ITC 4/53

Information Technology
and Control
Vol. 53 / No. 4/ 2024
pp. 1221-1235
DOI 10.5755/j01.itc.53.4.36947

Deep Learning-Based Trajectory Tracking Method for Intelligently Network-Connected Driverless Vehicles in Narrow Areas

Received 2024/04/10 | Accepted after revision 2024/07/13

HOW TO CITE: Han, YJ., Kim, B. C., Xu, HC. (2024). Deep Learning-Based Trajectory Tracking Method for Intelligently Network-Connected Driverless Vehicles in Narrow Areas. *Information Technology and Control,* 53(4), 1221-1235. https://doi.org/10.5755/j01.itc.53.4.36947

Deep Learning-Based Trajectory Tracking Method for Intelligently Network-Connected Driverless Vehicles in Narrow Areas

YaJun Han

Department of Automotive and Traffic Engineering, Jiangsu University of Technology,
Jiangsu Changzhou 213001, China

Byung Cheul Kim

School of Electronic Engineering , Gyeongsang National University, Dongjin-ro 33, Jinju 52725, South Korea

HaiChao Xu

Department of Business Administration, Gyeongsang National University, Dongjin-ro 33, Jinju 52725, South Korea

Corresponding author: vickyyajun@163.com

Driverless vehicles are the development direction of intelligent transportation. In recent years, the rapid development of driverless transportation technology, especially the practical performance of intelligently network-connected driverless vehicles has improved rapidly. However, due to problems with traffic planning, many roads are still relatively narrow. When an intelligently networked driverless car moves in a narrow area, the lack of precision in trajectory tracking can easily cause traffic accidents due to small trajectory changes. In this paper, for the driving characteristics of intelligently networked driverless vehicles in narrow areas, an improved Faster R-CNN target detection network is proposed that introduces a deep residual network ResNet-50, a dual attention mechanism CBAM, and an ROI-Pooling to estimate the position information of driverless vehicles in the video of the traffic scene. Based on the target detection results of driverless vehicles and the appearance characteristics of vehicles, the novel DeepSORT vehicle tracking algorithm improved by OSNet full-scale network and complete intersection over union (CIoU), is employed to derive a vehicle trajectory within a single camera on a real road. The UA-DETRAC dataset in real scenarios is selected to run experiments, and the results demonstrate that the proposed target detection and tracking algorithms perform well, and effectively realize target detection and trajectory tracking of intelligently internet-connected driverless vehicles in narrow areas, which can help realize the further performance enhancement. The improved DeepSORT achieves an impressive MOTA of 96.1% and MOTP of 0.115.

KEYWORDS: Driverless vehicles, Trajectory tracking, Narrow area, Intelligent network-connected, Faster R-CNN, DeepSORT.

1. Introduction

Driverless vehicles have gradually become an important part of intelligent transportation systems and can potentially substitute, partially or entirely, the responsibilities associated with human driving. This can enhance road capacity for vehicle accommodation and plays a crucial role in enhancing both the safety and efficiency of road traffic. The use of driverless technology in an intelligent network environment to improve traffic safety and efficiency has been one of the most widely researched topics in academia and the industry. In the development process of driverless vehicles, intelligent networking technology realizes the interconnection between vehicles and road traffic infrastructure, pedestrians, and surrounding vehicles, thus greatly enhancing the level of vehicle intelligence and effectively improving the safety of driving [21]. Intelligently network-connected technology can provide driving assistance information for vehicles, and the interconnection between vehicles and road infrastructure can allow the vehicle's sensors and cameras to obtain information about the surroundings, to adjust its driving route. It can also provide timely warnings and alerts when vehicles engage in unsafe driving manner.

Due to issues with traffic planning, many lanes still have narrow driving areas. The problem faced by driverless vehicles when moving in narrow areas is more complex than those in wider areas. Especially in dynamic traffic environments, wrong decisions about lane changes, irrational path planning, or ineffective control algorithms may lead to traffic congestion and accidents, and it is necessary to research driving problems of intelligently network-connected driverless vehicles in narrow areas. Vehicle trajectory tracking, as a key technology of intelligent driving, can help driverless vehicles evaluate the safety distance in real-time, so that driverless vehicles can intelligently choose driving routes and take emergency braking measures in unexpected situations to achieve safe driving [20]. Hence, it becomes crucial to conduct studies on technology that tracks the trajectory of vehicles, ensuring autonomous safe operations.

Vehicle trajectory tracking in the study of driverless vehicles is mainly based on real-time environment modeling, and under certain constraints, the consideration of static or dynamic obstacles in surroundings, which directly affect the driving trajectory of vehicles, can make the vehicle drive stably along the desired trajectory, which can help enhance the controllability and reduce the incidence of traffic accidents. Therefore, the article researches trajectory tracking of intelligently network-connected driverless vehicles in narrow areas. Target detection for driverless vehicles is a prerequisite for vehicle trajectory tracking realized in an intelligent network environment. In the research, conventional visual vehicle detection algorithms rely on the hand-designed feature processor of the developer, so the processing effect and the performance of the algorithm will be affected. Conventional vehicle detection algorithms are based on statistical methods and various operators are designed to extract texture information. The corresponding spectral method is Fourier spectrum analysis, which specifically refers to identifying the high-frequency section of the spectrum to pinpoint the periodicity of images. Directional gradient histograms and Hal features are frequently implemented detection methods. With the advancement of deep learning, applications of algorithms to detect targets in the automotive domain have gained prominence. These algorithms can be categorized into 2 types: one-stage detection network (One-Stage) and two-stage detection network (Two-Stage) [5, 19]. In the two-stage detection networks, convolutional networks are utilized to devise region proposals. These proposals serve as inputs for the subsequent Convolutional Neural Network (CNN) to extract features. Subsequently, the extracted features undergo classification and regression to determine the boundaries of the predicted targets. The final step involves eliminating redundant detection targets through Non-Maximum Suppression (NMS). R-CNN initially introduced this method, and Faster R-CNN further refined it, assigning the task of generating candidate regions to the neural network, marking the inception of the region candidate network. However, the two-stage network has a drawback in terms of computational speed. To address this limitation, the one-stage network is proposed. Instead of generating candidate regions, the one-stage network defines boxes of varying sizes at anchor points within the image. These anchor frames are then processed through the CNN for feature extraction, ultimately yielding the probability of the object class and the coordinate point of the detected object. The target tracking task can only be realized after the completion of target detection. The mainstream algorithms for target tracking use correlation filtering procedures, including sum-of-squares filters, Kernelized Correlation Filters (KCF), and other algorithms based on MOSSE improvement. Regression-based deep learning frameworks are also driving the target tracking task. Current multi-target tracking tasks are more complex and difficult to model. While single-target tracking operations require continuous data filtering for a definite target, multi-target tracking requires the detection of targets under specific frames and also a correlation of the same target data between different frames, so there is still a huge challenge.

As such, the manuscript tackles the previously mentioned issues by proposing a trajectory-tracking method based on deep learning for intelligently network-connected driverless vehicles operating in confined spaces. Its primary contributions are outlined as follows:

1. Leveraging the deep residual network ResNet-50 to substitute VGG16, the underlying feature extraction network of Faster R-CNN. This substitution aims to preserve additional information regarding vehicle characteristics, thereby enhancing the feature extraction capability of the network for driverless vehicles.
2. Introducing the spatial and channel dual attention mechanism CBAM into Faster R-CNN to enhance the accuracy of automobile target detection. Additionally, replacing ROI-Pooling with ROI-Align to bolster the network's generalization capability and enable precise detection of unmanned vehicles.
3. Employing the OSNet full-scale network to refine the shallow residual network enhances DeepSORT's capacity to extract distinct features of driverless vehicles.
4. Optimizing the Intersection over Union (IoU) matching method in DeepSORT by adopting the Complete Intersection over Union (CIoU) matching method. This adjustment facilitates accurate tracking of driverless vehicles by assessing the degree of match between the detection frame and the bounding regression.

The rest of the article is constructed as follows: The related work is presented in Section 2. Section 3 presents the target detection algorithm, which is the proposed algorithm. The trajectory tracking algorithm is presented in Section 4. Section 5 presents the experimentations. The research is concluded in Section 6.

2. Related Work

The development of intelligent network technology in recent years has prompted video analysis-based automotive Multi-object Tracking (MOT) technology to gradually become a significant research area in the field of driverless vehicles [11, 14]. Single-camera multi-object tracking refers to analyzing the video to identify and track the targets belonging to different categories and identities. Current approaches can be broadly categorized into 2 types: one follows the step-by-step completion of the detection tracking paradigm, called Detection-Based Tracking, and the other takes detection as a prerequisite for tracking, called Detection-Free Tracking [4]. First, for the former tracking task, a target detector is required to process each frame in the video to detect the target position information in it. Second, based on the target position information, a tracker is employed to associate targets between video frames to achieve continuous tracking of the same target. In contrast, for the latter tracking task, prior information about the position of each target at its first appearance in the video is required [1]. Then, a separate tracking algorithm is implemented for each target to achieve continuous tracking of the target. Many scholars have also researched trajectory tracking of driverless vehicles based on the ideas presented by the two methods and have achieved a great number of results.

Wu et al. [17] addressed the problem that available UAV single trackers are difficult to accurately identify video targets by designing PFN to extract multi-level features and complete feature aggregation and subsequently introduced a channel transform enhancer (CTE) to simulate relationship and complete fine-grained identity prediction. Experiments show that the method has excellent performance in tracking self-driving cars. To resolve the UAV tracking accuracy problem, Li et al. [8] devised a tracking model with a focus on behavioral information. This model primarily relies on the long short-term memory (LSTM) network and self-attention network. Simultaneously, it integrates an association model based on the Hungarian algorithm. The objective is to formulate a trajectory prediction model and forecast paths of individual vehicles. The test results based on the new dataset of highway vehicles show that the algorithm has good robustness and real-time performance. Wang et al. [15] considered the effect of the false detection problem of 3D multi-target tracking (MOT) on the effect of trajectory tracking and designed the fused 3D MOT framework by combining camera and LiDAR data to reduce the probability of tracking failure. The framework contains a motion cost matrix to improve tracking accuracy, while multi-target tracking has been accomplished through multi-category cost. Experimental results on the KITTI test dataset show a better performance of the framework. Wang et al. [16] designed the StrongFusionMOT algorithm to achieve the fusion of 2D and 3D detection through absolute difference (AD) census to strengthen the robustness, designed the SDIoU cost function to enhance the correlation accuracy, and introduced a matching mechanism that traces back the past trajectories to lower the error. The outcomes of the StrongFusionMOT approach demonstrate superiority through the test with KITTI and real-scene datasets.

Although the mentioned algorithms have made breakthroughs in multi-target tracking, there is a lack of relevant research on the trajectory tracking of intelligently network-connected driverless vehicles in narrow areas. In the article, we present a trajectory-tracking algorithm based on deep learning to achieve the tracking of intelligently network-connected autonomous vehicles in confined spaces. The goal is to advance driverless vehicle technology in narrow road conditions.

3. Target Detection Model for Intelligently Network-connected Driverless Vehicles in Narrow Areas based on an Improved Faster R-CNN

3.1. Faster R-CNN

Fast R-CNN employs the construction of an ROI pooling layer to extract feature regions and accomplishes feature classification through the SoftMax function. The Faster R-CNN algorithm represents an optimization over the Fast R-CNN foundation. It serves as an end-to-end deep learning model for target detection, enabling the training of the entire algorithm within a single framework. Utilizing the addition of the neural network Region Proposal Network (RPN), Faster R-CNN facilitates edge extraction, thereby enhancing both the model's time efficiency and the accuracy of target detection. This approach directly acquires candidate regions, further improving model effectiveness and target detection accuracy [9]. Figure 1 illustrates the fundamental structure of Faster R-CNN.

The specific workflow of Faster R-CNN is as follows [2]:

Step 1: Feature Extraction. Initially, the image undergoes CNN processing to extract its feature data, resulting in a corresponding feature map. Subsequently, the obtained feature map feeds into the Region Proposal Network (RPN) to generate candidate regions.

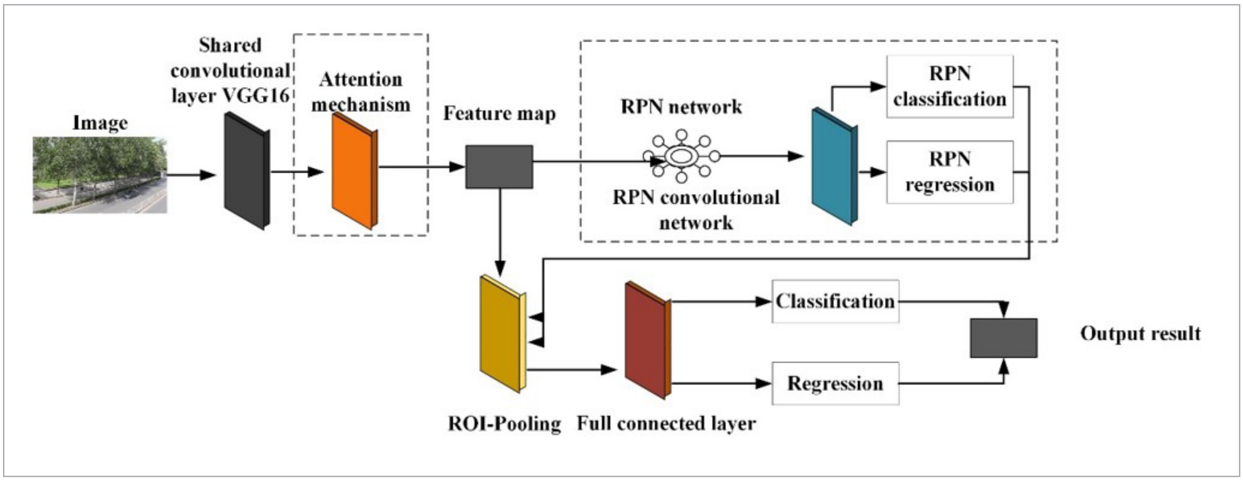
Step 2: Candidate Box Generation. The RPN's role involves producing candidate regions on the feature map, followed by binary classification of anchor boxes via the SoftMax function normalization. Further refinement of anchor boxes occurs through bounding box regression, enhancing the precision of proposal boxes.

Step 3: ROI Pooling. Proposal boxes acquired in the previous step, along with feature images from step 1, undergo aggregation in the ROI-Pooling layer. This process maps proposal boxes back to the feature maps, yielding Proposal Feature Maps. These maps are then fed into the fully connected layer for proposal categorization.

Step 4: Classification Regression. Boundary regression utilizes the target feature maps to compute prob-

Figure 1

The fundamental structure of Faster R-CNN



ability information for different target categories. The boundary regression network determines the precise location of the target object, resulting in detection outcomes.

Although Faster R-CNN can realize high-precision detection, it still has the following drawbacks: (1) The VGG16 network structure in the original model suffers from the problem of having many network parameters but few layers, which is prone to gradient vanishing. (2) The existence of 2-times rounding quantization in ROI-Pooling will cause a loss of accuracy. Aiming at these problems, the manuscript proposes improvements to Faster R-CNN.

3.2. Improvement of Faster R-CNN

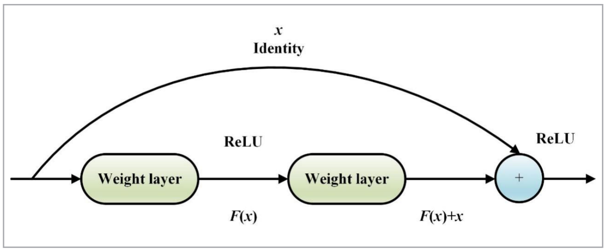
3.2.1. Selection of Feature Extraction Network

Faster R-CNN can better extract the features on the image, making the extracted features of paper disease finer, thus optimizing the subsequent detection efficiency.

The disadvantage of the VGG16 network structure is that the network parameters are many but the number of layers is small, while the ResNet-50 network is a deep CNN based on residual units, and the structure of residual units can be observed in Figure 2. Given an input x to the residual network, the initial convolutional layer conducts a convolution operation on the input. Subsequently, the second convolutional layer repeats the convolution operation on the preceding result. The outcome involves adding the convolution result to the original input x, yielding the ultimate output F(x)+x. When the CNN has the gradient dispersion problem, F(x)=0, the input and output are mapped to a constant mapping. The advantage of the residual network is that the information in the front layers is directly added to the back layers by connecting directly across the layers, thus effectively overcoming the problems of gradient vanishing and gradient explosion, and also making the algorithm easier to train and converge.

Figure 2

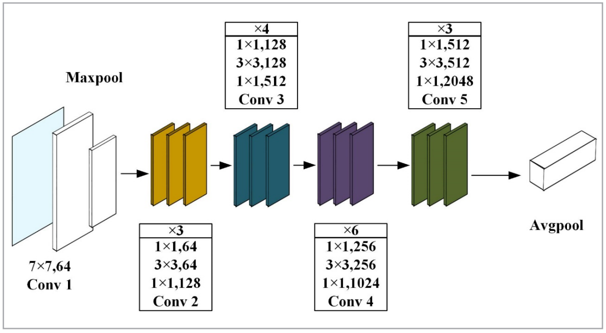
Residual element structure



The VGG16 feature selection network comprises 13 convolutional layers, with each followed by a pooling layer for feature map pooling. The repetition of pooling operations can potentially lead to the detriment of feature information. In Figure 3, the architecture of the substituted ResNet-50 network is illustrated. This alternative network integrates only 2 pooling

Figure 3

A structure of ResNet-50



layers, mitigating the issue of information loss associated with multiple pooling and preserving a greater amount of image feature information.

3.2.2. CBAM

CBAM serves as an attention module designed for feed-forward CNNs. It integrates spatial and channel attention mechanisms to emphasize crucial features in the feature map while filtering out non-essential ones. Illustrated in Figure 4 is the configuration of the CBAM module, encompassing both CAM and SAM.

Figure 4

CBAM module

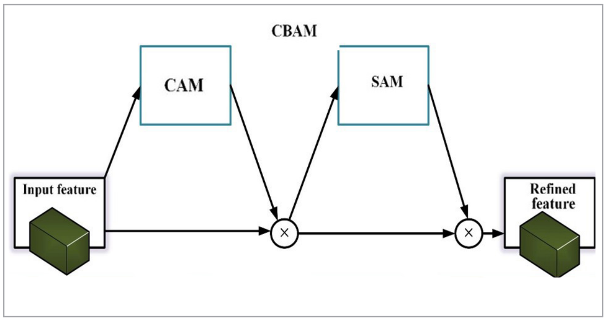
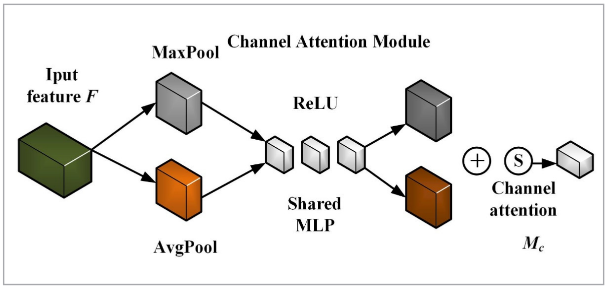


Figure 5

A CAM module



CAM module: The channel hierarchy remains unaltered, while the spatial hierarchy undergoes reduction. Illustrated in Figure 5, the initial step involves executing MaxPool and AvgPool operations in parallel on the input feature images to aggregate spatial information within the feature maps. This process generates 2 distinct spatial descriptions, representing average-pooled features and maximally-pooled features, respectively. The feature map size transitions from $C \times H \times W$ to $C \times 1 \times 1$, followed by input into a shared MLP module to produce 2 outputs. The shared network encompasses a hidden layer, with the hidden activation size set to $R^{C/r \times 1}$, where r denotes the reduction ratio. Subsequently, the two outputs undergo summation, and the channel attention output is derived through a sigmoid activation function. This result is then multiplied with the original image to revert to the size $C \times H \times W$. The rationale behind utilizing maximum pooling is to encode the most prominent features, allowing compensatory encoding of the average-pooled features. Simultaneous incorporation of these 2 features significantly enhances the network's representational power, surpassing the efficacy of relying on a singular pooling feature.

The CAM can be expressed by Equation (1).

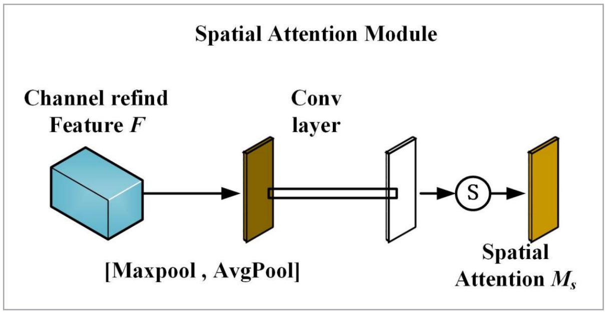
$$\begin{aligned} M_{\mathrm{C}}(F) &= sigmoid(MLP(AvgPool(F)) + \\ &+ MLP(MaxPool(F))) = \\ &= sigmoid\left(W_1\left(W_0\left(F_{avg}^{C}\right)\right) + W_1\left(W_0\left(F_{\max}^{C}\right)\right)\right), \end{aligned} \tag{1}$$

where F denotes the input feature. The sigmoid denotes the activation function. The weights belonging to MLP $W_0 \in R^{C/r \times C}$ as well as $W_1 \in R^{C/C \times r}$ are shared by 2 inputs.

A SAM module: The spatial hierarchy remains unaltered, while the channel hierarchy undergoes compression. As depicted in Figure 6, the results derived from the channel attention module undergo sequential maximum pooling and average pooling operations aligned with the CAM. These outcomes are interlinked to form a valid feature descriptor, resulting in 2 feature maps that are then seamlessly integrated. Following this, a 7×7 convolution operation is executed, yielding a single-channel feature map. The output of the spatial attention is subsequently derived through

Figure 6

A SAM module



the application of a sigmoid function. Finally, this result is multiplied with the original image to revert to the size $C \times H \times W$.

The SAM can be expressed by Equation (2).

$$M_S(F) = \text{sigmoid}\left(f^{7\times7}([AvgPool(F); \text{MaxPool}(F)])\right) = \\ = \text{sigmoid}\left(\left[F^S_{\text{avg}}; F^S_{\text{max}}\right]\right), \tag{2}$$

where $f^{7\times7}$ denotes a 7×7 convolution kernel.

3.2.3. Replacement of ROI-Pooling

Within Faster R-CNN, the RPN network produces candidate frames of varying sizes. To standardize the sizes of these diverse candidate frames, the ROI-Pooling layer is incorporated, facilitating uniformity through feature pooling. There are 2 rounding quantizations in ROI-Pooling, the first one is to quantize the positional coordinates of the input candidate frames' floating-point numbers into integers, and the second one is to divide the quantized candidate regions into M×M cells evenly, and then round the edges of each cell to quantize them. The first time is to quantize the positional coordinates of the input candidate frame floats into integers, and the second time is to divide the quantized candidate region into M×M cells, and then round the edges of each cell. These 2 quantization operations can easily introduce position errors in the measurement frame. This issue becomes less apparent when detecting small targets, ultimately leading to a reduction in detection accuracy. The research focuses on detecting driverless cars, where the planar geometric area of the car in the camera occupies a small portion of the photo and involves a limited number of image pixels. To address this, ROI-Align is employed to enhance the network's generalization ability.

The ROI-Align structure overcomes the quantization errors inherent in the ROI-Pooling structure. This is achieved by preserving the original floating-point scores and utilizing a bilinear interpolation algorithm to compute the pixel scores corresponding to the floating-point numbers of the image coordinates. The aim is to minimize quantization errors as much as possible. Unlike ROI-Align, ROI-Pooling introduces errors during the operation process since it relies on quantization operations. Consequently, ROI-Align's avoidance of quantization operations results in an enhancement of detection accuracy.

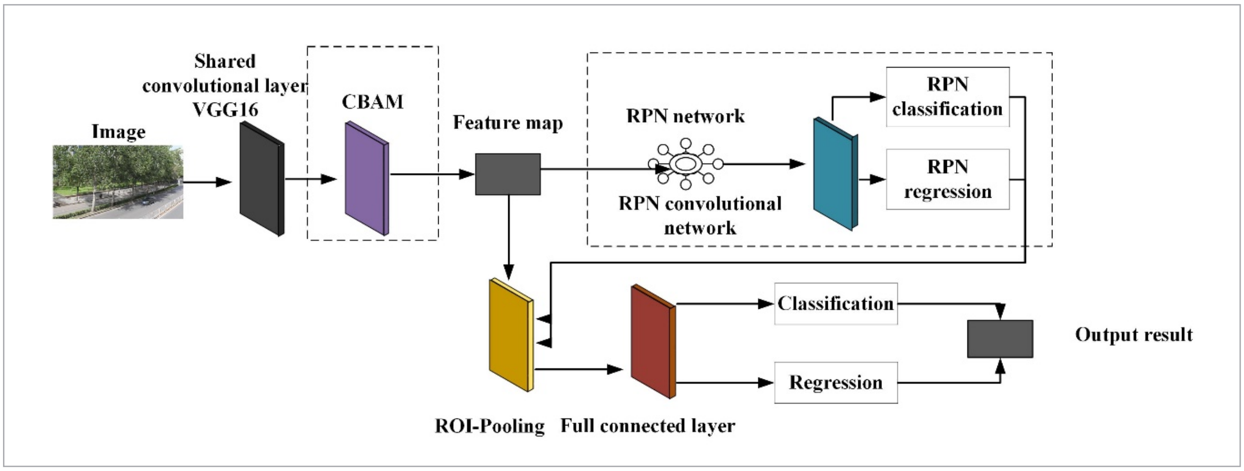
3.3. The Model Architecture of the Improved Faster R-CNN

The Faster R-CNN algorithm for detecting multiple targets in the context of vehicle recognition mirrors the fundamental structure of the Faster R-CNN algorithm, as elucidated in Figure 7. Image data, including labeling information, undergo processing within the feature extraction backbone network, resulting in the extraction of the image's feature map. This feature map serves as the input for the region candidate network, generating frames that are potential candidates for targets. These frames are then amalgamated with the feature map and directed into the domain-of-interest pooling layer, thereby ensuring a harmonized feature dimensionality. Ultimately, the fully connected layer receives multiple feature maps of standardized size, culminating in the conclusive steps of classification prediction and positional regression. Within the comprehensive framework, an uninterrupted flow of data extends from the input image to the final determination of the target's position and category. This design enables the end-to-end training of the entire model, fostering a seamless integration of the training and reasoning processes.

The network model's overall optimization loss function comprises 2 primary elements: the classification loss and the regression loss. These components are derived by comparing the network's predicted target categories with the manually labeled information on the location and detection frames of vehicles. This optimization process aims to enhance the network's proficiency in detecting vehicle targets. The initial

Figure 7

A model structure of the improved Faster R-CNN



loss function employed for computing the categorization loss is defined by

$$L_c = \frac{\sum_i -\left(y_i \log P_i + \left(1-y_i\right)\log 1 - P_i\right)}{N_c}, \tag{3}$$

where N_c denotes the number of candidate boxes generated by the regional candidate network. P_i represents the rate at which the candidate box with the serial number i predicts the category to be a vehicle in the outcome of the network. y_i Represents the real data of the candidate box with serial number i. The second loss function is employed to compute the location regression loss is defined as follows.

$$L_r = \frac{\sum y_i * S_{L1}\left(\beta_i^* - \beta_i\right)}{N_r} \tag{4}$$

$$S_{L1} = \begin{cases} \frac{x^2}{2}, & if\ |x|<1 \\ \frac{2\times|x|-1}{2}, & otherwise \end{cases}, \tag{5}$$

where N_r represents the total number of candidate boxes. β_i represents the predicted score of the position of the candidate box with the serial number i in the outcome of the network, which includes the coordinates of the four boundary points of the candidate box. β_i^* represents the real data of the candidate box with serial number i. The total loss for training the whole model is the weighted sum of the classification loss as well as the regression loss defined by

$$L = L_c + \varpi L_r, \tag{6}$$

where ϖ represents the weight parameter.

The training of the enhanced Faster R-CNN model involves 2 distinct stages. The initial stage focuses on training the feature extraction backbone network, while the subsequent stage is dedicated to the training of the region candidate network. The reason for doing so is that although the region candidate network shares some of the convolutional layer parameters with the backbone network, the roles of the 2 networks are not the same, and alternate training can make each part play its role better, and ultimately enhance the detection capability of the whole framework.

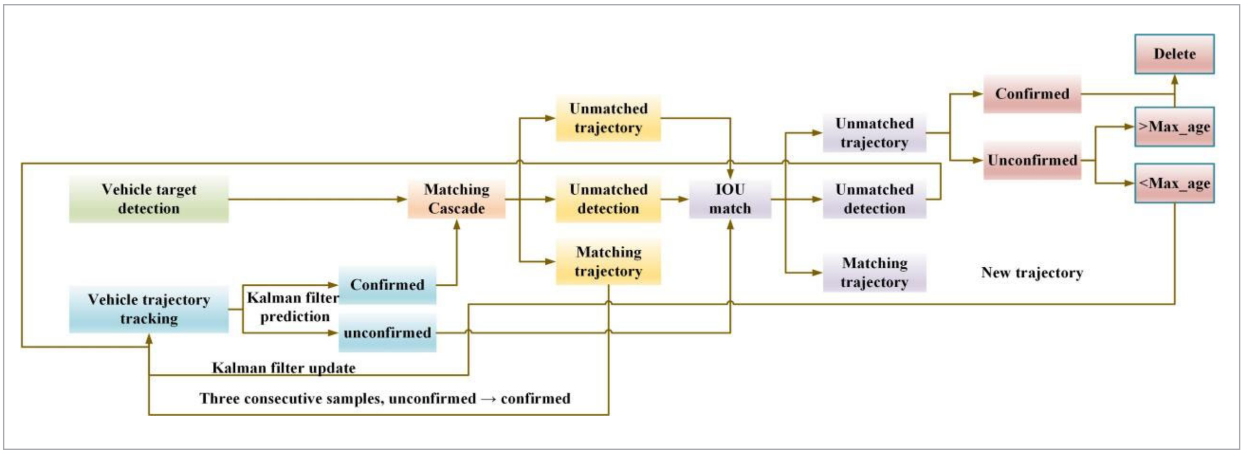
4. Trajectory Tracking Model for Intelligently Network-connected Driverless Vehicles in Narrow Areas based on an Improved DeepSORT

4.1. DeepSort

DeepSort stands out as a target-tracking algorithm rooted in deep learning principles, amalgamating deep learning and the conventional Sort algorithm

Figure 8

The DeepSORT process



for multi-target tracking within video sequences [3]. More up-to-date research can be found [6, 7, 12]. To enhance tracking performance, DeepSort introduces cascade matching on the foundation of the Sort tracking algorithm. The architectural representation of DeepSort is depicted in Figure 8.

The fundamental steps of DeepSort, outlined in [13], are as follows:

1. Employ the improved Faster R-CNN proposed in Section 3 to identify the driverless car within the image and retrieve the corresponding target detection frame.
2. Perform a cascaded matching by aligning the target detection frame with the prediction frame projected by the Kalman prediction from the preceding frame.
3. Reassess the matching results by re-evaluating Intersection over Union (IOUs) for failed detection frames and prediction frames within the cascade matching.
4. Assess whether the detection and prediction frames satisfy the conditions based on the matching outcome.
5. Revise the status of the target.

The core component of DeepSort, known as cascade matching, operates as follows: it initially identifies the detection frame A representing the driverless vehicle within the image, leveraging the enhanced Faster R-CNN. Subsequently, it employs the Kalman filter to forecast the target's motion from the preceding frame, yielding the prediction frame $\boldsymbol{B}$. Next, it integrates the detection frame A and prediction frame B along the target's motion trajectory. These frames are then fed into both the pedestrian reentry system and the driverless vehicle re-recognition network, facilitating the extraction of feature vectors for the pedestrian detection frame and prediction frame, respectively. Additionally, $\boldsymbol{A}$ as well as $\boldsymbol{B}$ undergo input into the pedestrian re-recognition network, enabling the extraction of feature vectors A and B for the driverless vehicle detection frame and prediction frame, correspondingly. The calculation of the minimum cosine distance between feature vectors A and B is performed utilizing Equation (7):

$$\boldsymbol{d}^{(1)}(i,j)=\min\left\{1-\boldsymbol{\lambda}_j^T\boldsymbol{\lambda}_k^{(i)}\mid\boldsymbol{\lambda}_k^{(i)}\in R_i\right\}, \tag{7}$$

where $\boldsymbol{\lambda}_j$ denotes the feature vector in the j-th detection frame and $\boldsymbol{\lambda}_k$ represents the k-th feature vector stored in the tracker. Afterward, we calculate the Mahalanobis distance (square) between the detection frames based on detection frames $\boldsymbol{A}$ and $\boldsymbol{B}$. Afterward, the Mahalanobis distance (square) matrix is calculated between detection frames $\boldsymbol{A}$ and $\boldsymbol{B}$ by Equation (8).

$$\boldsymbol{d}^{(2)}(i,j)=\left(\boldsymbol{d}_j-\overline{\boldsymbol{y}}_i\right)^T\sigma_i^{-1}\left(\boldsymbol{d}_j-\overline{\boldsymbol{y}}_i\right), \tag{8}$$

where $\boldsymbol{d}_j$ represents the eigenvector of a data point, $\overline{\boldsymbol{y}}_i$ shows the mean vector of the data set, and σ_i denotes the covariance matrix of the data set. The cost matrix is constructed based on the minimum cosine distance matrix and Mahalanobis distance matrix to construct the cost matrix.

$$\boldsymbol{\Omega}(i,j)=(1-\omega)\boldsymbol{d}^{(1)}(i,j)+\omega\boldsymbol{d}^{(2)}(i,j)\ , \tag{9}$$

where ω denotes the weight coefficient. The cost matrix is employed to represent the degree of match between each detected character frame and the predicted target position. Optimal matching is performed based on the cost matrix through the Hungary-aware algorithm to associate each detected target frame with its corresponding predicted target.

4.2. Improvement of DeepSort

4.2.1. The Improvement of Appearance Characterization

During the trajectory tracking of smart grid-connected driverless vehicles in narrow areas, the algorithm utilizes the appearance model to match the moving target when the target is occluded by objects for a long time. To attain more robust and efficient appearance features, the article introduces the full-scale network OSNet into the DeepSORT tracking algorithm to achieve a more accurate correlation between trajectory and detection.

The OSNet full-scale network is based on residual blocks to realize full-scale feature learning, and the generated multi-scale feature maps are fused by the channel weights generated by the Aggregation Gate (AG) dynamically. Furthermore, an additional crucial design tenet of the OSNet network involves crafting a lightweight architecture. To accomplish this, the standard convolution is deconstructed into point convolution and deep convolution within the support building module. Figure 9 illustrates the configuration of the OSNet network.

4.2.2. Optimization of IoU Matching Algorithm

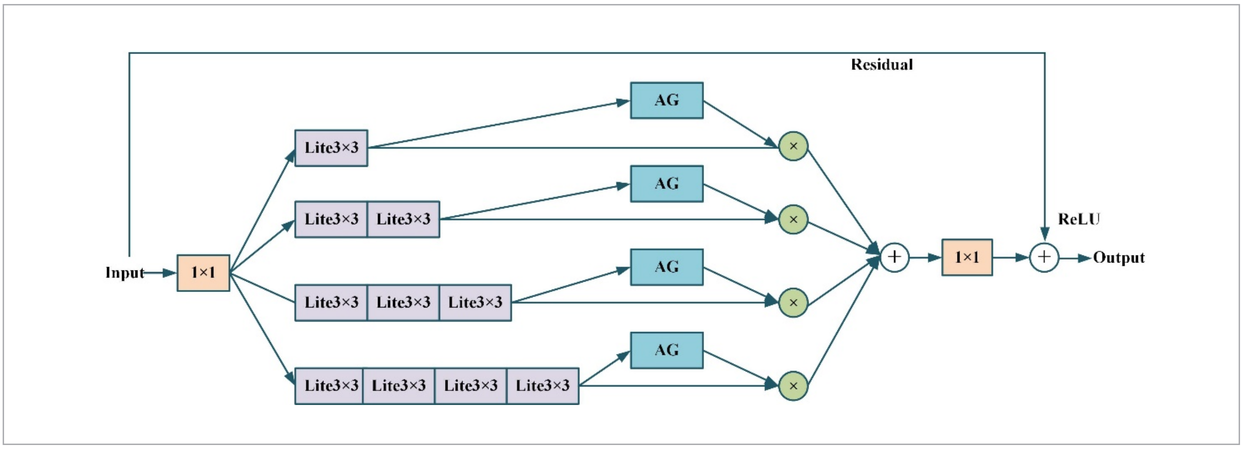
In the matching process of the DeepSORT algorithm, the result of matching through IoU is calculated by implementing the Hungarian algorithm to calculate its cost matrix and search for the optimal matching to judge the degree of matching when it is in a non-confirmation state.

Typically, IoU offers a more accurate portrayal of the relationship between 2 entities. However, it exhibits a lack of sensitivity to transformations in the target scale, particularly in the case of unmanned car targets susceptible to interference from similar targets. Moreover, when there is no overlap between the detection and the prediction frames, IoU remains consistently at 0, leading the algorithm to interpret the target as vanished, rendering further judgments impossible.

To address this issue, the research incorporates a Complete Intersection over Union (CIoU), augmenting IoU with a penalty term. CIoU factors in scale information regarding the overlap, center distance, and aspect ratio of the bounding box. Importantly, it

Figure 9

The structure of the OSNet network



retains the capability to indicate the movement direction of the bounding box even in scenarios where there is no overlap between target boxes. This enhancement serves to refine IoU matching and overcome the limitations posed by the previous scenarios.

Therefore, the computational process of CIoU matching is chosen in the trajectory tracking matching process as

$$CIoU = IoU - \frac{\rho^2(X,Y)}{l^2} - \mu\eta \tag{10}$$

$$\mu = \frac{IoU}{(1-IoU)+\eta} \tag{11}$$

$$\eta = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2, \tag{12}$$

where ρ^2 is the Euclidean distance between the two centers of the predicted and real frames. X and Y denote the predicted bounding box and the real bounding box, respectively. l represents the length of the shortest diagonal encompassing the predicted and real frames, v denotes a parameter for trade-off, and η represents a parameter measuring the similarity of aspect ratios.

w^{gt} and h^{gt} denote the width and height of the real frame, respectively. w and h represent the width and height of the predicted frame, respectively.

When m is 0, indicating that there is no intersection between the two, then the IoU will not directly remove the bounding regression box, but through the minimum enclosing box to calculate the minimum diagonal distance c between the bounding regression box and the detection box, the larger the c indicates that the 2 frames are farther away from each other.

$\frac{\rho^2(X,Y)}{l^2}$ is always greater than 0 and less than 1 and $\mu\eta$ is non-negative. The smaller the CIoU value, the worse the match, and vice versa.

In the process of driverless vehicle trajectory tracking, the CIoU matching method is employed to pinpoint the degree of match between the Kalman prediction and the detection results, which can reduce the phenomenon of state reconfirmation when the tracking process spans a large period, and decrease the number of ID switching.

5. Experimental Analysis

5.1. Data Source and Experimental Environment

The dataset, the UA-DETRAC, consists of 10 hours of video shot with a CannonEOS 550D camera at 24 different locations in Beijing and Tianjin, China [10] and is recorded at 25 (ps) per second with a resolution of 960x540 pixels, thus the video has more than 140,000 frames in the UA-DETRAC dataset, with 8,250 vehicles manually annotated and a total of 1.21 million labeled object bounding boxes, as shown in Figure 10.

The experimental environment is the lab host CPU model Intel Xeon(R) CPU E5-2686 v4 with 2.30 GHz, 16 GB×2 RAM, the GPU GeForce RTX 2080 Ti, the operating system Windows 10 64-bit system, and the framework Pytorch.

Figure 10

UA-DETRAC dataset

5.2. Evaluation Metrics for Algorithm Performance

The experiments are mainly categorized into target detection and trajectory tracking of intelligently network-connected driverless vehicles in a narrow area, and the evaluation metrics commonly implemented for these 2 tasks are presented in this section.

5.2.1. Evaluation Metrics for Target Detection

The main classification metrics used for vehicle target detection are Precision, Recall, and F-measure, which can be expressed in [18]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{14}$$

$$\text{F-measure} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{15}$$

True Positive, TP, signifies correct positive predictions, while False Positive, FP, denotes incorrect positive predictions. False Negative, FN, captures instances of missed positive predictions. The higher these three metrics are, the more accurate the target detection model is to classify the target of the detection frame.

The Mean Average Precision (mAP) serves as a crucial metric for assessing the comprehensive effectiveness of a target detection algorithm. mAP calculates the average precision across all classes and can be computed by Equation (16).

$$\text{mAP} = \frac{\sum P_{\text{classAve}}}{N_{\text{classes}}}, \tag{16}$$

where P_{classAve} represents the sum of the accuracies of all classes belonging to the test set. N_{classes} represents the number of target classes in the test set.

5.2.2. Evaluation Metrics for Trajectory Tracking

Intelligently internet-connected driverless vehicles in narrow areas mainly realize trajectory tracking by multi-target tracking within a single camera, which aims to differentiate each target from the others in each frame of the video, assign a unique identity ID to each moving target for identification, and record the motion trajectories of these targets at the same time. The article employs mainstream MOT evaluation metrics to assess the trajectory tracking performance. The evaluation metrics for trajectory tracking adhere to the standards set by the MOT evaluation criteria. The calculation for Multiple Object Tracking Accuracy (MOTA) is defined by Equation (17).

$$MOTA = 1 - \frac{\sum(\text{FP}+\text{FN}+\text{IDs})}{\sum \text{GT}}, \tag{17}$$

where FP indicates the total number of wrongly detected vehicle targets, and FN indicates the total number of omitted vehicle targets. The identity of the target vehicle in the tracking trajectory is ideally unique, but due to the complexity of the real scenario, the tracking algorithm will always make mistakes. IDs (ID Switch) represent the total number of times that the identity ID is assigned to the vehicle in the statistical trajectory is jumped. $\sum \text{GT}$ represents the total number of vehicle targets. The closer the value of MOTA is to 1, the higher the accuracy and better the performance of the tracking algorithm.

Multiple Object Tracking Precision (MOTP) denotes the degree of mismatch between the labeled and predicted frames of the actual tracked target and is calculated by Equation (18).

$$\text{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum \Omega_t}, \tag{18}$$

where t represents the current number of frames. $d_{t,i}$ represents the degree of overlap between the predicted frame and the real labeled frame of the i-th matching pair in frame t. Ω_t represents the number of actual matches between the predicted frame and the real labeled frame in frame t. MOTP is implemented to gauge the positional error in the results of the tracking algorithm, and the lower the value is, the better the tracking algorithm performs.

5.3. Experimental Results

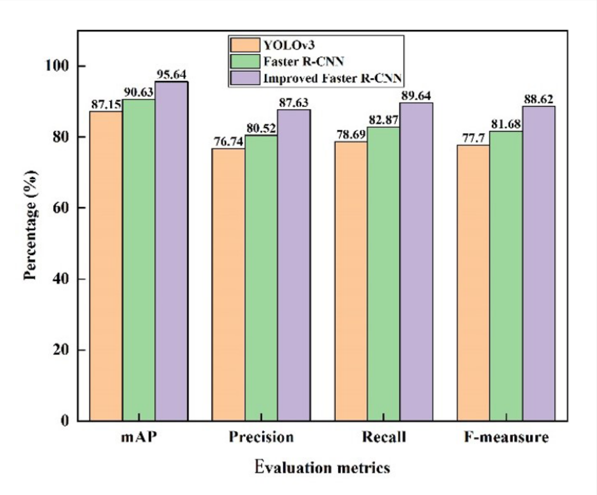
5.3.1. Experimental Results for Target Detection

The primary aim of assessing the performance of the target detection algorithm lies in evaluating the accuracy of algorithmic detection within the dataset. Therefore, it becomes imperative to appraise the metrics' performance, including Precision, Recall, F-measure, and mAP, for YOLOv3, Faster R-CNN, and the enhanced Faster R-CNN introduced in the research, as depicted in Figure 11.

As depicted in Figure 11, YOLOv3 attains a mAP of 87.15%, Precision of 76.74%, Recall of 78.69%, and F-measure of 77.7% in the test set. In comparison to YOLOv3, Faster R-CNN demonstrates enhanced detection accuracy, boasting a 3.38% increase in mAP and surpassing YOLOv3 in the remaining metrics.

Figure 11

Experimental results for target detection



The enhanced Faster R-CNN, introduced in the research, achieves notable results in the test set: 95.64% in mAP, 87.63% in Precision, 89.64% in Recall, and 88.62% in F-measure. This marks a substantial advantage over the preceding 2 algorithms, showcasing superior detection accuracy and heightened reliability. The improved Faster R-CNN lays a robust foundation for subsequent trajectory-tracking endeavors.

5.3.2. Experimental Results for Trajectory Tracking

The primary objective of assessing tracking algorithms is to evaluate their accuracy in tracking targets within a given dataset. The data utilized for detection comprises the target detection outcomes derived from the Faster R-CNN algorithm. The tracking accuracy results, as derived from SORT, DeepSORT, and the enhanced DeepSORT proposed in the research, are presented in Table 1.

Table 1 reveals that SORT achieves a MOTA of 85.4% and an MOTP of 0.173, respectively. In contrast, DeepSORT demonstrates a higher MOTA of 89.0% and a lower MOTP of 0.157. These metrics indicate a noteworthy enhancement in MOTA by 3.6% and a decrease in MOTP by 0.016 when compared to SORT. These improvements underscore the exceptional performance of the DeepSORT algorithm in trajectory tracking. Notably, the introduction of the appearance feature metric in DeepSORT contributes to the boosted tracking accuracy.

Moving on to the improved DeepSORT, it achieves an impressive MOTA of 96.1% and MOTP of 0.115. In comparison to DeepSORT, this represents a further 3.6% improvement in MOTA and a 0.016 reduction in MOTP, highlighting a substantial advantage. This improvement suggests an enhanced capability in apparent feature extraction by DeepSORT and effective optimization of the matching algorithm. Consequently, the improved algorithm enhances the tracking effectiveness for vehicle targets by minimizing identity switching and increasing stability throughout the tracking process, ultimately improving the completeness of the vehicle trajectory.

Table 1

Experimental results for trajectory tracking

Algorithms	MOTA	MOTP
SORT	85.4%	0.173
DeepSORT	89.0%	0.157
Improved Deep-SORT	96.1%	0.115

In summary, the proposed algorithms have good performance in practical applications and can be effectively implemented for trajectory tracking of intelligently network-connected driverless vehicles in a narrow area, which contributes to progressing driverless vehicle technology.

6. Conclusion

The manuscript introduces an enhanced target detection network based on Faster R-CNN for addressing the challenge of intelligent grid-connected driverless cars navigating through narrow spaces.

The proposed network incorporates ResNet-50, CBAM, and ROI-Pooling to amplify overall performance and enhance target detection accuracy. Additionally, an algorithm, comprising OSNet and CIoU, is integrated into the improved DeepSORT framework. This integration serves to boost feature extraction capabilities and optimize the matching algorithm, enabling precise tracking of vehicle trajectories utilizing a single camera in real road scenarios.

The findings highlight the notable accuracy achieved by the proposed target detection algorithm and the high precision in trajectory tracking.

Comparative analysis reveals superior application performance. The proposed approach effectively facilitates target detection and trajectory tracking of intelligently network-connected driverless vehicles within confined spaces. This achievement contributes to advancing the overall performance of driverless vehicles' movements, demonstrating significant potential for further enhancement in their capabilities.

Funding

This study was supported by the Foundation of the Natural Science of Universities in Jiangsu Province (No. 22KJB120002).

References

1. Bescos, B., Campos, C., Tardós, J. D., Neira, J. DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM. IEEE Robotics and Automation Letters, 2021, 6(3), 5191-5198. https://doi.org/10.1109/LRA.2021.3068640
2. Chen, S.-L., Ravi, S., Zebari, D. A., Zebari, N. A., Mohammed, M. A., Nedoma, J., Martinek, R., Deveci, M., Ding, W. Detection of Various Dental Conditions on Dental Panoramic Radiography Using Faster R-CNN. IEEE Access, 2023, 11, 127388-127401. https://doi.org/10.1109/ACCESS.2023.3332269
3. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T. StrongSORT: Make DeepSORT Great Again. IEEE Transactions on Multimedia, 2023, 25, 8725-8737. https://doi.org/10.1109/TMM.2023.3240881
4. Gündüz, G., Acarman, T. Efficient Multi-Object Tracking by Strong Associations on Temporal Window. IEEE Transactions on Intelligent Vehicles, 2019, 4(3), 447-455. https://doi.org/10.1109/TIV.2019.2919473
5. Jin, J., Wu, Y., Gao, J. Research on Target Detection and Tracking Algorithm for Campus Driverless Vehicle Based on Improved DeepSORT. In Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 2022, pp 1304-1308. https://doi.org/10.1109/CAC57257.2022.10055166
6. Lakhan, A., Mohammed, M. A., Abdulkareem, K. H., Deveci, M., Marhoon, H. A., Memon, S., Nedoma, J., Martinek, R. BEDS: Blockchain Energy-Efficient IoE Sensors Data Scheduling for Smart Home and Vehicle Applications. Applied Energy, 2024, 369, 123535. https://doi.org/10.1016/j.apenergy.2024.123535
7. Lakhan, A., Mohammed, M. A., Abdulkareem, K. H., Deveci, M., Marhoon, H. A., Nedoma, J., Martinek, R. A Multi-Objectives Framework for Secure Blockchain in Fog-Cloud Network of Vehicle-to-Infrastructure Applications. Knowledge-Based Systems, 2024, 290, 111576. https://doi.org/10.1016/j.knosys.2024.111576
8. Li, M., Zhai, D., Yang, D., Xu, L. BVTracker: Multivehicle Tracking Based on Behavioral-Visual Features. IEEE Sensors Journal, 2023, 23(11), 11815-11824. https://doi.org/10.1109/JSEN.2023.3265659
9. Li, Y., Zhang, S., Wang, W.-Q. A Lightweight Faster R-CNN for Ship Detection in SAR Images. IEEE Geoscience and Remote Sensing Letters, 2022, 19, 1-5. https://doi.org/10.1109/LGRS.2020.3038901
10. Luo, J., Fang, H., Shao, F., Hu, C., Meng, F. Vehicle Detection in Congested Traffic Based on Simplified Weighted Dual-Path Feature Pyramid Network with Guided Anchoring. IEEE Access, 2021, 9, 53219-53231. https://doi.org/10.1109/ACCESS.2021.3069216
11. Ma, Y., Zhang, J., Qin, G., Jing, J., Zhang, K., Pan, D. 3D Multi-Object Tracking Based on Dual-Tracker and D-S Evidence Theory. IEEE Transactions on Intelligent Vehicles, 2023, 8(3), 2426-2436. https://doi.org/10.1109/TIV.2022.3216102
12. Mostafa, S. A., Ravi, S., Zebari, D. A., Zebari, N. A., Mohammed, M. A., Nedoma, J., Martinek, R., Deveci, M., Ding, W. A YOLO-Based Deep Learning Model for Real-Time Face Mask Detection via Drone Surveillance in Public Spaces. Information Sciences, 2024, 676, 120865. https://doi.org/10.1016/j.ins.2024.120865
13. Sangsuwan, K., Ekpanyapong, M. Video-Based Vehicle Speed Estimation Using Speed Measurement Metrics. IEEE Access, 2024, 12, 4845-4858. https://doi.org/10.1109/ACCESS.2024.3350381
14. Tian, W., Lauer, M., Chen, L. Online Multi-Object Tracking Using Joint Domain Information in Traffic Scenarios. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(1), 374-384. https://doi.org/10.1109/TITS.2019.2892413
15. Wang, L., et al. CAMO-MOT: Combined Appearance-Motion Optimization for 3D Multi-Object Tracking with Camera-LiDAR Fusion. IEEE Transactions on

Intelligent Transportation Systems, 2023, 24(11), 11981-11996. https://doi.org/10.1109/TITS.2023.3285651

16. Wang, X., Fu, C., He, J., Wang, S., Wang, J. StrongFusionMOT: A Multi-Object Tracking Method Based on LiDAR-Camera Fusion. IEEE Sensors Journal, 2023, 23(11), 11241-11252. https://doi.org/10.1109/JSEN.2022.3226490
17. Wu, H., He, Z., Gao, M. GCEVT: Learning Global Context Embedding for Vehicle Tracking in Unmanned Aerial Vehicle Videos. IEEE Geoscience and Remote Sensing Letters, 2023, 20, 1-5. https://doi.org/10.1109/LGRS.2022.3228527
18. Wu, Y., Zhang, Z., Xiao, R., Jiang, P., Dong, Z., Deng, J. Operation State Identification Method for Converter Transformers Based on Vibration Detection Technology and Deep Belief Network Optimization Algorithm. Actuators, 2021, 10, 10030056. https://doi.org/10.3390/act10030056
19. Xie, Y., Wu, Y., Gao, J., Song, C., Chai, W., Xi, J. Emergency Obstacle Avoidance System of Driverless Vehicle Based on Model Predictive Control. In Proceedings of the 2021 International Conference on Advanced Mechatronic Systems (ICAMechS), Tokyo, Japan, 2021, 22-27. https://doi.org/10.1109/ICAMechS54019.2021.9661515
20. Yuan, D., Wang, Y. An Unmanned Vehicle Trajectory Tracking Method Based on Improved Model-Free Adaptive Control Algorithm. In Proceedings of the 2020 IEEE 9th Data-Driven Control and Learning Systems Conference (DDCLS), Liuzhou, China, 2020, 996-1002. https://doi.org/10.1109/DDCLS49620.2020.9275050
21. Zhu, Z., Li, R., Liu, H., Liu, R., Zhuang, W., Yin, G. Trajectory Tracking Control Design for Driverless Racing Car Considering Longitudinal Load Transfer. In Proceedings of the 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), Nanjing, China, 2022, 1-6. https://doi.org/10.1109/CVCI56766.2022.9964695