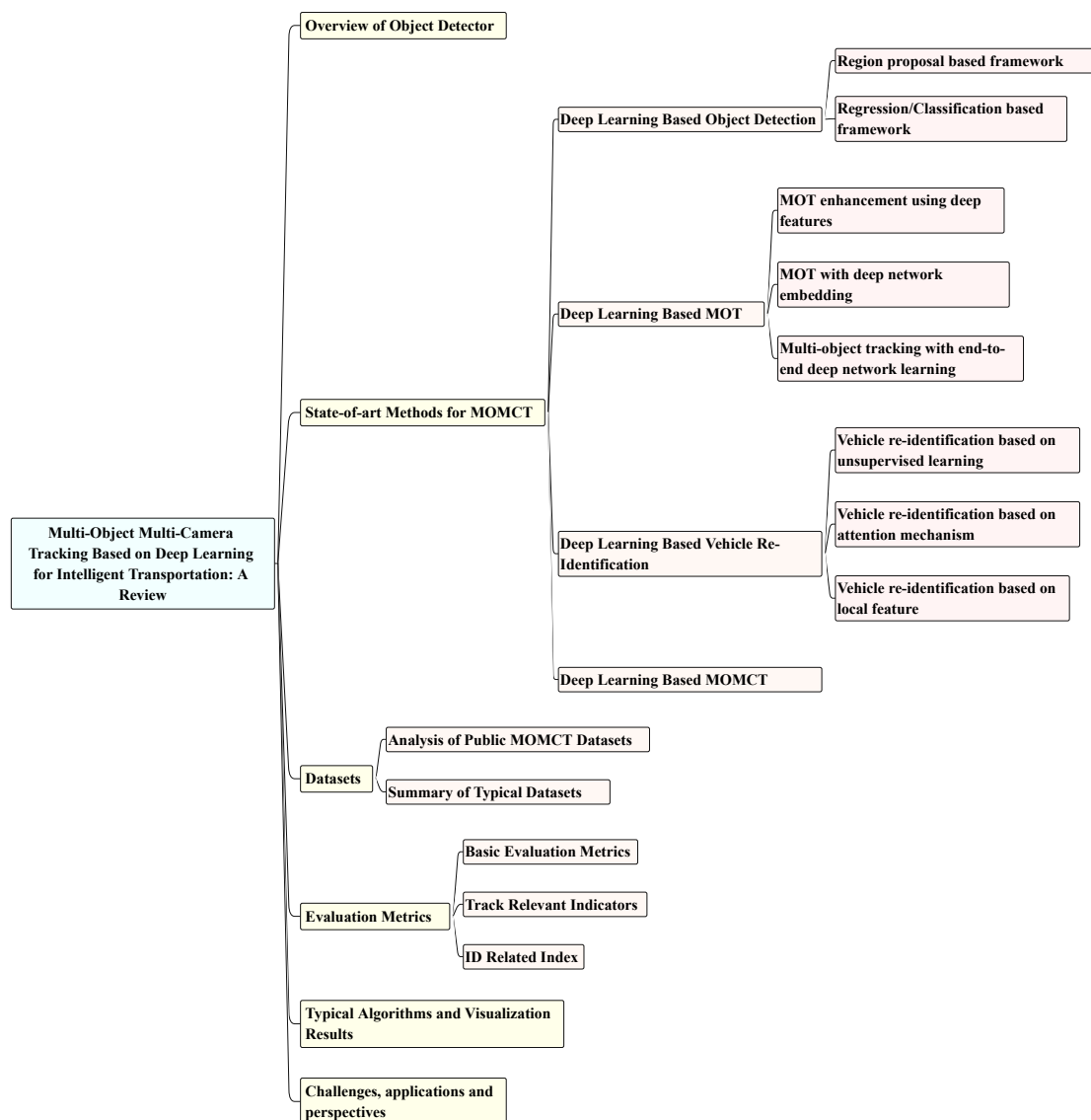


Figure 2. Hierarchical structure of the MOMCT based on deep learning.

Section 2 presents a typical baseline of MOMCT; Section 3 describes the classification of MOMCT-related algorithms based on deep learning; Section 4 describes the main datasets based on the MOMCT task; Section 5 describes in detail the evaluation criteria for MOMCT tasks; Section 6 shows typical algorithms and visualisation results for MOMCT; Section 7 details future challenges, practical application scenarios and future directions for MOMCT tasks; Finally, Section 8 concludes our work.

2. Overview of Object Detector

Object detection consists of two sub-tasks: localisation and classification. Localisation aims to determine the position of the object object in the video or image, while classification refers to the assignment of categories to the detected object objects (e.g., “vehicle”, “pedestrian”, “house”, etc.). Figure 3 illustrates a detailed classification of deep learning-based object detectors. The classification and role of these object detectors will be discussed in this section.

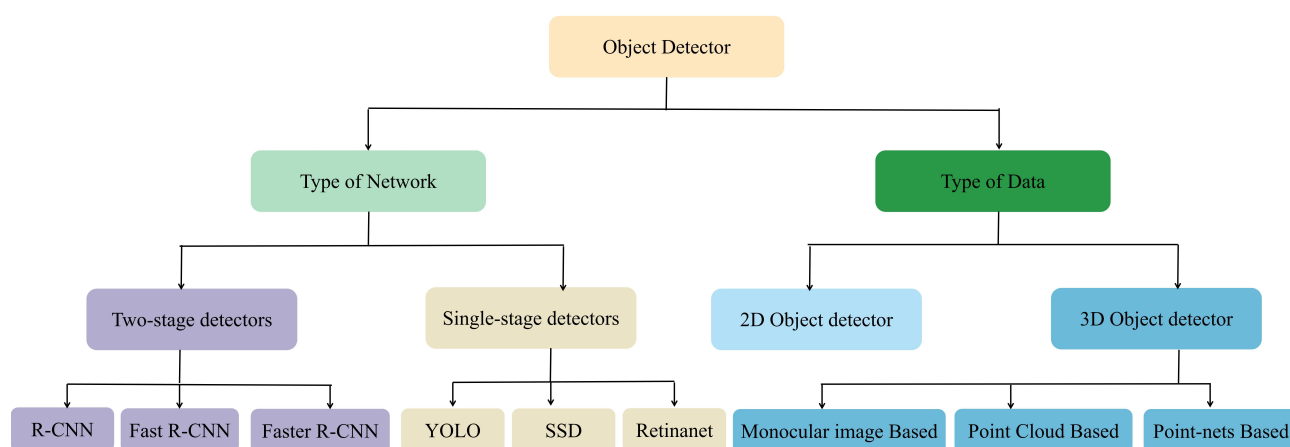


Figure 3. Classification of object detectors.

2.1. Two-Stage vs. Single-Stage Object Detectors

In the field of object detection, the secondary object detector, such as R-CNN [21], Fast R-CNN [22] and Faster R-CNN [23], consists of two processes: candidate regions and object classification. In the stage of candidate region, the object detector selects regions of interest (ROIs) in input image contained object objects. In the stage of object classification, the most possible ROI is selected, other ROI is discarded, and the selected object is classified [24]. In contrast to two-stage detectors, single-stage object detectors create bounding boxes during object detection and perform classification operations on the detected objects. The advantage of single-stage detectors is that they are faster than two-stage detectors, but the disadvantage is that they are less accurate in comparison. Popular single-stage detectors include YOLO, SSD, and RetinaNet.

Figure 4 illustrates the differences between the two types of object detectors. The evaluation metrics for both object detectors are generally evaluated using IoU and mAP. Among them, R-CNN is one of the first object detectors based on deep learning, and ROI is obtained by efficient and simple selective search algorithm. Fast R-CNN is an improvement of R-CNN, which solves the problems of low detection accuracy and slow network reasoning. Fast R-CNN uses convolution during training neural networks to detect the input image and generate ROI projections on feature maps. The ROI is then combined with the feature map for prediction. Fast R-CNN differs from R-CNN in that Fast R-CNN processes the feature map directly with the input image during the detection stage [25]. Faster R-CNN uses a separate detection network, an approach similar to Fast R-CNN, which combines the ROI directly with the ROI pooling layer and feature maps in combination with prediction [26].

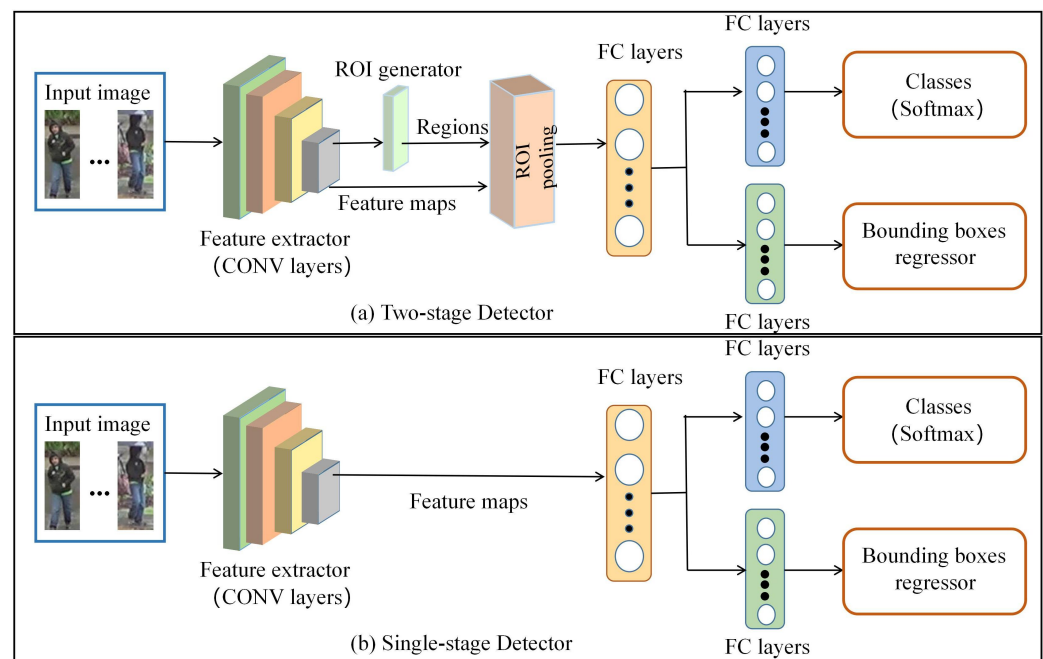


Figure 4. Two-stage vs. Single-stage object detector diagram.

Single-stage object detectors are faster than two-stage object detectors because they predict the input object once. YOLOv1 [27] was the first YOLO variant that learned the salient features of objects and could detect them at a faster speed. In 2016, YOLOv2 [28] creates bounding boxes by anchoring them and adds a high-resolution classifier as well as batch normalization. In 2018, Redmon et al. [29] proposed YOLOv3, which consists of a 53-layer backbone network, an independent logical classifier and cross entropy loss to predict bounding boxes and smaller objects. Single-detector SSD model [30] has good inference performance for real-time applications, because it builds object grids in images to generate feature maps. SSD shares features when performing localization and classification tasks on input images. The YOLO model is superior to the SSD in terms of speed but inferior to the SSD in terms of accuracy. Although the SSD and YOLO models have decent inference speeds, there is a class imbalance problem in detecting small objects. To solve this problem, RetinaNet [31] focuses on loss function using a separate network to solve bounding box regression and classification. The performance of each model is summarised in Table 1.

2.2. 2D vs. 3D Object Detectors

For object detection, 2D image data is usually obtained by a 2D object detector. In [32], data from radar and camera are fused by learning sensor detection methods. The depth information of objects can play a key role in predicting the position, size and shape of the objects, while 2D object detection can only obtain information in the 2D plane.

Data from radar or laser can be applied to 3D object detectors [33]. These object detectors can use methods such as frustum pointnets [34] and point clouds [35] to predict objects in real-time. In compensating for the loss of object information, some networks often use 2D to 3D augmentation, as it is very expensive and complex to calculate directly using 3D data. The 2D bounding boxes in the dot network are obtained in 2D images and these boxes work well in generating ROIs for 3D object detection, effectively reducing the search effort [36].

With the booming development of deep learning, researchers are increasingly interested in 3D object detection. Complex-YOLO [37] uses the Euler Region Proposal Network (E-RPN) based on YOLOv2 to obtain 3D candidate regions. It achieved 3D object detection and background semantic segmentation by random finite sets (RFS). Wen et al. [38] proposed a lightweight 3D object detection model, which consists of three modules: (1) a point transformation module, which extracts point features from RGB images achieved by the original point cloud; (2) voxelization, which combines the acquired voxel grid with the 3D point cloud to generate a many-to-one mapping; (3) point fusion module, which fuses extracted features for output detection. The performance of these models is summarised in Table 1.

Table 1. 2D and 3D object detector models and their performance.

Name	Year	Type	Dataset	mAP	Inference Rate (FPS)
YOLOv1 [27]	2016	2D	Pascal VOC	63.4%	45
YOLOv2 [28]	2016		Pascal VOC	78.6%	67
YOLOv3 [29]	2018		COCO	44.3%	95.2
YOLOv4 [39]	2020		COCO	65.7%	62
YOLOv5 [40]	2021		COCO	56.4%	140
YOLOX [41]	2021		COCO	51.2%	57.8
YOLOR [42]	2021		COCO	74.3%	30
R-CNN [21]	2014		Pascal VOC	66%	0.02
Fast R-CNN [22]	2015		Pascal VOC	68.8%	0.5
Faster R-CNN [23]	2016		COCO	78.9%	7
SSD [30]	2016		Pascal VOC	74.3%	59
RetinaNet [31]	2018		COCO	61.1%	90
Complex-YOLO [37]	2018	3D	KITTI	64%	50.4
Complexer-YOLO [37]	2019		KITTI	49.44%	100
Wen et al. [38]	2021		KITTI	73.76%	17.8

3. State-of-Art Methods for MOMCT

Unlike MOT, the camera network consisting of multiple cameras has a much broader view and application prospect than a single camera. This technology specifically contains four key technologies, including object detection, single-camera MOT, vehicle re-identification and multi-camera object tracking association. This section reviews the current state of development in detail and analyses the problems and shortcomings.

3.1. Deep Learning Based Object Detection

Object detection is the key part of MOMCT task. The object detector mentioned in Section 2 can be combined with scene classifier, which can learn rich semantic information in images and mine more advanced and deeper features. Frameworks for object detection methods fall into two main types: One is the traditional object detection pipeline, which first generates regional suggestions in the image, and then it classifies each suggestion into different object categories. The other is to treat object detection as a classification or regression problem and then use a unified detection framework to directly derive the final result (location and category). In this section, the object detection task is discussed in terms of the above two types.

3.1.1. Region Proposal Based Framework

The region-proposal based framework has two-step stage, similar to the now prevalent attention mechanism, which first scans the entire image and then focuses on the region of interest. The most prominent of the prior related work [43–45] is Overfeat [43]. This model uses CNNs in a sliding window approach to obtain confidence in the underlying object class, then predicts the bounding box from the location of the topmost feature map.

(1) R-CNN

R-CNN achieved an average accuracy of 53.3% on Pascal VOC 2012. It can generate high-quality candidate bounding boxes and then use depth architecture to extract advanced features from images. The main process can be divided into the following three stages.

Region proposal generation. The R-CNN combines saliency cues and bottom-up grouping with selective search methods [46]. It is able to generate candidate frames of arbitrary size accurately and quickly, thus reducing the search space in object detection [47,48]. On each image, the R-CNN can provide approximately 2000 region suggestions using the selective search method. Combined with R-CNN, Xie et al. [49] proposed region-oriented proposal networks to generate good region-oriented proposals for identifying them in a cost-free manner. Hong et al. [50] proposed a sparse R-CNN incorporating the Hungarian algorithm to assign learning suggestion frames one-to-one for each positive sample. It generates better features and initial suggestion frames for the training phase.

CNN based deep feature extraction. Using the CNN framework, each region is proposed to be tuned to a fixed resolution as the final representation [51]. Due to the remarkable hierarchical structure and expressive power of the neural network, a semantic, robust and high-level feature representation of each region proposal can be obtained. Wang et al. [52] proposed a two-stage detection method for dynamic R-CNNs, using a self-calibrating convolutional module in a convolutional network to extract rich object features. Alsharekh et al. [53] used a deep R-CNN architecture to extract features from the data.

Classification and localization. In multiple category pre-training, a multi-category specific linear SVM [54] is used to score different region proposals on a set of negative background regions and positive regions, respectively. Then, within the labelled area, adjustments are made using bounding box regression and filtered using greedy non-maximum suppression (NMS) to produce a final bounding box containing the object location. Zhang et al. [55] proposed a point-to-point regression grouping R-CNN to predict a reasonable bounding box for each point annotation in the image.

(2) R-FCN

Although the Faster R-CNN is an order of magnitude faster than the Fast R-CNN, the computation after the RoI merge layer cannot be shared. Therefore, Dai et al. [56] presented a fully convolutional R-FCN detector to build a shared RoI subnet. However, this simple design has poor detection accuracy, presumably because the deeper network is more sensitive to category semantics. Based on this phenomenon, Vijaya Kumar et al. [57] used the R-FCN ensemble layer to extract predicted scores for each region in the image. This method facilitates the computation of shared regions of interest and improves the detection accuracy of the network. Zhang et al. [58] constructed a region-based full convolution network (R-FCN) model. The model realizes the accurate classification of targets, and improves the recognition accuracy of the model by extracting fine-grained features from images.

3.1.2. Regression/Classification Based Framework

The region-proposal based framework consists of several interrelated phases such as region proposal generation, CNN feature extraction, classification and bounding box regression, which are usually trained separately. Even in R-CNNs with faster end-to-end modules, alternative training is required to obtain the convolution parameters between the detection network and the RPN. In this section, we present two important applications of the framework: YOLO [27] and SSD [30].

(1) YOLO

Redmon et al. [27] proposed YOLO, a method for predicting confidence in bounding boxes and object classes from the topmost feature map. Based on this, Roy et al. [59] proposed a deep learning-based detection model, WilDect-YOLO, in which a residual block was added to the depth space feature extraction. Karaman et al. [60] proposed an artificial bee colony (ABC) optimisation algorithm based on the YOLOv5, which improves the sensitivity of the system for real-time detection by pairing hyperparameters and optimal

activation functions. Xue et al. [61] presented an improved YOLOv5, which improves the feature extraction capability by adding self-attentive convolution and convolutional block-attentive modules. Mittal et al. [62] proposed a hybrid model combined YOLO and Fast R-CNN. It employs migration learning methods to reduce class imbalance problems and enhances the image quality through a sharpening process.

(2) SSD

Due to the strong spatial constraints of bounding boxes, YOLO lacks robustness in detecting multiple small objects. It also generates relatively coarse object features during the downsampling operation. To address these issues, Liu et al. [30] proposed SSD based on anchor points employed in RPN [63] and multiscale representation [64]. Contrasted to the fixed grid used in YOLO, SSD can use anchor boxes of different scales to discretize the bounding boxes of a given feature map. The network handles objects with different sizes by fusing the predictions of feature maps in various resolutions. On this basis, Jia et al. [65] used the positive and negative functions of SSD network to solve the shortcoming of insufficient sensitivity for small object detection. Chen et al. [66] enhanced the SSD using MobileNetv2 and the attention mechanism to improve the performance of the algorithm. Gao et al. [67] proposed the R-SSD which based on SSD and ResNet to improve the feature extraction quality of the algorithm. Ma et al. [68] proposed the anchorless 3D object detection model CG-SSD, which mines deeper features through a convolutional backbone network consisting of sparse convolutional layers and residual layers. Cheng et al. [69] used a hybrid attention mechanism in SSD to improve the accuracy of object detection and further combined it with a focal loss function to improve robustness.

3.2. Deep Learning Based MOT

MOT needs to be completed after object detection. In this section, we will roughly classify deep learning-based MOT approaches into three categories based on the different tracking framework: (i) MOT using deep network feature enhancement. Deep neural networks are used to extract semantic features for the task of interest and replace the previous traditional manual features. (ii) MOT with deep network embedding. Deep neural networks can classify the different object trajectories acquired and construct deep classifiers to detect whether they belong to the same object. (iii) MOT with end-to-end deep neural network learning. In the general case, there will be intertwined modules in motion object tracking and the MOT results can be obtained directly using deep networks.

3.2.1. MOT Enhancement Using Deep Features

Deep features extracted by deep neural networks are rich in semantic information and are clearly differentiated between different classes. These deep features not only improve the performance of MOT, but are also effective for tasks such as image segmentation and object detection [70].

Similar to object detection using CNN to extract object features [71], AlexNet [72] is mainly used to extract depth features in the multiple hypothesis tracking (MHT) framework. The MHT tracking framework creates a hypothesis tree and uses different associated hypotheses. Kim et al. [73] improved the MHT method with appearance features using regularized least squares. This method reduces the dimensionality of the extracted appearance depth features. Wang et al. [74] proposed a circular tracking unit to model the object by acquiring long-term information. Wojke et al. [75] used the wide residual network (WRN) to extract the depth features in the image to increase the discrimination ability. These features were also used to calculate the minimum cosine distance between tracking and detection. As shown in Figure 5, the whole tracking framework utilizes a cascade matching process. The tracking method has depth features from the WRN that improve the real-time speed of the model while maintaining competitive performance.

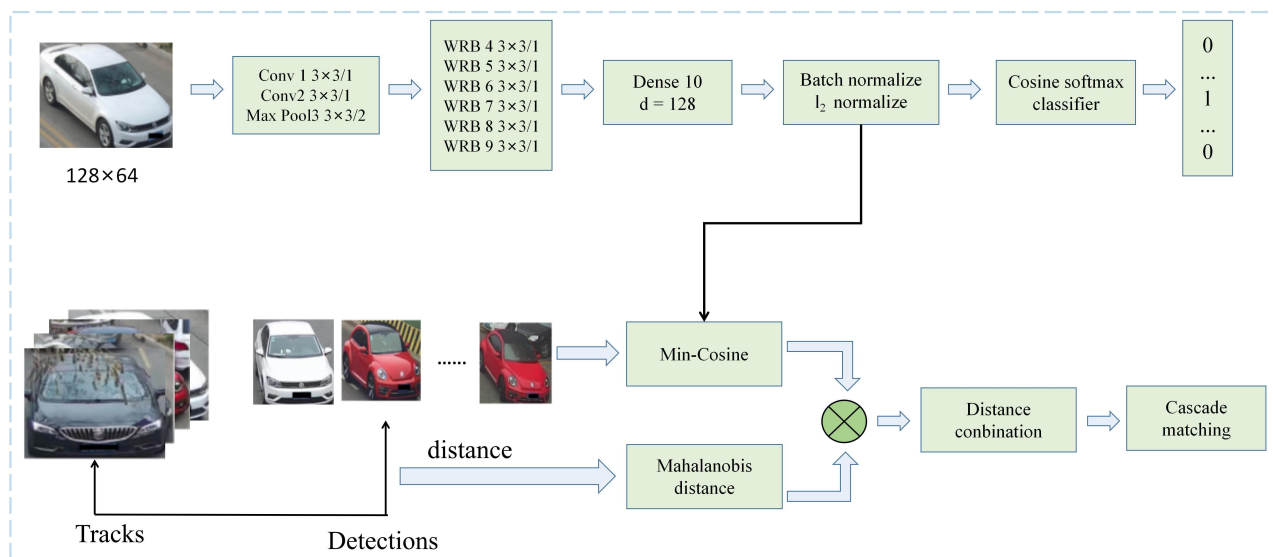


Figure 5. The framework of depth SORT [75].

The goal of MOT feature learning is to evaluate the similarity between tracking target features, and the CNN structure with two branches is well suited for extracting matching features of moving objects. Leal-Taixé et al. [76] used gradient advancing algorithm to fuse motion and depth features, so that the tracking problem was transformed into linear programming. Zhang et al. [77] proposed a framework that combines filtered tracking and conjoined object tracking. The framework combines artificial features with depth features to improve the performance and robustness of the tracker. Su et al. [78] used anchor-less networks to predict the position of objects in the search element domain, improving the correlation between search frames and template frames.

3.2.2. MOT with Deep Network Embedding

Compared to the enhanced tracking method of depth feature, it is more effective to combine deep neural networks and embedded designs as key components of the tracking framework, where samples of tracking data are used in the training process. Based on the task of network learning, we broadly classify MOT methods for deep network embedding into three types: discriminative deep network learning (DN-MOT), deep measurement learning (DM-MOT), and generative deep network learning (GN-MOT).

DN-MOT: In the DN-MOT method, object trackers optimise discriminative models and search the best locations in the next frames according to the models. Due to deep networks are widely used for discriminative tasks, it is easy to extend discriminative deep network models for tracking tasks. For example, Chen et al. [79] proposed an object particle filtering framework, specifically constructing models of VGG-16 [80] and Faster R-CNN [23] as object classifiers to track each object. Similar to [79], Chu et al. [81] used an object-specific tracker to construct the MOT framework and used an updated classifier to find the best candidate to complete the tracking, as shown in Figure 6. Firstly, the captured video frames are fed into a shared CNN layer to generate feature maps, which are subsequently fed into a RoI pool to generate candidate features. At the same time, positive, negative and historical samples are selected from the tracked samples. These samples are used to calculate the classification score and weight loss of candidate features. Secondly, the features in the RoI pool are passed through a fully-connected convolutional layer to generate a visibility map, and then spatial attention is extracted from the visibility map. Finally, the state of candidate features is extracted for object tracking. In addition, Liu et al. [82] provided a multivariate polynomial kinematic forward solution algorithm that effectively improves the accuracy and real-time performance of the model.

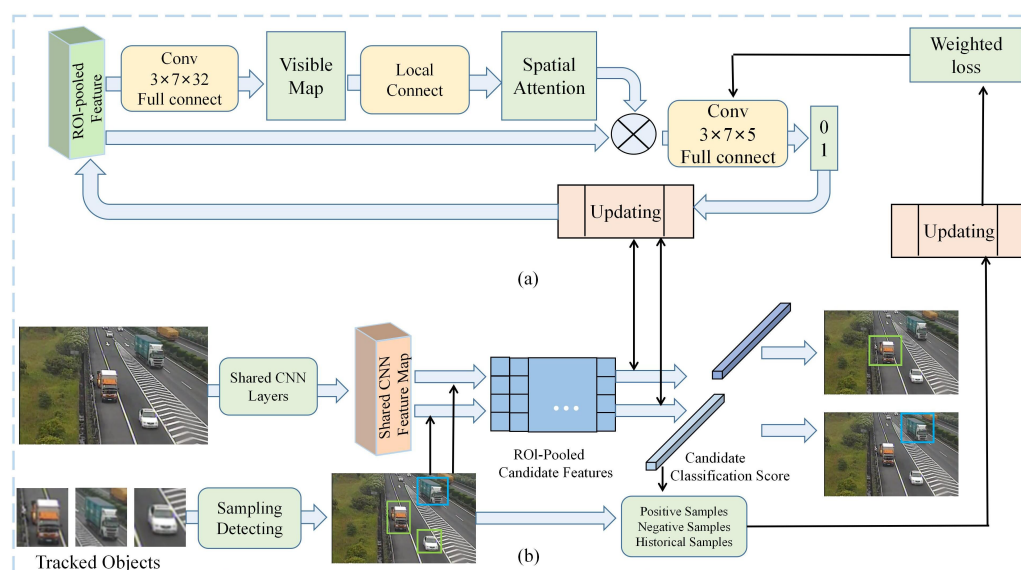


Figure 6. Framework of STAM-MOT [81]. In this framework, (a) a deep neural network-based spatial attention and object-specific classifier, and (b) a sampling-based best candidate classifier.

DM-MOT: It involves extracting image features and mapping them to a high-dimensional space. This method aims to calculate the similarity between different objects based on the feature information. It can be regarded as an image block verification problem [83]. Similar to pedestrian re-identification [84] and face recognition [85], it is important to learn accurate distance metrics and similarity models in such problems. Therefore, it is desirable to use suitable depth metric learning networks [86] for DM-MOT. Son et al. [87] used multiple image blocks as Siamese network inputs to achieve precise localisation by extracting appearance and motion features. Xiang et al. [88] designed a CNN network based on triadic loss to acquire the appearance features of the object and through, which the distance metric between the detector and the tracker was learned and calculated. Unlike [87], Cheng et al. [89] employed the appearance and motion features of the object as input for DM-MOT and used triple loss to optimize the model. In general, it usually applies the Hungarian algorithm [90] to solve the distance metric cost incurred by the detectors and trackers during the tracking process.

GN-MOT: It aims to generate the object model by learning its shape, motion and other characteristics and then using this model to track its trajectory. Some works have used deep generative learning to improve the performance of MOT. Fang et al. [91] presented a recursive auto-regressive network (RAN) model to improve the performance by modeling the motion features. Fernando et al. [92] predicted object trajectories by using an LSTM-based generative model. It associates the object tracker with the GAN model in the prediction module to enable the tracking of new objects.

3.2.3. MOT with End-to-End Deep Network Learning

It is difficult to apply a single model for learning multiple key modules in an MOT task such as target detection, target matching, trajectory tracking etc. Recently, researchers have presented many end-to-end learning methods to achieve this goal. Inspired by Posner and Ondruska [93], Milan et al. [94] modeled these procedures on the basis of RNNs. As shown in Figure 7, the inputs to the network are the matching matrix, state, and presence probability of the object, while the updated result, predicted state, and new presence probability are used as outputs, which determine whether the object should be terminated. In addition, an LSTM-based network was designed for computing the matching matrix, while modeling the matching process between the current observation and the object state to train the RNN in an end-to-end manner. On this basis, Sadeghian et al. [95] employed a hierarchical RNN structured network to obtain interaction, appearance and motion features of objects. The network uses an end-to-end approach to train matching classifiers,

which improves the probability of matching between objects and trackers. With a time-synchronous stability mechanism, Li et al. [96] enabled all parts of the controller's agent states to reach consensus simultaneously, effectively improving the performance of the model. Moreover, Kim et al. [97] proposed an end-to-end training method to improve the performance by using bilinear LSTM module.

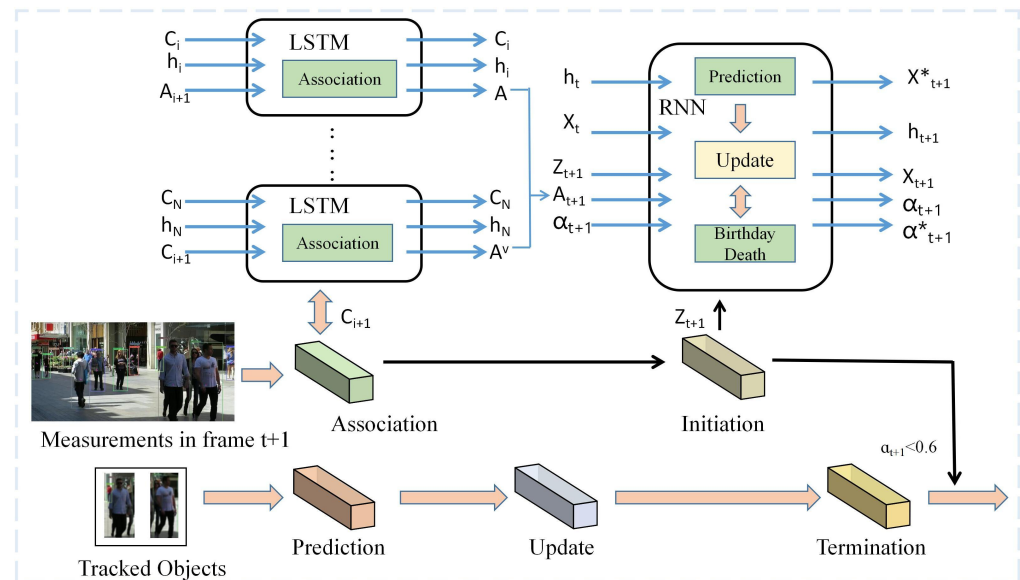


Figure 7. Framework of RNN-LSTM tracking [94]. An LSTM-based network is constructed in this framework for deriving the best association between object and detection, and an RNN-based network for updating states, learning predictions and termination probabilities.

3.3. Deep-Learning-Based Vehicle Re-Identification

Following single camera multi-objective tracking, a deep learning based re-identification method is used to extract embedded features of each trajectory for vehicle re-identification. This section will focus on unsupervised learning-based methods, attention mechanism-based methods and local feature-based methods.

3.3.1. Vehicle Re-Identification Based on Unsupervised Learning

Compared to supervised techniques, unsupervised learning aims to make inferences directly from unlabeled input data, which addresses the limited generalisation capability of the model and the high cost of manually labelled data [98]. This technology has been widely used in vehicle re-ID task. Deng et al. [99] presented a cross-domain adaptive unsupervised method from image to image. This method maintains similarity by combining neural network and improves the recognition accuracy of the model. Wang et al. [100] used an identity-based joint attribute learning method to improve the efficiency by the key attributes and semantic information.

In recent years, unsupervised methods have been widely used in vehicle re-identification. Shen et al. [101] used clustering features to simulate global and local features for improving the accuracy of unsupervised vehicle re-ID. Zhu et al. [102] trained convolutional neural networks with a deep feature learning module to improve the model's ability in feature differentiation. Wang et al. [103] proposed a new contrast learning framework that uses reliable discrete sample clustering to build a memory dictionary for object re-identification. Gao et al. [104] used an unsupervised framework based on data synthesis. By adjusting the target and source domain to adapt to the pre-training model, the domain generalization ability of the re-recognition model was improved.

Generated Antagonistic Network (GAN) [105] is widely used in unsupervised learning. A simulation of the GAN framework is shown in Figure 8. The framework consists of a data discriminator and a generator. The generator obtains the synthetic data through transformation after obtaining the random variables. The discriminator receives the data from the generator and judges the data, and finally reaches a balanced state. Zhou et al. [106] presented a GAN-siame network to solve the unsupervised V-reID cross-domain problem. The algorithm learns the distance measurement between two domains by connecting the network, which improves the performance of model matching.

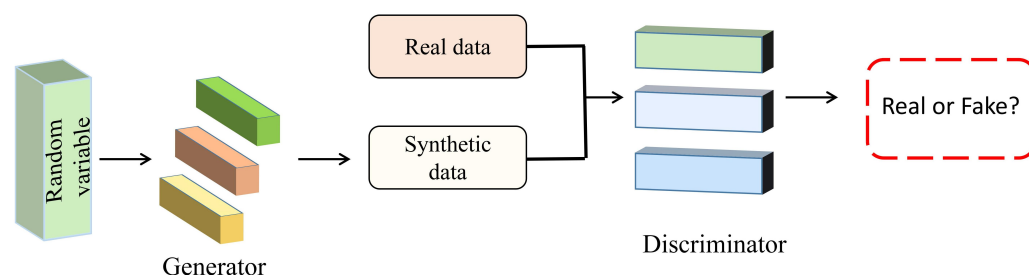


Figure 8. GAN architecture simulation diagram.

3.3.2. Vehicle Re-Identification Based on Attention Mechanism

Attention mechanism mainly focuses on selective actions/things related to tasks and ignores other irrelevant actions/things. Researchers are working on designing an effective attention-based neural network for vision-related applications such as fine-grained image recognition [107,108], image classification [109,110], image captioning [111,112], and vehicle re-identification [113]. The process of vehicle re-identification based on spatio-temporal attention is shown in Figure 9. Trajectory features are extracted for re-ID by the spatial attention mechanism, then the features are weight ranked by the temporal attention mechanism, and finally the key features of objects are output according to the ranking.

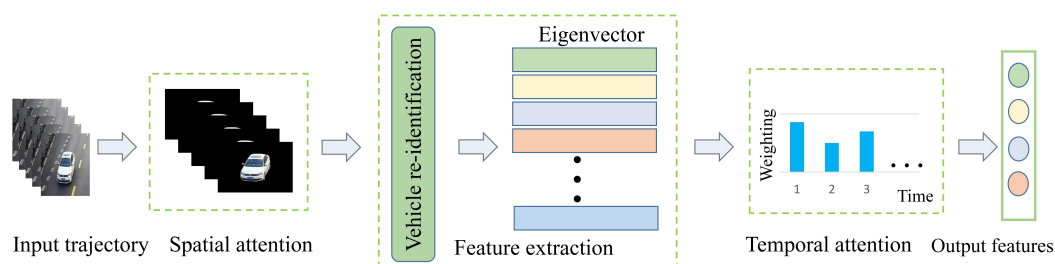


Figure 9. Vehicle Re-identification Based on Spatio-temporal Attention.

These approaches typically follow a strategy of integrating hard partial selection subnets or soft mask branches into deep networks. For example, Zhu et al. [113] added self-attentive models to each branch of the CNN network for fine-grained recognition of vehicles. To reduce the influence of noise in the image, Lian et al. [114] used the attention network based on transformer to extract the global features of vehicle re-ID. Jiang et al. [115] studied a global reference attention network. By mining distinguishing features, the difficulty of distinguishing vehicles with similar appearance is solved. Tian et al. [116] proposed an adaptive attention network. The network captures the global structural information of the vehicle through a global relational attention module to improve the accuracy of re-ID. Li et al. [117] investigated a CAM network with a contrast attention module. It enhanced the recognition ability of the re-ID model by refining the local features. Song et al. [118] introduced the global attention mechanism based on two-branch network. It trains the network by combining global and local features to improve the performance of vehicle recognition. Li et al. [119] presented a model with region attention and orthogonal view generation. The process of re-identification is simplified by extracting

distinguishable regional features to differentiate vehicles. Tang et al. [120] studied an attention network for extracting multi-scale features, which reduced the difficulty of vehicle re-ID task. Liu et al. [121] used a multiple soft attention network to extract robust features. Shen et al. [122] presented a graphical interaction converter to extract differentiated local features for the robustness of re-ID model.

3.3.3. Vehicle Re-Identification Based on Local Feature

Early research on vehicle re-ID focused on the global features of images. After encountering the bottleneck of accuracy, many studies began to pay attention to local features, because the differences between similar vehicles are mainly reflected in local areas. At present, it is common to extract local features by means of key point localisation and region segmentation. Wang et al. [123] segmented the image vertically and horizontally to extract features, which improves the accuracy of vehicle re-recognition. Rong et al. [124] fuse local-global features to obtain more vehicle information and enhance the learning ability of vehicle recognition. Yang et al. [125] studied the two-branch network based on pyramid feature learning. It solves the problem of learning and recognising model information by learning local and global features of the vehicle. Fu et al. [126] utilized local attention to facilitate the learning of local attentional features for vehicle re-ID. Liu et al. [127] designed a vehicle information module. The module improves the ability to recognize similar vehicles under the same camera.

3.4. Deep Learning Based Multi-Object Multi-Camera Tracking

Re-ID is carried out through the information of cameras, and the cross-regional tracking of moving objects is completed through the temporal and spatial correlation of multi-camera trajectories. The MOMCT with trajectory to object method is as shown in Figure 10. Input the object information detected in different cameras into the camera network, and then accurately match the object trajectories, and create a complete global cross-camera trajectory for each object.

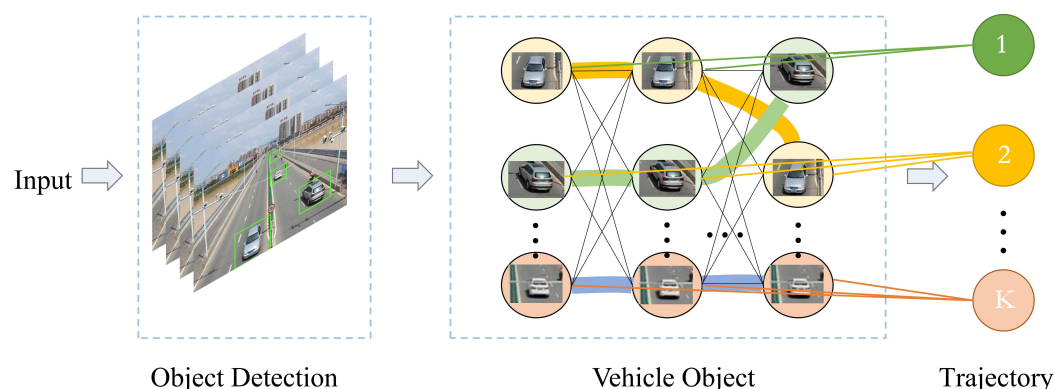


Figure 10. Schematic diagram of algorithm from trajectory to object.

Vehicle trajectory matching is the key part of MOMCT. Due to variations in lighting and viewing angles, it is susceptible to disruptions in vehicle tracking trajectories while vehicles are obscured and hence continuous vehicle tracking cannot be accomplished. For such problems, Hsu et al. [128] proposed a camera linkage model based on trajectory. By extracting the appearance and topological features of different cameras, the accuracy of vehicle trajectory matching is improved. Hsu et al. [129] provided a reliable framework for vehicle MOMCT using a hierarchical clustering algorithm. Li et al. [130] simplified the process of object trajectory matching in overlapping space by dynamically coding visual features. Liu et al. [131] used Markov decision to model the vehicle trajectory, which improves the accuracy of trajectory matching. Zhao et al. [132] presented a channel

estimation method which uses sensing, communication and control technologies to obtain the information needed for trajectory generation.

During tracking, it is easy to mistake tracks generated by different vehicles as the same ID. To solve this problem, Yang et al. [133] designed the trajectory re-connection technology. By reconnecting the segmented trajectories, an accurate vehicle trajectory is generated. Li et al. [134] developed a MOMCT vehicle tracking system which eliminates unreliable trajectories by assessing cross-view matching of vehicle trajectories. Liang et al. [135] used Kalman filter to predict the motion of the object, which improved the analysis and matching ability of the model. To match the local trajectory of the same object in different cameras, He et al. [136] employed the spatio-temporal attention mechanism to generate the vehicle trajectory representation, which improves the matching success rate of the object allocation algorithm. Tran et al. [137] presented a spatially constrained framework to improve the robustness of the model by using cross-awareness of the tracker.

4. Datasets

Datasets play an important role in the MOMCT task, not only to strategise and compare the performance of various algorithms, but also to help solve complex and challenging problems in the field. This section describes the main datasets in the MOMCT task.

4.1. Analysis of Public MOMCT Datasets

4.1.1. BDD100K Dataset

The BDD100K [138] dataset contains 100,000 videos with IMU/GPS information captured by mobile phones, each video lasts approximately 40 s at 30 frames per second and keyframes are extracted and annotated at the 10th second, mainly in terms of road object boundaries, driveable areas, and lane markings. The dataset annotates the boundary boxes for common objects on the road in 100,000 keyframe images to understand the location and distribution of the objects. The different traffic scenes included in the BDD100K dataset are shown in Figure 11.

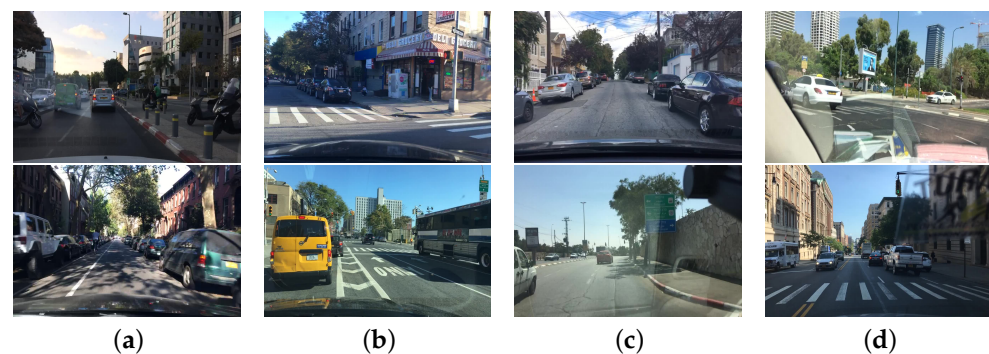


Figure 11. Traffic scene images of BDD100K data set, where (a–d) are images of traffic captured in different street scenes.

4.1.2. VehicleX Dataset

The VehicleX dataset [139] is a dataset synthesized from 3D models of various vehicles. It is taken from real-world scenes and used to synthesize images with a total of 1362 vehicles and 192,150 images. In addition, colour and type labels were also annotated. The traffic scene images of vehicleX dataset is shown in Figure 12.

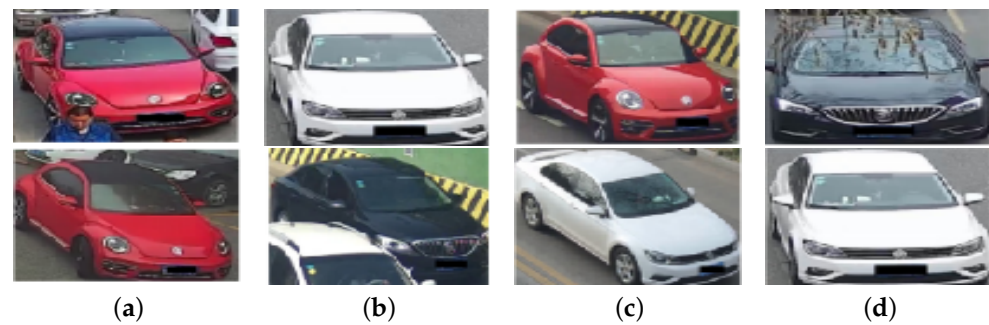


Figure 12. Traffic scene images of UA-DETRAC data set, where (a–d) are images of different vehicles in a traffic scene.

4.1.3. UA-DETRAC Dataset

The UA-DETRAC data set [140] contains 10 h of video from 24 different locations, including more than 140,000 frames of data. Up to 8250 vehicle objects in the scene are manually marked with more than 1.21 million object bounding boxes containing labels. The vehicle categories in the data set are cars, buses and trucks, and there are also four weather types: cloudy, night, sunny and rainy. As shown in Figure 13, UA-DETRAC dataset images with different congestion situations in traffic scenes.

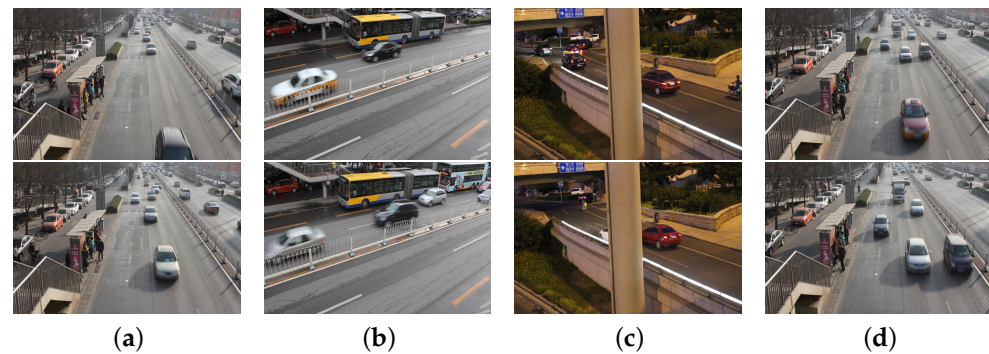


Figure 13. Traffic scene images of UA-DETRAC data set. (a,b,d) show the traffic scenes in daytime conditions. (c) shows the traffic scenes on the viaduct under night conditions.

4.1.4. KITTI Dataset

The KITTI dataset [141] was used to evaluate the performance of computer vision techniques such as stereo imagery, optical flow, visual ranging, 3D object detection, and 3D tracking in an in-vehicle environment. It contains real image data collected from urban, and rural and motorway scenes with up to 15 vehicles and 30 pedestrians per image, as well as various levels of occlusion and truncation. The traffic scene images of KITTI dataset is shown in Figure 14.

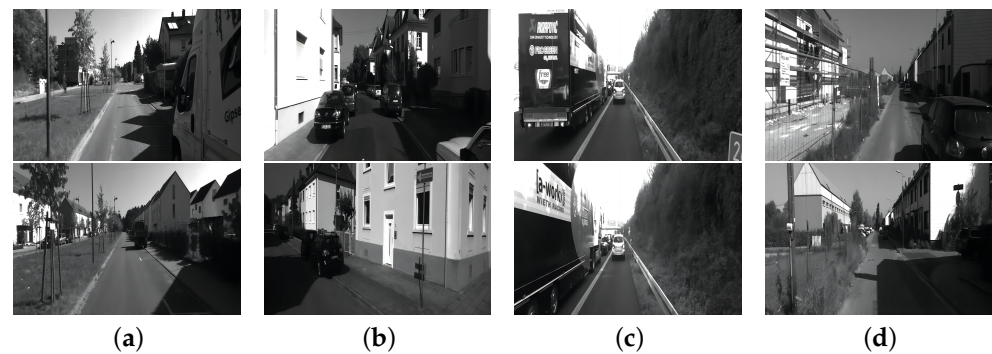


Figure 14. Traffic scene images of KITTI data set, where (a–d) are images of different urban and rural roads.

4.1.5. Nuscenes Dataset

The nuScenes dataset [142] is a shared large dataset for autonomous driving. The dataset has 1000 driving scenarios, each with 20 s of video, for a total of approximately 15 h of driving data. The scenarios were selected with due consideration for diverse driving maneuvers, traffic conditions, and accidents. The traffic scene images of nuscenes dataset is shown in Figure 15.



Figure 15. Traffic scene images of nuscenes data set, where (a–d) are images of intersections, office buildings and residences on different streets.

4.2. Summary of Typical Datasets

With the detailed description of the data related to the MOMCT dataset above, this paper summarises and tabulates the vehicle dataset in recent years, as shown in Table 2.

Table 2. Popular dataset for MOMCT.

Dataset	Year	Total Images	Categories	Image Size	Objects of Image	Size	Highlights
OpenData	2017	10,000	10	400 × 424	Varied	16 G	A great variety.
Stanford Cars	2013	16,185	5	720 × 540	3	10 G	Automobile model verification.
CompCars	2015	136,726	5	540 × 540	4	18 G	Fine-grained classification.
ImageNet	2009	14,197,122	21,841	500 × 400	1.5	138 G	Image classification, detection and location.
PASCAL VOC	2009	11,540	20	470 × 380	2.4	8 G	One of the mainstream data sets of computer vision.
MS COCO	2015	328,000+	91	640 × 480	7.3	18 G	Very high industry status and huge data set.
Open image	2020	9 million+	6000+	Varied	8.3	500 G	Very diverse.

Table 2. Cont.

Dataset	Year	Total Images	Categories	Image Size	Objects of Image	Size	Highlights
KITTI	2012	500+	5	1240 × 376	1.7	180 G	Evaluate vehicle performance.
BD100K	2018	100,000	10	1280 × 720	2.4	7 G	One of the largest driving data sets.
UA-DETRAC	2020	140,000	8	960 × 540	2.3	14.5 G	Challenging data set.
ILSVRC	2012	170,000+	1000	1280 × 720	Varied	16 G	The most popular machine vision competition.
vehiclex	2020	192,150	10	960 × 540	Varied	16 G	Accurate data.
CityFlow	2019	229,680	6	1080 × 540	Varied	8 G	Large-scale.
VehicleID	2019	221,763	11	840 × 840	Varied	7 G	Large-scale.

5. Evaluation Metrics

Certain evaluation metrics can measure the performance of cross-camera MOT tasks. It plays a key role in the evaluation analysis and selection of algorithms as a criterion for evaluating the performance of MOMCT algorithms. In this section, the MOMCT algorithm performance evaluation metrics are described in detail.

5.1. Basic Evaluation Metrics

- (1) TP: True Positive is a positive sample that is predicted to be positive by the model, which can be referred to as the percentage of correct judgments that are positive.
- (2) TN: True Negative is a negative sample that is predicted to be negative by the model and can be referred to as the percentage of correct judgments that are negative.
- (3) FP: False Positive is a negative sample that is predicted to be positive by the model and can be referred to as the false positive rate.
- (4) FN: False Negative refers to positive samples that are predicted to be negative by the model and can be referred to as the under-reporting rate.
- (5) Accuracy: This refers to the weighting of the correct decision by the classifier, and is publicly expressed as.

$$A = \frac{TP + TN}{TP + TN + FP + FN}. \quad (1)$$

- (6) Precision: Its the proportion of true positive samples among the positive examples determined by the classifier, expressed publicly as.

$$P = \frac{TP}{TP + FN}. \quad (2)$$

- (7) Recall: Its the proportion of positive cases correctly determined by the classifier to the total number of positive cases, expressed publicly as.

$$R = \frac{TP}{TP + FN}. \quad (3)$$

5.2. Track Relevant Indicators

- (1) MOTA [143]: MOT Accuracy is a measure of single-camera MOT accuracy and is publicly represented as.

$$\text{MOTA} = 1 - \frac{FN + FP + \Phi}{T}, \quad (4)$$

where FN is the sum of the false negatives of all frames, i.e., assuming that fn_t is the false negative of frame t , then $FN = \sum_t fn_t$. Similarly, $FP = \sum_t fp_t$. T is the sum of the number of real objects in all frames, i.e., assuming that there are g_t objects in frame t , then $T = \sum_t g_t$. Φ is the number of object jumps in all frames, ϕ_t is the number of object jumps in frame t , then $\Phi = \sum_t \phi_t$. In other words, these three items represent

the missing rate, the false positive rate, and the mismatch rate in that order. The closer MOTA is to 1 the better the tracker performance.

- (2) MOTP [143]: MOT accuracy is a measure of single-camera MOT position error, expressed by the formula.

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}, \quad (5)$$

where c_t denotes the number of matches at frame t . For each pair of matches, the matching error d_t^i , represents the distance between the object O_i , and its pairing hypothesis position at frame t .

- (3) MT: Mostly tracked is the number of tracks where the tracked portion is greater than 80%, the larger the value the better.
 (4) ML: Mostly lost is the number of tracks where the lost portion is greater than 80%, the smaller the value the better.
 (5) Frag: The number of jumps is the number of track changes from “tracking” to “not tracking” state.

5.3. ID Related Index

- (1) IDP: Identification Precision is the accuracy of vehicle ID identification in each bounding box. The formula is:

$$\text{IDP} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFP}}, \quad (6)$$

where IDIP and IDFP are the number of true IDs and the number of false positive IDs, respectively.

- (2) IDR: Identification Recall is the recall rate of vehicle ID identification in each bounding box. The formula is:

$$\text{IDR} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFN}}, \quad (7)$$

where IDFN is the negative ID number.

- (3) IDF1: Identification F-Score is the F-value of the vehicle ID identification in each bounding box. The formula is:

$$\text{IDF1} = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFP} + \text{IDFN}}. \quad (8)$$

In general, IDF1 is the first default metric used to evaluate the performance of the tracker. These three metrics can be inferred from any two of them, so it is also possible to show only two of them, although it is preferable that these two include IDF1.

- (4) IDS: The number of ID switches is the number of instantaneous vehicle ID transitions in the tracking track, usually reflecting the stability of the tracking, the smaller the value the better.

6. Typical Algorithms and Visualization Results

6.1. Comparison and Analysis of Algorithms

To describe MOMCT algorithm models more intuitively in Section 3, these algorithm models are listed in Table 3. It not only shows their performance of using different object detectors and tracking methods on BDD100K data set, but also shows their results based on IDP, IDR, IDF1 and other indicators on the data set, and makes the following analysis and comparison of typical algorithms in Section 3.

By comparing these algorithms, we can draw the following summary:

- (1) GCNM: After associating the object trajectory, the algorithm uses the graph convolution network to form the global trajectory. Then on the trajectory level, instant erasure and random horizontal flip are used to expand the data, which enhances the data

robustness of the camera. Finally, the new loss function improves the generalization ability of the model, thus obtaining good performance in data accuracy.

- (2) UWIPL: This method generates a motion track by using its appearance and time information. The system takes ResNet50 as the backbone network and combines Xent loss and Htri loss for training. The tracking accuracy is improved based on road channelization and road condition information, and the applicability of this method in different scenarios is realized.
- (3) ANU: It provides fine-grained features by using road spatio-temporal information and camera topology information. Removing overlapping bounding boxes by non-maximum suppression. At the same time, it also uses the color dithering mechanism to improve the performance of the model.
- (4) BUPT: It utilizes ResNet network as the backbone network and uses random filling and erasing methods to fill data. Then, it trains the framework by combining trajectory consistency loss and clustering loss. Finally, a higher IDF1 is obtained by introducing temporal and spatial clues.
- (5) DyGLIP: It has better lost trajectory recovery and better feature representation during camera overload. By adding correlation regression and attention module in the experiment, the scalability of the model in large-scale data sets is improved.
- (6) Online-MTMC: It solves the MOMCT problem by using the detection-clustering method. The feature pyramid network is used as the backbone network, and the quality of features is improved by Gaussian blur and contrast disturbance mechanism. This method also employs the minimum loss function to optimize the network parameters.
- (7) ELECTRICITY: It applies a cluster loss strategy to remove isolated tracks and synchronise track ID based on the re-identification results. Meanwhile, depth ranking is considered as a tracking model and Adagrad is applied as a loss function to optimise the model, which makes the algorithm suitable for large-scale realistic intelligent traffic scenes.
- (8) NCCU: It adopts vehicle image features and geometric factors for collaborative optimization matching. Then, FBG analysis is used to generate the mask of road region of interest, which effectively solves the problem of finding broken down vehicles on the road.

Table 3. Performance of Typical Algorithms on BDD100K Datasets.

Method	Object Detector	SCT	IDP↑	IDR↑	IDF1↑
GCNM [144]	SSD	TNT	71.95	92.81	81.06
UWIPL [145]	SSD	TNT	70.21	92.61	79.87
ANU [146]	SSD	custom	67.53	81.99	74.06
BUPT [147]	FPN	custom	78.23	63.69	70.22
DyGLIP [148]	Mask-RCNN	DeepSORT	-	-	64.90
Online-MTMC [149]	EfficientDet	Custom	55.15	76.98	64.26
ELECTRICITY [150]	Mask-RCNN	DeepSORT	-	-	53.80
NCCU [151]	FPN	DaSiamRPN	48.91	43.35	45.97

IDP: the accuracy of vehicle ID identification in each bounding box. IDR: the recall rate of vehicle ID identification.

IDF1: the F-value of the vehicle ID identification in each bounding box.

6.2. Visualization Results and Analysis

The visualization results of the listed classical algorithms are shown from Figures 16–19. Through visual analysis, we can get the following results.

- (1) ANU adjusts the thresholds of positive and negative sample pairs by increasing the perception of locality in small scenes. The non-maximum suppression mechanism also removes some of the overlapping bounding boxes and retains those close to the camera, improving the success rate of tracking to the vehicle.
- (2) UWIPL combines camera linking and deep feature re-identification of trajectories, uses the appearance and time information of trajectories for high confidence matching, and uses a greedy algorithm to select the smallest pairwise distance to match the vehicle being tracked, resulting in accurate tracking results in different scenarios.

- (3) ELECTRICITY combines MOMCT strategy and aggregation loss to eliminate the erroneous trajectories. It tracks objects mainly through re-identification, and further improves the robustness of the algorithm through image flipping and random erasure.
- (4) BUPT system combines the loss of trajectory consistency with the loss of clustering, and extracts more obvious features. The cluster loss used in the method improves the tracking accuracy.



Figure 16. Visualization results of ANU tracking system on BDD100K data set. (a) shows the results of vehicle detection and tracking during the day, (b–e) show the results of vehicle detection and tracking at different times of the night for the same video.

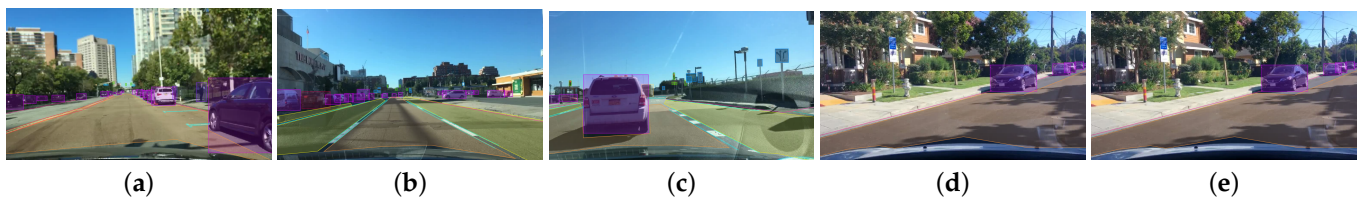


Figure 17. Visualization results of UWIPT tracking network on BDD100K data set, where (a–e) show the results of detecting and tracking vehicles in different scenes of the same video.



Figure 18. Visualization results of ELECTRICITY tracking algorithm on BDD100K data set, where (a–e) are the results of vehicle detection and tracking in the same scene in the same video.

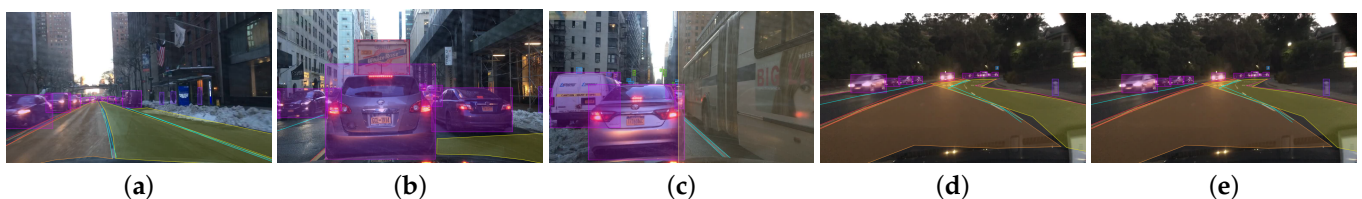


Figure 19. Visualization results of BUPT system on BDD100K data set. (a–c) are urban scenes, (d,e) are countryside scenes.

7. Challenges, Applications and Perspectives

7.1. Challenges and Opportunities

(1) Real-time processing

In the training stage, object detection in the sequences increases the training time. It can better detect new objects by increasing the correlation in the sequences, while discarding irrelevant frames. It may reduce the waiting time in training. It has become an open question that how to establish spatio-temporal relationships between consecutive frames in

order to improve the efficiency of detecting objects. Although some recent research works have begun to address this problem, there is still a great demand for further works.

(2) Semi-supervised object detection

The existing methods in object detection basically need to label data sets to train models. The supervised object detection model faces the challenge of the change of labeled data caused by scene change. Due to the changing nature of object tracking and detection, introducing semi-supervised learning into the model can effectively reduce the training time of object detection. Researchers recently used semi-supervised transformer model to improve the accuracy of object detection. However, it is still a challenge to apply them to the detection model, because they occupy a lot of memory and need further study.

(3) Publicly available datasets

Changing environmental conditions (such as weather) affect the performance of the object detector, but data from different environments can improve this situation during training. Using new data to shape and test the model can make the model adapt to the changes of weather environment. Therefore, there is a fundamental requirement for a dataset containing a wide range of data to train the model in order to ensure better robustness.

7.2. Applications

With the advancement and development of deep learning and various new technologies in recent years, the application of MOMCT in smart city has become more widespread and encompasses all aspects of life.

(1) Intelligent transportation

MOMCT technology enables real-time continuous tracking of vehicles on highways and is able to solve practical tracking problems in complex environments, such as occlusion, weather, light changes, etc. This technology obtains the traffic parameters of the vehicle and enables high precision continuous tracking of the vehicle on the highway. MOMCT technology can get the data of road congestion in time and provide people with more convenient travel modes. Meanwhile, traffic workers can monitor key road conditions by using MOMCT technology. In case of accidents such as traffic accidents, report to the traffic police in time and inform other car owners who are about to pass the accident section to avoid. It not only improves the efficiency of accident handling, but also reduces the waiting time for other travelers.

(2) Intelligent surveillance

Traditional monitoring technology usually consumes a lot of human and material resources, which is not in line with the development direction of the times. Because of its flexibility and accuracy, MOMCT technology is widely used for real-time monitoring in scenic spots, hospitals, banks, supermarkets and other public places with high traffic. This technology can track and detect moving objects such as pedestrians and vehicles in real time, and it identifies objects according to semantic information such as facial expressions and postures.

(3) Automated driving

With the development of technology, advanced autonomous driving has received extensive attention. This technology brings convenience and reduces the probability of accidents. The application of MOMCT technology in autonomous driving makes self-driving cars popular. It uses radar sensors and laser rangefinders to observe the surrounding traffic conditions, and keeps real-time, multi-directional and multi-viewpoint attention to moving objects and changes in the surrounding environment. It is realized that the vehicle keeps a relatively safe distance from other objects at all times, and traffic accidents can be effectively avoided.

7.3. Outlook

(1) Learning-based active tracking

Currently, some methods deal with some occlusion and ambiguity by reducing the reliance on trackers. However, they still need each camera to track the object, and then make motion control judgment and calculation. This method is easily influenced by the tracker. At present, it is difficult to obtain real value through labelling, so it is worth exploring further how to construct effective incentive functions in the real world.

(2) Multi-view information fusion

Some methods combine multi-view single object tracking methods to fuse information by directly combining features. Although these methods learn a more complete object image via a multi-view object model to achieve object tracking, camera movements can lead to image blurring. At the moment, there is no well-developed solution to solve this problem and it still requires continuous exploration by researchers.

8. Conclusions

In this paper, we present a review of recent advances in techniques and algorithms related to deep learning for multi-object multi-camera tracking tasks, including object trackers for MOMCT, analysis of different types of MOMCT methods, benchmark datasets, and evaluation metrics. In addition, several classical approaches and visualization results are presented to compare their performance. Although research on MOMCT tasks has made great progress in recent years, there are still significant challenges such as real-time processing problems, semi-supervised object detection problems, and open dataset problems. Therefore, we provide some perspectives on the future direction of MOMCT, including learning-based active tracking, multi-view information fusion, 3D object tracking, etc. The practical application of MOMCT in smart cities can both contribute to technological reform and help people create a better life. It is believed that this paper can help researchers gain insight into MOMCT and its application in practical scenarios, thus furthering its progress and development.

Author Contributions: Conceptualization, B.H.; software, L.F.; investigation, L.F.; formal analysis, B.H.; writing—original draft preparation, L.F.; writing—review and editing, B.H.; supervision, B.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science and Technology Project of Jiangxi Provincial Department of Transport (2022C0004, 2022C0005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets are available on Github at <https://github.com/wwwj795/datasets>, accessed on 14 March 2023.

Acknowledgments: We would like to thank anonymous reviewers for their supportive comments to improve our manuscript.

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

References

1. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards real-time multi-object tracking. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*; Springer: Cham, Switzerland, 2020; pp. 107–122.
2. Tang, Z.; Naphade, M.; Liu, M.Y.; Yang, X.; Birchfield, S.; Wang, S.; Kumar, R.; Anastasiu, D.; Hwang, J.N. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–19 June 2019; pp. 8797–8806.
3. Wang, W.; Wang, L.; Zhang, C.; Liu, C.; Sun, L. Social interactions for autonomous driving: A review and perspectives. *Found. Trends Robot.* **2022**, *10*, 198–376. [\[CrossRef\]](#)
4. Bendali-Braham, M.; Weber, J.; Forestier, G.; Idoumghar, L.; Muller, P.A. Recent trends in crowd analysis: A review. *Mach. Learn. Appl.* **2021**, *4*, 100023. [\[CrossRef\]](#)

5. Cao, J.; Weng, X.; Khirodkar, R.; Pang, J.; Kitani, K. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv* **2022**, arXiv:2203.14360.
6. Zhang, Y.; Wang, Q.; Zhao, A.; Ke, Y. A multi-object posture coordination method with tolerance constraints for aircraft components assembly. *Assem. Autom.* **2020**, *40*, 345–359. [[CrossRef](#)]
7. Liu, Q.; Chen, D.; Chu, Q.; Yuan, L.; Liu, B.; Zhang, L.; Yu, N. Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. *Neurocomputing* **2022**, *483*, 333–347. [[CrossRef](#)]
8. Parashar, A.; Shekhawat, R.S.; Ding, W.; Rida, I. Intra-class variations with deep learning-based gait analysis: A comprehensive survey of covariates and methods. *Neurocomputing* **2022**, *505*, 315–338. [[CrossRef](#)]
9. Zhang, Z.; Wang, S.; Liu, C.; Xie, R.; Hu, W.; Zhou, P. All-in-one two-dimensional retinomorph hardware device for motion detection and recognition. *Nat. Nanotechnol.* **2022**, *17*, 27–32. [[CrossRef](#)]
10. Jiang, D.; Li, G.; Tan, C.; Huang, L.; Sun, Y.; Kong, J. Semantic segmentation for multiscale target based on object recognition using the improved Faster-RCNN model. *Future Gener. Comput. Syst.* **2021**, *123*, 94–104. [[CrossRef](#)]
11. Li, X.; Zhao, H.; Yu, L.; Chen, H.; Deng, W.; Deng, W. Feature extraction using parameterized multisynchrosqueezing transform. *IEEE Sens. J.* **2022**, *22*, 14263–14272. [[CrossRef](#)]
12. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [[CrossRef](#)]
13. Jiménez-Bravo, D.M.; Murciego, Á.L.; Mendes, A.S.; San Blás, H.S.; Bajo, J. Multi-object tracking in traffic environments: A systematic literature review. *Neurocomputing* **2022**, *494*, 43–55. [[CrossRef](#)]
14. Khan, S.D.; Ullah, H. A survey of advances in vision-based vehicle re-identification. *Comput. Vis. Image Underst.* **2019**, *182*, 50–63. [[CrossRef](#)]
15. Dong, C.; Zhou, J.; Wen, W.; Chen, S. Deep Learning Based Multi-Target Multi-Camera Tracking System. In Proceedings of the 8th International Conference on Computing and Artificial Intelligence, Tianjin, China, 18–21 May 2022; pp. 419–424.
16. Luo, R.; Peng, Z.; Hu, J. On Model Identification Based Optimal Control and It's Applications to Multi-Agent Learning and Control. *Mathematics* **2023**, *11*, 906. [[CrossRef](#)]
17. Iguernaissi, R.; Merad, D.; Aziz, K.; Drap, P. People tracking in multi-camera systems: A review. *Multimed. Tools Appl.* **2019**, *78*, 10773–10793. [[CrossRef](#)]
18. Sufi, F.B.; Gazzano, J.D.D.; Calle, F.R.; Lopez, J.C.L. Multi-camera tracking system applications based on reconfigurable devices: A review. In Proceedings of the 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, 11–12 July 2019; pp. 1–5.
19. Wang, X. Intelligent multi-camera video surveillance: A review. *Pattern Recognit. Lett.* **2013**, *34*, 3–19. [[CrossRef](#)]
20. Olagoke, A.S.; Ibrahim, H.; Teoh, S.S. Literature survey on multi-camera system and its application. *IEEE Access* **2020**, *8*, 172892–172922. [[CrossRef](#)]
21. Bharati, P.; Pramanik, A. Deep learning techniques—R-CNN to mask R-CNN: A survey. In *Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019*; Springer: Singapore, 2020; pp. 657–668.
22. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
23. Jiang, M.; Gu, L.; Li, X.; Gao, F.; Jiang, T. Ship Contour Extraction from SAR images Based on Faster R-CNN and Chan-Vese model. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5203414. [[CrossRef](#)]
24. 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](#)]
25. Chen, M.; Yu, L.; Zhi, C.; Sun, R.; Zhu, S.; Gao, Z.; Ke, Z.; Zhu, M.; Zhang, Y. Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization. *Comput. Ind.* **2022**, *134*, 103551. [[CrossRef](#)]
26. Maity, M.; Banerjee, S.; Chaudhuri, S.S. Faster r-cnn and yolo based vehicle detection: A survey. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 1442–1447.
27. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
28. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
29. Redmon, J.; Farhadi, A. Yolo3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*; Springer: Cham, Switzerland, 2016; pp. 21–37.
31. Tiwari, V.; Singhal, A.; Dhankhar, N. Detecting COVID-19 Opacity in X-ray Images Using YOLO and RetinaNet Ensemble. In Proceedings of the 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 11–13 February 2022; pp. 1–5.
32. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

33. Peng, X.; Sun, B.; Ali, K.; Saenko, K. Learning deep object detectors from 3d models. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1278–1286.
34. Wang, L.; Chen, T.; Anklam, C.; Goldluecke, B. High dimensional frustum pointnet for 3D object detection from camera, lidar, and radar. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1621–1628.
35. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Pan, X.; Xia, Z.; Song, S.; Li, L.E.; Huang, G. 3D object detection with pointformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7463–7472.
37. Simon, M.; Milz, S.; Amende, K.; Gross, H.M. Complex-yolo: Real-time 3d object detection on point clouds. *arXiv* **2018**, arXiv:1803.06199.
38. Wen, L.H.; Jo, K.H. Fast and accurate 3D object detection for lidar-camera-based autonomous vehicles using one shared voxel-based backbone. *IEEE Access* **2021**, *9*, 22080–22089. [\[CrossRef\]](#)
39. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
40. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PLoS ONE* **2021**, *16*, e0259283. [\[CrossRef\]](#)
41. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
42. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206.
43. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
44. Hinton, G.E.; Krizhevsky, A.; Wang, S.D. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning—ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14–17, 2011, Proceedings, Part I 21*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 44–51.
45. Taylor, G.W.; Spiro, I.; Bregler, C.; Fergus, R. Learning invariance through imitation. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2729–2736.
46. Aliasghar, O.; Kanani Moghadam, V. Selective search and new-to-market process innovation. *J. Manuf. Technol. Manag.* **2022**, *33*, 1301–1318. [\[CrossRef\]](#)
47. Li, Y.; Shen, Y.; Zhang, W.; Zhang, C.; Cui, B. VolcanoML: Speeding up end-to-end AutoML via scalable search space decomposition. *VLDB J.* **2022**, *32*, 389–413. [\[CrossRef\]](#)
48. Daulton, S.; Eriksson, D.; Balandat, M.; Bakshy, E. Multi-objective bayesian optimization over high-dimensional search spaces. In Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, Eindhoven, The Netherlands, 1–5 August 2022; pp. 507–517.
49. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3520–3529.
50. Hong, Q.; Liu, F.; Li, D.; Liu, J.; Tian, L.; Shan, Y. Dynamic sparse r-cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4723–4732.
51. Ali, R.; Chuah, J.H.; Talip, M.S.A.; Mokhtar, N.; Shoaib, M.A. Structural crack detection using deep convolutional neural networks. *Autom. Constr.* **2022**, *133*, 103989. [\[CrossRef\]](#)
52. Wang, X.; Wang, L.; Zheng, P. SC-dynamic R-CNN: A self-calibrated dynamic R-CNN model for lung cancer lesion detection. *Comput. Math. Methods Med.* **2022**, *2022*, 9452157. [\[CrossRef\]](#) [\[PubMed\]](#)
53. Alsharekh, M.F.; Habib, S.; Dewi, D.A.; Albattah, W.; Islam, M.; Albahli, S. Improving the Efficiency of Multistep Short-Term Electricity Load Forecasting via R-CNN with ML-LSTM. *Sensors* **2022**, *22*, 6913. [\[CrossRef\]](#) [\[PubMed\]](#)
54. Ma, W.; Zhou, T.; Qin, J.; Zhou, Q.; Cai, Z. Joint-attention feature fusion network and dual-adaptive NMS for object detection. *Knowl.-Based Syst.* **2022**, *241*, 108213. [\[CrossRef\]](#)
55. Zhang, S.; Yu, Z.; Liu, L.; Wang, X.; Zhou, A.; Chen, K. Group R-CNN for weakly semi-supervised object detection with points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9417–9426.
56. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1–9.
57. Vijaya Kumar, D.; Mahammad Shafi, R. A fast feature selection technique for real-time face detection using hybrid optimized region based convolutional neural network. *Multimed. Tools Appl.* **2022**, *82*, 13719–13732. [\[CrossRef\]](#)
58. Zhang, R.; Song, Y. Non-intrusive load identification method based on color encoding and improve R-FCN. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102714. [\[CrossRef\]](#)
59. Roy, A.M.; Bhaduri, J.; Kumar, T.; Raj, K. WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection. *Ecol. Inform.* **2023**, *75*, 101919. [\[CrossRef\]](#)
60. Karaman, A.; Pacal, I.; Basturk, A.; Akay, B.; Nalbantoglu, U.; Coskun, S.; Sahin, O.; Karaboga, D. Robust real-time polyp detection system design based on YOLO algorithms by optimizing activation functions and hyper-parameters with artificial bee colony (ABC). *Expert Syst. Appl.* **2023**, *221*, 119741. [\[CrossRef\]](#)

61. Xue, Z.; Xu, R.; Bai, D.; Lin, H. YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5. *Forests* **2023**, *14*, 415. [\[CrossRef\]](#)
62. Mittal, U.; Chawla, P.; Tiwari, R. EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models. *Neural Comput. Appl.* **2023**, *35*, 4755–4774. [\[CrossRef\]](#)
63. Han, G.; Huang, S.; Ma, J.; He, Y.; Chang, S.F. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 780–789. [\[CrossRef\]](#)
64. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific machine learning through physics-informed neural networks: Where we are and what is next. *J. Sci. Comput.* **2022**, *92*, 88. [\[CrossRef\]](#)
65. Jia, D.; Zhou, J.; Zhang, C. Detection of cervical cells based on improved SSD network. *Multimed. Tools Appl.* **2022**, *81*, 13371–13387. [\[CrossRef\]](#)
66. Chen, Z.; Guo, H.; Yang, J.; Jiao, H.; Feng, Z.; Chen, L.; Gao, T. Fast vehicle detection algorithm in traffic scene based on improved SSD. *Measurement* **2022**, *201*, 111655. [\[CrossRef\]](#)
67. Gao, X.; Xu, J.; Luo, C.; Zhou, J.; Huang, P.; Deng, J. Detection of Lower Body for AGV Based on SSD Algorithm with ResNet. *Sensors* **2022**, *22*, 2008. [\[CrossRef\]](#)
68. Ma, R.; Chen, C.; Yang, B.; Li, D.; Wang, H.; Cong, Y.; Hu, Z. CG-SSD: Corner guided single stage 3D object detection from LiDAR point cloud. *ISPRS J. Photogramm. Remote Sens.* **2022**, *191*, 33–48. [\[CrossRef\]](#)
69. Cheng, L.; Ji, Y.; Li, C.; Liu, X.; Fang, G. Improved SSD network for fast concealed object detection and recognition in passive terahertz security images. *Sci. Rep.* **2022**, *12*, 12082. [\[CrossRef\]](#)
70. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-unet: Unet-like pure transformer for medical image segmentation. In *Computer Vision—ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*; Springer: Cham, Switzerland, 2023; pp. 205–218.
71. Kim, H.; Jung, W.K.; Park, Y.C.; Lee, J.W.; Ahn, S.H. Broken stitch detection method for sewing operation using CNN feature map and image-processing techniques. *Expert Syst. Appl.* **2022**, *188*, 116014. [\[CrossRef\]](#)
72. Chen, H.C.; Widodo, A.M.; Wisnujati, A.; Rahaman, M.; Lin, J.C.W.; Chen, L.; Weng, C.E. AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics* **2022**, *11*, 951. [\[CrossRef\]](#)
73. Kim, C.; Li, F.; Ciptadi, A.; Rehag, J.M. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015*; pp. 4696–4704.
74. Wang, S.; Sheng, H.; Yang, D.; Zhang, Y.; Wu, Y.; Wang, S. Extendable multiple nodes recurrent tracking framework with RTU++. *IEEE Trans. Image Process.* **2022**, *31*, 5257–5271. [\[CrossRef\]](#) [\[PubMed\]](#)
75. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017*; pp. 3645–3649.
76. Leal-Taixé, L.; Canton-Ferrer, C.; Schindler, K. Learning by tracking: Siamese CNN for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 27–30 June 2016*; pp. 33–40.
77. Zhang, J.; Sun, J.; Wang, J.; Li, Z.; Chen, X. An object tracking framework with recapture based on correlation filters and Siamese networks. *Comput. Electr. Eng.* **2022**, *98*, 107730. [\[CrossRef\]](#)
78. Su, Q.; Tang, J.; Zhai, M.; He, D. An intelligent method for dairy goat tracking based on Siamese network. *Comput. Electron. Agric.* **2022**, *193*, 106636. [\[CrossRef\]](#)
79. Chen, L.; Ai, H.; Shang, C.; Zhuang, Z.; Bai, B. Online multi-object tracking with convolutional neural networks. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017*; pp. 645–649.
80. Theckedath, D.; Sedamkar, R. Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Comput. Sci.* **2020**, *1*, 79. [\[CrossRef\]](#)
81. Chu, Q.; Ouyang, W.; Li, H.; Wang, X.; Liu, B.; Yu, N. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017*; pp. 4836–4845.
82. Liu, M.; Gu, Q.; Yang, B.; Yin, Z.; Liu, S.; Yin, L.; Zheng, W. Kinematics Model Optimization Algorithm for Six Degrees of Freedom Parallel Platform. *Appl. Sci.* **2023**, *13*, 3082. [\[CrossRef\]](#)
83. Katz, S.M.; Corso, A.L.; Strong, C.A.; Kochenderfer, M.J. Verification of image-based neural network controllers using generative models. *J. Aerosp. Inf. Syst.* **2022**, *19*, 574–584. [\[CrossRef\]](#)
84. Lu, J.; Wan, H.; Li, P.; Zhao, X.; Ma, N.; Gao, Y. Exploring High-order Spatio-temporal Correlations from Skeleton for Person Re-identification. *IEEE Trans. Image Process.* **2023**, *32*, 949–963. [\[CrossRef\]](#)
85. Hasan, M.R.; Guest, R.; Deravi, F. Presentation-Level Privacy Protection Techniques for Automated Face Recognition—A Survey. *ACM Comput. Surv.* **2023**, Accepted. [\[CrossRef\]](#)
86. Tang, W.; Chouzenoux, E.; Pesquet, J.C.; Krim, H. Deep transform and metric learning network: Wedding deep dictionary learning and neural network. *Neurocomputing* **2022**, *509*, 244–256. [\[CrossRef\]](#)
87. Son, J.; Baek, M.; Cho, M.; Han, B. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; pp. 5620–5629.
88. Xiang, J.; Zhang, G.; Hou, J.; Sang, N.; Huang, R. Multiple target tracking by learning feature representation and distance metric jointly. *arXiv* **2018**, arXiv:1802.03252.

89. Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; Zheng, N. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1335–1344.
90. Aggarwal, R.; Singh, N. An Approach to Learn Structural Similarity between Decision Trees Using Hungarian Algorithm. In *Proceedings of 3rd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications: ICMISC 2022*; Springer: Singapore, 2023; pp. 185–199.
91. Fang, K.; Xiang, Y.; Li, X.; Savarese, S. Recurrent autoregressive networks for online multi-object tracking. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 466–475.
92. Fernando, T.; Denman, S.; Sridharan, S.; Fookes, C. Tracking by prediction: A deep generative model for multi-person localisation and tracking. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1122–1132.
93. Ondruska, P.; Posner, I. Deep tracking: Seeing beyond seeing using recurrent neural networks. *Proc. AAAI Conf. Artif. Intell.* **2016**, *30*, 10413. [\[CrossRef\]](#)
94. Milan, A.; Rezatofghi, S.H.; Dick, A.; Reid, I.; Schindler, K. Online multi-target tracking using recurrent neural networks. *Proc. AAAI Conf. Artif. Intell.* **2017**, *31*, 11194. [\[CrossRef\]](#)
95. Sadeghian, A.; Alahi, A.; Savarese, S. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 300–311.
96. Li, D.; Ge, S.S.; Lee, T.H. Fixed-time-synchronized consensus control of multiagent systems. *IEEE Trans. Control Netw. Syst.* **2020**, *8*, 89–98. [\[CrossRef\]](#)
97. Kim, C.; Li, F.; Rehg, J.M. Multi-object tracking with neural gating using bilinear lstm. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 200–215.
98. Bashir, R.M.S.; Shahzad, M.; Fraz, M. Vr-proud: Vehicle re-identification using progressive unsupervised deep architecture. *Pattern Recognit.* **2019**, *90*, 52–65. [\[CrossRef\]](#)
99. Deng, W.; Zheng, L.; Ye, Q.; Kang, G.; Yang, Y.; Jiao, J. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 994–1003.
100. Wang, J.; Zhu, X.; Gong, S.; Li, W. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2275–2284.
101. Shen, F.; Du, X.; Zhang, L.; Tang, J. Triplet Contrastive Learning for Unsupervised Vehicle Re-identification. *arXiv* **2023**, arXiv:2301.09498.
102. Zhu, W.; Peng, B. Manifold-based aggregation clustering for unsupervised vehicle re-identification. *Knowl.-Based Syst.* **2022**, *235*, 107624. [\[CrossRef\]](#)
103. Wang, Y.; Wei, Y.; Ma, R.; Wang, L.; Wang, C. Unsupervised vehicle re-identification based on mixed sample contrastive learning. *Signal Image Video Process.* **2022**, *16*, 2083–2091. [\[CrossRef\]](#)
104. Gao, Z.; Wu, T.; Lin, L.; Zhao, J.; Zhang, A.; Wu, J. Eliminating domain deviation via synthetic data for vehicle re-identification. In Proceedings of the International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2022), Zhuhai, China, 25–27 February 2022; Volume 12288, pp. 6–11.
105. Chai, X.; Wang, Y.; Chen, X.; Gan, Z.; Zhang, Y. TPE-GAN: Thumbnail preserving encryption based on GAN with key. *IEEE Signal Process. Lett.* **2022**, *29*, 972–976. [\[CrossRef\]](#)
106. Zhou, Z.; Li, Y.; Li, J.; Yu, K.; Kou, G.; Wang, M.; Gupta, B.B. Gan-siamese network for cross-domain vehicle re-identification in intelligent transport systems. *IEEE Trans. Netw. Sci. Eng.* **2022**, *2022*, 3199919. [\[CrossRef\]](#)
107. Yan, T.; Li, H.; Sun, B.; Wang, Z.; Luo, Z. Discriminative feature mining and enhancement network for low-resolution fine-grained image recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5319–5330. [\[CrossRef\]](#)
108. Fayou, S.; Ngo, H.; Sek, Y. Combining multi-feature regions for fine-grained image recognition. *Int. J. Image Graph. Signal Process* **2022**, *14*, 15–25. [\[CrossRef\]](#)
109. Ning, X.; Tian, W.; He, F.; Bai, X.; Sun, L.; Li, W. Hyper-sausage coverage function neuron model and learning algorithm for image classification. *Pattern Recognit.* **2023**, *136*, 109216. [\[CrossRef\]](#)
110. Cenggoro, T.W.; Pardamean, B. A systematic literature review of machine learning application in COVID-19 medical image classification. *Procedia Comput. Sci.* **2023**, *216*, 749–756.
111. Salaberria, A.; Azkune, G.; de Lacalle, O.L.; Soroa, A.; Agirre, E. Image captioning for effective use of language models in knowledge-based visual question answering. *Expert Syst. Appl.* **2023**, *212*, 118669. [\[CrossRef\]](#)
112. Li, Z.; Wei, J.; Huang, F.; Ma, H. Modeling graph-structured contexts for image captioning. *Image Vis. Comput.* **2023**, *129*, 104591. [\[CrossRef\]](#)
113. Zhu, W.; Wang, Z.; Wang, X.; Hu, R.; Liu, H.; Liu, C.; Wang, C.; Li, D. A Dual Self-Attention mechanism for vehicle re-Identification. *Pattern Recognit.* **2023**, *137*, 109258. [\[CrossRef\]](#)
114. Lian, J.; Wang, D.; Zhu, S.; Wu, Y.; Li, C. Transformer-based attention network for vehicle re-identification. *Electronics* **2022**, *11*, 1016. [\[CrossRef\]](#)

115. Jiang, G.; Pang, X.; Tian, X.; Zheng, Y.; Meng, Q. Global reference attention network for vehicle re-identification. *Appl. Intell.* **2022**, 1–16. [\[CrossRef\]](#)
116. Tian, X.; Pang, X.; Jiang, G.; Meng, Q.; Zheng, Y. Vehicle Re-Identification Based on Global Relational Attention and Multi-Granularity Feature Learning. *IEEE Access* **2022**, *10*, 17674–17682. [\[CrossRef\]](#)
117. Li, M.; Wei, M.; He, X.; Shen, F. Enhancing Part Features via Contrastive Attention Module for Vehicle Re-identification. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 1816–1820.
118. Song, L.; Zhou, X.; Chen, Y. Global attention-assisted representation learning for vehicle re-identification. *Signal Image Video Process.* **2022**, *16*, 807–815. [\[CrossRef\]](#)
119. Li, H.; Wang, Y.; Wei, Y.; Wang, L.; Li, G. Discriminative-region attention and orthogonal-view generation model for vehicle re-identification. *Appl. Intell.* **2023**, *53*, 186–203. [\[CrossRef\]](#)
120. Tang, L.; Wang, Y.; Chau, L.P. Weakly-supervised Part-Attention and Mentored Networks for Vehicle Re-Identification. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 8887–8898. [\[CrossRef\]](#)
121. Liu, Y.; Hu, H.; Chen, D. Attentive Part-Based Alignment Network for Vehicle Re-Identification. *Electronics* **2022**, *11*, 1617. [\[CrossRef\]](#)
122. Shen, F.; Xie, Y.; Zhu, J.; Zhu, X.; Zeng, H. Git: Graph interactive transformer for vehicle re-identification. *arXiv* **2021**, arXiv:2107.05475.
123. Wang, H.; Peng, J.; Jiang, G.; Xu, F.; Fu, X. Discriminative feature and dictionary learning with part-aware model for vehicle re-identification. *Neurocomputing* **2021**, *438*, 55–62. [\[CrossRef\]](#)
124. Rong, L.; Xu, Y.; Zhou, X.; Han, L.; Li, L.; Pan, X. A vehicle re-identification framework based on the improved multi-branch feature fusion network. *Sci. Rep.* **2021**, *11*, 20210. [\[CrossRef\]](#)
125. Yang, J.; Xing, D.; Hu, Z.; Yao, T. A two-branch network with pyramid-based local and spatial attention global feature learning for vehicle re-identification. *CAAI Trans. Intell. Technol.* **2021**, *6*, 46–54. [\[CrossRef\]](#)
126. Fu, X.; Peng, J.; Jiang, G.; Wang, H. Learning latent features with local channel drop network for vehicle re-identification. *Eng. Appl. Artif. Intell.* **2022**, *107*, 104540. [\[CrossRef\]](#)
127. Liu, Y.; Zhang, X.; Zhang, B.; Zhang, X.; Wang, S.; Xu, J. Multi-camera vehicle tracking based on occlusion-aware and inter-vehicle information. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 3257–3264.
128. Hsu, H.M.; Wang, Y.; Cai, J.; Hwang, J.N. Multi-Target Multi-Camera Tracking of Vehicles by Graph Auto-Encoder and Self-Supervised Camera Link Model. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 489–499.
129. Hsu, H.M.; Wang, Y.; Hwang, J.N. Traffic-aware multi-camera tracking of vehicles based on reid and camera link model. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 964–972.
130. Li, Y.J.; Weng, X.; Xu, Y.; Kitani, K.M. Visio-temporal attention for multi-camera multi-target association. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9834–9844.
131. Liu, C.; Zhang, Y.; Chen, W.; Wang, F.; Li, H.; Shen, Y.D. Adaptive Matching Strategy for Multi-Target Multi-Camera Tracking. In Proceedings of the ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 2934–2938.
132. Zhao, J.; Gao, F.; Jia, W.; Yuan, W.; Jin, W. Integrated Sensing and Communications for UAV Communications with Jittering Effect. *IEEE Wirel. Commun. Lett.* **2023**, *2023*, 3243590. [\[CrossRef\]](#)
133. Yang, K.S.; Chen, Y.K.; Chen, T.S.; Liu, C.T.; Chien, S.Y. Tracklet-refined multi-camera tracking based on balanced cross-domain re-identification for vehicles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3983–3992.
134. Li, Y.L.; Chin, Z.Y.; Chang, M.C.; Chiang, C.K. Multi-camera tracking by candidate intersection ratio tracklet matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4103–4111.
135. Liang, N.S.J.; Srigrarom, S. Multi-camera multi-target drone tracking systems with trajectory-based target matching and re-identification. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 1337–1344.
136. He, Y.; Han, J.; Yu, W.; Hong, X.; Wei, X.; Gong, Y. City-scale multi-camera vehicle tracking by semantic attribute parsing and cross-camera tracklet matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 576–577.
137. Tran, D.N.N.; Pham, L.H.; Jeon, H.J.; Nguyen, H.H.; Jeon, H.M.; Tran, T.H.P.; Jeon, J.W. A robust traffic-aware city-scale multi-camera vehicle tracking of vehicles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3150–3159.
138. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2636–2645.

139. Yao, Y.; Zheng, L.; Yang, X.; Naphade, M.; Gedeon, T. Simulating content consistent vehicle datasets with attribute descent. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*; Springer: Cham, Switzerland, 2020; pp. 775–791.
140. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.C.; Qi, H.; Lim, J.; Yang, M.H.; Lyu, S. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [[CrossRef](#)]
141. 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.
142. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
143. Bernardin, K.; Stiefel, R. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP J. Image Video Process.* **2008**, *2008*, 246309. [[CrossRef](#)]
144. Luna, E.; Miguel, J.C.S.; Martínez, J.M.; Escudero-Viñolo, M. Graph Convolutional Network for Multi-Target Multi-Camera Vehicle Tracking. *arXiv* **2022**, arXiv:2211.15538.
145. Hsu, H.M.; Huang, T.W.; Wang, G.; Cai, J.; Lei, Z.; Hwang, J.N. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA, 15–20 June 2019; pp. 416–424.
146. Hou, Y.; Du, H.; Zheng, L. A locality aware city-scale multi-camera vehicle tracking system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA, 15–20 June 2019; pp. 167–174.
147. He, Z.; Lei, Y.; Bai, S.; Wu, W. Multi-Camera Vehicle Tracking with Powerful Visual Features and Spatial-Temporal Cue. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA, 15–20 June 2019; pp. 203–212.
148. Quach, K.G.; Nguyen, P.; Le, H.; Truong, T.D.; Duong, C.N.; Tran, M.T.; Luu, K. Dyglip: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, 20–25 June 2021; pp. 13784–13793.
149. Luna, E.; SanMiguel, J.C.; Martínez, J.M.; Escudero-Viñolo, M. Online clustering-based multi-camera vehicle tracking in scenarios with overlapping FOVs. *Multimed. Tools Appl.* **2022**, *81*, 7063–7083. [[CrossRef](#)]
150. Qian, Y.; Yu, L.; Liu, W.; Hauptmann, A.G. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, WA, USA, 14–19 June 2020; pp. 588–589.
151. Chang, M.C.; Wei, J.; Zhu, Z.A.; Chen, Y.M.; Hu, C.S.; Jiang, M.X.; Chiang, C.K. AI City Challenge 2019-City-Scale Video Analytics for Smart Transportation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA, 15–20 June 2019; pp. 99–108.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.