




Review

# Multiple Object Tracking in Deep Learning Approaches: A Survey

Yesul Park <sup>1</sup> , L. Minh Dang <sup>2</sup>, Sujin Lee <sup>1</sup>, Dongil Han <sup>1</sup>  and Hyeonjoon Moon <sup>1,\*</sup> 

<sup>1</sup> Department of Computer Science and Engineering, Sejong University, Seoul 143-747, Korea; ys@sju.ac.kr (Y.P.); genegraphy@sejong.ac.kr (S.L.); dihan@sejong.ac.kr (D.H.)

<sup>2</sup> Department of Information Technology, FPT University, Ho Chi Minh City 70000, Vietnam; minhdl3@fe.edu.vn

\* Correspondence: hmoon@sejong.ac.kr

**Abstract:** Object tracking is a fundamental computer vision problem that refers to a set of methods proposed to precisely track the motion trajectory of an object in a video. Multiple Object Tracking (MOT) is a subclass of object tracking that has received growing interest due to its academic and commercial potential. Although numerous methods have been introduced to cope with this problem, many challenges remain to be solved, such as severe object occlusion and abrupt appearance changes. This paper focuses on giving a thorough review of the evolution of MOT in recent decades, investigating the recent advances in MOT, and showing some potential directions for future work. The primary contributions include: (1) a detailed description of the MOT's main problems and solutions, (2) a categorization of the previous MOT algorithms into 12 approaches and discussion of the main procedures for each category, (3) a review of the benchmark datasets and standard evaluation methods for evaluating the MOT, (4) a discussion of various MOT challenges and solutions by analyzing the related references, and (5) a summary of the latest MOT technologies and recent MOT trends using the mentioned MOT categories.

**Keywords:** multiple object tracking; occlusion; ID switch; appearance; association



check for updates

**Citation:** Park, Y.; Dang, L.M.; Lee, S.; Han, D.; Moon, H. Multiple Object Tracking in Deep Learning Approaches: A Survey. *Electronics* **2021**, *10*, 2406. <https://doi.org/10.3390/electronics10192406>

Academic Editors: Amir Mosavi and Jungong Han

Received: 13 July 2021

Accepted: 23 September 2021

Published: 2 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



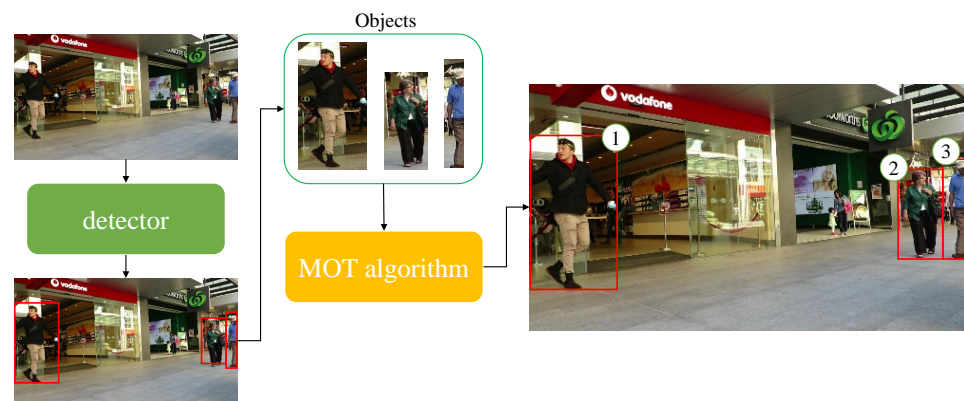
**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

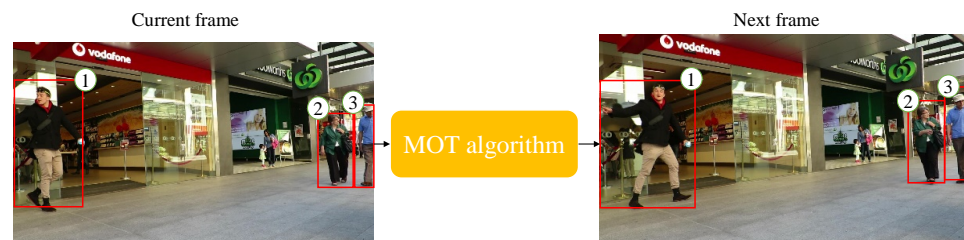
The recent advances in deep learning [1–3] and the availability of computing power [4,5] has revolutionized several fields, such as computer vision and natural language processing (NLP). Object detection [6,7] is a well-developed field in computer vision. Object tracking is usually the next process after object detection, which receives an initial set of detected objects, puts a unique identification (ID) for each of the initial detections, and then tracks the detected objects as they move between frames.

Multiple Object Tracking (MOT) is a subgroup of object tracking, which is proposed to track multiple objects in a video and represent them as a set of trajectories with high accuracy. However, object tracking usually has one big challenge when the same object is not given the same ID in all the frames, which is usually caused by ID switching and occlusion. ID switching is a phenomenon in which an object *X* with an existing ID *A* is assigned a different ID *B*, which can be caused by many scenarios, such as the tracker assigning another object *Y* the ID *A* as it resembles object *X*. Another problem is occlusion, which is when another object obscures one object partly or totally during a short period.

Figure 1 illustrates the MOT process. Initially, the objects in the current frame are detected by a detector. The objects are then tracked when they are fed into an MOT algorithm. After that, Figure 2 visualizes the object tracking process of multiple tracked objects from the current frame to the following frame. The two figures introduce how MOT aims to accurately track a large number of objects in a single frame.



**Figure 1.** The explanation of the ID assignment method, which is one of the main concepts of MOT using the MOT15 benchmark dataset [8]. Objects in the current frame are first detected using the detector. The detected result is then fed into an MOT algorithm to assign the ID for each object.



**Figure 2.** The visualization of the object tracking for the next frame using the MOT algorithm on the MOT15 benchmark dataset [8]. Objects in the following frame are first detected. The detected result is then fed into the MOT algorithm to compare the objects from the current frame with the objects from the next frame. Finally, the IDs are assigned for each object in the next frame based on the current frame.

In recent years, numerous novel MOT studies have been proposed to address the existing tracking problems, such as real-time tracking, ID switching, and occlusion. In addition, deep learning has been increasingly applied to MOT to improve its performance and robustness. Table 1 describes, in detail, the contributions of some of the previous surveys on the MOT topic. Overall, each survey focused on a specific problem of the MOT. Most lately, Pal et al. focused on the deep learning method and explained detection and tracking separately [9], so that the readers can easily concentrate on their part of interest. However, due to the description of many detection-related parts, the description of tracking is insufficient.

On the other hand, Ciaparrone et al. reviewed deep learning-based MOT papers that were published in the past three years [10]. They described online methods that perform in real-time and batch methods that can use global information and compare the experimental results. However, they focused only on the MOT benchmarks and provided no comparison for other benchmarks. In another review, Luo et al. described MOT methodology in two categories [11], and the evaluation focused on the PETS2009-S2L1 sequence of the PETS [12] benchmark. Finally, Kalake et al. reviewed MOT papers during the last 5 years [13]. Although they covered many aspects of the MOT, it was difficult to determine the exact evaluation for each tracking method due to the limited evaluation.

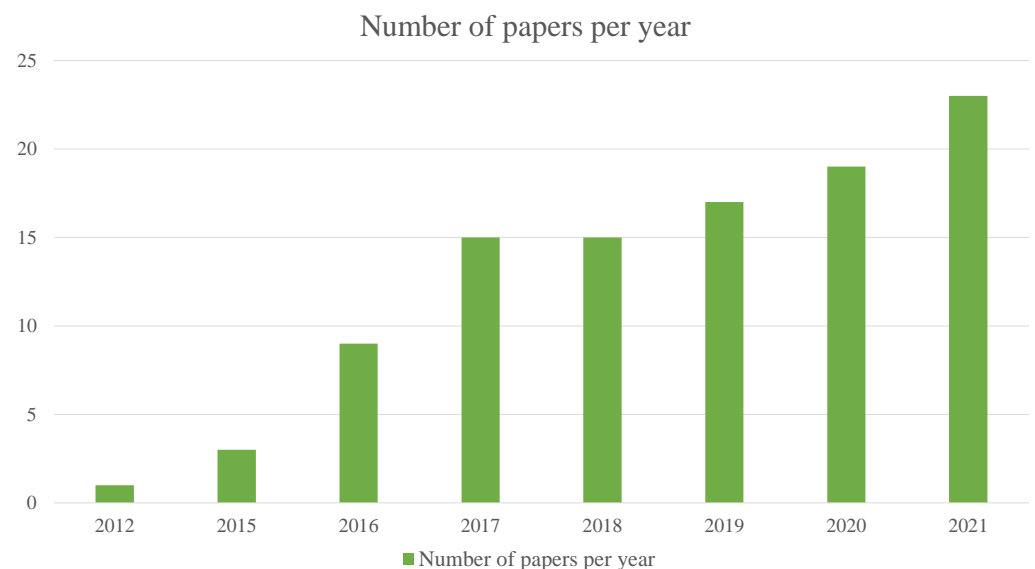
Figure 3 shows the total number of papers investigated in this survey. Overall, there is an increasing trend in the number of MOT papers, which introduced various deep learning-based MOT frameworks, new hypotheses, procedures, and applications, although previous surveys partly addressed the tracking and particularly the MOT topic. However, some existing parts of the MOT have not been covered in those reviews, for instance, (1) most of the surveys concentrated on the detection part rather than the tracking part [9,10], and (2) a limited number of benchmarks were mentioned [11]. As a result, a comprehensive survey

on recent MOT work is meaningful for stakeholders and researchers who want to integrate MOT into the existing systems or start new MOT research. This survey summarizes the previous work and covers many aspects of MOT. The main contributions are as follows.

- Describe the most basic techniques applied to the MOT.
- Categorize MOT methods and organize and explain the techniques used with deep learning methods.
- Include state-of-the-art papers and discuss MOT trends and challenges.
- Describe various benchmark datasets and evaluation metrics of the MOT.

**Table 1.** List of contributions from previous surveys on MOT.

ID	Ref.	Year	Contributions
1	[9]	2021	<ul style="list-style-type: none"> <li>• Offers a comprehensive review of object detection models.</li> <li>• Shows the development trends of both object detection and tracking</li> <li>• Describes various comparative results for getting the best detector and tracker.</li> <li>• Categorizes deep learning based on object detection and tracking into three groups.</li> </ul>
2	[10]	2020	<ul style="list-style-type: none"> <li>• Reviews the previous deep learning-based MOT research in the past 3 years</li> <li>• Divides previous papers into five main sections, which include detection, feature extraction and motion prediction, affinity computation, association/tracking, and other methods.</li> <li>• Shows the main MOT challenges</li> </ul>
3	[11]	2020	<ul style="list-style-type: none"> <li>• Shows the key aspects in a multiple object tracking system.</li> <li>• Categorizes previous work according to various aspects, and explains the advances and drawbacks of each group.</li> <li>• Provides a discussion about the challenges of MOT research and some potential future directions.</li> </ul>
4	[13]	2020	<ul style="list-style-type: none"> <li>• Reviews the past five years of multi-object tracking systems.</li> <li>• Compares the results of online MOTs and public datasets environment in the deep learning model.</li> <li>• Focuses mainly on deep-learning-based approaches</li> </ul>

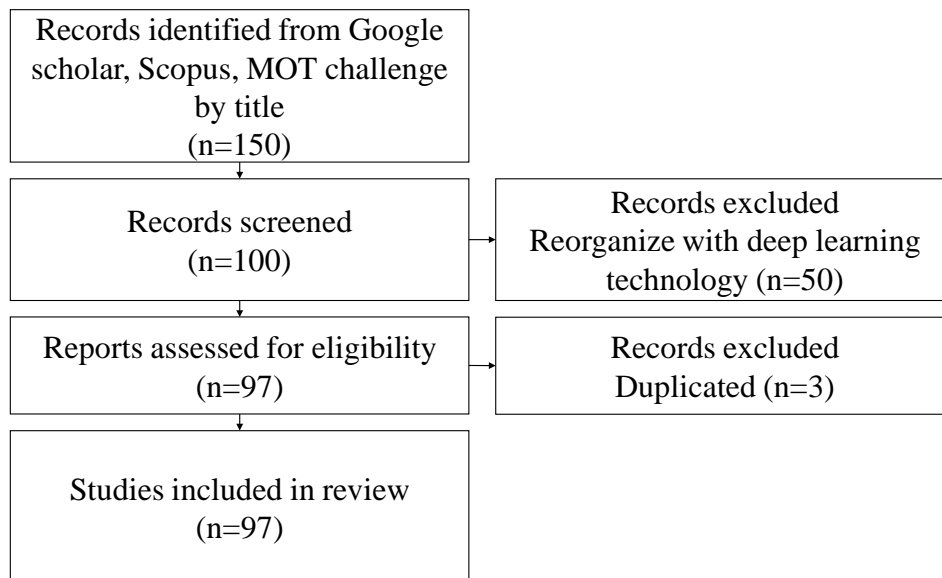


**Figure 3.** The number of published MOT papers yearly since 2012 that are discussed in this review.

## 2. Methodology

The method used to survey this paper summarizes according to TRANSPARENT REPORTING of SYSTEMATIC REVIEWS and META-ANALYSES (PRISMA) [14]. Figure 4 shows the process of how papers are collected for review. The papers used in the analysis were searched for only between 2017 and 2021. The search title used terms, such as 'multiple object tracking' and 'multiple object tracking deep learning' and included deep learning-based papers among the listed papers on the MOT challenge site. We included

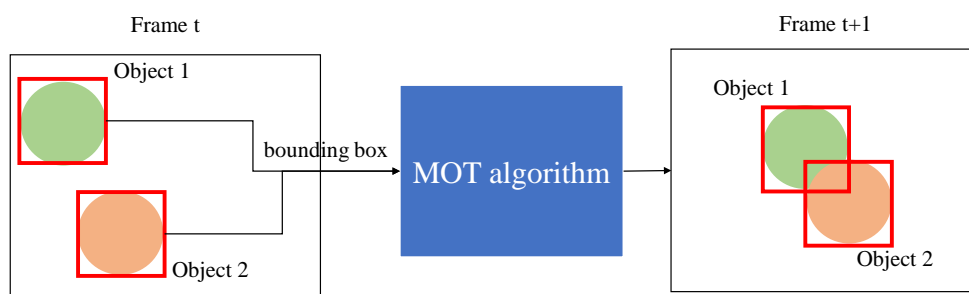
journals and conference papers, for a total of 150 papers. Some of the papers were removed because they did not use deep learning techniques; thereby, the number of papers became 100. Next, three overlapping papers were excluded; thus, finally resulting in 97 papers.



**Figure 4.** The search methodology for this paper.

### 3. Multiple Object Tracking Analysis

Figure 5 shows that most of the MOT algorithms track the location of the object in the next frame using the information of the detected objects from the current frame.



**Figure 5.** Description of a basic MOT process that includes (1) the detection of an object in frame  $t$ , (2) the exact position of the object is extracted and fed into an MOT algorithm, and (3) the object is tracked, and the object location at frame  $t+1$  is predicted.

Before reviewing the MOT research, Section 3.1 describes the MOT concepts, reviews two main problems that are mainly addressed in MOT, which are occlusion and id switching. After that, Section 3.2 shows common concepts that are frequently mentioned in the MOT research. Finally, Section 3.3 provides an in-depth overview of the previous MOT research.

#### 3.1. Multiple Object Tracking Main Challenges

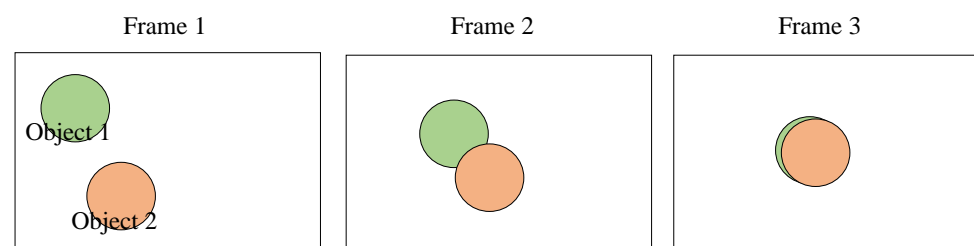
This section describes two common problems in MOT, which are occlusion and id switch. After that, four common deep learning approaches that are widely implemented in MOT, Recurrent Neural Network (RNN), Deep Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), and Attention, are described. Finally, Intersection Over Union (IoU), which is a common evaluation metric for object detection and object tracking is explained.

### 3.1.1. Occlusion

This section describes occlusion, which is a primary problem that usually happens during the MOT implementation. When the MOT algorithm is implemented only by the cameras without using other sensor data, it is difficult for the algorithm to track the location of the objects when they overlap each other [15]. Occlusion occurs when a part of one object obscures a part of another object in the same frame. It is even more challenging to solve the occlusion if one object completely occludes the other object. As a result, the object information from different frames is required to recognize different objects in the same location.

Figure 6 illustrates an example of occlusion. Occlusion is still among the existing challenge of MOT, because when the occlusion happens, it is challenging to predict the object's current position with only a simple tracking algorithm [16,17]. Occlusion mainly occurs in the frames that have many objects, which are solved by extracting appearance information using CNN [18–20] or using the graph information to find global attributes [21]. Huo's paper increased the resolution and appearance feature extraction resolution and robustness and then performed tracking by associating this data with the detection result [18].

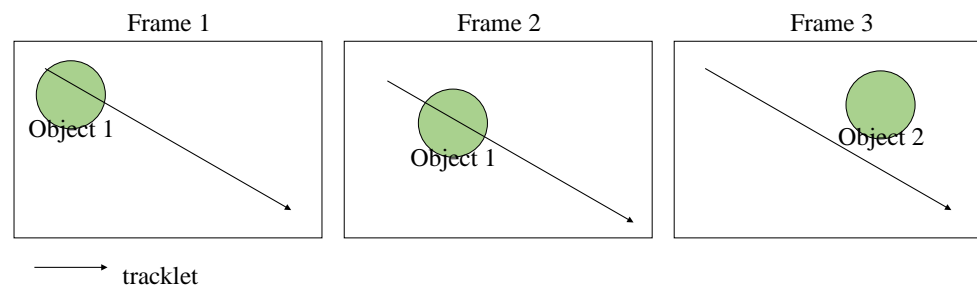
Milan et al. model improved performance by using a robust association strategy that integrates speed, acceleration, and appearance models [19]. Tian's model allocates costs, including appearance and motion directions, when each node in the network calculates the probability that new nodes are the same ID [20]. Global attributes are the properties found in the graph method for long-term tracking.



**Figure 6.** Visualization of the occlusion for two objects (orange and green). The objects in frame 1 are not overlapping. The objects in frame 2 are slightly overlapped, which is called occlusion. For frame 3, the green object is almost fully occluded by the orange object.

### 3.1.2. ID Switch

A tracklet is the calculation of the path of an object in a short period, usually under 10 frames [22,23], which is used by the tracking algorithm to predict the object's position in the next frame. As shown in Figure 7, when the same object is not included in the predicted tracklet in the current frame, this is considered an object that disappears in the current frame and is assigned the new ID. The object outside of the path is given a new ID, where the problem of changing the ID of the same object in the entire video is called ID Switching. A unique ID is assigned to an object by the tracker during the tracking process, but the ID will be removed if the tracker decides that the object is no longer within the frame.



**Figure 7.** Example of the ID switch problem. The object in frame 1 and the object in frame 2 are considered to have the same id because they are in the tracklet. On the other hand, frame 3 is judged to contain a different object because, there, the object is outside of the tracklet, and thus a new id is given.

### 3.2. Multiple Object Tracking Main Concepts

Some concepts commonly appear in the latest MOT research, such as RNN, LSTM, CNN, IoU, and Attention. The motivation for using ML is because it is challenging to learn the high-dimensional data with only simple algorithms. Although complex non-ML pattern recognition algorithms can be applied to perform the tracking algorithm, because using a simple method to estimate the location of an object, an estimation result frequently occurs as a lost object [24], and thus many researchers have used deep learning approaches to extract more robust features.

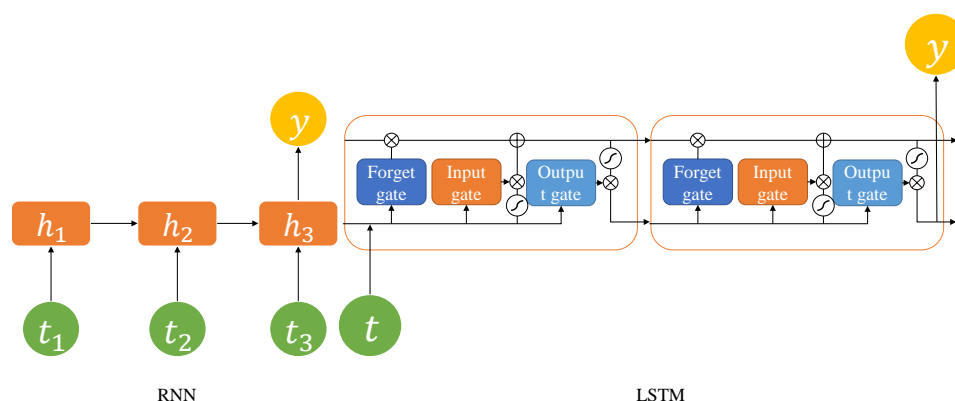
#### 3.2.1. Recurrent Neural Network (RNN)

RNN performs classification or prediction by learning sequential data from deep learning algorithms. CNN models used the filters within convolutional layers to transform data before being passed to the next layer. On the other hand, RNN models make use of the activation functions from other data points in the sequence to create the following output. As a result, RNN can process temporal data well. Thus, the current output results are influenced by the results of the previous time sequence, and the hidden layer acts as a type of memory that remembers features to send features to the next sequence using features from the input layer and the previously hidden layer. The hidden layer receives weights from the input layer and generates an output through an activation function [25].

#### 3.2.2. Long Short-Term Memory (LSTM)

LSTM [1] is a deep learning model derived from RNN. One of the main problems of RNN is that it fails to deliver important gradient information from the model's output to the layers near the model's input. A vanishing gradient means that the weight of the top layers has little effect on the output layer because the gradient vanishes further as the model becomes deeper [2]. LSTM has a structure similar to that of RNN but has two states in the hidden layer, called short-term states and long-term states, and LSTM has that input data passes through the hidden layer with a short-term state and a long-term state and generates output data through the output layer.

To solve gradient vanishing, LSTM is introduced with a forget gate that decides whether to remember the last information in RNN. The structure of the LSTM is shown in Figure 8. LSTM controls the current node's state information with three gates (input, forget, and output). The Forget gate determines whether to store old state information, the input gate determines whether to store new information being entered, and the output gate controls the output of updated cells. LSTM is used in both feature and sequence data in MOT.



**Figure 8.** Visualization of the LSTM architecture [10].

### 3.2.3. Convolutional Neural Network (CNN)

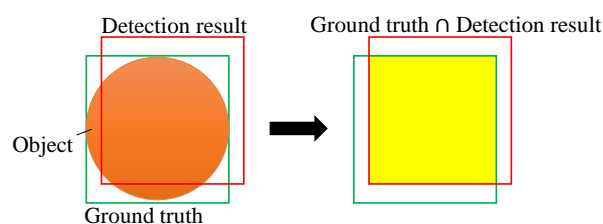
CNN [2,26,27] is a deep learning-based structure that is most commonly used to interpret visual imagery. It accepts the input as a multi-channelled image. On the other hand, RNN [25] is proposed to recognize patterns in data sequences, such as text, genomes, handwriting, the spoken word, and numerical times series data. Most research has used CNN to extract object appearance [4,28–32], which is the category of papers including CNN during the tracking process to extract features or using CNN as backbone during the detection.

### 3.2.4. Intersection over Union (IoU)

IoU is one of the evaluation methods shown in Figure 9. An orange circle is an object, the green bounding box indicates the ground truth annotation, and the red bounding box represents the predicted result. IoU is the intersection of the actual and detected result values divided into all actual and detected results areas. In the tracking part, some researchers used this to predict the tracklet. Bewley et al. used the Hungarian algorithm [17] with IoU [33].

The Hungarian algorithm is usually applied to solve the assignment problem by placing the predicted bounding boxes, and the detected bounding boxes in rows and columns of a cost matrix and allocates detection pairs with minimal cost by calculating the cost of having a row and column pairs. This method achieved better performance than the original IoU based tracker, which proved that a standard detector would not always obtain a good result.

In most cases, the used of a robust detector for the object tracking remarkably improved the tracking performance [4,10]. However, Bochinski suggested that any robust detector could be a problem if the detector predicted false-positives or false-negatives in the real world [34]. In this case, the track quality was greatly degraded due to a huge number of ID switch cases and fragments. Therefore, the authors solved the ID switch problem through a false-positive filtering process. The experimental results showed that the ID switch and fragments were significantly reduced on the VisDrone-VDT2018 test set. In addition, the model achieved a high Frames Per Second (FPS) rate, which indicated that it could process many frames per second.



**Figure 9.** Main concept of the IoU tracker [34].

### 3.2.5. Attention

Attention [35] is a technique that increases the performance of a model by making it focus on a specific vector, and this maps the query and key-value pairs to the output by input three vectors, Query (Q), Key (K), and Value (V). Attention and CNN [36–38] highlights a specific part of the feature map or describes the image by image captioning [39]. Image captioning methodology highlights some part of the node, making the feature map one-dimensional vector by flattening instead of using a fully connected layer. In RNN, attention is used to prevent important information loss when compressing input by encoders, and Transformer is used for encoders and decoders.

### 3.3. Techniques Used in the Paper about Multiple Object Tracking

In Section 3.1, this paper describes the methodology used in previous studies and presents the different methodologies for MOT as shown in Table 2.

**Table 2.** Common MOT methods and their corresponding references.

Methodology	Deep Learning Algorithms + Networks	Year
Appearance Learning	Spatial Temporal Information + Template Matching [24]	2017
	CNN + Appearance Model [40]	2019
	CNN + Compression Network [28]	2019
	LSTM [41]	2017
	Target appearance + Linear Subspace [42]	2016
	Generalized Minimum Clique Graphs (GMCP) [43]	2012
Occlusion Handling	CNN + FDH [44]	2017
	CNN + Semantic Information [29]	2018
	GMPHD [45]	2017
Detection and Prediction	CNN + Template Matching + Optical Flow [46]	2015
	CNN + Poisson Multi-Bernoulli Mixture (PMBM) Filter [47–49]	2018
	Box regression + Siamese Region Proposal (SiamRPN) [50]	2019
	CNN + Tracking-by-detection (TBD) + Reinforcement Learning [51]	2018
	F-RCNN + MTC [52]	2017
	Detector Confidence + PHDP [53]	2016
Computational Cost Minimization	Segmentation + CRF [54]	2015
	3D CNN + Kalman Filter [55]	2019
	CNN + Global Context Distancing (GCD) + Guided Transformer Encoder (GTE) [56]	2021
	Segmentation + Single Convolutional Neural Network (SCNN) [57]	2019
	spatial- temporal attention mechanism (STAM) + CNN [58]	2017
Motion Variations	Subgraph Decomposition [59]	2015
	TBD without Image information [60]	2017
	CNN + kernelized filter [61]	2020
	Pairwise Cost [62]	2018
	CNN + Motion Segmentation [63,64]	2018
LAC filters + CNN [65]	2017	



Table 2. Cont.

Methodology	Deep Learning Algorithms + Networks	Year
Appearance Variations, Drifting and Identity Switching	CNN + Data Association [30]	2018
	Joint inference network [66]	2021
	cross correlation CNN + scale aware attention network [31]	2020
	LSTM + Bayesian filtering network [67]	2019
	CNN + LSTM + Attention network [32]	2018
	CNN [4]	2017
Distance and Long Occlusions Handling	CNN [68]	2018
	CNN [69]	2018
Detection and Target Association	Kalman Filter + Hungarian Algorithm [33]	2016
	CNN [70]	2019
	CNN + GMPHD [71]	2019
Affinity	Appearance Learning [72]	2017
	CNN [73]	2018
	R-CNN [19]	2017
	CNN + Online transfer learning [74]	2017
	Spatial Temporal and Appearance Modeling [75]	2016
	Siamese Network [76]	2016
Tracklet Association	Visual Sensor Networks [77]	2018
	GNN [78]	2021
	CNN + decision making algorithm [79]	2018
	Single Camera Tracking + CNN [80]	2017
	Fast Constrained Domain Sets [81]	2016
	Sequential Monte Carlo (SMC) + Labeled Multi-Bernoulli (LMB) filter [82]	2016
Automatic Detection Learning	CNN [83]	2017
	Region-based Fully Convolutional Neural network (R-FCN) [84]	2018
	Quadruplet Convolutional Neural Networks (QCNN) [85]	2017
	CNN + Lucas-Kande Tracker (LKT) [86]	2016
Transformer	CNN + Query Learning Networks (QLN) [87]	2021
	CNN + GTE [56]	2021
	CNN + query key [88]	2021
	CNN + continuous query passing [89]	2021

### 3.3.1. Appearance Learning

In the first row of the table, one can see appearance learning algorithms. Appearance learning is a methodology that tracks objects by extracting features from CNN, mainly using the detectors. Zhang et al. proposed two approaches [90] of appearance learning, which are CNN features and the cascaded correlation filter. Wei et al. applied appearance, shape, and motion to decide the matching result between trajectory and detection [24], which prevents missed detection by including spatial information, such as appearance.

Fagot-Bouquet et al. used a sliding window to find the relationship between the already estimated trajectory and detection and made function E use the appearance, motion,

and interaction in the association problem [42]. Sliding window is an algorithm that moves the array element by a specific length and finds the maximum value. Ning et al. used Long Short-Term Memory (LSTM) [41], which makes a regression to enable the prediction of tracking locations in convolutional layers and LSTM [91].

Wang et al. proposed TrackletNet, which combines trajectory and appearance information [40], which obtains competitive Multi-Object Tracking Accuracy (MOTA) through occlusion processing, the generation of tracklets using epipolar geometry, and robustness to appearance features. Epipolar geometry is that objects have a geometric correlation. Zamir et al. achieved high MOTA and Multiple Object Tracking Precision (MOTP) in PET 09 sequences using optimization using a method of combining motion and appearance information and associating time-integrated data with Generalized Minimum Clique Graphs (GMCG) [43].

Kim et al. proposed a multi hypothesis tracking framework [92] based on both appearance and behavior using Bilinear LSTM. Multiple Hypothesis Tracking solves the multidimensional allocation problem through the Breadth-First Search (BFS) process. Sun et al. solved the ID switching problem in occlusion [28] to increase the multi-object tracking accuracy and achieved an average 6.3 frame rate per second.

Azimi et al. used AerialMPTNet [93] that model uses appearance, temporal, and graphical information, and which includes Siamese Neural Network, LSTM, and GNN. Siamese networks consist of two sub-networks. Two input images are put into each sub-network to create two outputs and calculate distance through two outputs. Zhou et al. conducted research on defensive and unmanned driving [94] and proposed a Fusion-Residual Predictive Network (FRPN) framework to measure the degree of risk on the road.

Zhang et al. proposed a high-resolution Siamese network [95] to address the low-resolution feature extraction of patches, which shares information of multiple resolutions and maintains high-resolution features by connecting convolutional streams in parallel, and they also attempted to improve the features of CNN as an attention mechanism. Tang et al. proposed an MDSPF method [96], shared convolutional units, and particle filter methods. Particle filters used scale-adaptive particle filters for robustness. The particle filter recursively predicts the state of the object based on prior information.

### 3.3.2. Occlusion Handling

The second technique is occlusion handling. Ray and Chakraborty separated foreground and background from video sequence [44]. Adding the current frame, they detected objects in the foreground, which refines the object region, and last, they tracked objects using the Kalman filter [16]. Xiang et al. tracked using Markov Decision Processes (MDPs) [46] for online tracking, which manages an object's birth, death, appearance, and disappearance to track objects, and this increased at least 7% of MOTA compared to other studies.

Kutschbach et al. applied the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter [45] for tracking objects, which was also combined with Kernelized Correlation Filters (KCF). This model sacrifices high sensitivity to increase the runtime and false positivity, which obtained competitive results on the UA-DETRAC benchmark [97]. Zhao et al. combined the Correlation Filter tracker with CNN features to enable re-identification (ReID) [29] when tracked objects are lost, which achieved a low tracking time of 0.07 with a competitive MOTA and high MOTP.

Hidayatullah et al. tracked using angles and grids to solve blurring or occlusion problems in fast motion [98]. Xia et al. used CNN and correlation filter [99] to solve occlusion, pose changing, and movement.

### 3.3.3. Detection and Prediction

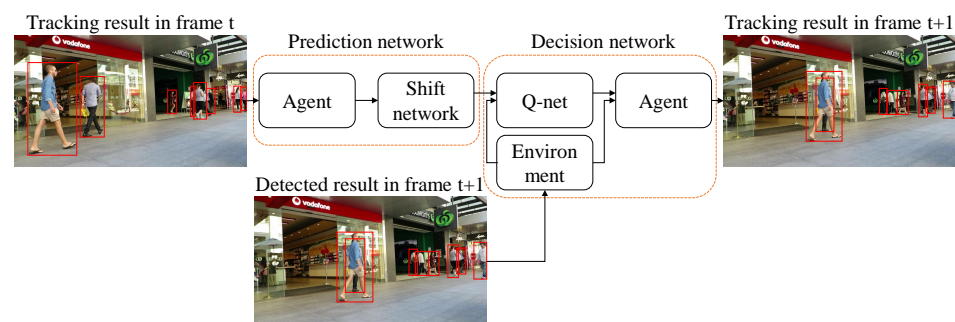
Detection and prediction compare the detection results with the prediction results. Scheidegger et al. combined the PMBM filter result to the detector [49], which uses 3D vector information for tracking, and PMBM filters predict world coordinates and achieved high FPS, and minimizing ID switching. Milan et al. used instance-based segmentation for their tracking

method [54], which used superpixel association with the low-level image, and they proposed a new conditional random field (CRF) that uses high-level information and low-level superpixel information.

Sanchez-Matilla et al. used low confidence, and high confidence from target detections [53] and also proposed the Probability Hypothesis Density Particle (PHDP) Filter framework, which framework makes reduction computation cost. Zhang et al. used Faster R-CNN in the detection part [52], which extracts appearance features for re-identification and which cluster for trajectories.

Li et al. solved the traditional translation invariance to Using ResNet as a backbone in Siamese networks [50], which predicts using a feature map, Siamese Region Proposal (Siam RPN) blocks, and Bounding box regression. Furthermore, the network reduces computational costs and redundant parameters using the correlation layer based on the online model.

Ren et al. proposed Collaborative Deep Reinforcement Learning (C-DRL) [51] as shown in Figure 10, and which solved occlusion, missed, or false detection with a reinforcement learning approach. C-DRL first used detection object location in the current frame, and they predicted the next frame object location and combined this in the Decision network. Even if the agent is blocked, they can update the location of the agent. Moreover, if candidate agents have noise, they will be ignored, have a new object location, update the new agent.



**Figure 10.** The proposed detection and prediction system [51] in MOT15 benchmark [8].

Madasamy et al. proposed Deep YOLO v3 [100], which includes a regression method for object location probability, which uses CNN with upsampling to detect the small object. Dao and Fremont used 3D information for MOT [101], and which uses the Hungarian algorithm for track-by-detection. Yin et al. proposed a CNN-based light neural network [102] in tracking-by-detection, which tracks using a graph matching process.

Song et al. tracked objects from self-driving vehicles to Deepsort using YOLOv3 [103]. In YOLOv3, they removed 32-times subsampling and added four-times subsampling for traffic signs. Padmaja et al. proposed a human activity tracking system for intelligent video [104], which uses the Deep landmark model and YOLOv3 detector. Chou et al. proposed Mask-Guided Two-Streamed Augmentation Learning (MGTSAL) [105] to complement the information of instance semantic segmentation.

Zhou et al. proposed a multi-scale network in Synthetic Aperture Radar (SAR) images [106] and selected Region Proposal Network (RPN) for classification and regression. Liu et al. detected 3D skeleton key points to used YOLOv4 [107], which used the Meanshift target tracking algorithm that converts to spatial RGB and CNN for recognition. Xie et al. utilized affine transformations [108] to spatial information models using CNN, which also refines the bounding box using a multi-task loss function including affine transformations and used Non-Maximum Suppression (NMS).

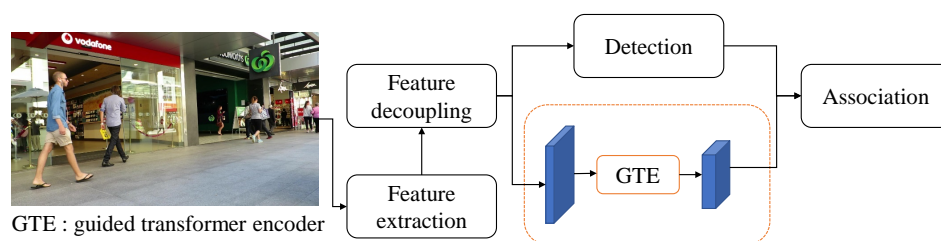
Shao et al. made use of autoencoder networks [109], motion generation networks, and location detection networks to generate trajectories. In Nobis et al., the proposed model fused sensor data [110], which is radar data, with Camera Radar Fusion Net (CRF-Net) to detect objects. Wu et al. composed the multi-level same-resolution compress (MSC) feature using a network, such as DSNet, to refine it using encoding and channel reliability

measurement (CRM) to form a tracking framework [111]. Zhu et al. proposed a tracking framework [112] using vehicle fine-grained vehicle classification and detection.

### 3.3.4. Computational Cost Minimization

The Weng and Kitani model is for 3D multi-object tracking and approaches the 3D object detector using by LiDAR point cloud [55], and this model combines the 3D Kalman filter and Hungarian algorithm with data association, which obtains increased MOTA while maintaining real-time performance. Tang et al. proposed the Subgraph Multicut model and heuristic solution using the Kernighan–Lin algorithm [59]. To solve the occlusion problem, Chu et al. used the spatial-temporal attention mechanism (STAM) model [58] that uses shared CNN features and ROI-pooling.

Voigtlaender et al. proposed Track R-CNN, which extracts temporally enhanced image features using CNN and relabel KITTI and MOT challenge for instance segmentation [57]. This benchmark was released. Yu et al. proposed a framework that incorporates GCD and GTE, called RelationTrack [56], which is shown in Figure 11. GCD is a module that separates by sense and by ReID to avoid contradictions that optimize during learning. The GTE module combines transformer encoders with deformable attention for global information consideration. GTE can capture global information with limited amounts of resources. They attempted computational cost-minimizing.



**Figure 11.** Computational cost minimization using the model [56] with the MOT15 benchmark [8].

Avola et al. used Multi-Stream architecture with Faster R-CNN [113] backbone and used Deep Association Metric (Deep SORT), which includes Simple Online and Real-time Tracking. Zhou et al. proposed a framework for the real-time security surveillance of smart IoT systems [114], and proposed a tracking algorithm for lightweight models by combining MTCNN and YOLO. In Hossain and Lee, they developed an association metric by integrating it into Deep SORT [115], which combines Kalman filtering and deep learning for tracking in small flight drones with limited computing power.

### 3.3.5. Motion Variations

Bochinski et al. tracked by reducing complexity and computational cost through IoU trackers [60], and this approach is tracking-by-detection that relies on detector performance. IoU tracker to make possible 100K fps. Ruchay et al. proposed a locally adaptive correlation filter with CNN [65], which adapted scene frame information into filters. Sharma et al.'s model is for urban driving using a single camera, which complements the error in detection using pairwise costs using 3D cues [62] and performs in real-time.

Whin et al. proposed a kernelized correction filter (KCF) tracking [61] method with three modules integrated to increase tracking accuracy at high speed in real-time streaming condition environments. First, they tracked failure detection and search multiple windows using re-tracking. This paper also analyzed the motion vector for the searching window. Keuper et al. used bottom-up motion segmentation [63] as a way to group point trajectories with multi-object tracking through clustering of top-down detect tracking.

Chen and Ren used Multi-Appearance Local Control segmentation (MALC) [64] for segmenting merge detection, and proposed Track-Oriented Multiple Hypothesis Tracking (TOMHT) to improved speed and performance. Their main technique is motion variations.

He et al. proposed a tracking framework [116] in an end-to-end manner for using unlabeled data, and this framework includes Reprioritized Attentive Tracking with Tracking-By-Animation. Lee and Kim proposed a Feature Pyramid Siamese Network (FPSN) [117] to extract multi-level feature information and to add Spatio-temporal motion features to consider both appearance and motion information.

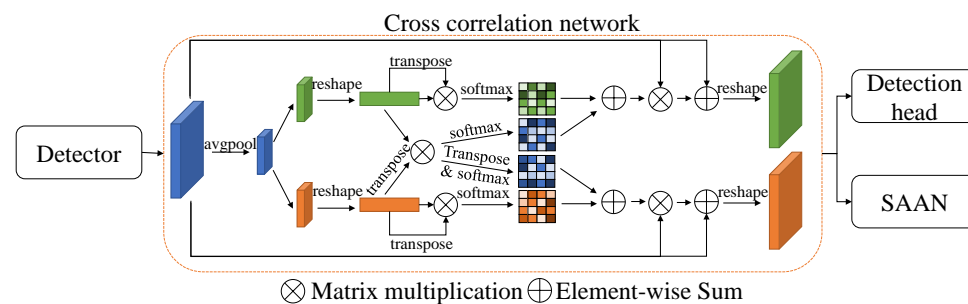
### 3.3.6. Appearance Variations, Drifting and Identity Switching

Zhu et al. extracted detection features by using CNN, calculating affinity using spatial attention [32], which also applied temporal attention to LSTM to use a Dual Matching Attention Network (DMAN). Wojke et al. solved the occlusion problem using CNN [4], a deep application descriptor for existing SORT algorithms to solve the re-identification problem.

In Xiang et al., they used three architectures for their affinity model: A-Net, M-Net, and Metric-Net [67]. A-Net extracts their appearance, and M-Net extracts Motion. Metric-Net uses three-channel CNN-LSTM networks, which share weights. For comparison similarity, Metric-Net uses triplet loss in their framework. A-Net consists of CNN and bounding box regression. CNN performs with VGG-16 Net for using a pre-trained model and then fine-tunes it with MOT and person identity datasets.

M-Net has two modules, that is, LSTM and a Bayesian filtering network (BF-Net). After predicting trajectory and position, they put this into BF-NET and Metric-Net. Metric-Net inference with data association using Hungarian algorithms, and they put this to BF-Net. BF-Net estimates their trajectory. Yoon et al. used appearance models based on joint inference networks [66] in multi-object environments where features need to compare with other objects.

To learn special features, Liang et al. proposed a cross-correlation network, which exploits the correlation of CNN features [31]. The cross-correlation network from this model is shown in Figure 12. This paper addresses the ReID feature using scale-aware attention networks and approaches appearance variations, drifting, and identity switching.



**Figure 12.** Example of a model using appearance variations, drifting, and identity switching [31].

Dike and Zhou proposed a Deep Quadruplet network (DQN) [118] that uses a new loss function for feature space. Gomez-Silva et al. computed affinity using an appearance preference model [119]. The loss function used in this model is the triple loss function.

Li et al. proposed a tracking model capable of hierarchical time series prediction and used constrained mix sequential Monte Carlo (CMSMC) [120] to solve the re-id problem. This paper consists of two modules: a behavior recognition module and a state evolution module. Lv et al. proposed a depthwise separable convolution neural network (DS-CNN) [121], which included pointwise convolution (P-Conv2D) and depthwise convolution (D-Conv2D). Xu et al. proposed the Group Feature Selection Method for Discriminative Correlation Filters (GFS-DCF) [122] to select group features at the channel and spatial dimensions.

### 3.3.7. Distance and Long Occlusions Handling

There are two approaches the distance and long occlusion handling. Based on single CNNs, Gan et al. combined cues of multiple features to assign IDs to tracked targets and stores them [68] in memory for model updates. If the object is missing, the target is removed, and the target is tracked until target-out. This framework is in Figure 13. Kampker et al. proposed a framework [69] in urban scenarios with grid-based techniques and object-based techniques using Lidar raw data. Shahbazi et al. used 3D information to determine the speed and estimate the location of trackers with detection and tracking [123].

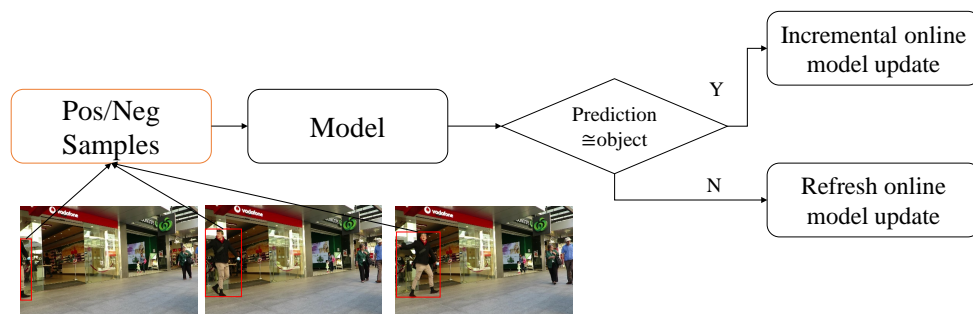


Figure 13. The model proposed in the paper of category distance and long occlusion handling [68] using the MOT15 Benchmark [8].

### 3.3.8. Detection and Target Association

In the detection and target association approach, Bewley et al. had a simple process [33] for object tracking. They used only the Kalman filter and Hungarian Algorithm to predict motion. Wang et al. proposed an appearance embedding model for data connection with single-shot detectors using the Joint Learning of Detection and Embedding (JDE) [70] method, which are for real-time tracking, and this speed was 22–40 FPS. Figure 14 presents the comparison between JDE and other detectors.

Baisa applied the Gaussian mixture Probability Hypothesis Density (GM-PHD) [71] filter to visual similarity CNN to track multiple targets in video sequences, and which uses a cost-minimizing approach to CNN appearance features and bounding boxes using Hungarian algorithms. The GM-PHD filter is divided into two steps. One is prediction, and the other is the update.

Following the linear Gaussian model, they initialize velocity to zero to remove prior knowledge, which sets the target state independently of survival and detection probabilities. Visual-similarity CNN is constructed with two steps. First, detect patch from frame  $k - 1$  and  $k$ , and concatenate after resizing. That input patch is put into visual-similarity CNN. Second, they get similarity confidence from binary cross-entropy loss.

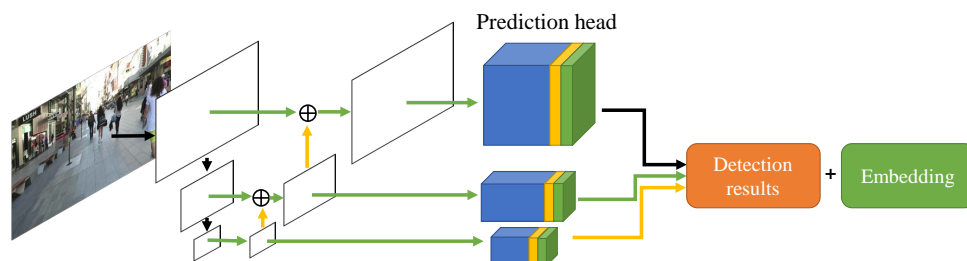


Figure 14. A detection and target association model [70] in MOT15 benchmark [8], which compares Separate Detection and Embedding (SDE), two-stage model, and Joint Detection and Embedding (JDE).

Pegoraro and Rossi used cloud sequence by Mm-wave radars and used the extended Kalman filter [124], and the platform was evaluated in an edge-computing system. In Liu, tracking using deep learning-based detectors was performed using the Deep Associated

Elastic Tracker (DAE-Tracker) [125]. Wen used faster-CNN and Hungarian matching algorithms to track objects [126]. Ullah and Alaya Cheikh proposed an inDuctive Sparse Graphic Model (DSGM) [21], a graph model that reduced computational complexity by minimizing connectivity in the multi-target scene, resulting in competitive results compared to other models.

### 3.3.9. Affinity

Affinity solves the problem of ID switching and occlusion by comparing the application of the detected object and the feature of the tracked object. Ju et al. proposed a process using a novel affinity model and appearance features [72], which model performs track fragmentation processing when the object is initialized to connect occluded objects. KC et al. proposed a graph-based model and approached the problem of allocating identical or distinct labels to the detection of graphs [75].

Each graph assigns a unique label if the spatio-temporal or appearance cues are the same in detection pairs. Leal-Taixe et al. used the Siamese Network approach [76], which includes the same two models in one. When two detectors belong to the same tracking entity, Siamese networks estimate from CNNs.

Milan et al. proposed a tracking method [19] based on recurrent neural networks for the state of changing by time, and which found that LSTM can learn a one-to-one assignment. The one-to-one constraint ensures that the same measurement is not assigned to multiple targets in joint data connection for the task of uniquely classifying measurements in data association. Yoon et al. used the Siamese network [73] to track discriminative appearance features through appearance matching.

Bae and Yoon improved MOTA using an online detector of information and online transfer learning [127] and tracklet confidence to update the online multi-object tracking framework [74]. Tracklet confidence is measured using length, occlusion, and effectiveness of the track, and transfer learning used a pre-trained model for training a model using many datasets.

This paper used appearance, shape, and motion models to set tracklet elements. A tracklet is divided into two ways. One is a high-confidence tracklet, which is locally linked to the HC-association (High Confidence association) phase, and another is a low-confidence tracklet, which is globally linked to the LC-association (Low Confidence association) phase. For the distinction of multiple objects, relating tracklets with appearance modeling is essential, and, for this, they proposed a deep appearance model and adapted online transfer learning. The discriminative Deep Appearance Model has a simple layer because of learning complexity, which the method calculates for minimizing positive objects and maximizing negative objects.

Xu et al. used augmented Lagrangian in Discriminative Correlation Filters (DCF) [128] to focus on channel selection. Huang et al. focused on the detector to small object detection and class imbalance and, thus, used a Hierarchical Deep High-resolution network (HDHNet) [129] for a prediction network. They used a new loss function that combines focal loss and GIoU loss.

Chen et al.'s system is an autonomous system [130], which used 3D information from 3D point clouds [131] and relation conv for pair of objects correlation. Wang et al. proposed unsupervised learning with a Siamese correlation filter network [132], which uses a multi-frame validation scheme and cost-sensitive loss and which have real-time speed using unsupervised learning.

Wu et al. used dimension adaptation correction filters (DACF) to extract features using CNN from conventional correction filters (CF) [133]. Yang et al. model proceeds in real-time, and they proposed a tracking method using long-term and short-term features [134], which optimizes the position of the object using boundary box regression using the cosine window of the correlation filter.

Mauri et al. conducted research on two approaches for smart mobility [135]: monodepth2 and MADNet. This research exploited the extended Kalman filter to improved the

SORT approach in tracking. Akhloufi et al. used deep reinforcement learning and IoU to increase tracking accuracy for tracking in Unmanned Aerial Vehicles (UAVs) [136]. Huo et al. accurately extracted feature information when the target is obscured, based on the Reid pedestrian re-identification network (RFB) [18], solved call detection, and linked to distinguish targets.

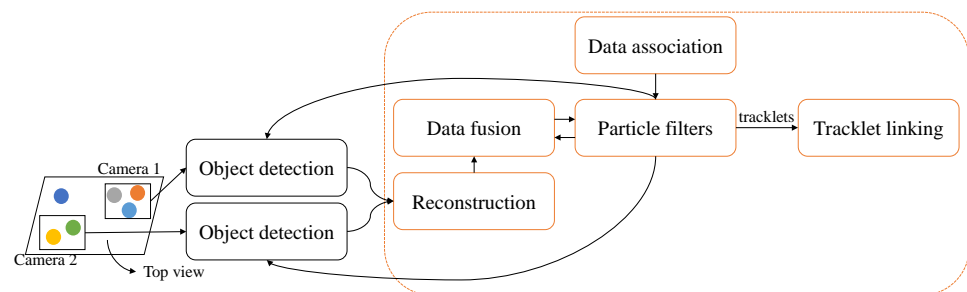
Tian et al. combined the learning tracker [20] structured for multi-target tracking and segmentation with the segmentation algorithm, resulting in accurate segmentation results.

### 3.3.10. Tracklet Association

Some research needs to associate tracklet when tracking objects, and we show an example in Figure 15. Jiang et al.'s paper estimated observations by combining 2D features and 3D features in multiple views [77], such as multi-camera systems. Tracklets are formed using particle filters are modeled by graphs, and integrated into the full track. Wu et al. used track-based multi-camera tracking (T-MCT) [80] with clustering and proposed distributed online framework, including T-MCT re-identification (Re-id).

Le et al. proposed target tracking between cameras as a synchronized overlapping camera network environment [79]. The decision algorithm performs the tracking, which allows the camera to track the target in the case of occlusion in each view. Kieritz et al. proposed Multiple-Instance Learning (MIL) [81] for the training appearance model, which uses the Integrate Channel Features (ICF) in fast-detected pedestrians.

Scheel et al. used multiple high-resolution radar in their method [82] that has a Labeled Multi-Bernoulli filter to track vehicles, which also implements the Sequential Monte Carlo (MC) Algorithm. Weng et al. predicted and tracked using multi-agent interaction, which uses Graph Neural Networks (GNN) [78] for multi-agent interaction and also uses the diversity sampling function to avoid duplicated trajectories. GNN is used to visualize relationships using graphs, which is used for the association between objects in tracking.



**Figure 15.** An example of a model of Tracklet association [77] with the MOT15 benchmark [8].

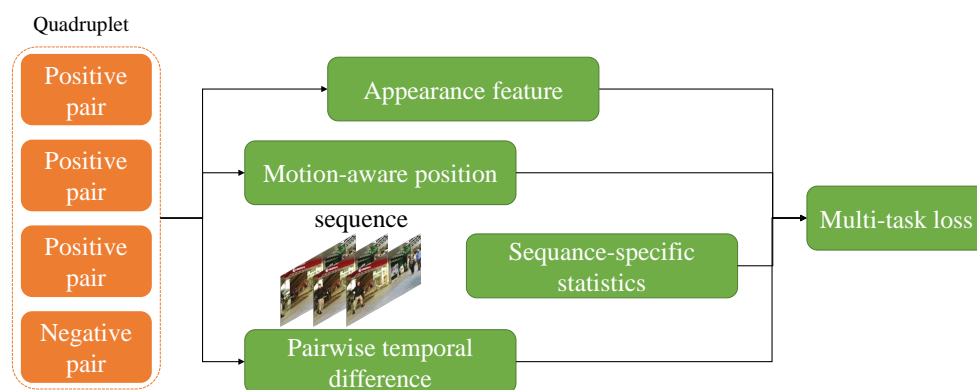
### 3.3.11. Automatic Detection Learning

This section deals with automatic detection learning. Schuster et al. proposed a data association network [83] using backpropagation, which tracks over the cost of pairwise association. In Son et al., the proposed tracker was an end-to-end Quadruplet Convolutional Neural Network (CNN) [85] as shown in Figure 16, where the tracker exploits quadruple loss and gives constraints that places it closer to adjacent detectors.

Lee et al. used ensemble network construct CNN with Lucas Kanade Tracker(LKT) [86] in motion detection. Their model has robust multi-class multi-object tracking (MCMOT) with a Bayesian filtering framework. Chen et al. solved occlusions in track prediction using candidates to handle unreliable detection in existing tracks [84]. Scoring was carried out by using an R-FCN to select the best among many candidates during tracking.

Voeikov et al.'s goal was tracking real-time processing in high-resolution video [137]. To solve this problem, they used an auto-referee system and the occlusion problem by using CNN with the appearance feature, which also uses the spatial attention mechanism of insertion and location. Jiang et al. proposed multi-agent deep reinforcement learning (MADRL) [138], which uses a learning method that uses Q-Learning (QL) to treat other agents as part of the current agent's environment.





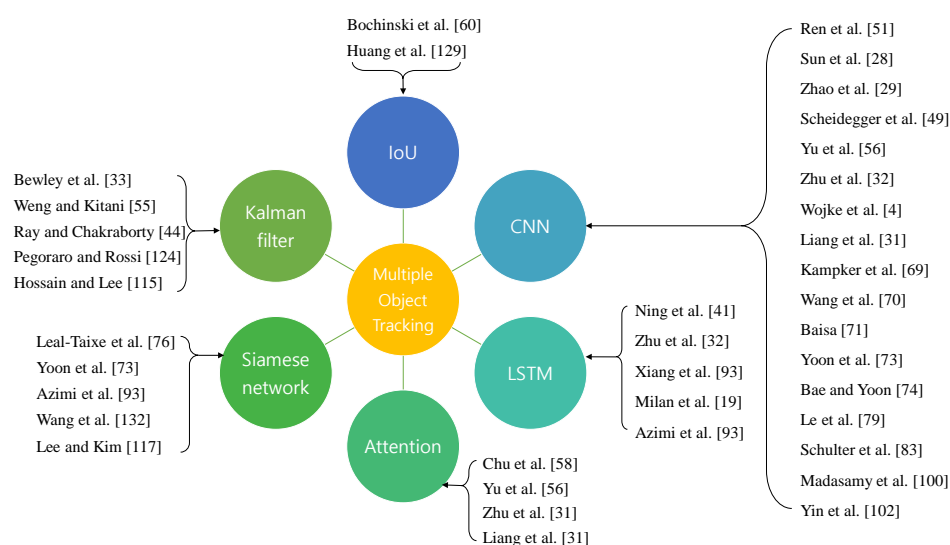
**Figure 16.** The automatic detection learning model proposed by Son et al. [85] using the MOT15 benchmark [8].

### 3.3.12. Transformer

Xu et al. proposed TransCenter [87], which consists of the tracking and detection. Each part includes a Deformable Encoder and Decoder. As this paper describes in Section 3.3.4, Yu et al. used a transformer in GTE [56], which included a deformable attention to transformer encoder. GTE uses defensible attention to overcome slow speeds and limited resolution during learning and obtains robust embedding for the subsequent association. Sun et al. used “object query” [88] generated with learned object detectors and “track query” about objects from previous frames. After a bounding box using each query prediction, an IoU match generates a final set of objects.

Zeng et al. introduced MOTR [89], which created a transformer and DETection TRANSformer (DETR) [139]. MOTR models long-range temporal relations through a continuous query delivery mechanism. In Belyaev et al., they used the Deep Object Tracking model with Circular Loss Function (DOTCL) [140], a loss function that takes into account boundary box overlap and orientation, and which uses the Transformer Multi-head Attention architecture.

We organized the MOT methodology from Table 2. We present MOT algorithms in six categories as shown in Figure 17.



**Figure 17.** The categorization of MOT algorithms into six main categories, IoU, CNN, LSTM, attention, Siamese network, and Kalman filter.

### 3.3.13. IoU

Bochinski et al. assumed that the detector can detect all objects, resulting in a small gap in detection [60]. In this case, the objects in the current frame and the previous frame have high overlap IoU. However, there are disadvantages to IoU. In Huang et al., the two boxes cannot reflect similarity if the objects in the previous frame do not intersect with those in the current frame [129]. In addition, not all objects have the same location, even if they have the same IoU. Therefore, to address this, in this paper, they complement it by using Generalized Intersection over Union (GIoU) to compute the similarity between two objects in IoU.

### 3.3.14. CNN

As we can see in Section 3.1, CNN is often used for extract appearance similar with feature map. Most research uses this feature for tracking enhancement [28–30]. Ren et al. used CNN in their prediction network [51]. The network trains the movement of objects. Zhao et al. combined Correlation Filter (CF) and CNN [29]. For feature extraction, Scheidegger et al. used CNN for detection [49]. Zhu et al. used the attention module when they extract features from CNN [32] and used a matching layer after CNN. Wojke et al. used the wide residual network in CNN [4] for deep appearance descriptor.

Liang et al. adapted scale-aware network (SAAN) [31] through CNN. In Wang et al., they used CNN with the Embedding model to extract appearance [70]. Baisa also used CNN for visual computing similarity from the patch [71]. Yoon et al. proposed a Siamese network into CNN [73]. Bae and Yoon model used Discriminative Deep Representation Learning from CNN [74]. Yin et al. used CNN for feature extraction in the associate network [102]. When the associate network finish, they matched the network using by hyper-target graph. Other research is discussed in Section 3.3.

### 3.3.15. LSTM

In MOT, LSTM is mainly used to estimate motion after feature extraction. Ning et al. used LSTM to temporal construct [41] a detected object. In Zhu et al., this research used Bi-directional Long-Short Term Memory (Bi-LSTM) [32] after extracting features. Xiang et al.'s model used M-Net, which includes LSTM to the Affinity model [67]. For data association [19], Milan et al. used LSTM. Azimi et al. combined a LSTM module with a GCNN module [93].

### 3.3.16. Attention

Attention is used to both CNN and sequence models, like LSTM and RNN. Attention is often used to emphasizes certain nodes in networks. In Chu et al., they used spatial attention [58] when they extracted features. Yu et al. proposed a global attraction [56] that considers inter-pixel interactions, unlike previous models. Zhu et al. used Dual Matching Attention [32], which uses both spare and temporary attraction. Liang et al. used scale-aware attention [31] when they were extracting features in a multi-scale environment.

### 3.3.17. Siamese Network

Attention is used in both sequence data and CNN, which is the same for extracting sensitive information. In Leal-Taixe et al., they used a CNN-based Siamese network [76], where they exploited pixel values and optical information. Yoon et al. compared the appearance from Siamese network [73]. Azimi et al. applied the Siamese network in the model they used for accurate tracking AerialMPTNet [93]. Wang et al. used the Siamese network in the Unsupervised learning method [132]. To address the lack of features, Lee and Kim proposed a Feature pyramid Siamese network (FPSN) [117].

### 3.3.18. Kalman Filter

The Kalman filter is suitable to solve the problem of low memory in computational environments. Therefore, this method is ideal for real-time processing systems, such as autonomous driving. Bewley et al. used the most common method [33] of the Kalman filter and the Hungarian algorithm. Weng and Kitani also followed Bewley et al.'s method [33], which performed in the 3D system [55]. Ray and Chakraborty used only the Kalman filter in their tracking system [44]. Pegoraro and Rossi used the Kalman filter, classifier, and radar point cloud in their tracking system [124]. In Hossain and Lee, they combined CNN and the Kalman filter [115] for tracking.

## 4. Multiple Object Tracking Benchmarks

Although there are various MOT-related benchmarks, this paper focuses on the MOT benchmark [8,141] and KITTI [142] benchmark, because they are both well-developed benchmarks and have been used to evaluate the performance of many state-of-the-art MOT models [141,142]. The examples of these benchmarks are displayed in Figure 18. Each benchmark will be addressed in detail in the following subsections.



**Figure 18.** MOT15 [8], MOT16 [141], MOT17, and KITTI [142] benchmark datasets.

### 4.1. KITTI Benchmark

The KITTI benchmark includes a set of vision tasks collected using an autonomous driving platform. KITTI includes 'Car', 'Van', 'Truck', 'Pedestrian', 'Person (sitting)', 'Cyclist', 'Tram', and 'Misc' classes. The main goal of the KITTI object tracking task is to calculate object tracklets only for the 'Car' and 'Pedestrian' classes. In total, the KITTI object tracking benchmark consists of 21 training sequences and 29 test sequences with a variety of data, such as left color image, right color images, Velodyne point clouds, GPS/IMU data, camera calibration metrics, L-SVM reference, and Region reference. In the Velodyne point clouds case, it contains manually labeled 3D points. All of the collected data and calibration were done by experts.

### 4.2. MOT Benchmark

The MOT benchmark has had many versions, which include MOT15 [8], MOT16 [141], MOT17, and others. Compared to other benchmarks, the MOT benchmark contains various sequences that are challenging for the MOT. Each sequence has fundamental information, such as frame number, identity number, bounding box left, bounding box top, bounding box width, bounding box height, confidence score, x position, y position, and z position [8].

#### 4.2.1. MOT15

MOT15 combines both the Performance Evaluation of Tracking and Surveillance (PETS) [12], and KITTI benchmarks. The weather condition of the training dataset includes cloudy, sunny, and night, yet the test dataset does not have a night weather dataset. The training set has 11 sequences, 5500 frames, 500 tracks, 39,905 boxes, and 7.3 density, while the testing set contains 11 sequences, 5783 frames, 721 tracks, 61,440 boxes, and 10.6 density. The dataset resolution is varied from  $640 \times 480$  to  $1920 \times 1080$ .

#### 4.2.2. MOT16

After MOT 15, MOT16 was introduced with some new improvements, which included the annotation of personal gestures. The training set has seven sequences, 5316 frames, 512 tracks, 110,407 boxes, and a density of 20.8, whereas the test set includes seven sequences, 5919 frames, 830 tracks, 182,326 boxes, and a density of 30.8. Moreover, with two videos excluded, the entire dataset was constructed from the start without the involvement of any other datasets. Finally, the MOT16 dataset has more weather conditions than the MOT15, which include cloudy, night, sunny, indoor in the training dataset, and cloudy, night, sunny, shadow, and indoor in the test dataset.

#### 4.2.3. MOT17

MOT17 uses the same video as MOT16 but has a different data structure. They focus on more accurate ground truth from MOT16 benchmark raw data. MOT17 has 21 sequences in the training set and the test set, with 15,948 frames, 1638 for the training set, 336,891 for the box, 21.1 for the Density, 17,757 for the test set, 2355 for the track, and 31.8 for the Density.

### 5. Evaluation of Multiple Objects Tracking Benchmark Datasets

Although there are various evaluation metrics for the MOT, MOTA, and MOTP [143] are the two most important metrics that show a tracker's characteristics and can be calculated for evaluating MOT performance. This section describes how the previous state-of-the-art systems were evaluated using those evaluation metrics on the two benchmark datasets (KITTI and MOT).

Firstly, MOTA combines three types of tracking errors, as described below.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (1)$$

where  $t$  is a frame.  $m_t$  is the respectively misses for frame  $t$ .  $fp_t$  is the number of false positives.  $mme_t$  is the number of mismatch errors for frame  $t$ .  $g_t$  is the total number of objects at time  $t$ .

Secondly, MOTP can be computed to check whether the tracker performed properly or not.

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t} \quad (2)$$

$c_t$  is the number of matches found for time  $t$ .  $d_{i,t}$  is the distance between the object and corresponding hypothesis.

Table 3 shows that for the KITTI benchmark, RRC-IITH [62] achieved the highest MOTP of 85.73 and ML of 2.77. PC3T model [144] also obtained good performance with the MOTA of 88.8, MT of 80, and high fps at 222. On the other hand, AB3DMOT [145] did not report the ID switch, and FR was low at 15.

**Table 3.** The previous state-of-the-art models on the KITTI benchmark.

Name	Year	Class	MOTA	MOTP	IDF1	IDS	MT	ML	FR	Tracking Time(s)	fps	Ref.
	2018		32.7		38.9		26.2	19.6		0.09		[29]
	2018	car class	80.4	81.3		121	62.8	6.2	613		73	[49]
	2019	car class	83.34	85.23		10	65.85	11.54	222		214.7	[55]
RRC-IIITH	2018		84.24	85.73		468	73.23	2.77	944			[62]
	2017		67.36	78.79		65	53.81	9.45	574			[83]
PC3T	2021		88.8	84.37		208	80	8.31	369		222	[144]
DiTNet	2021		84.62	84.18		19	74.15	12.92	196	0.01		[146]
AB3DMOT	2021	car class	86.24	78.43		0			15			[145]

Note: IDSw and IDS are the total numbers of switching ID. IDF1 is F1 Score [147] of ID, which is the ratio to the average number of correctly identified and calculated detection. MT is the percentage of trajectories tracked over some ratio of time. ML is the opposite of MT, that is, lost targets. Fps is the number of frames that can be processed in one second. FP is the number of false positives, and FN is the number of false negatives. Frag and FR are interrupted the number of objects. Hz is speed about the process.

The performances of the previous systems on the MOT benchmark datasets are shown in Table A1 from the Appendix A. For the MOT15 dataset, FairMOT [148] demonstrated the highest performance in terms of the evaluation metrics with MOTA of 59, IDF1 of 62.2, MT of 45.6, and ML of 11.5. The SORT [33] algorithm is a widely accepted real-time tracking algorithm, because it showed an impressive component speed of 260 Hz and small FN of 11.7. STAM [58] achieved the best IDSw 348, whereas EA-PHD-PF [53] obtained the highest MOTP of 75.3 and Frag of 1269. DAN [28] showed that it had the best FP of 1290.25.

For the MOT16 benchmark dataset, RelationTrack [56] achieved high performance with MOTA of 75.6, MOTP of 80.9, FN of 34214, and MT of 43.1. EA-PHD-PF [53], on the other hand, focused on the detector, which got the lowest FP of 407. oICF [81] obtained the best IDSw of 380, while JDE [70] achieved the highest ML of 16.7 and FPS of 30.3. FairMOT [148] performed well and achieved the highest IDF1 of 70.4 in the MOT16 benchmark.

For the MOT17 benchmark dataset, the RelationTrack [56] model continued to show the best performance among the existing models, with MOTA of 75.6, MOTP of 80.9, IDF1 of 75.8, MT of 43.1, and FN of 34,214. GAN et al. implemented long occlusion handling [68], which obtained IDSw of 560, FP of 7912, and Frag of 1212.

## 6. MOT Trends

Although object detection and object tracking have been studied for a long time, the recent development of deep learning and computer vision has led to more advanced models being introduced in order to solve some existing challenges that the previous models failed to address. Nevertheless, there remain several challenges in the MOT that need to be addressed. Figure 19 shows various MOT trends that are attracting the research community.

Figure 19 and Table 4 demonstrates that, in general, detection, appearance, and affinity are the most common trends, which are implemented to solve MOT problems. Detection/prediction is the most common trend lately, which took 32% of the total related research published in 2020. Another interesting trend is affinity, which has witnessed a stable rise since 2019 and becomes the most prevalent MOT trend in 2021. Most MOT research tried to construct a robust detector because it significantly affects the tracking performance. Basic trackers used only the object's location to estimate the direction.

However, as it is challenging to track only this information, tracking with the appearance category was proved to offer better performance and has thus received more attention. Most research from the computational costs minimization trend was used in autonomous driving cars and Internet of Things (IoT) devices [149].

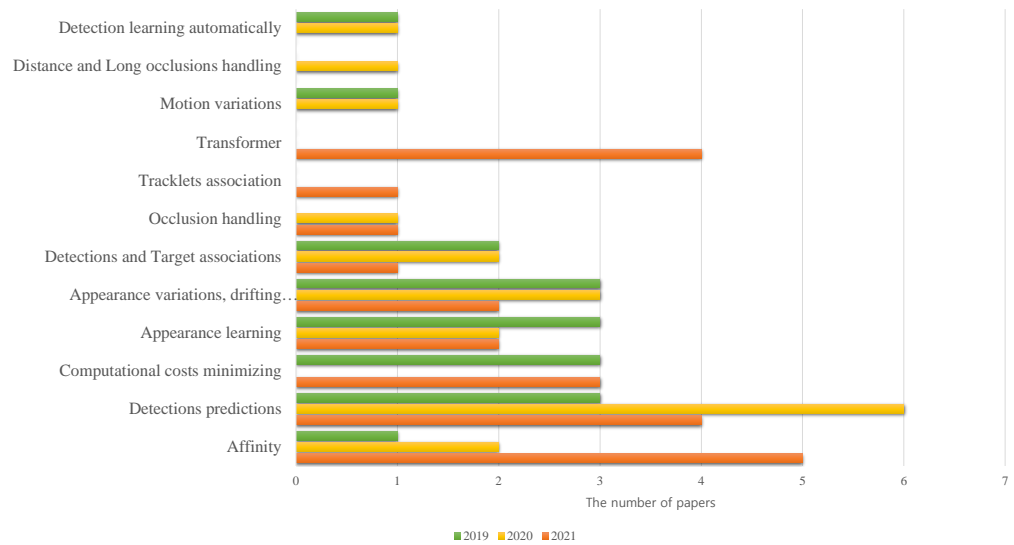


Figure 19. Notable MOT trends from the last three years (2019–2021).

Table 4. Organized issues by trends in MOT papers from the past 3 years.

Category	Issue	2019	2020	2021
Affinity	The number of papers is increasing	6%	10%	22%
Appearance learning		17%	11%	9%
Appearance Variations, Drifting, and Identity Switching	<ul style="list-style-type: none"> <li>An important factor in identifying the id information of the object.</li> <li>A lot of relevant papers.</li> </ul>	17%	16%	9%
Detection prediction		18%	32%	18%
Detection and target associations	<ul style="list-style-type: none"> <li>Detection performance affects the tracking</li> <li>Most of the papers attempted to make strong detectors for the tracking systems</li> </ul>	12%	11%	4%
Automatic Detection Learning		6%	5%	-
Transformer	Models including deep learning and transformers have been published recently	-	-	17%

The Kalman filter and Hungarian algorithm are used for speed improvement because they could effectively construct light detectors. Unlike conventional methods that used application features only using CNN, recent methods proposed focusing on essential features through an attention mechanism. A new method that became available in 2021 is the application of the transformer [35]. Even though this category was newly introduced in 2021, it appears that a large number of papers were published.

### 7. Conclusions

Organizations and research communities worldwide are closely collaborating to revolutionize how visual systems track various moving objects. This research is helpful for readers who are interested in studying the object tracking problem, especially MOT.

Driven by the ongoing developments of object tracking, especially MOT, this study offers a comprehensive view of the field with up-to-date information for (i) MOT's main approaches and the common techniques for each approach, which include state-of-the-art results achieved by the most representative techniques, (ii) benchmark datasets and evaluation methods for the MOT research, and (iii) several challenges in the current MOT research and many open problems to be studied.

In particular, this paper analyzed two main problems of the MOT, which included the occlusion problem and the identity switch problem. Moreover, this review concentrated on studying the latest deep learning-based methods that were proposed to solve those problems efficiently. After that, the standard MOT benchmark datasets and evaluation techniques were listed and discussed in detail. Finally, this survey analyzed the main challenges of MOT and provided potential techniques that can be further studied to cope with the challenges.

**Author Contributions:** Conceptualization Y.P. and L.M.D.; Data curation L.M.D. and D.H.; Methodology S.L.; Validation S.L.; Visualization D.H.; Supervision H.M.; Writing—original draft preparation, Y.P.; Writing—review and editing, L.M.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable

**Informed Consent Statement:** Not applicable.

**Acknowledgments:** This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540) and National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (2021R1F1A1046339).

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

## Appendix A

Table A1. Performance on the MOT benchmark, which includes the MOT15 [8], MOT16 [141], and MOT17 benchmarks.

Benchmark	Name	Year	MOTA	MOTP	IDF1	IDS <sub>w</sub>	IR	IS	FM	FAF	FP	FN	MT	ML	Frag	Hz	FPS	Ref.
MOT15	LINF1	2016	40.5	74.9	NA	426	9.4	NA	953	1.4	8401	99,715	10.7	56.1	NA	NA	NA	[42]
	DAN	2019	38.3	71.1	45.6	1648.08	NA	NA	NA	NA	1290.25	2700	17.6	41.2	1515.6	6.3	NA	[28]
	MDP OFL	2015	30.1	71.6	NA	690	NA	NA	NA	NA	8789	33,479	10.4	41.3	1301	0.8	NA	[46]
	MDP REL	2015	30.3	71.3	NA	680	NA	NA	NA	NA	9717	32,422	13	38.4	1500	1.1	NA	[46]
		2018	32.7	NA	38.9	NA	NA	NA	NA	NA	NA	NA	26.2	19.6	NA	NA	NA	[29]
	EA-PHD-PF	2016	53	75.3	NA	776	NA	NA	NA	1.3	7538	20,590	35.9	19.6	1269	NA	NA	[53]
	C-DRL	2018	37.1	71	NA	NA	NA	NA	NA	1.2	7036	30,440	14	31.3	NA	NA	NA	[51]
	STAM	2017	34.3	70.5	NA	348	NA	NA	NA	NA	5154	34,848	11.4	43.4	1463	NA	NA	[58]
	CCC	2018	35.6	NA	45.1	457	NA	NA	969	NA	10,580	28,508	23.2	39.3	NA	NA	NA	[63]
	SORT	2016	33.4	72.1	NA	1001	NA	NA	NA	1.3	7318	11.7	30.9	NA	1764	260	NA	[33]
FairMOT	2020	59	NA	62.2	582	NA	NA	NA	NA	NA	NA	45.6	11.5	NA	30.5	NA	[148]	
MOT16	LINF1	2016	40.5	74.9	NA	426	9.4	NA	953	1.4	8401	99,715	10.7	56.1	NA	NA	NA	[42]
	EA-PHD-PF	2016	52.5	78.8	NA	910	NA	NA	NA	0.7	4407	81,223	19	34.9	1321	12.2	NA	[53]
	C-DRL	2018	47.3	74.6	NA	NA	NA	NA	NA	1.1	6375	88,543	17.4	39.9	NA	NA	NA	[51]
	STAM	2018	46.0	74.9	NA	473	NA	NA	NA	NA	6895	91,117	14.6	43.6	1422	NA	NA	[58]
	RelationTrack	2021	75.6	80.9	NA	448	NA	NA	NA	NA	9786	34,214	43.1	21.5	NA	NA	NA	[56]
	TNT	2019	49.2	NA	NA	606	NA	NA	NA	NA	8400	83,702	17.3	40.3	882	NA	NA	[40]
	DMAN	2018	46.1	73.8	NA	532	NA	NA	NA	NA	7909	89,874	17.4	42.7	1616	NA	NA	[32]
	Deep SORT	2017	61.4	79.1	NA	781	NA	NA	2008	NA	12,852	56,668	32.8	18.2	NA	20	NA	[4]
	Deep-TAMA	2021	46.2	NA	NA	598	NA	NA	1127	NA	5126	92,367	14.1	44	NA	NA	2	[66]
	CSTrack++	2020	70.7	NA	NA	1071	NA	NA	NA	NA	NA	NA	38.2	17.8	NA	NA	15.8	[31]
		2018	44	78.3	NA	560	NA	NA	NA	NA	7912	93,215	15.2	45.7	1212	NA	NA	[68]
	JDE	2019	62.1	NA	NA	1608	NA	NA	NA	NA	NA	NA	34.4	16.7	NA	NA	30.3	[70]
	oICF	2016	42.8	74.3	NA	380	NA	NA	NA	NA	NA	NA	10.4	53.1	1397	NA	NA	[81]
	MCMOT HDM	2016	62.4	78.3	NA	1394	NA	NA	NA	1.7	9855	57,257	31.5	24.2	1318	34.9	NA	[86]
	MOTDT	2018	47.6	NA	NA	792	NA	NA	NA	NA	9253	85,431	15.2	38.3	NA	NA	20.6	[84]
FairMOT	2020	68.7	NA	70.4	953	NA	NA	NA	NA	NA	NA	39.5	19	NA	25.9	NA	[148]	



Table A1. Cont.

Benchmark	Name	Year	MOTA	MOTP	IDF1	IDS <sub>w</sub>	IR	IS	FM	FAF	FP	FN	MT	ML	Frag	Hz	FPS	Ref.
MOT17	EB+DAN	2019	53.5	NA	62.3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	[28]
	RelationTrack	2021	75.6	80.9	75.8	448	NA	7.4	NA	NA	9786	34,214	43.1	21.5	NA	NA	NA	[56]
	TNT	2019	51.9	NA	58	2294	NA	NA	NA	NA	37,311	231,658	23.5	35.5	2917	NA	NA	[40]
	DMAN	2018	48.2	75.9	NA	2194	NA	NA	NA	NA	26,218	263,608	19.3	38.3	5378	NA	NA	[32]
		2021	50.3	NA	53.5	2192	NA	NA	NA	NA	25,479	252,996	19.2	37.5	NA	NA	1.5	[66]
		2020	70.6	NA	71.6	3465	NA	NA	NA	NA	NA	NA	37.5	18.7	NA	NA	15.8	[31]
		2018	44	78.3	NA	560	NA	NA	NA	NA	7912	93,215	15.2	45.7	1212	NA	NA	[68]
	FairMOT	2020	67.5	NA	69.8	2868	NA	NA	NA	NA	NA	NA	37.7	20.8	NA	25.9	NA	[148]

Note: IDS<sub>w</sub> and IDS are the total numbers of switching ID. IDF1 is the F1 Score [147] of ID, which is the ratio to the average number of correctly identified and calculated detection. MT is the percentage of trajectories tracked over some ratio of time. ML is the opposite of MT, that is, lost targets. Fps is the number of frames that can be processed in one second. FP is the number of false positives, and FN is the number of false negatives. Frag and FR are interrupted the number of objects. Hz is the speed of the process.

## References

1. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27 June 2016; pp. 770–778.
3. Gui, Y.; Zhou, B.; Xiong, D.; Wei, W. Fast and robust interactive image segmentation in bilateral space with reliable color modeling and higher order potential. *J. Electron. Imaging* **2021**, *30*, 033018. [[CrossRef](#)]
4. 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.
5. Wang, H.; Li, Y.; Dang, L.; Moon, H. Robust Korean License Plate Recognition Based on Deep Neural Networks. *Sensors* **2021**, *21*, 4140. [[CrossRef](#)]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [[CrossRef](#)] [[PubMed](#)]
7. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
8. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv* **2015**, arXiv:1504.01942.
9. Pal, S.K.; Pramanik, A.; Maiti, J.; Mitra, P. Deep learning in multi-object detection and tracking: State of the art. *Appl. Intell.* **2021**, *51*, 6400–6429. [[CrossRef](#)]
10. Ciaparrone, G.; Sánchez, F.L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020**, *381*, 61–88. [[CrossRef](#)]
11. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.K. Multiple object tracking: A literature review. *Artif. Intell.* **2020**, *293*, 103448. [[CrossRef](#)]
12. Ellis, A.; Ferryman, J. PETS2010 and PETS2009 evaluation of results using individual ground truthed single views. In Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, Boston, MA, USA, 29 August 2010; pp. 135–142.
13. Kalake, L.; Wan, W.; Hou, L. Analysis Based on Recent Deep Learning Approaches Applied in Real-Time Multi-Object Tracking: A Review. *IEEE Access* **2021**, *9*, 32650–32671. [[CrossRef](#)]
14. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G.; Prisma Group. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Med.* **2009**, *6*, e1000097. [[CrossRef](#)] [[PubMed](#)]
15. Dang, L.M.; Min, K.; Wang, H.; Piran, M.J.; Lee, C.H.; Moon, H. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognit.* **2020**, *108*, 107561. [[CrossRef](#)]
16. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Technical Report; University of North Carolina: Chapel Hill, NC, USA, 1995.
17. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [[CrossRef](#)]
18. Huo, W.; Ou, J.; Li, T. Multi-target tracking algorithm based on deep learning. *J. Phys. Conf. Ser. IOP Publ.* **2021**, *1948*, 012011. [[CrossRef](#)]
19. Milan, A.; Rezatofighi, S.H.; Dick, A.; Reid, I.; Schindler, K. Online multi-target tracking using recurrent neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
20. Tian, Y.; Dehghan, A.; Shah, M. On detection, data association and segmentation for multi-target tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2146–2160. [[CrossRef](#)]
21. Ullah, M.; Alaya Cheikh, F. A directed sparse graphical model for multi-target tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18 June 2018; pp. 1816–1823.
22. Mousavi, H.; Nabi, M.; Kiani, H.; Perina, A.; Murino, V. Crowd motion monitoring using tracklet-based commotion measure. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Québec City, QC, Canada, 27–30 September 2015; pp. 2354–2358.
23. Butt, A.A.; Collins, R.T. Multiple target tracking using frame triplets. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 163–176.
24. Wei, J.; Yang, M.; Liu, F. Learning spatio-temporal information for multi-object tracking. *IEEE Access* **2017**, *5*, 3869–3877. [[CrossRef](#)]
25. Rodriguez, P.; Wiles, J.; Elman, J.L. A recurrent neural network that learns to count. *Connect. Sci.* **1999**, *11*, 5–40. [[CrossRef](#)]
26. Lee, H.; Grosse, R.; Ranganath, R.; Ng, A.Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 609–616.
27. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
28. Sun, S.; Akhtar, N.; Song, H.; Mian, A.; Shah, M. Deep affinity network for multiple object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 104–119. [[CrossRef](#)]
29. Zhao, D.; Fu, H.; Xiao, L.; Wu, T.; Dai, B. Multi-object tracking with correlation filter for autonomous vehicle. *Sensors* **2018**, *18*, 2004. [[CrossRef](#)]
30. Khan, G.; Tariq, Z.; Khan, M.U.G. Multi-person tracking based on faster R-CNN and deep appearance features. In *Visual Object Tracking with Deep Neural Networks*; IntechOpen: London, UK, 2019. [[CrossRef](#)]

31. Liang, C.; Zhang, Z.; Lu, Y.; Zhou, X.; Li, B.; Ye, X.; Zou, J. Rethinking the competition between detection and ReID in Multi-Object Tracking. *arXiv* **2020**, arXiv:2010.12138.
32. Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; Yang, M.H. Online multi-object tracking with dual matching attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 366–382.
33. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
34. Bochinski, E.; Sens, T.; Sikora, T. Extending IOU based multi-object tracking by visual information. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–6.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
36. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21 July 2017; pp. 3156–3164.
37. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18 June 2018; pp. 7132–7141.
38. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
39. You, Q.; Jin, H.; Wang, Z.; Fang, C.; Luo, J. Image captioning with semantic attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Amsterdam, The Netherlands, 8–16 October 2016; pp. 4651–4659.
40. Wang, G.; Wang, Y.; Zhang, H.; Gu, R.; Hwang, J.N. Exploit the connectivity: Multi-object tracking with trackletnet. In Proceedings of the 27th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2019; pp. 482–490.
41. Ning, G.; Zhang, Z.; Huang, C.; Ren, X.; Wang, H.; Cai, C.; He, Z. Spatially supervised recurrent convolutional neural networks for visual object tracking. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4.
42. Fagot-Bouquet, L.; Audigier, R.; Dhome, Y.; Lerasle, F. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 774–790.
43. Zamir, A.R.; Dehghan, A.; Shah, M. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 343–356.
44. Ray, K.S.; Chakraborty, S. An efficient approach for object detection and tracking of objects in a video with variable background. *arXiv* **2017**, arXiv:1706.02672.
45. Kutschbach, T.; Bochinski, E.; Eiselein, V.; Sikora, T. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–5.
46. Xiang, Y.; Alahi, A.; Savarese, S. Learning to track: Online multi-object tracking by decision making. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7 December 2015; pp. 4705–4713.
47. Williams, J.L. Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based MeMBeR. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 1664–1687. [[CrossRef](#)]
48. García-Fernández, Á.F.; Williams, J.L.; Granström, K.; Svensson, L. Poisson multi-Bernoulli mixture filter: Direct derivation and implementation. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 1883–1901. [[CrossRef](#)]
49. Scheidegger, S.; Benjaminsson, J.; Rosenberg, E.; Krishnan, A.; Granström, K. Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, Suzhou, China, 26–30 June 2018; pp. 433–440.
50. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. Siamrpn++: Evolution of Siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4282–4291.
51. Ren, L.; Lu, J.; Wang, Z.; Tian, Q.; Zhou, J. Collaborative deep reinforcement learning for multi-object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 586–602.
52. Zhang, Z.; Wu, J.; Zhang, X.; Zhang, C. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *arXiv* **2017**, arXiv:1712.09531.
53. Sanchez-Matilla, R.; Poiesi, F.; Cavallaro, A. Online multi-target tracking with strong and weak detections. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 84–99.
54. Milan, A.; Leal-Taixé, L.; Schindler, K.; Reid, I. Joint tracking and segmentation of multiple targets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7 June 2015; pp. 5397–5406.
55. Weng, X.; Kitani, K. A baseline for 3d multi-object tracking. *arXiv* **2019**, arXiv:1907.03961.
56. Yu, E.; Li, Z.; Han, S.; Wang, H. RelationTrack: Relation-aware Multiple Object Tracking with Decoupled Representation. *arXiv* **2021**, arXiv:2105.04322.

57. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A.; Leibe, B. Mots: Multi-object tracking and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 20 June 2019; pp. 7942–7951.
58. 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 October 2017; pp. 4836–4845.
59. Tang, S.; Andres, B.; Andriluka, M.; Schiele, B. Subgraph decomposition for multi-target tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7 June 2015; pp. 5033–5041.
60. Bochinski, E.; Eiselein, V.; Sikora, T. High-speed tracking-by-detection without using image information. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
61. Shin, J.; Kim, H.; Kim, D.; Paik, J. Fast and robust object tracking using tracking failure detection in kernelized correlation filter. *Appl. Sci.* **2020**, *10*, 713. [[CrossRef](#)]
62. Sharma, S.; Ansari, J.A.; Murthy, J.K.; Krishna, K.M. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3508–3515.
63. Keuper, M.; Tang, S.; Andres, B.; Brox, T.; Schiele, B. Motion segmentation and multiple object tracking by correlation co-clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 140–153. [[CrossRef](#)]
64. Chen, L.; Ren, M. Multi-appearance segmentation and extended 0–1 programming for dense small object tracking. *PLoS ONE* **2018**, *13*, e0206168. [[CrossRef](#)] [[PubMed](#)]
65. Ruchay, A.; Kober, V.; Chernoskulov, I. Real-time tracking of multiple objects with locally adaptive correlation filters. In Proceedings of the Information Technology and Nanotechnology 2017, Samara, Russia, 25–27 April 2017; pp. 214–218.
66. Yoon, Y.C.; Kim, D.Y.; Song, Y.M.; Yoon, K.; Jeon, M. Online multiple pedestrians tracking using deep temporal appearance matching association. *Inf. Sci.* **2021**, *561*, 326–351. [[CrossRef](#)]
67. Xiang, J.; Zhang, G.; Hou, J. Online multi-object tracking based on feature representation and Bayesian filtering within a deep learning architecture. *IEEE Access* **2019**, *7*, 27923–27935. [[CrossRef](#)]
68. Gan, W.; Wang, S.; Lei, X.; Lee, M.S.; Kuo, C.C.J. Online CNN-based multiple object tracking with enhanced model updates and identity association. *Signal Process. Image Commun.* **2018**, *66*, 95–102. [[CrossRef](#)]
69. Kampker, A.; Sefati, M.; Rachman, A.S.A.; Kreisköther, K.; Campoy, P. Towards Multi-Object Detection and Tracking in Urban Scenario under Uncertainties. In Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), Funchal, Madeira, Portugal, 16–18 March 2018; pp. 156–167.
70. Wang, Z.; Zheng, L.; Liu, Y.; Wang, S. Towards real-time multi-object tracking. *arXiv* **2019**, arXiv:1909.12605.
71. Baisa, N.L. Online multi-object visual tracking using a GM-PHD filter with deep appearance learning. In Proceedings of the 2019 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019; pp. 1–8.
72. Ju, J.; Kim, D.; Ku, B.; Han, D.K.; Ko, H. Online multi-object tracking with efficient track drift and fragmentation handling. *JOSA A* **2017**, *34*, 280–293. [[CrossRef](#)] [[PubMed](#)]
73. Yoon, Y.C.; Song, Y.M.; Yoon, K.; Jeon, M. Online multi-object tracking using selective deep appearance matching. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Jeju, Korea, 24–26 June 2018; pp. 206–212.
74. Bae, S.H.; Yoon, K.J. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 595–610. [[CrossRef](#)]
75. KC, A.K.; Jacques, L.; De Vleeschouwer, C. Discriminative and efficient label propagation on complementary graphs for multi-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 61–74.
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 June 2016; pp. 33–40.
77. Jiang, X.; Fang, Z.; Xiong, N.N.; Gao, Y.; Huang, B.; Zhang, J.; Yu, L.; Harrington, P. Data fusion-based multi-object tracking for unconstrained visual sensor networks. *IEEE Access* **2018**, *6*, 13716–13728. [[CrossRef](#)]
78. Weng, X.; Yuan, Y.; Kitani, K. PTP: Parallelized Tracking and Prediction with Graph Neural Networks and Diversity Sampling. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4640–4647. [[CrossRef](#)]
79. Le, Q.C.; Conte, D.; Hidane, M. Online multiple view tracking: Targets association across cameras. In Proceedings of the 6th Workshop on Activity Monitoring by Multiple Distributed Sensing (AMMDS 2018), Newcastle upon Tyne, UK, 6 September 2018.
80. Wu, C.W.; Zhong, M.T.; Tsao, Y.; Yang, S.W.; Chen, Y.K.; Chien, S.Y. Track-clustering error evaluation for track-based multi-camera tracking system employing human re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21 July 2017; pp. 1–9.
81. Kieritz, H.; Becker, S.; Hübner, W.; Arens, M. Online multi-person tracking using integral channel features. In Proceedings of the 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Colorado Springs, CO, USA, 23 August 2016; pp. 122–130.
82. Scheel, A.; Knill, C.; Reuter, S.; Dietmayer, K. Multi-sensor multi-object tracking of vehicles using high-resolution radars. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gotenburg, Sweden, 19–22 June 2016; pp. 558–565.

83. Schuster, S.; Vernaza, P.; Choi, W.; Chandraker, M. Deep network flow for multi-object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21 July 2017; pp. 6951–6960.
84. Chen, L.; Ai, H.; Zhuang, Z.; Shang, C. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
85. 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 July 2017; pp. 5620–5629.
86. Lee, B.; Erdenee, E.; Jin, S.; Nam, M.Y.; Jung, Y.G.; Rhee, P.K. Multi-class multi-object tracking using changing point detection. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 68–83.
87. Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; Alameda-Pineda, X. TransCenter: Transformers with Dense Queries for Multiple-Object Tracking. *arXiv* **2021**, arXiv:2103.15145.
88. Sun, P.; Jiang, Y.; Zhang, R.; Xie, E.; Cao, J.; Hu, X.; Kong, T.; Yuan, Z.; Wang, C.; Luo, P. Transtrack: Multiple-object tracking with transformer. *arXiv* **2020**, arXiv:2012.15460.
89. Zeng, F.; Dong, B.; Wang, T.; Chen, C.; Zhang, X.; Wei, Y. MOTR: End-to-End Multiple-Object Tracking with TRansformer. *arXiv* **2021**, arXiv:2105.03247.
90. Zhang, J.; Sun, J.; Wang, J.; Yue, X.G. Visual object tracking based on residual network and cascaded correlation filters. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *12*, 8427–8440. [[CrossRef](#)]
91. Dang, L.M.; Kyeong, S.; Li, Y.; Wang, H.; Nguyen, T.N.; Moon, H. Deep Learning-based Sewer Defect Classification for Highly Imbalanced Dataset. *Comput. Ind. Eng.* **2021**, *161*, 107630. [[CrossRef](#)]
92. 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.
93. Azimi, S.M.; Kraus, M.; Bahmanyar, R.; Reinartz, P. Multiple Pedestrians and Vehicles Tracking in Aerial Imagery Using a Convolutional Neural Network. *Remote Sens.* **2021**, *13*, 1953. [[CrossRef](#)]
94. Zhou, Y.F.; Xie, K.; Zhang, X.Y.; Wen, C.; He, J.B. Efficient Traffic Accident Warning Based on Unsupervised Prediction Framework. *IEEE Access* **2021**, *9*, 69100–69113. [[CrossRef](#)]
95. Zhang, D.; Zheng, Z.; Wang, T.; He, Y. HROM: Learning High-Resolution Representation and Object-Aware Masks for Visual Object Tracking. *Sensors* **2020**, *20*, 4807. [[CrossRef](#)] [[PubMed](#)]
96. Tang, Y.; Liu, Y.; Huang, H.; Liu, J.; Xie, W. A Scale-Adaptive Particle Filter Tracking Algorithm Based on Offline Trained Multi-Domain Deep Network. *IEEE Access* **2020**, *8*, 31970–31982. [[CrossRef](#)]
97. 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](#)]
98. Hidayatullah, P.; Mengko, T.L.; Munir, R.; Barlian, A. Bull Sperm Tracking and Machine Learning-Based Motility Classification. *IEEE Access* **2021**, *9*, 61159–61170. [[CrossRef](#)]
99. Xia, H.; Zhang, Y.; Yang, M.; Zhao, Y. Visual tracking via deep feature fusion and correlation filters. *Sensors* **2020**, *20*, 3370. [[CrossRef](#)] [[PubMed](#)]
100. Madasamy, K.; Shanmuganathan, V.; Kandasamy, V.; Lee, M.Y.; Thangadurai, M. OSDDY: Embedded system-based object surveillance detection system with small drone using deep YOLO. *EURASIP J. Image Video Process.* **2021**, *2021*, 1–14. [[CrossRef](#)]
101. Dao, M.Q.; Frémont, V. A two-stage data association approach for 3D Multi-object Tracking. *Sensors* **2021**, *21*, 2894. [[CrossRef](#)] [[PubMed](#)]
102. Yin, Y.; Feng, X.; Wu, H. Learning for Graph Matching based Multi-object Tracking in Auto Driving. *J. Phys. Conf. Ser. IOP Publ.* **2021**, *1871*, 012152. [[CrossRef](#)]
103. Song, S.; Li, Y.; Huang, Q.; Li, G. A New Real-Time Detection and Tracking Method in Videos for Small Target Traffic Signs. *Appl. Sci.* **2021**, *11*, 3061. [[CrossRef](#)]
104. Padmaja, B.; Myneni, M.B.; Patro, E.K.R. A comparison on visual prediction models for MAMO (multi activity-multi object) recognition using deep learning. *J. Big Data* **2020**, *7*, 1–15. [[CrossRef](#)]
105. Chou, Y.S.; Wang, C.Y.; Lin, S.D.; Liao, H.Y.M. How Incompletely Segmented Information Affects Multi-Object Tracking and Segmentation (MOTS). In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 2086–2090.
106. Zhou, Y.; Cai, Z.; Zhu, Y.; Yan, J. Automatic ship detection in SAR Image based on Multi-scale Faster R-CNN. *J. Phys. Conf. Ser. IOP Publ.* **2020**, *1550*, 042006. [[CrossRef](#)]
107. Liu, Y.; Zhang, S.; Li, Z.; Zhang, Y. Abnormal Behavior Recognition Based on Key Points of Human Skeleton. *IFAC PapersOnLine* **2020**, *53*, 441–445. [[CrossRef](#)]
108. Xie, Y.; Shen, J.; Wu, C. Affine Geometrical Region CNN for Object Tracking. *IEEE Access* **2020**, *8*, 68638–68648. [[CrossRef](#)]
109. Shao, Q.; Hu, J.; Wang, W.; Fang, Y.; Xue, T.; Qi, J. Location Instruction-Based Motion Generation for Sequential Robotic Manipulation. *IEEE Access* **2020**, *8*, 26094–26106. [[CrossRef](#)]
110. Nobis, F.; Geisslinger, M.; Weber, M.; Betz, J.; Lienkamp, M. A deep learning-based radar and camera sensor fusion architecture for object detection. In Proceedings of the 2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 15–17 October 2019; pp. 1–7.

111. Wu, Q.; Yan, Y.; Liang, Y.; Liu, Y.; Wang, H. DSNet: Deep and shallow feature learning for efficient visual tracking. In Proceedings of the Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Springer: Cham, Switzerland, 2018; pp. 119–134.
112. Zhu, W.; Yu, S.; Zheng, X.; Wu, Y. Fine-grained Vehicle Classification Technology Based on Fusion of Multi-convolutional Neural Networks. *Sens. Mater.* **2019**, *31*, 569–578. [[CrossRef](#)]
113. Avola, D.; Cinque, L.; Diko, A.; Fagioli, A.; Foresti, G.L.; Mecca, A.; Pannone, D.; Piciarelli, C. MS-Faster R-CNN: Multi-Stream Backbone for Improved Faster R-CNN Object Detection and Aerial Tracking from UAV Images. *Remote Sens.* **2021**, *13*, 1670. [[CrossRef](#)]
114. Zhou, X.; Xu, X.; Liang, W.; Zeng, Z.; Yan, Z. Deep Learning Enhanced Multi-Target Detection for End-Edge-Cloud Surveillance in Smart IoT. *IEEE Internet Things J.* **2021**, *8*, 12588–12596. [[CrossRef](#)]
115. Hossain, S.; Lee, D.J. Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. *Sensors* **2019**, *19*, 3371. [[CrossRef](#)] [[PubMed](#)]
116. He, Z.; Li, J.; Liu, D.; He, H.; Barber, D. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15 June 2019; pp. 1318–1327.
117. Lee, S.; Kim, E. Multiple object tracking via feature pyramid Siamese networks. *IEEE Access* **2018**, *7*, 8181–8194. [[CrossRef](#)]
118. Dike, H.U.; Zhou, Y. A Robust Quadruplet and Faster Region-Based CNN for UAV Video-Based Multiple Object Tracking in Crowded Environment. *Electronics* **2021**, *10*, 795. [[CrossRef](#)]
119. Gómez-Silva, M.J.; Escalera, A.D.L.; Armingol, J.M. Deep Learning of Appearance Affinity for Multi-Object Tracking and Re-Identification: A Comparative View. *Electronics* **2020**, *9*, 1757. [[CrossRef](#)]
120. Li, J.; Zhan, W.; Hu, Y.; Tomizuka, M. Generic tracking and probabilistic prediction framework and its application in autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3634–3649. [[CrossRef](#)]
121. Lv, X.; Dai, C.; Chen, L.; Lang, Y.; Tang, R.; Huang, Q.; He, J. A robust real-time detecting and tracking framework for multiple kinds of unmarked object. *Sensors* **2020**, *20*, 2. [[CrossRef](#)]
122. Xu, T.; Feng, Z.H.; Wu, X.J.; Kittler, J. Joint group feature selection and discriminative filter learning for robust visual object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October 2019; pp. 7950–7960.
123. Shahbazi, M.; Simeonova, S.; Lichti, D.; Wang, J. Vehicle Tracking and Speed Estimation from Unmanned Aerial Videos. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 623–630. [[CrossRef](#)]
124. Pegoraro, J.; Rossi, M. Real-time People Tracking and Identification from Sparse mm-Wave Radar Point-clouds. *IEEE Access* **2021**, *4*, 78504–78520. [[CrossRef](#)]
125. Liu, K. Deep Associated Elastic Tracker for Intelligent Traffic Intersections. In Proceedings of the 2nd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things, Seoul, Korea, 16 November 2020; pp. 55–61.
126. Wen, A. Real-Time Panoramic Multi-Target Detection Based on Mobile Machine Vision and Deep Learning. *J. Phys. Conf. Ser. IOP Publ.* **2020**, *1650*, 032113. [[CrossRef](#)]
127. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *arXiv* **2014**, arXiv:1411.1792.
128. Xu, T.; Feng, Z.; Wu, X.J.; Kittler, J. Adaptive Channel Selection for Robust Visual Object Tracking with Discriminative Correlation Filters. *Int. J. Comput. Vis.* **2021**, *129*, 1359–1375. [[CrossRef](#)]
129. Huang, W.; Zhou, X.; Dong, M.; Xu, H. Multiple objects tracking in the UAV system based on hierarchical deep high-resolution network. *Multimed. Tools Appl.* **2021**, *80*, 13911–13929. [[CrossRef](#)]
130. Li, Y.; Wang, H.; Dang, L.M.; Nguyen, T.N.; Han, D.; Lee, A.; Jang, I.; Moon, H. A deep learning-based hybrid framework for object detection and recognition in autonomous driving. *IEEE Access* **2020**, *8*, 194228–194239. [[CrossRef](#)]
131. Chen, C.; Zhanotti Fragonara, L.; Tsourdos, A. Relation3DMOT: Exploiting Deep Affinity for 3D Multi-Object Tracking from View Aggregation. *Sensors* **2021**, *21*, 2113. [[CrossRef](#)]
132. Wang, N.; Zhou, W.; Song, Y.; Ma, C.; Liu, W.; Li, H. Unsupervised deep representation learning for real-time tracking. *Int. J. Comput. Vis.* **2021**, *129*, 400–418. [[CrossRef](#)]
133. Wu, L.; Xu, T.; Zhang, Y.; Wu, F.; Xu, C.; Li, X.; Wang, J. Multi-Channel Feature Dimension Adaption for Correlation Tracking. *IEEE Access* **2021**, *9*, 63814–63824. [[CrossRef](#)]
134. Yang, Y.; Xing, W.; Zhang, S.; Gao, L.; Yu, Q.; Che, X.; Lu, W. Visual tracking with long-short term based correlation filter. *IEEE Access* **2020**, *8*, 20257–20269. [[CrossRef](#)]
135. Mauri, A.; Khemmar, R.; Decoux, B.; Ragot, N.; Rossi, R.; Trabelsi, R.; Bouteau, R.; Ertaud, J.Y.; Savatier, X. Deep Learning for Real-Time 3D Multi-Object Detection, Localisation, and Tracking: Application to Smart Mobility. *Sensors* **2020**, *20*, 532. [[CrossRef](#)] [[PubMed](#)]
136. Akhloufi, M.A.; Arola, S.; Bonnet, A. Drones chasing drones: Reinforcement learning and deep search area proposal. *Drones* **2019**, *3*, 58. [[CrossRef](#)]
137. Voikov, R.; Falaleev, N.; Baikulov, R. TNet: Real-time temporal and spatial video analysis of table tennis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–16 June 2020; pp. 884–885.
138. Jiang, M.; Hai, T.; Pan, Z.; Wang, H.; Jia, Y.; Deng, C. Multi-agent deep reinforcement learning for multi-object tracker. *IEEE Access* **2019**, *7*, 32400–32407. [[CrossRef](#)]

139. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 213–229.
140. Belyaev, V.; Malysheva, A.; Shpilman, A. End-to-end Deep Object Tracking with Circular Loss Function for Rotated Bounding Box. In Proceedings of the 2019 IEEE XVI International Symposium “Problems of Redundancy in Information and Control Systems” (REDUNDANCY)m, Moscow, Russia, 21–25 October 2019; pp. 165–170.
141. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A Benchmark for Multi-Object Tracking. *arXiv* **2016**, arXiv:1603.00831.
142. 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 June 2012; pp. 3354–3361.
143. Stiefelhagen, R.; Bernardin, K.; Bowers, R.; Garofolo, J.; Mostefa, D.; Soundararajan, P. The CLEAR 2006 evaluation. In Proceedings of the International Evaluation Workshop on Classification of Events, Activities and Relationships, Southampton UK, 6–7 April 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–44.
144. Wu, H.; Han, W.; Wen, C.; Li, X.; Wang, C. 3D Multi-Object Tracking in Point Clouds Based on Prediction Confidence-Guided Data Association. *IEEE Trans. Intell. Transp. Syst.* **2021**, early access. [[CrossRef](#)]
145. Weng, X.; Wang, J.; Held, D.; Kitani, K. 3d multi-object tracking: A baseline and new evaluation metrics. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 10359–10366.
146. Wang, S.; Cai, P.; Wang, L.; Liu, M. DiTNet: End-to-End 3D Object Detection and Track ID Assignment in Spatio-Temporal World. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3397–3404. [[CrossRef](#)]
147. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 17–35.
148. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. A simple baseline for multi-object tracking. *arXiv* **2020**, arXiv:2004.01888.
149. Dang, L.M.; Piran, M.; Han, D.; Min, K.; Moon, H. A survey on internet of things and cloud computing for healthcare. *Electronics* **2019**, *8*, 768. [[CrossRef](#)]