Vehicle recognition and tracking method based on roadside perception

Wenguan Cao[a,b*]

[a] The Architectural Design & Research Institute of Tongji University, Shanghai, China; [b] Shanghai Research Center for Smart Mobility and Road Safety, Shanghai, China

ABSTRACT

In vehicle detection and tracking methods based on roadside cameras, due to factors such as motion blur, vehicle occlusion, and target scale changes in the video, problems such as missed detection, false detection, and low positioning accuracy may occur during vehicle detection and tracking. This paper proposes to use the improved target detection model YOLOv5s as the target detector, then combined with the classical Deep SORT target tracking method, to achieve end-to-end vehicle detection and tracking. By integrating the attention mechanism with the detection network, and modifying the loss function, the ability of the model to extract features is strengthened, and the final vehicle detection accuracy is 96.7%. During tracking, the vehicle IDs are reduced to 28 times and the operating speed reaches 32 Hz.

Keywords: Roadside sensing, vehicle detection and tracking, YoloV5, attention mechanism

1. INTRODUCTION

In the intelligent transportation system, it is of great meaning to study the driving training project and automatic driving test project based on virtual reality. The use of virtual reality technology to digitize the real environment into a three-dimensional model makes driving training and automatic driving testing more convenient[1-2]. However, most of the traffic environment models in the current 3D models are generated by traditional tools based on analytical models, which are different from the real traffic environment. This paper perceives the real traffic environment based on roadside sensors, and extracts the vehicle's position trajectory and other information through deep learning algorithms. It is hoped that it can be imported into the 3D model to simulate the real traffic environment.

This paper mainly studies the field of multi-target tracking based on roadside vision sensors. Multi-target tracking normally includes two parts: target detection part and target tracking part. The difficulty of multi-target tracking lies in motion blur, vehicle occlusion, and target scale changes in the video. These factors will lead to problems such as missed detection, false detection, and low positioning accuracy during vehicle detection and tracking. In view of these difficulties, this paper uses the target detection model based on deep learning to detect vehicles, combines the detection results and the online tracking Deep SORT model to track the vehicle, and finally generates a real traffic environment.

2. RELATED WORK

In the field of object detection, traditional object detection algorithms use manual construction of feature descriptors of objects, and then use classification algorithms to classify and judge whether the objects exist. Common algorithms include Haar+Adaboost[3], Hog+SVM[4], and DPM[5]. However, these algorithms need to perform sliding window operation in the image. Sliding window operation will lead to low detection efficiency, large consumption of resources and other problems, and the artificially designed feature descriptors have poor generalization and low robustness effect, which is easy to lead to the phenomenon of missed detection and false detection. With the continuous development of deep learning and neural network technology, target recognition algorithms have gradually changed from traditional machine learning methods to deep learning-based methods, which are mainly divided into One-Stage structure and Two-Stage structure[4]. The Two-Stage algorithm first generates candidate regions during detection, and then classifies and calibrates on the basis of the candidate regions. The accuracy rate is relatively high, and the representative model has the R-CNN series[6-8]. The One-Stage algorithm does not need to generate candidate regions during detection, and directly regresses the target category and boundary, and the detection speed is fast. The representative models include SSD[9], RetinaNet[10], and YOLO[11-13] series.

* caowenguan@126.com

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 125063N
 · 0277-786X · doi: 10.1117/12.2662534

In the field of multi-target tracking, Bewley A proposed the SORT algorithm. The tracking part uses Kalman filter for state prediction and joint IOU to construct a cost matrix, and then uses the Hungarian algorithm to detect the relationship between the frame and the trajectory[14]. The tracking speed is fast, but the inside of the frame is not considered. Target characteristics, prone to identity switching. Yu Fengwei proposed the POI algorithm, using the improved Faster R-CNN for detection, introducing a pedestrian re-identification network in the tracking part to reduce pedestrian identity switching, using Kalman filtering and the output vector of the re-identification network to construct a similarity matrix, and finally combined with KM The algorithm associates the detection frame with the trajectory, and the tracking accuracy is high, but the real-time performance is slightly poor[15]. Wang Z proposes the JDE algorithm, which combines the pedestrian re-identification model and the detector, directly uses the detection network to extract features, and then combines Kalman filtering and data association algorithm for matching. In the case of mutual occlusion, the detection accuracy is reduced and the model is difficult to train and optimize end-to-end[16]. Pang B proposed the TubeTK algorithm, which uses 3-dimensional convolution to extract spatial and temporal dimension information in multi-target tracking tasks, and performs regression, which can effectively solve problems such as occlusion, inability to use motion information, and end-to-end training difficulties[17]. But the speed on MOT16 is only around 1Hz. On the basis of SORT, the DeepSORT[18] algorithm adds a shallow deep apparent feature extraction network, which greatly reduces the identity switching and improves the tracking accuracy.

Based on the above work, this paper proposes to use the classic light target detection network model YOLOv5s as the target detector, combined with the common Deep SORT target tracking method, to achieve end-to-end vehicle target detection and tracking. Aiming at the problem of low vehicle recognition rate, we combine the target attention mechanism with the object detection network to promote the ability of the model to analyze and extract vehicle features, and make the model pay more attention to the features of the detected target itself. In this paper, DIOU-NMS is used to replace the original NMS to improve the missed detection problem caused by vehicle occlusion. At the same time, the input size of the target feature extraction network of Deep SORT is adjusted and retrained on the vehicle re-recognition dataset. Finally, the detector and tracker are connected, and after debugging the parameters on the public dataset, the application is tested in the actual road surveillance video.

3. PROPOSED METHOD

3.1 The principle and improvement of YOLOv5 algorithm

3.1.1 YOLOv5 Algorithm Principle. YoloV5s is mainly composed of four parts, as shown in Figure 1, including: Input, Backbone network, Neck, and Precaution. Input mainly preprocesses the data through Mosaic data enhancement, adaptive image filling, and automatic setting of the initial anchor box size. The backbone network is mainly composed of BottleneckSCP[19] and SPP[20] (spatial pyramid pooling). BottleneckSCP is used to improve the inference speed and reduce the amount of calculation. SPP is used to extract multi-scale features from the feature map, so as to complete the extraction of different levels of features from the image. The Neck network layer includes path aggregation structure PAN and feature pyramid FPN. FPN uses a top-down approach to convey semantic information in the network, while PAN uses a bottom-up approach to convey localization information, and integrates different information from various network layers in the backbone network to improve the ability of object detection. As the final detection part, target prediction is primarily to predict objects of diversiform sizes on feature maps of different sizes.

YOLOv5 fused attention mechanism. For the image based on the roadside information source, the vehicle target is small and occupies less pixels. The Yolov5 model algorithm is insensitive to the vehicle feature information of the small vehicle target during convolution sampling, and the detection effect is poor[21]. The coordinate attention mechanism is introduced in this paper to reconstruct the attention of the final feature map so as to highlight the important and sensitive information in the feature map while suppressing the general information. For small targets and dense targets, the feature information can be effectively extracted and the accuracy class of detection can be further improved. Coordinate Attention (CA)[22] adds location information to channel attention, so that the network can pay attention to a larger area. In order to alleviate the problem of location information loss caused by two-dimensional global pooling proposed by previous attention mechanisms such as SENet[23] and CBAM[24], we decompose the channel attention mechanism divided into two parallel unidimensional feature encoding systems, which aggregate features in two directions respectively. In one direction, remote dependencies can be obtained, and in the other direction, precise location information can be obtained. The resulting feature maps are encoded to form a pair of direction-aware and position-sensitive features.

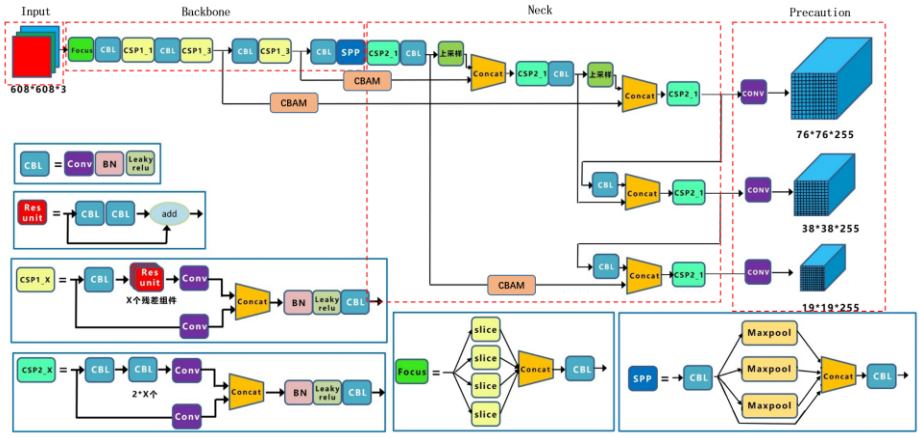

Figure 1. YoloV5 algorithm structure diagram

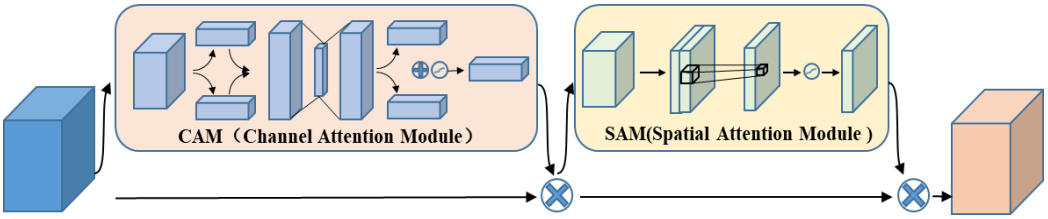

Figure 2. CBAM (Caption centered convolution Block Attention Module).

As shown in Figure 2, CBAM pays attention to both spatial and channel information, reconstructs the feature map among the total network through two sub-modules CAM (channel attention module) and SAM (spatial attention module), emphasizes important features, suppresses general vehicle features, and achieves the goal of improving the vehicle target detection effect. This paper integrates CBAM behind Backbone and before Neck network. The main reason for this is that YOLOv5s completes the feature extraction process in Back-bone, and then predicts the output on a large number of feature maps after Neck feature fusion. CBAM performs attention reconstruction here, which can play a crucial role in connecting the above and the next. The specific structure is shown in Figure 3.

3.1.2 Loss Function Improvements. At present, the main application of the minimum target detection based on the Anchor prediction mechanism is to continuously improve the measurement accuracy of the objects in the target prediction frame by measuring the height mean square error distance between the coordinates of the object in the minimization target prediction and the object coordinates of the enlarged target prediction frame. In the original YOLOv5 algorithm, the IoU Loss loss function uses the GIoU Loss loss function. GIoU Loss can also be a distance metric of a loss function, which can directly meet the metric requirements of the basic loss distance function. At the same time, because GIoU Loss also has a strong scale invariance, the expression is as follows:

$$L_{GIoU} = 1 - IoU + \frac{|c - b \cup b^{gt}|}{|c|} \tag{1}$$

However, when GIoU Loss contains two boxes, GIOU Loss will degenerate into IOU Loss and GIOU Loss need to iterate many times to converge, considering the shortcomings of GIou, the article introduces CIoU Loss, the expression is as follows:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{2}$$

Among them: α is the weight, v measures the similarity of the depth width ratio, b, b^{gt} separately represent the center point of the prediction frame and the target frame, and the distance between the two adopts the Euclidean distance ρ. c represents the slant distance of the smallest bounding box that can contain both the prediction box and the target box.

CIoU Loss can directly minimize the distance among the central point of the predicted point and the real point to accelerate concentration. At the same time, it also increases the distance loss that can detect different scales of the real frame, and increases the loss of length and width, so that the entire predicted frame will be more completely consistent. real box. So the CIoU Loss in the article replaces the original GIoU Loss, the effect will be better.

3.2 DeepSORT tracking algorithm

The DeepSORT algorithm employs recursive Kalman filtering to cope with data correlation frame-by-frame and uses the *Hungarian algorithm* to perform target screening and cross-frame matching on the output of the detector, and trains the vehicle re-identification model by using the residual network. The algorithm flow is shown in Figure 3. This paper selects the improved YOLO v5 target detection algorithm as the detector in the DeepSORT algorithm.

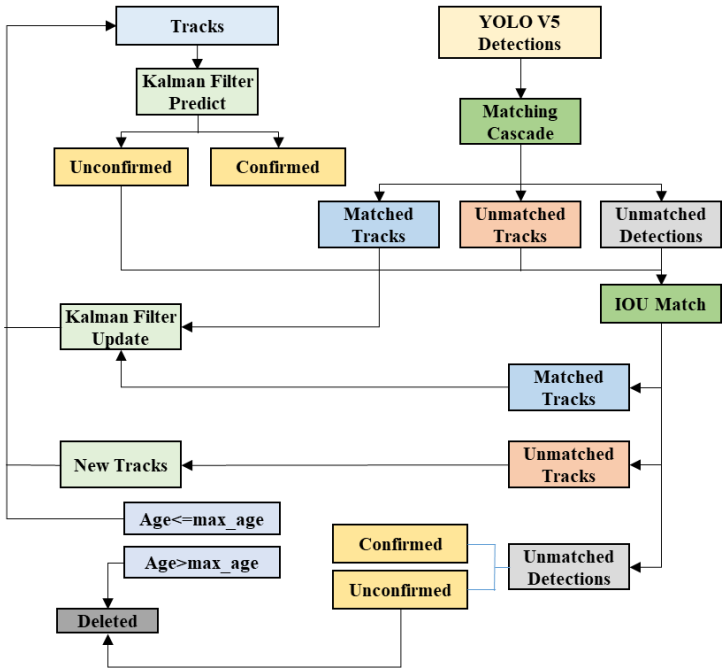

Figure 3. Algorithm flow chart of Deep SORT.

3.2.1 Data Association and Matching. Deep SORT merge appearance information with motion information to match predicted and tracked boxes using a Hungarian algorithm. For motion information, Deep SORT algorithm usually uses *Mahalanobis distance* to characterize the correlation coefficient between *Kalman* filter prediction results and detector results, as shown in formula (3):

$$d^{(1)}(i,j) = \left(d_j - y_i\right)^T S_i^{-1}(d_j - y_i) \qquad (3)$$

In the formula, d_j and y_i represent the state vector of the j-th detection result and the i-th prediction result, and S_i represents the covariance matrix among the detection result and the average tracking result. The *Mahalanobis distance* measures the standard deviation of the detection results from the average tracking results, taking the uncertainty of the state estimation into account, and can exclude low-probability associations.

When the certainty of target vehicle motion information is high, *Mahalanobis distance* performs well as an appropriate correlation factor. However, when there is target occlusion or camera Angle jitter in the environment, only using *Mahalanobis distance* correlation will make the identity of the target switch and eventually lead to target loss. Therefore, consider adding appearance information, calculate the corresponding appearance feature descriptor r_j for each detection frame d_j, and set $\|r_j\| = 1$. For each tracked trajectory $R_k = \{r_k^{(i)}\}_{k=1}^{L_k}$, set the feature warehouse $L_k = 100$, which is used to store the feature descriptors, L, which are successfully associated with the last 100 targets. Calculate the minimum cosine distance always among the *i*-th tracking frame and the *j*-th detection frame, as in formula (4):

$$d^{(2)}(i,j) = min\{1 - r_j{}^T r_k{}^{(i)} | r_k{}^{(i)} \in R_i\} \quad (4)$$

When $d^{(2)}(i,j)$ is less than the specified threshold, the association is considered successful.

Mahalanobis distance can give reliable target location information in the case of short-term prediction. The cosine similarity of appearance features can be used to restore target ID when target occlusion reappears. In order to complement the advantages of the two metrics, a linear weighting method is used. To combine:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \quad (5)$$

3.2.2 Vehicle Re-identification Model. The vehicle re-identification model calculates the cosine distance between classes through deep learning, trains the deep learning network offline, and obtains the deep learning weight[25]. The input object is based on the weight of the network, the direction of the decision boundary and the closest distance to find the bunch to which the object belongs, to achieve Tracking object object re-identification (Re-ID), the unification of metric learning and classification, while improving the recognition accuracy[26]. Using CNN to perform offline training on large-scale re-identification datasets, the CNN network structure is shown in Table 1. The network adopts a wide residual module, all convolution kernels are 3×3 in size, and convolution with stride 2 is used instead of max pooling[27]. When the spatial resolution is reduced, the number of channels is increased to avoid the bottleneck problem, Use an exponential liner unit (ELU) as the activation function throughout the network. The network structure is shown in Table 1.

Table 1. Re-identify network structure.

Layer name	Weight size	Output size
Conv_1	3×3/1	32×128×64
Conv_2	3×3/1	32×128×64
Max Pool _3	3×3/2	32×64×32
Res _4	3×3/1	32×64×32
Res _5	3×3/1	32×64×32
Res _6	3×3/2	64×32×16
Res _7	3×3/1	64×32×16
Res _8	3×3/2	128×16×8
Res _9	3×3/1	128×16×8
Dense _10		128
BN		128

4. EXPERIMENTS

4.1 Experiment preparation

Vehicle Detection and Tracking Dataset: The UA-DETRAC[28] dataset was collected in 24 different locations in Beijing and Tianjin, containing more than 140,000 video frames and 8,250 manually labeled car objects, totaling 1.21 million labeled object detection boxes. The dataset contains different traffic scenes, including highways, intersections, T-junctions, and different environmental backgrounds, including day, night, cloudy, and raining. Since the shooting angle of UA-DETRAC is close to the monitoring probe and the vehicle types are diverse, this dataset is selected as the experimental dataset of the detector YOLOv5 and the Deep SORT multi-target tracking dataset.

Vehicle re-identification dataset: The VeRi dataset is derived from 20 different real surveillance cameras. The cameras are installed at any position and direction within 1 km, and the shooting scenes include intersections, two-lane roads, and four-lane roads. The dataset contains more than 50,000 images of 776 vehicles, each captured by at least two cameras from different angles, lighting conditions, and environmental backgrounds.

The test platform is mainly composed of two main parts, the main hardware platform configuration includes: Intel(R) Core(TM) i7-7700CPU@3.60 GHZ, NVIDIA GeForce GTX 1080Ti. Software platform the first deep learning environment, including: Ubuntu18.04 OS, Cuda 9.0, Cudnn v7.5, Tensorflow 1.8.0-gpu, OpenCV 3.4.0. For the multi-target tracking of the front vehicle studied in this paper, perform vehicle re-identification pre-training and vehicle detector YOLO v5 training, and write Connect the script to realize the online tracking of the vehicle.

4.2 Model training

4.2.1 YOLO v5 Vehicle Detection Model Training. The backbone network of the detector model selects Darknet-53 for feature extraction, and YOLO v5 is trained using the UA-DETRAC dataset. Modify the original label, only keep the Car, van, Truck categories. According to the specific hardware information, the batch size is set to 16, the filter size is adjusted to 36, and the rest of the hyper parameters remain unchanged. The change trend of the loss function in the training process is shown in Figure 4. In the figure, L is the loss value.

4.2.2 Vehicle Re-identification Model Training. The training is based on cosine metric learning, using the constructed vehicle re-identification dataset, setting the batch α size to 32, the learning rate to 0.001, and the balance parameter γ to 0.085. The training results are shown in Figure 5. The training results show that after 400,000 iterations, the classification accuracy Ac is stable at 96.52%. At this time, the vehicle re-identification network has good classification ability and can accurately re-identify vehicles that have disappeared and reappeared in the field of vision.

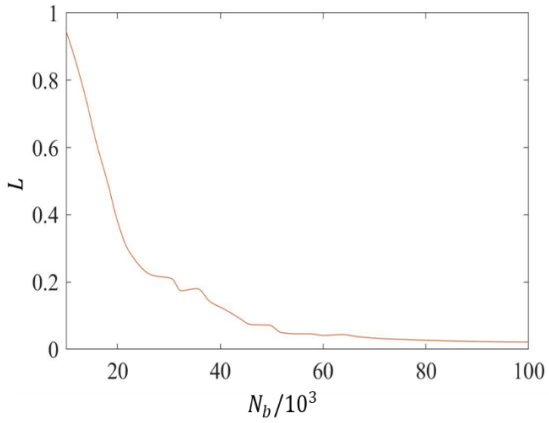

Figure 4. YOLOv5s loss curve.

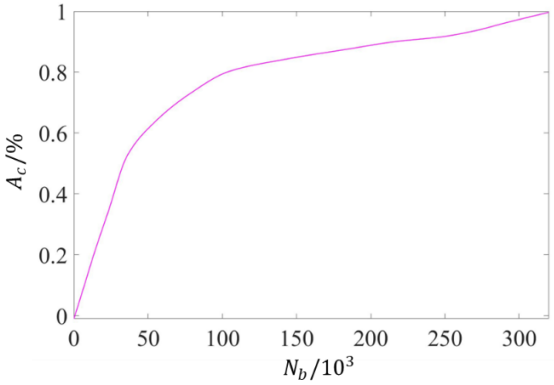

Figure 5. Change trend of training accuracy of vehicle rerecognition model.

4.3 Vehicle detection and tracking experiment and result analysis

In order to verify the three improvement strategies for YOLOv5s proposed in this paper, ablation experiments were carried out on the above data sets to judge the effectiveness of each improvement point, and CBAM and CIoU Loss were added to the initial YOLOv5s in turn. The pre-training model is not used in all experiments, and equivalent experimental parameters are configured during the training process. The experimental results are shown in Table 2.

Row 1 of Table 2 represents the base performance of the original YOLOv5s on the dataset, with an average detection accuracy of 93.8. After introducing CBAM and CIoU Loss respectively, it can be seen that CBAM improves the detection results significantly, Precision, Recall, and AP all improve significantly, while CIoU Loss improves slightly. The analysis believes that this is related to the different functions of the two modules. The attention mechanism aims to improve the network's ability to extract important features, and the result is the improvement of the accuracy rate, while the CIoU Loss is to speed up the regression of the prediction frame and improve the regression accuracy, so there is only

a small increase in detection accuracy. After introducing CBAM and CIoU Loss at the same time, the detection network achieves the best results, and the average accuracy AP is improved by 2.9% compared with the original network.

Table 2. Ablation of YOLOv5s.

CBAM	CIoU Loss	Precision	Recall	AP@0.5/%
--	--	90.3	94.3	93.8
	--	92.6	96.8	95.6
--		91.1	94.2	94.2
		94.2	97.1	96.7

For horizontal comparison, this paper selects Faster R-CNN, mobilenetv2-YOLOv4, and YOLOv5s three networks to train and test on the same dataset, all using pre-training models. The final experimental results are shown in Table 3, from which it can be clearly seen that the improved YOLOv5s is ahead of the other three networks in terms of weight size, detection rate Frames Per Second(FPS), and average precision AP. For the network weight size, the original YOLOv5s is 7.3 MB, and the network has only increased by 0.9 MB after the improvement.

Table 3. Comparison of different detection networks.

	FPS	Weight size/MB	AP@0.5/%
Faster R-CNN	12	316.00	89.2
Mobilenetv2-YOLOv4	80	53.15	90.8
YOLOv5s	103	7.30	93.2
Proposed	120	8.20	96.7

The algorithm in this paper is tested on MVI_20052, MVI_40163 and MVI_63521 in UA-DETRAC training set by connecting the improved YOLOv5s and the Deep SORT after vehicle re-recognition. The results are shown in Table 4.

Table 4. Vehicle tracing experiment.

	IDs/Times	Speed/Hz
SORT	76	60
DeepSort	40	36
YOLOv5-DeepSORT	28	32

Since the SORT algorithm only uses motion features as the basis for target association, a total of 92 occurred in the vehicle tracking of the above three data segments, Deep SORT is 57% lower than SORT, and the model after vehicle re-identification in this paper is further reduced. IDs, not only IDs are reduced to 28 times, but also the detection speed on the local test platform can reach 32 Hz, which meets the standard of real-time detection. The visualization results are shown in Figure 6. The vehicle disappeared at frame 1045 due to occlusion by obstacles, and the target could not be tracked. At frame 1076, the target reappeared, the target identity was restored, and the tracking was successful.

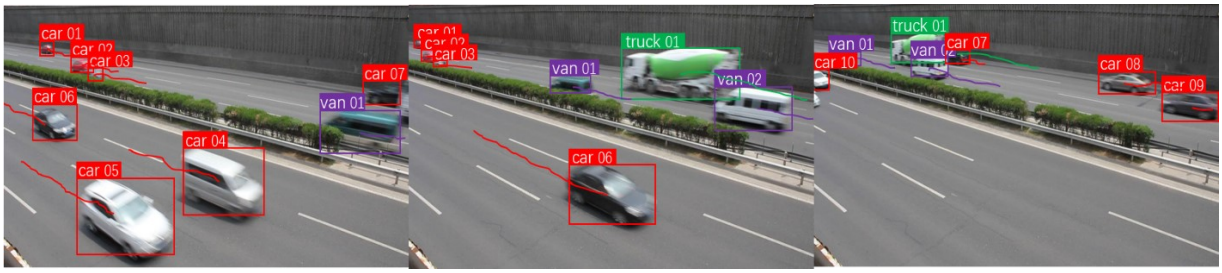

Figure 6. Vehicle tracking effect and ID recovery.

5. CONCLUSION

This paper uses YOLOv5s as the detector, combined with Deep SORT tracking for vehicle detection and tracking. Integrating the attention mechanism CBAM with YOLOv5s effectively improves the accuracy of the detector; using the improved CIoU Loss loss function further improves the positioning accuracy of the detector, and effectively improves the missed detection phenomenon in the crowded vehicle scene; the Deep The original feature extraction network in SORT performs input adjustment and re-identification training to make the algorithm more suitable for vehicle class applications. The experiments in this paper are all performed on devices with high computing power. Although the weight of the model is already small, it is still too heavy for edge devices. Therefore, the future direction is to further compress and prune the model and transplant it to embedded devices.

ACKNOWLEDGEMENT

This paper was funded by Shanghai Science and Technology Program 20DZ2252100 and Shanghai Young Science and Technology Talents Yang Fan Program 22Y2252100.

REFERENCES

[1] Lang, Y., et al. “Synthesizing personalized training programs for improving driving habits via virtual reality,” 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 297-304(2018).
[2] Salgado Fernández, J., Lara, F. J. S. and Granados, M. D. M., “Spatial skills training proposal in virtual reality learning environments,” International Conference on the Digital Transformation in the Graphic Engineering, 277-283(2021).
[3] Viola, P., “Rapid object detection using a boosted cascade of simple features,” Proc. IEEE CVPR 2001, (2001).
[4] Dalal, N., “Histograms of oriented gradients for human detection,” Proc of CVPR, (2005).
[5] Forsyth, D. A., “Object detection with discriminatively trained part-based models,” Computer, 32(9), 1627-1645(2014).
[6] Girshick, R., et al. “Rich feature hierarchies for accurate object detection and semantic segmentation,” IEEE Computer Society, (2017).
[7] Girshick, R., “Fast R-CNN,” Proceedings of the IEEE International Conference on Computer Vision, 1140-1448(2015).
[8] Ren, S., et al., “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Transactions on Pattern Analysis & Machine Intelligence,39(6), 1137-1149 (2017).
[9] Liu, W., et al., “SSD: Single shot multibox detector,” European Conference on Computer Vision, Springer, 21-37(2015).
[10] Lin, T. Y., et al., “Focal loss for dense object detection,” Proceedings of the IEEE international Conference on Computer Vision, 2980-2988(2020).
[11] Redmon, J. and Farhadi, A., “YOLOv3: An Incremental Improvement.” arXiv preprint arXiv, (2018).
[12] Bochkovskiy, A., Wang, C. Y. and Liao, H., “YOLOv4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv, (2020).
[13] Kuznetsova, Anna, Tatiana Maleva, and Vladimir Soloviev. “Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images,” International Symposium on Neural Networks, 233-243(2020).
[14] Bewley, A., et al., “Simple online and realtime tracking,” 2016 IEEE International Conference on Image Processing (ICIP), IEEE, (2016).
[15] Yu, F., et al. “Poi: Multiple object tracking with high performance detection and appearance feature,” European Conference on Computer Vision, Springer, Cham, (2016).

[16] Wang, Z., et al., “Towards real-time multi-object tracking,” European Conference on Computer Vision, Springer, Cham (2019).
[17] Pang, B., et al., “TubeTK: Adopting tubes to track multi-object in a one-step training Model,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, (2020).
[18] Wojke, N., Bewley, A. and Paulus, D., “Simple online and realtime tracking with a deep association metric,” 2017 IEEE international conference on image processing (ICIP), 3645-3649(2017).
[19] Wang, C. Y., et al., “CSPNet: A new backbone that can enhance learning capability of CNN,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 390-391(2019).
[20] He, K., et al., “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904-1916(2015).
[21] Liu, S., et al., “Path aggregation network for instance segmentation,” Proceedings of the IEEE Conference on Computer vision and Pattern Recognition, 8759-8768 (2018).
[22] Hou, Q., Zhou, D. and Feng, J., “Coordinate attention for efficient mobile network design,” Proceedings of the IEEE/CVF, (2021).
[23] Jie, H., Li, S. and Gang, S. “Squeeze-and-excitation networks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7132-7141 (2018).
[24] Woo, S., et al. “Cbam: Convolutional block attention module,” Proceedings of the European Conference on Computer Vision, 3-19 (2018).
[25] Wojke, N. and Bewley, A., “Deep cosine metric learning for person re-identification,” 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 748-756(2018).
[26] Krizhevsky, A., Sutskever, I. and Hinton, G., “ImageNet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems, 25(2), 324-335(2012).
[27] Zagoruyko, S. and Komodakis, N., “Wide residual networks,” British Machine Vision Conference 2016, (2016).
[28] Wen, L., et al. “UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking,” Computer Vision and Image Understanding, (2020).