electronics

MDPI

Article

Multiple Moving Vehicles Tracking Algorithm with Attention Mechanism and Motion Model

Jiajun Gao [1], Guangjie Han [1,2,*], Hongbo Zhu [3] and Lyuchao Liao [1]

[1] School of Transportation, Fujian University of Technology, Fuzhou 350118, China; 2211308028@smail.fjut.edu.cn (J.G.); achao@fjut.edu.cn (L.L.)
[2] Department of Information and Communication System, Hohai University, Changzhou 213022, China
[3] School of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China; zhuhongbo@mail.neu.edu.cn
* Correspondence: hanguangjie@hhu.edu.cn

Abstract: With the acceleration of urbanization and the increasing demand for travel, current road traffic is experiencing rapid growth and more complex spatio-temporal logic. Vehicle tracking on roads presents several challenges, including complex scenes with frequent foreground–background transitions, fast and nonlinear vehicle movements, and the presence of numerous unavoidable low-score detection boxes. In this paper, we propose AM-Vehicle-Track, following the proven-effective paradigm of tracking by detection (TBD). At the detection stage, we introduce the lightweight channel block attention mechanism (LCBAM), facilitating the detector to concentrate more on foreground features with limited computational resources. At the tracking stage, we innovatively propose the noise-adaptive extended Kalman filter (NSA-EKF) module to extract vehicles' motion information while considering the impact of detection confidence on observation noise when dealing with nonlinear motion. Additionally, we borrow the Byte data association method to address unavoidable low-score detection boxes, enabling secondary association to reduce ID switches. We achieve 42.2 MOTA, 51.2 IDF1, and 364 IDs on the test set of VisDrone-MOT with 72 FPS. The experimental results showcase our approach's highly competitive performance, attaining SOTA tracking performance with a fast speed.

Keywords: multi object tracking; vehicle tracking; intelligent transportation; motion model

Citation: Gao, J.; Han, G.; Zhu, H.; Liao, L. Multiple Moving Vehicles Tracking Algorithm with Attention Mechanism and Motion Model. *Electronics* **2024**, *13*, 242. https://doi.org/10.3390/electronics13010242

Academic Editor: Shiho Kim

Received: 19 November 2023
Revised: 26 December 2023
Accepted: 28 December 2023
Published: 4 January 2024

1. Introduction

The field of computer vision (CV) encompasses a crucial study domain known as multi-object tracking (MOT) [1]. The primary objective of this task is to accurately detect and locate multiple objects within a given video and assign unique identities to them. The tracked objects can encompass a wide range of entities, such as pedestrians on urban streets, athletes on sports fields, moving vehicles, and even cells under microscopic observation [2].

In recent years, MOT research has experienced rapid development due to the successful implementation of advanced target detectors and various correlation algorithms. Pedestrians, being the main tracking objects in MOT methods, have gained significant attention due to the abundance of pedestrian datasets and their commercial value. It has been observed that 70% of current MOT methods specifically focus on pedestrians [2]. Pedestrian MOT datasets often have the feature of relatively homogeneous scenes [3], slow pedestrian movement, and few scene transitions [4]; in addition, the motion state of pedestrians is relatively simple in most scenarios [3]. Therefore, treating pedestrians in multi-object tracking as linear and uniformly moving objects is completely feasible in order to simplify the motion model and reduce the computational complexity for short-term tracking.

The intelligent transportation field is rapidly developing, and trackers that only focus on pedestrians are insufficient for the needs of intelligent transportation [5]. The multi-

object tracking of vehicles on roads holds significant research importance and practical value in constructing an integrated intelligent transportation system encompassing people, vehicles, roads, and cloud collaboration. However, the traffic road scene of vehicles is highly complex. The video frames of traffic roads contain numerous objects and intricate backgrounds. Objects frequently undergo foreground–background switches, which makes pre-tracking detection very challenging [6]. Consequently, there are numerous low-confidence detection boxes. Moreover, there are significant disparities between the motion states of vehicles and pedestrians. First, vehicles have much higher moving speeds compared to pedestrians [7]. Then, the vehicles are influenced not only by their own power systems and friction coefficients but also by various complex factors in their surroundings, including traffic conditions, traffic rules, and human driving factors on the road [8]. As a result, vehicles in motion often need to change lanes, execute large turns, encounter overlapping obstructions with other vehicles, and perform frequent accelerations and decelerations. The nonlinear nature of these factors renders the motion states of vehicles extraordinarily complex, thereby making it challenging to describe them using simple linear motion models [9] like the traditional Kalman filter (KF) [10], as mentioned in the ByteTrack method [11], and this is the primary reason for tracking failures in autonomous driving scenes. Although advanced multi-object tracking methods have exhibited excellent performance on pedestrian datasets [1], only a few have ventured to tackle the difficult task of tracking moving vehicles in road traffic datasets [12,13]. Consequently, we emphasize that enhancing the robustness of detectors in traffic road scenes, extracting information on the nonlinear motion of vehicles to estimate their positions, and effectively employing low-confidence detection boxes to associate targets are crucial factors in improving the performance of the TBD paradigm for moving vehicles.

Based on the analysis above, we propose a simple and effective MOT method called AM-Vehicle-Track, which is designed to track vehicles on complex roads. Specifically, we propose the lightweight convolutional block attention module (LCBAM) to mitigate the impact of complex road traffic backgrounds and frequent foreground–background switches on vehicle detection. This module incorporates two attention mechanisms: lightweight channel attention mechanism (L-CAM) and spatial attention mechanism (SAM). By combining L-CAM and SAM, foreground feature weights are emphasized, enabling the detector network to focus more on foreground objects without significant computational overhead. Furthermore, we introduce the noise-adaptive extended Kalman filter (NSA-EKF) module to address the nonlinear motion problem in vehicles. This module replaces the widely used Kalman filter [11,14,15] applied to linear Gaussian motion models in many MOT methods. The NSA-EKF module transforms the nonlinear problem into a linear one by retaining the first-order Taylor expansion of the nonlinear function [16]. It also considers the influence of detection confidence on observation noise to more accurately extract vehicle motion information and predict their positions. Lastly, to handle inevitable low-score detection boxes encountered in traffic road scenes, we apply the Byte association algorithm [11], a straightforward and effective approach that performs a second match between low-score detection boxes and trajectories in cases where the initial matching is unsuccessful. This algorithm effectively identifies true targets and maintains trajectory tracking continuity. To validate the effectiveness of AM-Vehicle-Track, we perform numerous experiments on a challenging road scene benchmark dataset VisDrone-MOT [17]. The experimental results reveal the superior effectiveness of our approach by achieving a favorable balance between MOT accuracy and its real-time performance (Figure 1).

The main contributions of this paper are as follows:

1. We propose AM-Vehicle-Track, a simple and effective multi-object tracker, to address the tracking problem of vehicles in complex traffic road scenes.
2. We design a novel attention module called LCBAM, helping our object detector extract foreground features more effectively without incurring excessive computational costs.
3. To better predict the potential future positions of nonlinear moving targets, we design the NSA-EKF module to extract their motion information.

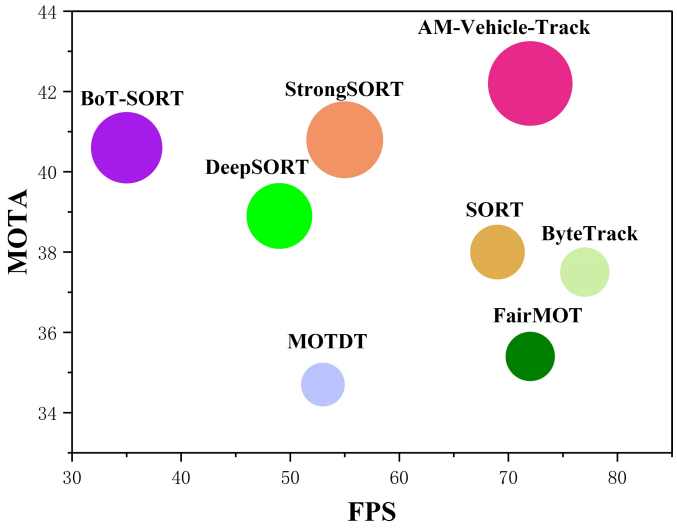

Figure 1. Comparative analysis of various trackers on the VisDrone-MOT test sets based on MOTA-IDF1-FPS metrics. The *x*-axis represents the running speed (FPS), the *y*-axis represents the MOTA, and the circle's radius corresponds to IDF1. Our AM-Vehicle-Track achieves the best MOTA and IDF1 and comparable FPS performance.

2. Related Work

2.1. Object Detection

The field of CV has historically placed significant emphasis on the study of object detection. Continuous progress in this field has been driven by the use of open benchmark tests like COCO [18] and ImageNet VID. Object detection serves as a crucial foundation for comprehending high-level semantic information present in images. Furthermore, it is crucial in advancing other CV technologies, including image segmentation [19], stylization [20], object tracking [2], and so on. Additionally, it finds widespread applications in scene understanding [21], intelligent interaction [22], and autonomous driving [23]. Deep learning-based object detection methods can be broadly categorized as either anchor-based or anchor-free approaches [24]. The former includes two-stage algorithms like R-CNN [25] and Faster R-CNN [26], as well as single-stage methods such as SSD [27], RetinaNet [28], and lightweight YOLO [29], which are currently more prevalent. In this study, the YOLOv7 [30] object detector is used, due to its harmonious combination of model accuracy, inference performance, and speed. Compared to other existing object detectors, YOLOv7 offers higher detection accuracy without compromising on inference speed.

2.2. Tracking by Detection (TBD)

Tracking by Detection (TBD) is a detection-based framework for MOT. It employs object detectors to identify target objects in each frame and then uses matching algorithms, such as the Hungarian algorithm, to associate these targets and determine which objects, at different times and positions in the video sequence, are the same. This enables the extraction of object trajectories and identifiers [2]. The classic SORT [14] algorithm is a multi-object tracking algorithm that is built upon TBD, which utilizes the Faster R-CNN two-stage object detector to detect targets. It is the first method to employ Kalman filter based on linear Gaussian motion model for predicting the position of each object in the next frame. Additionally, the similarity and data association of the objects are determined by calculating the IOU distance and employing the Hungarian algorithm [31]. With the advancements in object detection, more powerful one-stage object detectors have become commonly used for MOT. The ByteTrack algorithm, based on TBD, makes use of the popular one-stage object detector YOLOX [32] to detect the objects of interest. It additionally performs a second matching using low-scoring detection boxes to filter out background information, effectively optimizing the tracking process by reducing excessive ID switching.

The advantage of this multi-object tracking framework lies in its efficient handling of changes in the number of targets. The framework typically exhibits a high tracking speed, while its performance can be further enhanced through optimized object detection. However, a notable disadvantage is the heavy reliance of this framework on the accuracy and robustness of the object detector [2]. Complications arise when the background becomes complex or when issues like object occlusion or crossing motion occur, potentially resulting in missed detections or false detections by the object detector. Consequently, the overall effectiveness of this MOT framework becomes limited.

2.2.1. Separation Detection and Embedding (SDE)

The method based on separation detection and embedding (SDE) separates detection and feature extraction, using two independent networks for implementation. First, a detection network is used to locate the target, followed by feature extraction from the target. Finally, a data association algorithm calculates the affinity between targets, establishing target associations. DeepSORT [15], a classic SDE algorithm, employs a two-stage object detector based on Faster R-CNN [26] to output detection boxes for objects. It then utilizes Kalman filter to extract object motion information and a re-identification (Re-ID) network to further extract appearance features of objects within the detection boxes. Finally, it computes the similarity between Re-ID features and uses the Hungarian algorithm to generate trajectories. Subsequently, several methods incorporating feature pyramids and deep affinity networks have emerged to enhance target discrimination and extraction of target appearance features. Tracking algorithms that integrate motion and appearance features typically depend on pre-existing detectors and Re-ID networks, resulting in improved tracking accuracy and greater robustness against various challenges in complex scenes [33,34]. However, this approach's drawback is that the overall tracking speed of the algorithm is relatively slower due to the high complexity and computational overhead of the network.

2.2.2. Joint Detection and Embedding (JDE)

The joint detection and embedding (JDE) method, in contrast to the SDE method, simultaneously produces the positional and appearance features of objects by incorporating a parallel feature extraction branch into the detection network. By incorporating common features, it avoids redundant computations, enhances the tracking speed of the model, and ensures real-time performance [1]. The classical FairMOT algorithm [35], based on the JDE framework, conducts object detection and identity embedding concurrently within a single network. It abstains from using anchors and solely relies on detecting objects through the center point. Furthermore, a balance between object detection and ReID tasks is achieved through the utilization of a multi-layer feature aggregation approach. This approach effectively combines features from diverse depths and receptive fields. These optimizations have enabled FairMOT to achieve SOTA results on various public datasets. The JDE method accelerates MOT speed and optimizes both detection and feature extraction performance. However, a drawback of the JDE method is its often simplistic design for the feature extraction branch, which leads to an inability to learn critical target representations and lowers the overall robustness of the framework.

2.3. Motion Model

In multi-object tracking, object motion can be categorized as linear motion or nonlinear motion based on patterns [2]. The linear motion refers to the approximate movement trajectory of an object that can be represented by a straight line or a first-degree polynomial, such as uniform linear motion or uniform acceleration/deceleration motion [36]. On the other hand, nonlinear motion models encompass curved motion, accelerated or decelerated motion, and so on. Currently, linear motion models are widely adopted in motion tracking and serve as the foundation for most multi-object tracking algorithms. Assumptions of linear motion simplify the model design, reduce computational complexity, and enable

real-time tracking [37]. For instance, the classical multi-object tracking algorithm SORT incorporates the Kalman filter (KF) to estimate object motion. The KF, based on Bayesian theory [38], updates the position and velocity of objects by leveraging historical states and current observations, providing inspiration for subsequent MOT research [11,14,15].

However, in reality, the motion of objects is rarely linear, especially for vehicles driving on the road [39]. Nonlinear factors, such as friction, air resistance, nonlinear engine output torque–vehicle speed relationship, and the interactions between the driver and the environment, significantly contribute to nonlinear vehicle motion. Lane changes, U-turns, frequent acceleration, deceleration, and emergency stops unequivocally invalidate the possibility of adopting a simple linear motion model for driving vehicles. For example, the experimental results of the ByteTrack [11] algorithm on the BDD100K [40] road dataset demonstrate that the poor performance of SORT [14], DeepSORT [15], MOTDT [33], and other algorithms in road traffic scenes can be attributed to the utilization of the Kalman filter. Consequently, a range of multi-object tracking systems have emerged, employing alternative methods to extract target motion information. One such example is Tracktor [41], which uses Faster R-CNN as the detector and leverages its regression module to predict the position of targets in the subsequent frame. Furthermore, several MOT algorithms have been proposed that utilize deep learning techniques like GCN [42] and Transformer [43,44] to handle target motion and estimate target positions [1]. In contrast, the extended Kalman filter (EKF) addresses the limitations of KF, which is only suitable for linear Gaussian systems, by adapting it for nonlinear systems. The main idea of EKF is to simplify the nonlinear problem by focusing on the first-order terms of the Taylor expansion, disregarding the higher-order terms. This transformation converts the problem from a nonlinear one to a linear one [38]. Besides overcoming the incorrect linear assumptions made by the KF, EKF also inherits the advantages of low computational complexity and accurate prediction.

3. Methods

3.1. Architecture Overview

For a given video sequence in a road scene, our objective is to first detect vehicles in each frame and track them by assigning unique identities. To accomplish this, we follow the paradigm of tracking by detection, propose AM-Vehicle-Track, where we embed the LCBAM attention mechanism in the detector network, and utilize the NSA-EKF algorithm in the tracking phase to extract vehicle motion information and estimate their positions. Specifically, in the detection phase, we employ the improved YOLOv7-LCBAM to obtain the bounding boxes and confidences of targets in every frame of the input image. This detector, with the LCBAM attention mechanism, effectively captures the features of foreground objects and filters out irrelevant backgrounds, which significantly enhances subsequent object tracking.

In the tracking phase, we use the NSA-EKF algorithm to process the motion information of the vehicle's nonlinear movement and predict the location, thereby obtaining the predicted box of the target. Instead of discarding the low-score detection boxes like previous methods, the Byte data association algorithm is borrowed to handle the unavoidable low-score detection boxes that are obtained from detection in complex scenes. The high-score detection boxes are initially matched with the target trajectory boxes predicted by the NSA-EKF algorithm. If a successful match is made, a unique identity is assigned, indicating a successful tracking. For the detection boxes that did not successfully match in the initial attempt and the low-score detection boxes, they undergo a second matching process with the remaining target trajectory boxes. Next, we will present a comprehensive elucidation of AM-Vehicle-Track along with the enhancements made.

3.2. Lightweight Convolutional Block Attention Module (LCBAM)

In the TBD paradigm, the performance of the detector significantly impacts the tracking results. To enhance the detection capability of the YOLOv7 detector, we introduce LCBAM in the object detection stage, which is based on CBAM [45]. In the channel atten-

tion, we replace two fully connected layers with a one-dimensional convolution having a convolutional kernel length of k, enabling local cross-channel interaction. This approach addresses the issues of computational complexity and low FPS caused by the use of fully connected layers in CBAM. LCBAM consists of two attention mechanisms: lightweight channel attention module (L-CAM) and spatial attention module (SAM).

The L-CAM primarily emphasizes the semantic information, specifically on the effective features within the feature map. To achieve this, first, the L-CAM module utilizes a combination of global average pooling and global maximum pooling techniques to effectively aggregate the spatial information from the input feature map $F \in \mathbb{R}^{W \times H \times C}$, allowing it to capture more refined target features. The simultaneous use of these two pooling operations not only reduces the size and computational cost of the feature map, but also improves the network's expressive power. Subsequently, the two one-dimensional vectors resulting from the pooling operations are fed into a one-dimensional convolution with a kernel length of k to extract features. This step is crucial, as it obtains the weights for each channel of the feature map, facilitating local inter-channel interaction and capturing interdependencies between channels. Then, the generated feature vectors undergo element-wise addition and activation using a sigmoid function, obtaining the weight of each channel of the input feature layer, denoted as $M_c \in \mathbb{R}^{1 \times 1 \times C}$, and the channel weight of the relevant information in the input feature map is large, while it is comparatively small for the opposite. Finally, the normalized weights M_c are then element-wise multiplied with the original input feature map $F \in \mathbb{R}^{W \times H \times C}$, resulting in the weighted output feature map $F_c \in \mathbb{R}^{W \times H \times C}$, as shown in Figure 2. The kernel length k is determined using a formula,

$$k = \psi(C) = \left| \frac{\log_2(C)}{m} + \frac{n}{m} \right|_{odd}, \tag{1}$$

in the given equation, C corresponds to the number of channels in the input feature map, $|a|_{odd}$ represents the nearest odd number to a, and m and n are assigned values of 2 and 1, respectively.

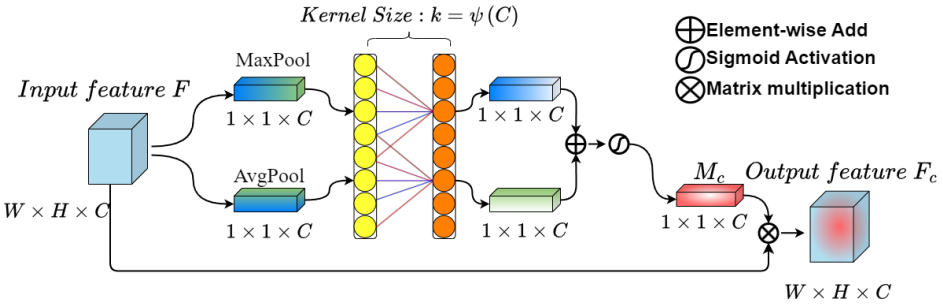

Figure 2. Structural diagram of L-CAM.

The SAM serves as a complementary component to channel attention by focusing on regions in the feature map that contain more effective features and paying greater attention to their positional information. Given a feature map $F_c \in \mathbb{R}^{W \times H \times C}$ from L-CAM, the SAM applies both maximum pooling and average pooling to compress it. The resulting two-dimensional feature maps are then concatenated to form a new feature map with two channels. After that, the concatenated feature map is convolved with a one-channel convolutional operation to ensure spatial consistency with the input. The sigmoid function is then applied to activate the feature map, thereby obtaining the weight of each feature point in the input feature layer, known as the spatial attention weight $M_S \in \mathbb{R}^{W \times H \times 1}$, as shown in Figure 3.

Finally, the weight $M_S \in \mathbb{R}^{W \times H \times 1}$ of SAM is multiplied channel-wise with the input feature map from L-CAM, represented as $F_c \in \mathbb{R}^{W \times H \times C}$, to produce the weighted final feature map $F_{cs} \in \mathbb{R}^{W \times H \times C}$, as shown in Figure 4.

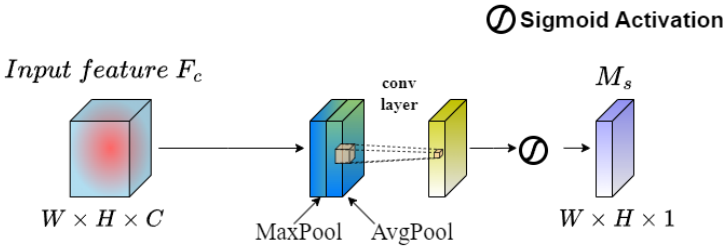

Figure 3. Structural diagram of SAM.

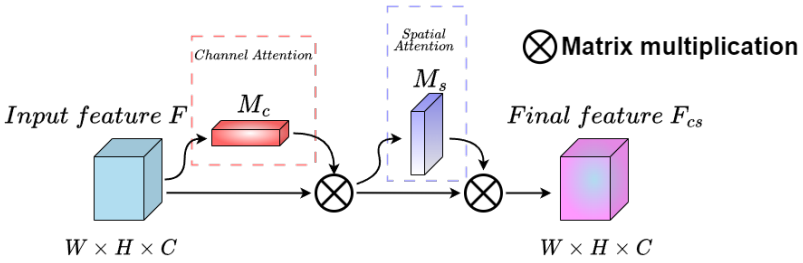

Figure 4. Structural diagram of LCBAM.

In the detection end of our multi-object tracking method, the LCBAM attention mechanism is added to the YOLOv7 network structure, as shown in Figure 5.

When LCBAM is incorporated into the backbone network, it reduces some of the original weights, resulting in inaccurate prediction outcomes. In order to tackle this matter, we have opted to introduce LCBAM solely to the feature extraction component of YOLOv7, ensuring the integrity of the originally extracted features.

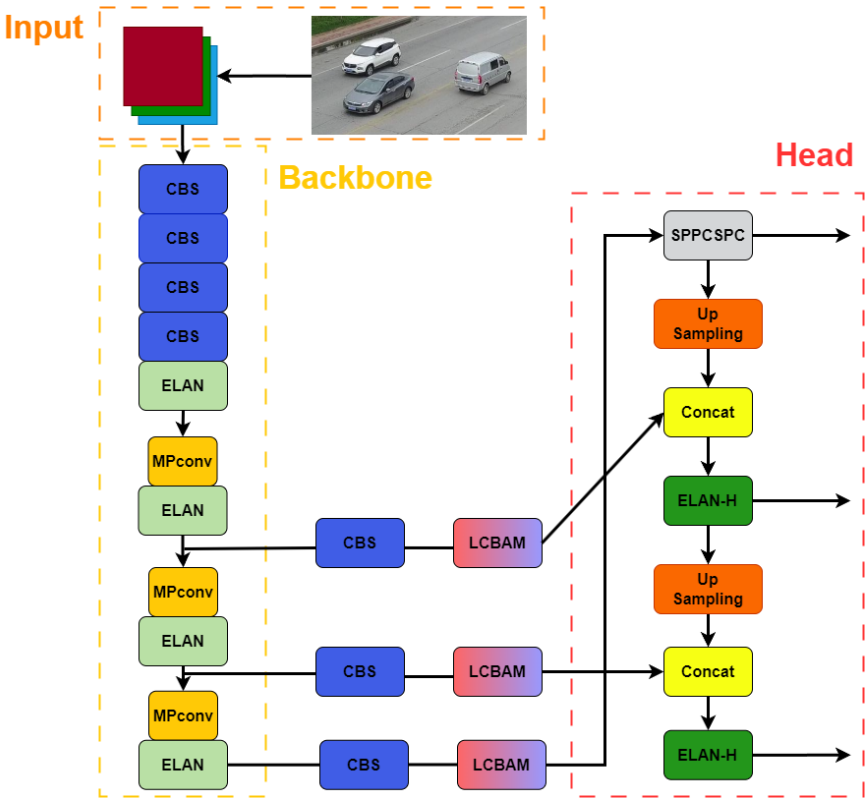

Figure 5. Part of the structural diagram of the improved YOLOv7-LCBAM. The attention mechanism is added to the feature extraction part of the YOLOv7 network.

3.3. Noise-Adaptive Extended Kalman Filter (NSA-EKF)

After object detection, the tracking phase of the objects is initiated. The common linear Kalman filter algorithm has been widely applied in object tracking. This algorithm estimates the input signals based on the previous moment, updates the state variables using observed system values, predicts the position for the next moment, and ultimately outputs the estimated values as the system output [14]. However, assuming that vehicle models on traffic roads only have linear and constant speed motion is not sufficiently robust and can lead to considerable prediction errors. Therefore, we propose using the extended Kalman filter [38] instead of the ordinary Kalman filter and enhancing it to noise-adaptive extended Kalman filter (NSA-EKF). The extended Kalman filter can provide more accurate predictions for the nonlinear variable speed motion of moving vehicles. Additionally, our improved adaptation of the noise allows the measurement noise scale to vary with the confidence of the detection, thereby aiding the algorithm in obtaining more accurate motion states. Despite appearing as a simple improvement, NSA-EKF significantly enhances the robustness and tracking performance of MOT, demonstrating targeted enhancement and improvement.

The linear Kalman filter (KF) algorithm employed in DeepSort [15] will be discussed first. DeepSort assumes that the state of the target can be represented by an eight-dimensional vector, denoted as

$$x_k = \left[c_x, c_y, r, h, v_{x_k}, v_{y_k}, v_{r_k}, v_{h_k}\right]^T, \tag{2}$$

where c_x and c_y represent the center coordinates of the bounding boxes, r is the aspect ratio, h is the height, and v_{x_k},v_{y_k},v_{r_k},v_{h_k} represent the velocity change values for each respective dimension. All velocity values are initialized to 0. DeepSort uses a constant velocity model to describe the motion of the target, which can be mathematically expressed as:

$$x_k = Fx_{k-1} + w_k, \tag{3}$$

where F represents the transition matrix, and w_k denotes the process noise, which arises from the inherent uncertainty associated with the movement of the target. DeepSort makes the assumption that the process noise conforms to a Gaussian distribution with zero mean, i.e.,

$$w_k \sim \mathcal{N}(0, Q), \tag{4}$$

where Q represents the covariance matrix associated with the process noise. Moreover, DeepSort employs a linear measurement model to depict the detection outcomes of targets, given by

$$z_k = Hx_k + v_k, \tag{5}$$

Here, z_k denotes the bounding boxes' coordinates, H represents the measurement matrix, and v_k signifies the measurement noise. The measurement noise can originate from various sources, including loss, overlap, and uncertainty of the bounding boxes. DeepSort assumes that the observation noise follows a zero-mean Gaussian distribution, i.e.,

$$v_k \sim \mathcal{N}(0, R), \tag{6}$$

where R is the covariance matrix of the observation noise.

Based on the given assumptions, the DeepSort algorithm utilizes the KF to estimate the state and covariance. The algorithm follows specific steps: Prediction: To forecast the current state and covariance, the linear system equation is employed using the estimated state value $\hat{x}_{k-1}$ and covariance estimate value $\hat{P}_{k-1}$ from the previous timestamp. The predicted state and covariance are denoted as follows:

$$\hat{x}_k^- = F\hat{x}_{k-1}, \tag{7}$$

$$\hat{P}_k^- = F\hat{P}_{k-1}F^T + Q. \tag{8}$$

Update: For each successfully matched track, the track state is updated by leveraging the position of the match with the detection. The measurement matrix H maps the mean vector $\hat{x}_k^-$ of the track to the detection space. Subsequently, utilizing the current observation z_k, the predicted value $\hat{x}_k^-$ and $\hat{P}_k^-$, the linear observation equation calculates the observation residual S_k, Kalman gain K_k, and the updated optimal estimation value as follows:

$$\widetilde{y}_k = z_k - H\hat{x}_k^-, \tag{9}$$

$$S_k = H\hat{P}_k^- H^T + R, \tag{10}$$

$$K_k = \hat{P}_k^- H^T S_k^{-1}, \tag{11}$$

$$\hat{x}_k = \hat{x}_k^- + K_k\widetilde{y}_k, \tag{12}$$

$$\hat{P}_k = (I - K_k H)\hat{P}_k^- \tag{13}$$

This section addresses the replacement of the linear Kalman filter, utilized in algorithms like DeepSort, with a nonlinear extended Kalman filter (EKF), and adaptively adjusts the noise scale based on the detection quality of objects. The fundamental concept behind the extended Kalman filter involves approximating nonlinear problems as linear ones, through the disregarding of higher-order terms in the Taylor expansion of the nonlinear function and retaining solely the first-order Taylor series. In the scenario where a more intricate nonlinear model is employed to describe target motion and observation, the following equations can serve as an example:

$$x_k = f(x_{k-1}) + w_k, \tag{14}$$

$$z_k = h(x_k) + v_k, \tag{15}$$

where $f(\cdot)$ and $h(\cdot)$ represent arbitrary nonlinear functions. The EKF algorithm requires performing a Taylor expansion on the nonlinear functions and approximating them to the first order around the current state. Some simulation results demonstrate that optimal filtering performance can be achieved by extending the Taylor series expansion up to the second order. However, this considerably increases computational complexity and runtime. Furthermore, the difference in results between the second-order and first-order approximations is relatively small [16]. Consequently, the EKF generally employs the first-order expansion, which can be expressed as follows:

$$f(x_{k-1}) \approx f(\hat{x}_{k-1}) + F_k(x_{k-1} - \hat{x}_{k-1}), \tag{16}$$

$$h(x_k) \approx h\left(\hat{x}_k^-\right) + H_k\left(x_k - \hat{x}_k^-\right), \tag{17}$$

here, F_k and H_k are the Jacobian matrices of $f(\cdot)$ and $h(\cdot)$ evaluated at $\hat{x}_{k-1}$ and $\hat{x}_k^-$, respectively, and they can be expressed as:

$$F_k = \frac{\partial f}{\partial x} \mid_{x=\hat{x}_{k-1}}, \tag{18}$$

$$H_k = \frac{\partial h}{\partial x} \mid_{x=\hat{x}_k^-}. \tag{19}$$

It is noteworthy that the extended Kalman filter (EKF) utilizes Taylor expansion for linearization. When linearizing a system, it is necessary to find an operating point to perform linearization around it. In a nonlinear system, the true point serves as the optimal choice. However, given the presence of numerous system errors, determining the true point becomes infeasible. Hence, the subsequent alternative is to linearize $f(x_k)$ around $\hat{x}_{k-1}$.

Based on these approximations, we can now use the EKF algorithm to estimate the state and covariance of the target. The particular steps are as described below: Prediction: Based on the estimated state $\hat{x}_{k-1}$ and covariance $\hat{P}_{k-1}$ at the previous time step, the current state and covariance are predicted using the nonlinear system equations. The state transition matrix is also calculated as:

$$\hat{x}_k^- = f(\hat{x}_{k-1}), \tag{20}$$

$$\hat{P}_k^- = F_k \hat{P}_{k-1} F_k^T + Q, \tag{21}$$

$$F_k = \frac{\partial f}{\partial x} \mid_{x=\hat{x}_{k-1}} . \tag{22}$$

Update: In this section, we propose improvements to the EKF. Unlike the KF algorithm used in DeepSort, which treats the observation noise covariance matrix R as a constant and does not consider detection performance, we suggest that the measurement noise scale ought to be adjusted according to the confidence level of detecting the current state, denoted as c_k. This is because the observation noise reflects the noise scale of the current frame, and increased uncertainty and detection noise should result in smaller weights during the state updating step. Intuitively, different measurements should incorporate noises of varying scales. To account for this, we introduce an adaptive noise covariance formula:

$$\widetilde{R} = (1 - c_k)R, \tag{23}$$

enabling a more realistic and fair calculation of observation residuals. Finally, utilizing the current observation value z_k, the predicted value $\hat{x}_k^-$, and $\hat{P}_k^-$, we employ a nonlinear observation equation to calculate the observation residuals S_k, the Kalman gain K_k, and the updated optimal estimation as follows:

$$\widetilde{y}_k = z_k - h\left(\hat{x}_k^-\right), \tag{24}$$

$$S_k = H_k \hat{P}_k^- H_k^T + \widetilde{R}, \tag{25}$$

$$K_k = \hat{P}_k^- H_k^T S_k^{-1}, \tag{26}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \widetilde{y}_k, \tag{27}$$

$$H_k = \frac{\partial h}{\partial x} \mid_{x=\hat{x}_k^-}, \tag{28}$$

$$\hat{P}_k = (I - K_k H_k)\hat{P}_k^-, \tag{29}$$

We have successfully replaced the Kalman filter (KF) with the noise-adaptive extended Kalman filter (NSA-EKF) to capture object motion information and predict their potential locations in multi-object tracking.

4. Experiments

4.1. Datasets and Metrics

We conduct experiments on the VisDrone-MOT dataset under the "private detection" protocol. The VisDrone-MOT dataset is a widely used multi-class multi-object tracking dataset that includes various target categories such as cars, buses, trucks, and pedestrians [17]. These targets are captured in different scenarios using unmanned aerial vehicles with a similar height and perspective to surveillance cameras on the traffic road, thus rendering VisDrone-MOT highly valuable in practical applications. The dataset presents challenging scenes with diverse environments and densities, including 56 video sequences with 24,201 frames in the training set, 7 video sequences with 2819 frames in the validation set, and 33 video sequences with 12,968 frames in the test set, which is divided into test-challenge (16 video sequences) and test-dev (17 video sequences) for debugging and further verification purposes. Additionally, since we employ the YOLO object detector [30], it is necessary to convert the dataset into the YOLO format.

In this experimental section, we adopted widely accepted CLEAR metrics [46] (MOTA, FP, FN, IDs, etc.) to assess the performance. The MOTA evaluation metric is commonly utilized for MOT, which is computed by integrating various measures including FP, FN, and IDs, therefore MOTA provides a comprehensive assessment of factors such as false

alarm rate, target loss, and target identity switching. Additionally, MOTA primarily emphasizes detection performance, as the number of FPs and FNs is larger than IDs. IDF1 aims to map predicted trajectories to actual trajectories and emphasizes the algorithm's correlation performance. MT denotes the proportion of successfully tracked targets, which accounts for more than 80% of the trajectories. ML represents the proportion of unsuccessfully tracked targets, which accounts for less than 20% of the true trajectories. IDs indicate the number of identity switches that occur for all tracked targets. FN denotes the count of false negatives in the entire video sequence, whereas FP represents the count of false positives. Additionally, we report FPS to measure tracking speed and the real-time performance of the MOT tracker by indicating the number of frames processed per second.

4.2. Implementation Details

Our network is built upon PyTorch 1.7.0 and CUDA 11.0 [47]; the operating system is Ubuntu 20.4. For training, we use our improved YOLOv7-LCBAM as the detection detector with the parameters pre-trained on COCO [18]. We do not use additional training data, the mini-batch size is set to 16 and the number of total training epochs is 80, the first epoch is warmed up by cosine annealing method. We use SGD [48] as the optimizer, the initial learning rate is set to 0.001, weight decay is set to 5×10^{-4}, and momentum is 0.9. The input size of VisDrone-MOT is 1088×640; data-augmentations include random resized cropping, Mosaic [49], and Mixup [50]. For the sake of fair comparisons, we present the results obtained by other previous methods that have also employed the same data-augmentation strategy. In our study, we employ UniTrack as the Re-ID model. All of our experimental procedures are executed on NVIDIA RTX3090 GPUs.

4.3. Ablative Studiess

In this section, we perform ablation experiments on the VisDrone-MOT dataset to assess the reliability of each component in AM-Vehicle-Track. The DeepSort algorithm with YOLOv7 target detector is employed as the baseline model for the ablation experiments.

4.3.1. The Effectiveness of LCBAM

In the TBD paradigm, the performance of the object detector plays a crucial role in MOT algorithms [2]. It not only impacts tracking accuracy but also correlates closely with the algorithm's real-time capability. Hence, we chose the YOLOv5 [51] detector, commonly utilized in previous MOT algorithms [52,53], along with YOLOv7 as the baseline object detector. We also conducted ablation experiments by comparing them with the YOLOv7-CBAM object detector and our improved YOLOv7-LCBAM object detector. As shown in Table 1, the performance analysis in the table demonstrates that using YOLOv5 as the object detector yields the worst algorithm performance, with both low algorithm accuracy and frame rate. It is the only algorithm in the experiment that falls below 50 frames. Although the inclusion of the CBAM in the YOLOv7 detector enhances the MOTA value compared to YOLOv7, the addition of the attention module complicates the entire detector network, resulting in a 17% decrease in the frame rate. However, utilizing the improved YOLOv7-LCBAM as the detector enables us to achieve more competitive tracking performance with a slightly lower frame rate. As analyzed earlier, LCBAM utilizes one-dimensional convolution to achieve inter-channel interaction. Not only does it prevent information loss caused by feature compression in fully connected layers, but it also reduces network complexity.

Table 1. Analysis of performance using different detectors (best in bold)..

Method	MOTA (↑)	FPS (↑)
YOLOv5s	30.4	45
YOLOv7	34.9	**74**
YOLOv7-CBAM	37.3	61
YOLOv7-LCBAM (Ours)	**37.5**	68

4.3.2. The Effectiveness of NSA-EKF

To investigate the impact of different motion state models on vehicle tracking in traffic scenarios, we compare the widely used KF and EKF in the baseline model with our improved NSA-EKF, as shown in Figure 6.

We evaluate their effectiveness in obtaining motion information and predicting the trajectories of road vehicles. Our findings indicate that the baseline model with KF exhibits the lowest MOTA and IDF1 values. The simple KF algorithm proves ineffective in accurately predicting and updating the complex motion of vehicles on the road, as noted in ByteTrack [11]. By employing EKF, as shown in Table 2, the MOTA metric improves from 34.9 to 37.8, and IDF1 improves from 41.5 to 45.5. Additionally, the multi-object tracking based on NSA-EKF further enhances the overall algorithm performance, with the MOTA improving to 39.2 and IDF1 to 46.1. Moreover, the algorithm demonstrates improved real-time capabilities compared with EKF. It is evident that the use of the improved adaptive noise extended Kalman filter, as a replacement for the standard Kalman filter, enables better extraction of motion information from nonlinear road vehicles and accurate estimation of their future positions. From the perspectives of MOTA and IDF1, this approach indeed enhances the prediction accuracy and association capability, making it more suitable for road traffic scenarios compared to the standard Kalman filter.

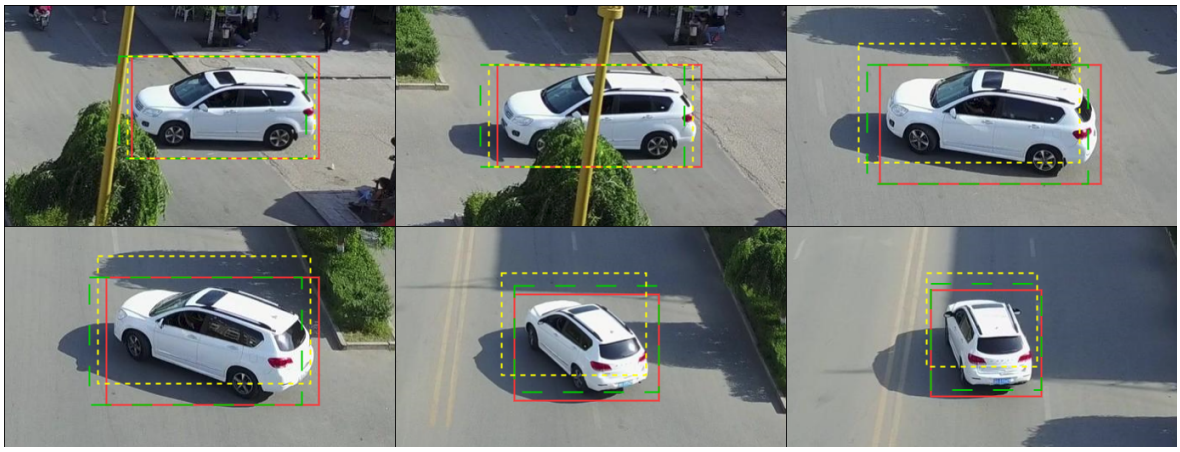

Figure 6. Visualization of the prediction box location compared with the widely used KF (dashed green) and our NSA-EKF (dashed yellow), the red box represents the bounding box of the object, the object is a car that undergoes a sequential process starting with deceleration, then acceleration and turning right. It seems that the prediction box location produced by the proposed NSA-EKF fits more accurately to the potential motion patterns of an object. It is important to highlight that the principles and effects of NSA-EKF remain consistent whether applied to a single car or multiple cars. Consequently, in order to minimize visual redundancy, we exclusively showcase the visual effects of NSA-EKF on a single car from the dataset.

Table 2. Analysis of different motion modules (best in bold).

Method	MOTA (↑)	IDF1 (↑)	FPS (↑)
KF	34.9	41.5	**74**
EKF	37.8	45.5	62
NSA-EKF (Ours)	**39.2**	**46.1**	69

4.3.3. The Effectiveness of Byte

In complex road traffic scenarios characterized by frequent occlusions and overlaps caused by vehicles, encountering multiple low-scoring detection boxes is inevitable. To address this issue, we propose the Byte correlation algorithm for secondary correlation of these low-scoring detection boxes. As demonstrated in Table 3, the introduction of the Byte algorithm remarkably reduces the interchanging of object IDs, resulting in a decrease in

the number of IDs from 1094 to 715 when employing the baseline model. This improvement is significant. It is evident that directly discarding low-scoring detection boxes in previous multi-object tracking algorithms leads to a substantial loss of information. Therefore, the correlation of low-scoring detection boxes holds significant importance in road traffic scenarios.

Table 3. Ablation studies on VisDrone-MOT test set (best in bold).

baseline	+LCBAM	+NSA-EKF	+Byte	MOTA (↑)	IDF1 (↑)	IDs (↓)
√				34.9	41.5	1104
√	√			37.5	44.1	1162
√	√	√		39.1	46.3	1094
√	√	√	√	**40.7**	**46.8**	**715**

4.4. Comparison to State-of-the-Art

In this section, we compare our approach with classical and contemporary methods, including the classical Sort method and ByteTrack [11] method based on the TBD paradigm, the DeepSort [15] method based on the SDE paradigm, the FairMOT [35] method based on the JDE paradigm, and the more advanced BoT-SORT [54] and SrongSORT [55] methods. As presented in Table 4, our AM-Vehicle-Track method achieves highly competitive performance across most key metrics. It secures the top rank in most metrics such as MOTA and IDF1, and obtains the second rank in metrics like IDs and FPS. Our method is designed to tackle the challenge of effectively tracking fast, nonlinearly moving, and frequently occluded vehicles in complex road traffic scenarios. By doing so, it enhances the accuracy of target association and strengthens the robustness of the tracker. Furthermore, our tracker demonstrates superior performance, outperforming the second-best multi-object tracker significantly in metrics that measure tracking accuracy and precision (i.e., +1.4 MOTA and +3.5 IDF1). It only slightly trails behind the ByteTrack algorithm (i.e., −5 FPS) in terms of identity switching and tracking speed, implying that the inclusion of more complex detection and tracking components does impact speed. Nonetheless, we believe that maintaining a favorable balance between tracking accuracy and speed while ensuring an improvement in accuracy is reasonable overall.

Table 4. Comparison with the state-of-the-art methods under the "private detector" protocol on the VisDrone-MOT test set. The best results for each metric are shown in bold. AM-Vehicle-Track ranks first in the most of metrics. (best in bold)

Method	MOTA (↑)	IDF1 (↑)	MT (↑)	ML (↓)	IDs (↓)	FP (↓)	FN (↓)	FPS (↑)
SORT [14]	38	44.2	24.1	42.9	702	10,073	60,996	69
DeepSORT [15]	38.9	44.9	25.7	35.4	655	12,548	61,534	49
FairMOT [35]	35.4	43.7	25.2	36.6	882	17,513	63,655	72
MOTDT [33]	34.7	42.8	23.5	37.3	754	15,301	60,072	53
ByteTrack [11]	37.5	43.7	28.4	43.6	377	6413	59,482	**77**
BoT-SORT [54]	40.6	47.1	27.2	38.4	396	6841	58,624	35
StrongSORT [55]	40.8	47.7	35.1	**32.3**	524	9072	62,583	55
Ours	**42.2**	**51.2**	**35.4**	33.1	**364**	**6157**	**56,481**	72

Figure 7 illustrates the tracking performance of our method on multiple vehicles exhibiting nonlinear motion, despite experiencing significant occlusion during their movement. The bounding boxes surrounding the vehicles are tracking boxes, and the top-left corner ID of each tracking box represents the assigned ID for the corresponding vehicle throughout the entire video stream. As shown in Figure 7, our tracker can perform accurate tracking when multiple cars undergo nonlinear motions, such as significant turns, and are occluded by foreground objects during the nonlinear motion process. For instance, vehicle ID 24 in the figure is observed to be executing a turning maneuver in

Figure 7a,b, despite being significantly obstructed by a billboard in Figure 7c. Interestingly, in Figure 7d, the tracking of the vehicle remains accurate and its identification number remains unchanged.

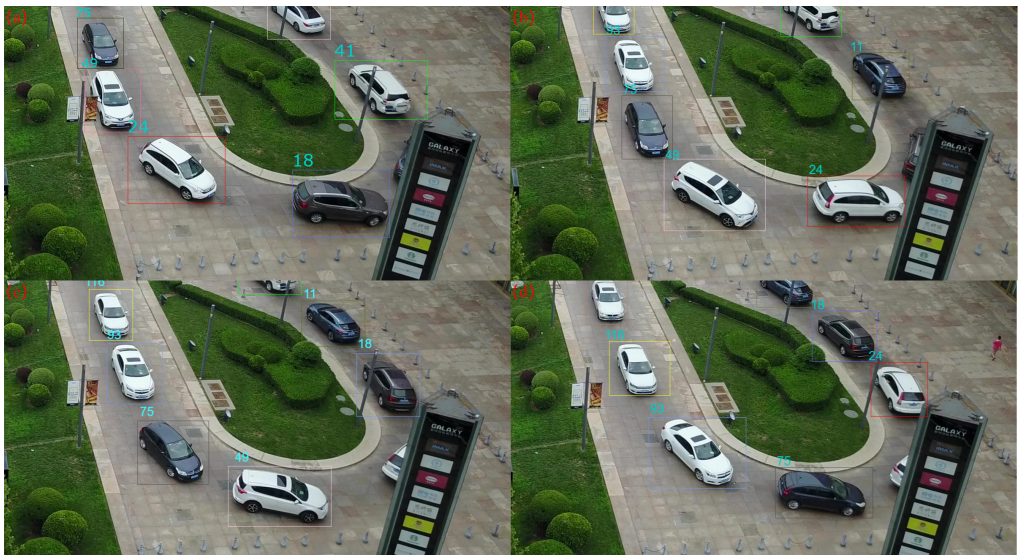

Figure 7. Qualitative results on VisDrone-MOT benchmark for multi-object tracking. Subfigures (**a**–**d**) display the tracking results of our tracker across various time sequences.

5. Conclusions

In this paper, we present AM-Vehicle-Track, a framework that follows the TBD paradigm to enhance vehicle tracking in road traffic scenes. In the detection phase, we introduce the LCBAM attention module, which is integrated into the Yolov7 detector for enhanced foreground feature extraction and suppression of irrelevant features, particularly in complex scenes and at higher frame rates. Subsequently, in the tracking phase, we propose the NSA-EKF module as an innovative replacement for the conventional KF module. This module effectively captures motion information for the vehicle's nonlinear motion and provides improved estimation of the vehicle's position in the next frame for tracking purposes. Additionally, we borrow the Byte data association method, which performs secondary association for low-confidence detection boxes to enhance the detector's association capability and reduce ID switches. Experimental results on the VisDrone-MOT road dataset demonstrate the competitive performance of our method, striking a balance between tracking accuracy and execution speed.

6. Limitations and Future Work

The primary focus of our method is its reliance on extracting motion information from vehicles to enable tracking; additionally, we employ existing Re-ID methods that also rely on local object information, which effectively meet the requirements for vehicle tracking in most of scenarios. However, video frames also contain valuable global information [56]. Extracting and leveraging temporal and spatial global information holds significant value [57]. In certain specific scenarios, particularly those involving highways, it is reasonable to assume that vehicles have similar motion directions and speeds. This not only simplifies detection and tracking but also reduces the likelihood of false matches and tracking loss. In the future, we plan to explore this information in our AM-Vehicle-Track to further improve the tracking performance. In addition, we aim to investigate various approaches to mitigate the computational requirements of our tracker, thereby facilitating its adoption in real-world scenarios with lower-tier hardware.

Author Contributions: Investigation, J.G. and H.Z.; methodology, J.G.; validation, J.G.; writing—original draft preparation, J.G.; writing—review and editing, H.Z. and G.H.; supervision, G.H. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: Funding was provided by the Fujian Key Lab for Automotive Electronics and Electric Drive, Fujian University of Technology, 350118, China. This work is supported in part by a project of the Fujian University of Technology, No. GY-Z19066.

Data Availability Statement: Data is contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

LCBAM	Lightweight Channel Block Attention Mechanism
NSA-EKF	Noise-adaptive Extended Kalman Filter
MOT	Multi-object tracking
L-CAM	Lightweight Channel Attention Mechanism
SAM	Spatial Attention Mechanism
TBD	Tracking by Detection
SDE	Separation Detection and Embedding
Re-ID	Re-identification
JDE	Joint Detection and Embedding
KF	Kalman Filter
CV	Computer Vision

References

1. Bashar, M.; Islam, S.; Hussain, K.K.; Hasan, M.B.; Rahman, A.; Kabir, M.H. Multiple object tracking in recent times: A literature review. *arXiv* **2022**, arXiv:2209.04796.
2. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.K. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [CrossRef]
3. Simsek, F.E.; Cigla, C.; Kayabol, K. SOMPT22: A Surveillance Oriented Multi-pedestrian Tracking Dataset. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 659–675.
4. Fabbri, M.; Brasó, G.; Maugeri, G.; Cetintas, O.; Gasparini, R.; Ošep, A.; Calderara, S.; Leal-Taixé, L.; Cucchiara, R. Motsynth: How can synthetic data help pedestrian detection and tracking? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10849–10859.
5. Creß, C.; Bing, Z.; Knoll, A.C. Intelligent transportation systems using external infrastructure: A literature survey. *arXiv* **2021**, arXiv:2112.05615.
6. Zahra, A.; Ghafoor, M.; Munir, K.; Ullah, A.; Ul Abideen, Z. Application of region-based video surveillance in smart cities using deep learning. *Multimed. Tools Appl.* **2021**, 1–26. [CrossRef]
7. Dollár, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: A benchmark. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 304–311.
8. Lefèvre, S.; Vasquez, D.; Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH J.* **2014**, *1*, 1. [CrossRef]
9. Dawood, M.; Abdelaziz, M.; Ghoneima, M.; Hammad, S. A nonlinear model predictive controller for autonomous driving. In Proceedings of the 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), Aswan, Egypt, 8–9 February 2020; pp. 151–157.
10. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
11. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-object tracking by associating every detection box. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–21.
12. Lu, Z.; Rathod, V.; Votel, R.; Huang, J. Retinatrack: Online single stage joint detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14668–14678.
13. Du, Y.; Wan, J.; Zhao, Y.; Zhang, B.; Tong, Z.; Dong, J. Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2809–2819.
14. 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.

15. 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.
16. Sunahara, Y. An approximate method of state estimation for nonlinear dynamical systems. *J. Basic Eng.* **1970**, *92*, 385–393. [CrossRef]
17. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7380–7399. [CrossRef]
18. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
19. Yao, R.; Lin, G.; Xia, S.; Zhao, J.; Zhou, Y. Video object segmentation and tracking: A survey. *ACM Trans. Intell. Syst. Technol. TIST* **2020**, *11*, 1–47. [CrossRef]
20. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.
21. Gupta, A.; Anpalagan, A.; Guan, L.; Khwaja, A.S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* **2021**, *10*, 100057. [CrossRef]
22. Pandey, A.; Puri, M.; Varde, A. Object detection with neural models, deep learning and common sense to aid smart mobility. In Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), Volos, Greece, 5–7 November 2018; pp. 859–863.
23. Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J.Z.; Langer, D.; Pink, O.; Pratt, V.; et al. Towards fully autonomous driving: Systems and algorithms. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 163–168.
24. Liu, S.; Zhou, H.; Li, C.; Wang, S. Analysis of anchor-based and anchor-free object detection methods based on deep learning. In Proceedings of the 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 13–16 October 2020; pp. 1058–1065.
25. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
26. 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]
27. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
28. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
30. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
31. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
32. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
33. 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 June 2018; pp. 1–6.
34. Yu, F.; Li, W.; Li, Q.; Liu, Y.; Shi, X.; Yan, J. Poi: Multiple object tracking with high performance detection and appearance feature. In Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016; Part II 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 36–42.
35. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
36. Breitenstein, M.D.; Reichlin, F.; Leibe, B.; Koller-Meier, E.; Van Gool, L. Robust tracking-by-detection using a detector confidence particle filter. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 1515–1522.
37. Murray, S. Real-time multiple object tracking-a study on the importance of speed. *arXiv* **2017**, arXiv:1709.03572.
38. Fang, H.; Tian, N.; Wang, Y.; Zhou, M.; Haile, M.A. Nonlinear Bayesian estimation: From Kalman filtering to a broader horizon. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 401–417. [CrossRef]
39. Dutta, S.; Subramaniam, A.; Mittal, A. Non-linear motion estimation for video frame interpolation using space-time convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1726–1731.

40. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2636–2645.
41. Bergmann, P.; Meinhardt, T.; Leal-Taixe, L. Tracking without bells and whistles. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 941–951.
42. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
43. Sun, P.; Cao, J.; Jiang, Y.; Zhang, R.; Xie, E.; Yuan, Z.; Wang, C.; Luo, P. Transtrack: Multiple object tracking with transformer. *arXiv* **2020**, arXiv:2012.15460.
44. Blatter, P.; Kanakis, M.; Danelljan, M.; Van Gool, L. Efficient visual tracking with exemplar transformers. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 1571–1581.
45. 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.
46. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP J. Image Video Process.* **2008**, *2008*, 246309. [CrossRef]
47. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
48. Bottou, L. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–436.
49. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
50. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
51. Jocher, G.; Stoken, A.; Borovec, J.; NanoCode012, C.; Changyu, L.; Laughing, H. ultralytics/yolov5: v3.0. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 December 2020).
52. Wang, Y.; Yang, H. Multi-target pedestrian tracking based on yolov5 and deepsort. In Proceedings of the 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 14–16 April 2022; pp. 508–514.
53. Zhang, K.; Wang, C.; Yu, X.; Zheng, A.; Gao, M.; Pan, Z.; Chen, G.; Shen, Z. Research on mine vehicle tracking and detection technology based on YOLOv5. *Syst. Sci. Control Eng.* **2022**, *10*, 347–366. [CrossRef]
54. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv* **2022**, arXiv:2206.14651.
55. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. Strongsort: Make deepsort great again. *IEEE Trans. Multimed.* **2023**, *25*, 8725–8737. [CrossRef]
56. Zhou, X.; Yin, T.; Koltun, V.; Krähenbühl, P. Global tracking transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8771–8780.
57. Wang, G.; Gu, R.; Liu, Z.; Hu, W.; Song, M.; Hwang, J.N. Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9876–9886.