Applied Mathematics, Modeling and Computer Simulation
C.-H. Chen et al. (Eds.)

doi:10.3233/ATDE221115

Research on Multi-Target Detection and Tracking Algorithm Based on Improved YOLOv5

Qian ZHOU [a] , Fuxin SUN [b], Junyou ZHANG [a,1]
[a] *Shandong University of Science and Technology, Qingdao, Shandong, China*
[b] *Shandong Port Qingdao Port Group Limited, Qingdao, Shandong, China*

Abstract. A detection and tracking algorithm based on improved YOLOv5 is proposed for the poor recognition and tracking of obscured targets and small-sized targets. The K-means ++ algorithm is used to cluster to obtain new anchor values; the CIOU-NMS is introduced to improve the missed detection problem when the target is obscured; the CBAM is proposed to be embedded into the Backbone and Neck part to improve the feature extraction capability of the algorithm for small targets. DeepSORT is chosen as the multi-target tracker to plot the motion trajectory of the target in real time. The experimental results show that the improved algorithm has a 2.1% improvement in detection accuracy and a detection speed of 32.32/s, satisfying real-time efficient detection with better tracking.

Keywords. Multi-object tracking, small-scale target, occlusion, YOLOv5, DeepSORT

1. Introduction

As a hotspot in the field of computer vision, multi-target detection and tracking has received extensive attention in the field of automatic driving[1].Real-time and accurate detection and tracking of vehicles, pedestrians and other targets is a prerequisite for autonomous driving. The commonly used Detection-Based Tracking (DBT) algorithm uses a detector to detect targets in the image sequence and then matches the detection results with the existing tracks to achieve tracking [2] .

Deep learning-based target detection algorithms can be divided into single-stage algorithms that directly generate object class probabilities and location coordinates, and two-stage methods that first generate pre-selected regions and then classify samples. Two-stage algorithms such as Faster RCNN have high detection accuracy but slow network computation speed, while single-stage methods such as YOLO and SSD can satisfy real-time detection but have lower accuracy. Currently, the YOLOv5[3] performs well in terms of detection speed and accuracy, providing a good balance between the two. Among the tracking algorithms, the SORT[4] uses Kalman filter and Hungarian algorithm to deal with motion estimation and data association problems to improve the efficiency of tracking, DeepSORT[5] enhances the association metric, uses CNN network training and extracts features from the dataset to improve the robustness of the network and effectively solved the occlusion problem.

1 Corresponding Author, Junyou ZHANG,Shandong University of Science and Technology, Qingdao, Shandong, China; E-mail: junyouzhang@sdust.edu.cn.

The difficulty of target detection and tracking lies in the missed or false detection caused by target occlusion and target scale change. To address these problems, Shi et al [6] optimised convolutional neural networks based on the Faster RCNN to improve detection accuracy, but at a slower rate. Yin Wang et al [7] proposed to fuse CBAM into YOLOv4 network and use PANet-D to fuse semantic information of multi-scale feature maps to improve the detection of small target vehicles. The detection model is large in scale and the k-means clustering is random in nature, which reduces the inference speed.

To further improve the robustness and accuracy of detection and tracking, this paper combines the YOLOv5 detection algorithm and DeepSORT tracking algorithm for multi-target detection and tracking of traffic scenes. Firstly, YOLOv5 is optimised and improved for the target occlusion and small size target problems that occur during detection. Secondly, in combination with DeepSORT, data correlation is performed using motion models and apparent information to achieve end-to-end multi-target vision tracking. The flow of the algorithm designed in this paper is shown in figure 1.

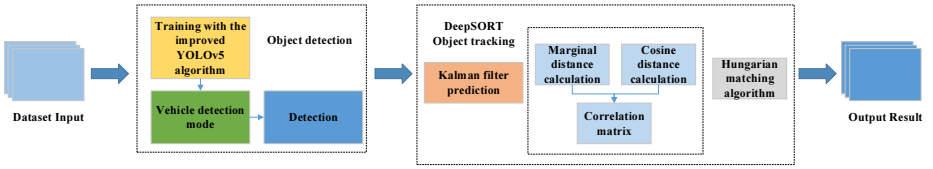


Figure 1. Flow chart of target detection and tracking algorithm

2. YOLOv5 Target Detection Algorithm

In traffic scenes with a high density of targets, feature information is more difficult to identify, which places greater demands on the detection algorithm.YOLOv5 has released several pre-training models, and this paper integrates the detection accuracy and speed of each pre-training model, and optimizes the YOLOv5l network on the basis of this to improve the detection accuracy of the algorithm for obscured targets and small-sized targets, and then lay a good foundation for accurate target tracking.

2.1. YOLOv5l Detection Model

The YOLOv5l network is divided into four parts: Input, Backbone, Neck and Prediction.

1) Input: Mosaic data enhancement, adaptive image filling and adaptive anchor frame calculation are used on the input side to pre-process images, increase sample diversity and improve inference speed.

2) Backbone: Backbone part mainly consists of Focus and CSP structures [8], which extract features from the image at different levels by deep convolution operations.

3) Neck: Neck network uses FPN (Feature Pyramid Networks), which conveys feature information top-down, and PAN (Pyramid Attention Network) structure, which conveys localization information bottom-up, both of which perform multi-scale fusion of features to enhance the feature information of the network.

4) Prediction: The prediction side outputs three feature maps of 19×19, 38×38 and 76×76 sizes, and uses GIOU to calculate the localization loss, combined with non-maximum suppression (NMS) to retain the information of the prediction frame with the highest confidence to complete the prediction.

2.2. YOLOv5 Algorithm Optimization

1) Optimized acquisition of Anchor based on driving scenarios.

The nine anchor frames in the YOLOv5 model were obtained by clustering the COCO dataset containing 80 categories by the K-means algorithm. This paper focuses on vehicles and pedestrians in traffic scenes, so the KITTI dataset, which focuses on vehicle images, is used for the experimental training set and testing. The K-means algorithm randomly selects K initial clustering centres at clustering time, which is not conducive to producing reasonable clustering results. Whereas the K-means + + algorithm uses the roulette method after randomly selecting an initial cluster center, to ensure that the distance between the cluster centers is as far as possible.The anchor obtained by the K-means++ algorithm can converge faster in the iterative process, so that a more suitable prior frame can be selected. The anchor values obtained after re-clustering are shown in table 1.

Table1. YOLOv5 feature map dimensions and corresponding Anchor

Feature map	Original Anchor	This article Anchor
19*19	(116,90)(156,198)(373,326)	(72,38)(99,61)(158,92)
38*38	(30,61)(62,45)(59,119)	(30,20)(47,27)(34,74)
76*76	(10,13)(16,30)(33,23)	(12,11)(20,14)(13,35)

2) Loss function and Non-maximum suppression optimization.

In the classical NMS algorithm, IOU is the only consideration. However, in practical application scenarios, target localization errors are prone to occur due to the dense target and the problem of missed detection. The original YOLOv5 network uses the GIOU loss function and NMS. GIOU is an improvement of IOU, which ensures that when there is no intersection area between the labeled and detected frames (IOU=0), the training model can still back-propagate to update the weight parameters; it also adds a measure of the overlap area between the predicted and real frames. However, when the prediction frame and the real frame are inclusion relationship, GIOU degenerates into IOU again, and the position of the labeled frame and the detection frame cannot be determined. To avoid filtering the effective prediction frame, the CIOU loss function [9] is introduced in this paper. CIOU not only considers the overlap degree of the detection frame and the labeling frame, but also considers the influence of the consistency of the aspect ratio and the distance of the center point of the two frames, and its calculation formula is shown in equation 1.

$$CIOU\_Loss = 1 - CIOU = 1 - \left(\left| \frac{A \cap B}{A \cup B} \right| - \frac{d^2}{c^2} - \alpha\tau \right) \tag{1}$$

where A and B denote the prediction frame and the real frame area, respectively; d denotes the distance between the center point of the prediction frame and the real frame, c denotes the diagonal distance of the minimum external rectangle; τ is the similarity of the aspect ratio of A and B.

In terms of the optimization of the NMS algorithm, this paper selects CIOU-NMS, which compares the difference between the IOU and the distance between the detection frame and the center point of the labeled frame, and keeps the score if the difference is less than the threshold, and removes the detection frame if it is greater than the threshold. The calculation formula is shown in equation 2.

$$s_i = \begin{cases} s_i, & IOU\text{-}R_{CIOU}\left(M,B_i\right) < \varepsilon \\ 0, & IOU\text{-}R_{CIOU}\left(M,B_i\right) \geq \varepsilon \end{cases} \tag{2}$$

where, ε is the NMS threshold, s_i is the classification score, M denotes the highest scoring predictor frame, and B_i denotes the i-th predictor frame.

3) Embedded attention mechanism.

In order to highlight the important information in the feature map and fully extract the location and semantic information of the target image, this paper introduces CBAM to introduce the location information by global pooling in the information encoding between channels. CBAM is a lightweight general-purpose module that can be integrated into existing network architectures as a plug-and-play module. It takes the feature map obtained from the network as input, inferring the attention weights along both spatial and channel dimensions in turn, and multiplying them with the original feature map to perform adaptive adjustment of the features.

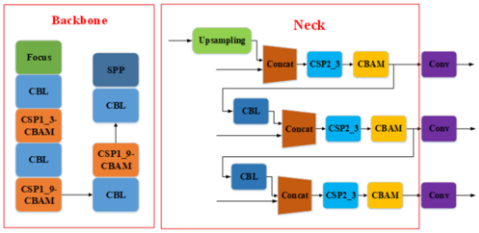


Figure 2. Embedded CBAM mechanism

The attention mechanism performs attention reconstruction on the feature map to highlight the important information in the feature map, and the Backbone network in YOLOv5 is responsible for extracting the main features, and the Neck network is responsible for Concat operation between the deep and shallow feature maps. Therefore, as shown in figure 2, the CBAM module is added to the residual structure of the backbone part and behind Neck's Concat operation, respectively, to increase the network perceptual field and refine the features.

2.3. DeepSORT Target Tracking Algorithm

DeepSORT uses recursive Kalman filtering to predict tracking targets frame by frame, associates data frame by frame using the Hungarian algorithm, and then calculates the IOU of the bounding box from the association metric, assigning a corresponding ID to each target [10],improving the identity jumping problem caused by occlusion. In this paper, we use a combination of motion and appearance information to improve the accuracy of the association.

1) Motion information matching: The Kalman predictions are matched with the actual measurements using the Mahalanobis distance, the covariance matrix is normalized, and the uncertainty of the state estimate is evaluated by calculating the detection and average track deviation to achieve motion information matching.

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

where: $d^{(1)}(i,j)$is the motion information matching result, y_i denotes the target prediction frame position of the i-th tracker, d_j is the j-th detection frame position, and S_i denotes the covariance matrix between the two. The association is considered successful if $d^{(1)}(i,j)$is less than the threshold $t^{(1)}$.

2) Appearance information matching: The cosine distance metric is used for correlation, and by extracting the appearance features of tracking target i and measuring the degree of matching its appearance information with that of detection target j, more accurate prediction is achieved. The calculation formula is：

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

where: $d^{(2)}(i,j)$ is the result of cosine distance metric, r_j is the feature vector of detection target j, the constraint is $\|r_j\|=1$, $R_i=\left\{r_k^{(i)}\right\}_{k=1}^{M}$ is the tracking target appearance feature vector library, storing M frames of appearance feature vector for each determined trajectory, generally M=100.If $d^{(2)}(i,j)$ is less than the association threshold $t^{(2)}$ then the association is considered successful.

3) Final matching: linear weighting of the mahalanobis distance and cosine distance as the final metric.

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

when $c_{i,j}$ lies within the intersection of the two metric thresholds, the correct association is considered to be achieved. The magnitude of the influence factors of the two metrics in the trajectory association is controlled by the hyperparameter λ.

3. Experiments and Results Analysis

The experiment is built using the deep learning framework Pytorch with Win10 operating system. The specific configuration and training parameters are shown in table 2. The KITTI dataset is selected for training and the performance of the improved YOLOv5 algorithm is evaluated. The performance of the detection model is evaluated using the mAP and the number of images processed per second (fps).

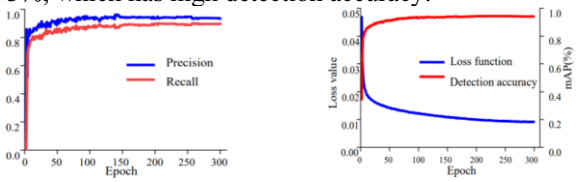
Table 2. Experimental environment configuration

Parameters	Configuration	Parameters	Configuration
Framework	Pytorch1.9	Training rounds	8
CPU	Intel Core I7-10700	Initial learning rate	0.01
GPU	NVIDIA GeForce RTX3060	Training iterations	300
GPU- environment	CUDA11.1 cudnn8.1	Weight decay	0.0005
Image input size	640×640	$t^{(1)}$	9.4877
λ	0	$t^{(2)}$	0.2

3.1. YOLOv5 Target Detection

1) YOLOv5 Network Training. In this paper, the learning rate is decayed by cosine annealing and the momentum is set to 0.937. The performance evaluation of the model obtained after training is shown in figure 3, which shows that the value of the loss function decreases sharply from iteration 0 to 200, and after 200 iterations, the loss

value tends to stabilize and the model reaches the optimal state, the accuracy P (precision) remains at 93%, the recall R (recall) is maintained at about 90%, and mAP is maintained at 93%, which has high detection accuracy.



(a) Accuracy and recall (b)Loss function and detection accuracy

Figure 3. Model performance evaluation

2) Comparative analysis of results. In this paper, Faster RCNN, YOLOv4-Tiny, YOLOv4-Mobilenet, and YOLOv5 are selected to train the KITTI training set, and the trained models are tested against the KITTI test set, and the final evaluation results after testing are shown in table 3.

Table 3. Performance test results of target detection algorithm

Models	Faster RCNN	YOLOv4-Tiny	YOLOv4 Mobilenet	YOLOv5	Ours method
fps(f/s)	12.57	88.48	41.35	35.30	32.32
mAP(%)	81.0%	72.2%	74.5%	93.3%	95.4%
AP	93.2%	83.4%	86.2%	95.7%	97.5%

From table 3, we can see that YoloV4-Tiny and YOLOv4-Mobilenet run fast but have low detection accuracy of 72.2% and 74.5%, respectively. Faster RCNN detection accuracy reaches 81.0%, but runs slowly at 12.57f/s, which cannot meet the real-time requirement. YOLOv5l detection accuracy has reached 93.3%, and our method improves 2.1% on this basis, and the detection speed reaches 32.32f/s, which meets the real-time requirements. In addition, analyzing various algorithms from the perspective of image detection for small size and occlusion, Faster RCNN and YOLOv5 can identify most of the occluded vehicles and small size vehicles with high detection accuracy; while YOLOv4-Tiny and YOLOv4-Mobilenet are difficult for the detection of occluded targets and small targets, and there are more missed detections; this paper improves algorithm has high accuracy and better improves the obscuration problem for obscured vehicles and small-sized vehicles.

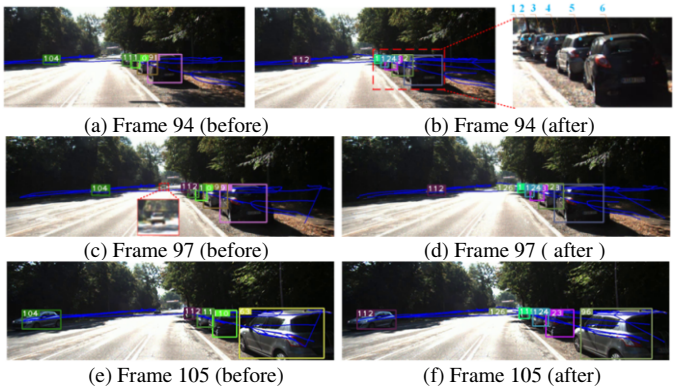
3.2. DeepSORT Tracking Results Comparison

To verify the effect of the optimized detection model on tracking, pedestrian and vehicle tracking experiments are done in this paper, and the results are shown in figures 4 and 5.

(a) Frame 20 (before) (b) Frame 19 (after) (c) Frame 237 (before) (d) Frame 224 (after)

Figure 4. Comparison of tracking results before and after improvement (pedestrian video)

In figure 4, (a), (c) and (b) (d) show the tracking results before and after the improvement, respectively, and the blue line shows the motion trajectory curve of the target. For the same target, (b) tracks the pedestrian one frame earlier than (a); for the occluded target, (c) resumes counting at frame 237 and (d) resumes counting at frame 224.



(a) Frame 94 (before) (b) Frame 94 (after)

(c) Frame 97 (before) (d) Frame 97 (after)

(e) Frame 105 (before) (f) Frame 105 (after)

Figure 5. Comparison of tracking results before and after improvements (vehicle video)

For vehicle detection, the zoomed-in image in frame 94 shows a total of 6 vehicles, of which car 6 is not obscured, cars 1, 3, 4 and 5 are more severely obscured, and car 2 is extremely obscured and almost featureless. Four were identified in (a) and five in (b); the small-sized vehicle in frame 97 was not identified in (c) and remained unidentified until the end of the video in frame 105, i.e. in (e), and the vehicle was successfully identified in (d) and tracked until the end of the video.

4. Conclusion

This paper proposes an efficient multi-objective detection and tracking algorithm by combining YOLOv5 with DeepSORT algorithm, targeting vehicles and pedestrians in traffic scenes. Firstly, K-means++ is used to replace K-means to improve the detection performance of the network. Secondly, the loss function of the network is optimised in order to reduce the missed detection caused by occlusion, while CBAM is introduced to improve the recognition of small targets by the algorithm. On the basis of this, combined with DeepSORT, the tracking frame centroids of the front and back frames of the target are connected to obtain the running trajectory of the target. After experimental validation, the improved algorithm in this paper achieves a 2.1% improvement in mAP on the KITTI dataset, while ensuring real-time detection speed. During the tracking process, the algorithm is able to detect the targets in advance, recover the counts of the occluded targets in time, and show better results for vehicle and pedestrian multi-target tracking.

References

[1] XIANG J, Xu G, et al. End-to-end Learning Deep CRF Models for Multi-object Tracking Deep CRF Models[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 31(1): 275-288.

[2] JIN S, LONG W, HU T et al. Research Progress of Detection and Multi-object Tracking Algorithm in Intelligent Traffic Monitoring System[J/OL].Control and Decision:1-13 [2022-06-09].
[3] WU T H, WANG T W, LIU Y Q. Real-time Vehicle and Distance Detection based on Improved Yolo v5 Network[C]. 2021 3rd World Symposium on Artificial Intelligence. IEEE, 2021: 24-28.
[4] BEWLEY A, GE Z, OTT L, et al. Simple Online and Realtime Tracking[C]. 2016 IEEE international conference on image processing. IEEE, 2016: 3464-3468.
[5] WOIKE N, BEWLEY A, PAULUS D. Simple Online and Realtime Tracking with a Deep Association Metric[C]. 2017 IEEE international conference on image processing. IEEE, 2017: 3645-3649.
[6] SHI Keijing, BAO Hong, XU Bingxin, et al. Forward Vehicle Detection Method of Intelligent Vehicle in Road Based on Faster RCNN[J]. Computer Engineering,2018,44(07):36-41.
[7] WANG Yin, WANG Feixiang, SUN Qianlai. Vehicle Detection Method Based on Multi Scale Feature Fusion[J].Journal of System Simulation:1-12[2022-06-09].
[8] WANG C Y, LIAO H Y M, YEH I H, et al. CSPNet: A New Backbone that can Enhance Learning Capability of CNN [J]. Computer Vision and Pattern Recognition, 2019: 11929.
[9] ZHENG Z, et al. Distance-IoU loss: faster and better Learning for Bounding Box Regression[C]. Proceedings of the AAAI Conference on Artificial Intelligence.2020, 34(07): 12993-13000.
[10] WOIKE N, BEWLEY A, PAULUS D. Simple Online and Realtime Tracking with a Deep Association Metric[J]. IEEE, 2017:3645-3649.