CIT. Journal of Computing and Information Technology, Vol. 30, No. 4, December 2022, 239–256
doi: 10.20532/cit.2022.1005718

Attention Mechanism and Detection Box Information Based Real-time Multi-Object Vehicle Detection

Hao Wu[1, 2, 3, 4], Wei Wu[5], Xiaoyan Sun[2], Jin Zhong[1] and Fengyun Cao[1]

[1]School of Computer Science and Technology, Hefei Normal University, Hefei, China
[2]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China
[3]Universities Joint Key Laboratory of Photoelectric Detection Science and Technology in Anhui Province, Hefei, China
[4]Key Laboratory of Philosophy and Social Science of Anhui Province on Adolescent Mental Health and Crisis Intelligence Intervention Hefei, China
[5]School of Economics and Trade, Anhui Business and Technology College, Hefei, China

Ensuring both the accuracy of vehicle target detection and meeting real-time requirements is crucial in traffic videos. The YOLOv5s target detection framework, known for its accuracy and efficiency, has attracted attention in academic circles. However, there are still some features that can be optimized. First of all, the detection subnet of the YOLOv5s framework cannot smoothly convert complex feature maps into relatively sparse target prediction boxes. To solve this, we integrate a self-attention-based gating mechanism into the detection subnet, forming the YOLOv5s-SAG network. Secondly, the loss function of CIoU used by YOLOv5s pays insufficient attention to the overlapping area of the detection frame, which can be used as metric for measuring target detection effectiveness. We add the loss term of area ratio to CIoU to further improve the modeling ability. Finally, the current multi-class Non-Maximum Suppression algorithm can cause high overlap of multi-class detection frames. To improve it, we propose a multi-class CS-NMS algorithm based on category suppression. Experimental results show an approximately 8% improvement in the mAP50 index on the UA-DETRAC dataset compared with YOLOv5s. The proposed algorithm also achieves better detection results compared to mainstream target detection algorithms and meets the real-time requirements of traffic video analysis.

ACM CCS (2012) Classification: Computing methodologies → Artificial Intelligence → Computer vision → Computer vision problems → Object detection

Keywords: YOLOv5s, AIoU Loss, multi-object detection, attention mechanism, CS-NMS

1. Introduction

Vehicle target detection refers to the rapid and precise acquisition of vehicle location and type information in traffic videos, which is a specific application of target detection algorithms for vehicle targets. As an advanced application of computer vision, target detection aims to find specific targets in image information and provide their locations.

In the actual traffic video scene, vehicle target detection can provide technical support for automatic driving, provided that vehicle target detection and automatic driving must be synchronized. Therefore, the detection algorithm is required to satisfy the requirements of both detection accuracy and real-time detection speed. In recent years, related algorithms for target detection can roughly be divided into two main types: detection algorithms based on traditional machine learning and those based on deep convolutional neural networks.

Traditional machine learning algorithms mainly divide the image into specific regions, perform feature extraction, and finally construct a classification model to determine whether the target exists and its specific type [1]. Early algorithms first used a fixed window to traverse each region of the image and then judged the target for each window. This type of algorithm

has a series of shortcomings such as long computation time, high overhead, and difficulty in adapting fixed windows to targets of different sizes.

To solve these problems, Uijlings *et al.* [2] proposed a non-fixed region selection algorithm, which divides the image into uneven regions according to specific rules, and filters the detection feature frames through the feature similarity of adjacent regions, greatly improving the efficiency of target detection.

For target feature extraction, many feature extraction techniques commonly used in image processing, such as wavelet decomposition [3], integral graph features [4], scale-invariant features [5], and directional gradient histogram features [6], have been applied to the field of target detection and achieved relatively good detection results. Classification models mainly classify the relevant features of the target and determine the category of the target. Common classification algorithms such as support vector machines are the most common classification algorithms.

However, traditional machine learning algorithms have relatively low detection efficiency, relatively poor robustness, and relatively high dependence on the environment for detection effectiveness, so they cannot meet the real-time vehicle detection requirements in different environments for traffic videos. With the continuous development of deep learning technology, especially deep convolutional neural networks in computer vision tasks, detection algorithms based on deep convolutional neural networks have gradually replaced traditional machine learning target detection algorithms due to their faster inference speed, higher robustness and environmental adaptability, and have received widespread attention in the field of target detection.

Deep learning-based target detection algorithms can be divided into two-stage target detection algorithms and single-stage target detection algorithms depending on the training method. Two-stage target detection algorithms preprocess the image, extract the candidate regions of the target, and then use CNN to perform target feature detection and classification operations on the candidate regions. This method has relatively high precision, but involves preprocessing operations, so the efficiency is relatively low. Typical two-stage target detection algorithms include SPP-Net [7], R-FCN [8], Faster R-CNN [9], Mask R-CNN [10], *etc.*

Single-stage target detection algorithms directly use CNN to extract features from images as well as perform target detection and classification operations. The detection efficiency is relatively high, but the detection precision is relatively weak. Typical single-stage target detection algorithms include SSD [11], YOLO [12–15], RetinaNet [16], YOLOX [17] *etc.*

In recent years, the YOLO algorithm has undergone multiple iterations and improvements, and its target detection accuracy has been greatly improved, attracting widespread attention in academia and industry. YOLOv5 uses CSP-Darknet53 as its backbone network, which has strong feature extraction ability and computational efficiency.

Moreover, a series of Intersection over Union (IoU) based target bounding box loss functions were used for model training. Rezatofighi *et al.* [18] and others proposed the Generalized Intersection Over Union (GIoU) based on IoU to solve the drawback that IoU cannot accurately reflect the size of the overlap between target and detection boxes. Zheng *et al.* [19] and others put forward DIoU loss and CIoU loss based on IoU respectively. DIoU introduces the distance between the center point of the target box and the detection box on the basis of IoU, which effectively solves the problem that GIoU makes it difficult to play its role due to the overlap of the target box and the detection box. CIoU introduces the length-width ratio of the target box and detection box on the basis of DIoU, which can achieve faster convergence speed and better detection effect. YOLOv5s can achieve excellent detection results with little inference time, which makes the algorithm suitable for real-time traffic video vehicle detection tasks.

This article applies the lightweight YOLOv5s network framework in the YOLOv5 [20] target detection algorithm to the task of vehicle target detection, proposes a vehicle target detection algorithm based on improved YOLOv5s, and improves the accuracy of vehicle target detection on the basis of ensuring the original YOLOv5s detection efficiency. The main contributions of this paper are as follows.

1. On the basis of the original YOLOv5s network structure, we integrate the threshold module based on the self-attention mechanism and propose the YOLOv5s-SAG network structure to enhance the effect of target detection;
2. We improve the CIoU bounding box loss function and propose a vehicle target detection loss function integrating the bounding box area ratio (AIoU), thereby enhancing the network's predictive ability of the target box position;
3. We improve the defects of the multi-category non-maximum suppression (NMS) algorithm in multi-category target detection and propose a category-suppression-integrated non-maximum (CS-NMS) target box screening algorithm to enhance the screening ability of detection boxes.

2. Related work

2.1. YOLOv5s Network Model

The YOLOv5s network model is a lightweight model in the YOLOv5 version that can be used for real-time target detection tasks. Its network structure is shown in Figure 1. The network is composed of Backbone, Head, and Detection modules. The input image first goes through the Backbone module, which is composed of CBS, C3, and SPPF submodules.

The CBS module is composed of a Conv layer, a BatchNorm layer, and a SiLU activation function. The C3 module is composed of several Conv layers, using bottleneck feature extraction [21] and residual connection [22] to obtain deep image features. The SPPF module integrates the above information using feature pyramid technology. The Head layer uses an FPN + PAN [23] network structure to deeply fuse the features, and its composition modules mainly consist of CBS and C3. The Detection module is responsible for outputting grid data of different scales. Since the channel numbers of grid data at different scales are different, a 1×1 Conv layer is used before each grid data output layer to adjust the channel number to the required number for detection features.

2.2. YOLOv5 Target Detection Loss Function

The loss function of YOLOv5 network training consists of calculating the sum of target classification loss, target confidence loss, and target bounding box location loss. Both target classification loss and target confidence loss generally use cross-entropy loss function. The calculation methods are shown in formulas (1) and (2), respectively.

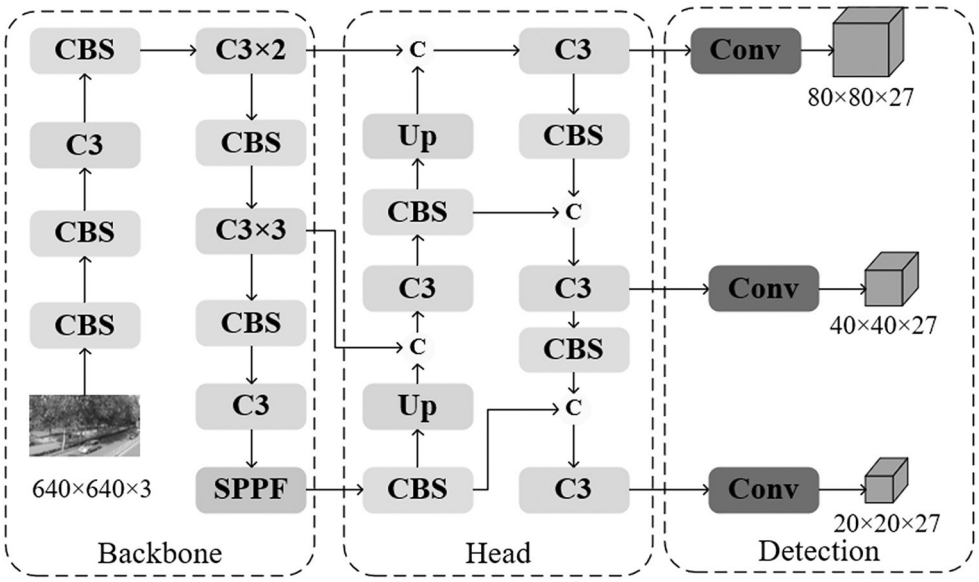

Figure 1. YOLOv5s Network structure.

$$L_{cls} = \sum_{i=0}^{S\times S}\sum_{j=0}^{B} 1_{ij}^{obj} \sum_{c\in classes} \Big[p_i(c)\log\big(p_i(c)\big) + \big(1-p_i(c)\log\big(1-p_i(c)\big)\big)\Big] \quad (1)$$

$$L_{obj} = \sum_{i=0}^{S\times S}\sum_{j=0}^{B} 1_{ij}^{obj}\Big[y_i\log\big(p_i\big)+\big(1-y_i\big)\log\big(1-p_i\big)\Big] - \sum_{i=0}^{S\times S}\sum_{j=0}^{B} 1_{ij}^{noobj}\Big[y_i\log\big(p_i\big)+\big(1-y_i\big)\log\big(1-p_i\big)\Big] \quad (2)$$

The target bounding box loss function mainly measures the difference between the detected bounding box and the actual bounding box. Initially, Intersection over Union (IoU) was used, which couldn't accurately measure the loss when the detection frame and the real frame did not overlap. Rezatofighi *et al.* [18] and others proposed a Generalized Intersection Over Union (GIoU) based on IoU, as shown in formula (3).

$$GIoU = IoU - \frac{A^c - u}{A^c} \quad (3)$$

GIoU has the shortcomings of slow convergence and inaccurate regression. Zheng *et al.* [19] and others put forward DIoU loss and CIoU loss based on IoU respectively. DIoU adds factors such as the ratio of the distance between the center points of the detection box and the real box to the diagonal of the minimum circumscribed rectangle on the basis of IoU, as shown in formula (4).

$$DIoU = IoU - \frac{\rho^2\left(b, b^{gt}\right)}{c^2} \quad (4)$$

Where $\rho^2(b, b^{gt})$ represents the Euclidean distance between the center points of the detection box and real box, and c^2 represents the diagonal distance of the minimum circumscribed rectangle of the detection box and real box.

On the basis of DIoU, CIoU incorporates the length-width ratio of the detection box and the real box, as shown in formula (5).

$$CIoU = IoU - \left(\frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \partial v\right) \quad (5)$$

Where ∂v represents the length-width ratio of the detection box and real box, and the calculation methods of v and ∂ are shown in formulas (6) and (7), respectively.

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \quad (6)$$

$$\partial = \frac{v}{\big(1-IoU\big)+v} \quad (7)$$

Here, ∂v represents the length-width ratio of the detection box and the real box. w, h, w^{gt}, $h^{gt}t$ respectively represent the width and height of the detection box and the real box. The loss function of CIoU is shown in formula (8).

$$L_{CIoU} = 1 - CIoU \quad (8)$$

2.3. Multi-category Non-Maximum Suppression (NMS) Target Box Screening Algorithm

The output of the model includes candidate boxes at different scale resolutions. At this time, a screening algorithm is needed to select appropriate detection boxes from the candidate boxes as the prediction results. Candidate box screening algorithms [24] generally use non-maximum suppression algorithms. Non-maximum suppression algorithms retain detection boxes with maximum confidence and delete those detection boxes that are similar to the maximum confidence detection box and have lower confidence than the maximum, thereby achieving the purpose of retaining suitable detection boxes from many candidate boxes.

The flow of this algorithm is described in Algorithm 1. In this context, B represents the candidate box list. To distinguish detection boxes that are close in position but belong to different categories, an offset value is added to the detection box according to the category, which is convenient for retaining candidate boxes of different categories in the subsequent NMS screening process. S denotes the list of confidence scores for each detection box in B, sorted in descending order. γ_1 represents the threshold

for candidate box screening. NMS generally uses IoU as the index to measure the similarity of candidate boxes. The algorithm deletes candidate boxes with IoU higher than γ_1.

Algorithm 1. Non-Maximum Suppression Algorithm.

Input: B, S, γ_1

Step 1: Get the index list of B, denoted as I, and create a new list K.
Step 2: If the length of I is greater than 0, otherwise exit the program and return K.
Step 3: Get the last element i of I and add it to K.
Step 4: If the length of I is 1, the program exits and returns K.
Step 5: Divide B into two groups according to the index i: the element b1 corresponding to the index i and the remaining elements b2.
Step 6: Calculate the IoU of each element in b1 and b2 separately, denoted as IoU.
Step 7: Retain the indices in IoU that are less than γ_1, denoted as idx.
Step 8: Assign idx to I and return to step 2.

Multi-class NMS algorithm can effectively suppress redundant candidate boxes and select as many suitable candidate boxes as possible as detection frames, so it is suitable for filtering the output of the YOLOv5 model. However, the multi-category NMS algorithm does not filter the candidate boxes of different categories, which could lead to the situation that the same object is marked as two categories and the detection boxes are very close. This situation is particularly common in vehicle target detection because different types of vehicles are similar.

3. Proposed Method

3.1. YOLOv5-SAG Vehicle Object Detection Algorithm with Fused Self-attention Gate

The output of the YOLOv5s [20] network is transformed from the complicated image feature layer to the target position detection layer through a simple 1×1 convolution layer. However, this method encounters two issues. Firstly, the dimension of the data before transformation is high, and it is not enough to simply use a convolution layer for channel reduction. These features should be further processed. On the other hand, compared to image features, the output detection box features should be "sparse". A gating mechanism can be added to filter the output data. Based on this, this article improves the YOLOv5s network, that is, a self-attention gating feature processing layer is added before the final detection box output to form the YOLOv5s-SAG structure. The network structure of YOLOv5s-SAG is shown in Figure 2(a).

Based on the original YOLOv5s network, a SAG module is added before each of the three feature output layers. As shown in Figure 2(b), the SAG module first passes the feature map through a 1×1 convolution layer for channel reduction. The data between channels is irrelevant, so the subsequent convolution layers of the SAG module are all deep-wise separable convolutions [25]. Then the feature map goes through a gating unit, which consists of two parallel branches composed of 1×1 and 3×3 depth convolution layers. One of the branches then goes through a sigmoid function for gating coefficients, and the product of these coefficients and the output of the other branch are passed through a 1×1 deep convolution layer to get the final result. The 1×1 convolution layer is used to adjust channels from the feature map to the model's output. The deep-wise separable convolution layer is mainly used to further extract the information from each channel. The gating mechanism is mainly realized by the sigmoid function, which can filter information by configuring weights. Suppose the input of SAG is $x \in R^{C\times H\times W}$, then the output of SAG can be obtained by formula (9). Where * represents two-dimensional convolution, ⊙ represents dot product.

$$y = W_2 * \left\{ \varnothing_1 \left(W_1 * x \right) \odot \sigma \left[\varnothing_2 \left(W_1 * x \right) \right] \right\} \quad (9)$$

Compared with the simple 1×1 convolution layer, the SAG module can filter and transform the picture representation information better by introducing the self-attention gating mechanism, and then output the results that are more suitable for the characteristics of the target boxes.

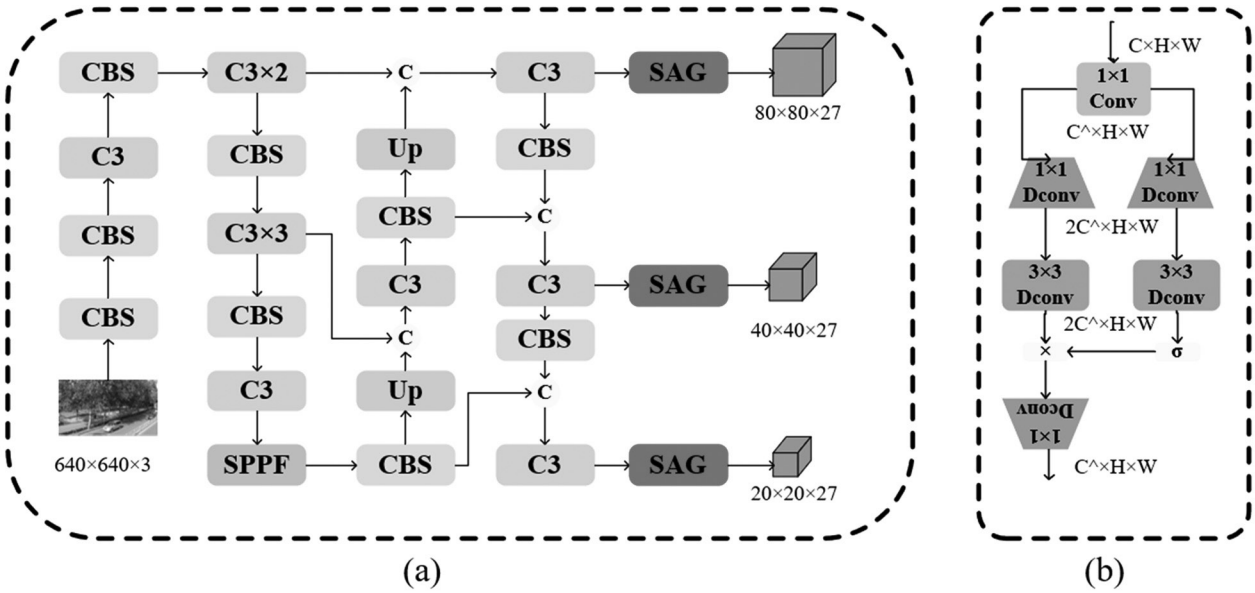

Figure 2. YOLOv5s-SAG network structure; (a) network body structure; (b) SAG module structure.

3.2. AIoU Vehicle Bounding Box Loss Function Based on Target Box Area Ratio

Based on the CioU [19] bounding box loss function, vehicle target detection tasks often encounter scenarios where the center point of the vehicle is accurately detected, but there is a certain deviation between the detection box and the area where the vehicle is located. The CIoU bounding box loss function integrates the deviation of the center point position and the length-width ratio of the detection box and the marked box based on the intersection over a union of the detection box and the marked box.

The target box and the detection box trained using the CIoU loss function can have close center points, high overlapping areas, and similar length-width ratios, but there are still some differences in the sizes of the target box and the detection box. A typical detection result trained using the CIoU loss is illustrated in Figure 3, where the target frame represented by green and the detection frame represented by red have obvious differences in area, although they have close center points, high overlapping areas, and similar length-width ratios.

Figure 3. Difference between target box and detection box trained by CIoU.

Based on this, this paper further incorporates the area deviation of the detection box and the marked box, and proposes an AIoU bounding box loss function based on target area ratio. This further enhances the accuracy of the detection box. The AIoU bounding box loss function is shown in formula 10:

$$AIoU = IoU - \left[\left(\frac{\rho^2\left(b, b^{gt}\right)}{c^2} \right) + \partial v + \ln\left(\min\left(\max\left(\frac{s_1}{s_2}, \frac{s_2}{s_1} \right), \gamma \right) \right) \right]. \tag{10}$$

The computation method for AIoU involves adding a loss based on area ratio and CIoU, where s_1 and s_2 represent the areas of the marked box and the detection box, respectively. After obtaining the larger area ratio, logarithmic operation is used to alleviate the instability of the loss function caused by the large difference in area ratio.

Furthermore, to prevent the situation where the network training is unstable due to the overweight of the area ratio loss, the maximum value of the area ratio is limited to a hyperparameter γ. To prevent the instability of the network training gradient caused by the larger bounding box loss function, we set the γ to 4.0 to limit the larger bounding box loss function.

3.3. Category Suppression Integrated Non-Maximum Suppression (CCNMS) Target Box Screening Algorithm

Determining the vehicle target detection box generally relies on the non-maximum suppression algorithm [24]. This algorithm selects the detection box with the highest confidence from several similar detection boxes for a certain object and deletes the detection box with a higher intersection over the union with the highest confidence detection box. For single-category target detection tasks, this algorithm can delete redundant detection boxes, thereby achieving better detection results. But for YOLOv5 multi-category target detection tasks, this algorithm may cause different category redundancy boxes to appear in the detection boxes. For example, in multi-class vehicle detection tasks, there are two nearly identical detection boxes for the same car, but these two boxes mark different categories, as shown in Figure 4. Here, the van-type truck is marked as both 'van' and 'others', resulting in two detection boxes that basically overlap in position.

The main reasons for this issue include two aspects. First, during the training process of YOLOv5, in the calculation part of the loss function, the loss calculation of each category is independent, and the existence of multiple categories for the same car is inevitable. On the other hand, after obtaining the candidate detection boxes, YOLOv5 will add a larger offset to the positions of different category detection boxes. When using the NMS algorithm to calculate the intersection over union, even if the areas where two different category detection boxes are located are roughly the same, their intersection over union is still 0, so it's not possible to discard them using the non-maximum suppression algorithm.

Figure 4. Instances of single targets marked by multi-category detection boxes.

Algorithm 2. Category Suppression Integrated Non-Maximum Suppression Algorithm.

Input: CB, B, S, γ_1, γ_2

Step 1: Get the index list of CB, denoted as I, and create a new list K

Step 2: Check whether the length of I is greater than 0, otherwise the program exits and returns K

Step 3: Get the last element i of I and add it to K

Step 4: Check whether the length of I is 1, if it is, the program exits and returns K

Step 5: Divide CB into two groups according to index i, the element cb1 corresponding to index i and the remaining elements cb2

Step 6: Divide B into two groups according to index i, the element b1 corresponding to index i and the remaining elements b2

Step 7: Calculate the intersection over union of cb1 and each element in cb2 respectively, denoted as IoU1

Step 8: Keep the indexes in IoU1 that are below γ_1, denoted as idx1

Step 9: Calculate the intersection over union of b1 and each element in b2 respectively, denoted as IoU2

Step 10: Keep the indexes in IoU2 that are below γ_2, denoted as idx2

Step 11: Assign the intersection of idx1 and idx2 to I, and return to step2

To effectively suppress defects of different category redundancy boxes appearing in multi-category vehicle target detection tasks, this paper improves the existing NMS algorithm and proposes a category suppression integrated non-maximum suppression (CCNMS) target box screening algorithm. This algorithm calculates IoU for both the original detection boxes and detection boxes with category offsets, uses different thresholds for screening, and obtains the final detection boxes after taking intersections. The flow of the algorithm is shown in Algorithm 2, where CB and B are lists of categories offset detection boxes and original detection boxes respectively, which are ranked from low to high according to scores, denoted as S. The program will separately calculate the non-maximum suppression filtering for CB and B, finally take the intersection of the results, and re-filter the intersection until there are no optional boxes left. γ_1 and γ_2 represent the non-same category IoU threshold and the same category IoU threshold respectively. γ_2 is generally set to 0.45, while γ_1 is determined by experiment in section 4.2.

4. Experiment Result Analysis

4.1. Experimental Environment and Dataset

Experimental Dataset: The experimental data is derived from the UA-DETRAC dataset [26]. The UA-DETRAC dataset is a large-scale multi-object vehicle detection database that documents a range of vehicle types in real-world road scenarios, including clear days, rainy days, and nighttime. There are four categories of vehicles, namely small cars (car), public transit buses (bus), vans (van), and other vehicles (others). The dataset identifies a total of 8,250 vehicles, with 1.21 million labeled frames. The images were extracted from continuous video at a rate of 25 frames per second, resulting in a large quantity of highly similar pictures. For the experiment, every tenth picture was sampled from the original dataset, yielding 8,208 images for the training set and 5,617 for the testing set. The proportion of each type of vehicle in the training and testing sets is displayed in Figure 5. According to Figure 5, 'cars' comprise the majority in both the training and test sets; the remaining categories are relatively less frequent, with 'others' being the least common, accounting for 0.6% and 2.5% in the training and validation sets respectively.

Model Training: Four models, namely YOLOv5s-CIoU, YOLOv5s-SGA-CIoU, YOLOv5s-AIoU, and YOLOv5s-SGA-AIoU, were trained based on the above training set and test set.

The output channel dimension in the network is set to 27 dimensions, which is the number of anchor box groups multiplied by the dimension of the detection box information. SAG module first reduces the number of input channels to 27, then increases the number of channels to 54 in the gating part, and finally reduces the output to 27. The parameter γ in AIOU loss is set to

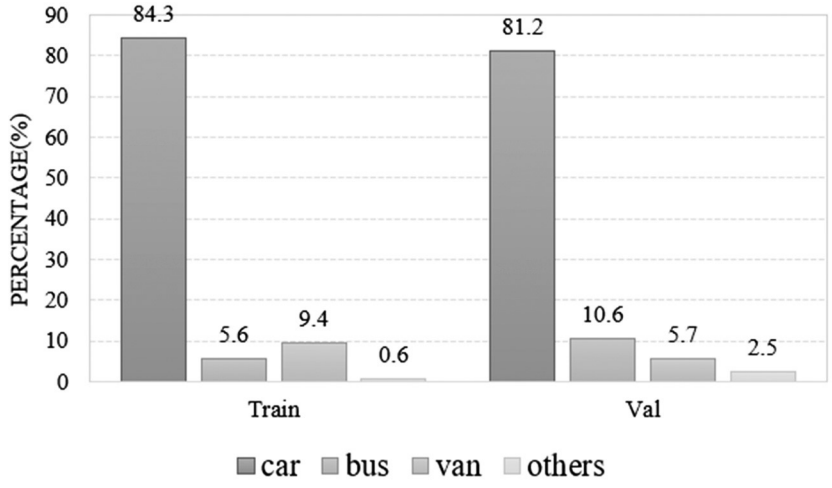

Figure 5. Percentage of vehicle categories.

4.0. The AdaM function was used to optimize the loss with an initial learning rate of 0.01. The experiment was iterated 100 times with a batch size of 32. The same category IoU threshold in CS-NMS is set to 0.45, and the non-same category IoU threshold is selected from 0.8 to 1.0. Evaluation Metrics: The evaluation metrics for object detection mainly consist of the PR curve composed of precision and recall, as well as the area constructed together with the coordinate axis. This area is referred to as the Average Precision (AP), with the Mean Average Precision (mAP) often used in multi-class object detection tasks serving as the evaluation metric for vehicle object detection in this study. mAP50 and mAP75 respectively represent the mAP value when IoU is 0.5 and 0.75, while mAP50-95 is an average of ten mAP values, ranging from an IoU of 0.5 to 0.95.

4.2. Experiment Results Analysis

4.2.1. The Impact of Network Structure on Model Performance

The paper presents an improved YOLOv5-SAG structure based on the YOLOv5s network structure. The network performance under the CIoU loss function is shown in Table 1.

From Table 1, the parameter count, and model size of the YOLOv5s-SAG network structure vary slightly from the original YOLOv5s, but the detection effect of the model has improved, with an overall mAP50 value increase of 1.6%. The mAP75 and mAP50-95 values are generally equivalent. In terms of detection efficiency, the average inference time of a single image for YOLOv5s is 3.9ms, with an average NMS time of 0.7ms, whereas the average inference time of a single image for YOLOv5s-SAG is 4.2ms, with an average NMS time of 0.6ms. Thus, their detection efficiencies are comparable.

4.2.2. The Impact of Loss Function on Model Performance

This study proposes an Area Ratio-based loss function (AIoU) that improves upon the CIoU loss function used in object detection in YOLOv5s. The performance of the models under different network structures is shown in Table 2.

According to Table 2, the loss function is applied during the model training stage, primarily adjusting the model parameters. As such, the number and size of model parameters do not change under different loss functions. The AIoU loss function proposed in this study significantly impacts the mAP50 accuracy of the model predictions. Compared with the CIoU loss function, the AIoU loss function can effectively improve the prediction accuracy of the model under different network structures.

Table 1. Comparison of Network Structure Effects.

Model	Number of Parameter	Model Size	Inference Time	NMS Time	mAP50	mAP75	mAP50-95
YOLOv5s	7030417	14.4MB	3.9ms	0.7ms	0.525	0.46	0.379
YOLOv5s-SAG	7036735	14.4MB	4.2ms	0.6ms	0.541	0.462	0.38

Table 2. Comparison of Loss Function Effects.

Model	Loss Function	mAP50	mAP75	mAP50-95
YOLOv5s	CIoU	0.525	0.46	0.379
YOLOv5s	AIoU	0.539	0.466	0.389
YOLOv5s-SAG	CIoU	0.541	0.457	0.377
YOLOv5s-SAG	AIoU	0.565	0.487	0.398

The mAP50 results of the YOLOv5s and the YOLOv5s-SAG network structure proposed in this study using AIoU improved by 1.4% and 2.4% compared to CIoU, respectively. This shows that the area ratio-based loss function can be effectively applied to vehicle object detection tasks. Overall, the YOLOv5s-SAG-AIoU algorithm significantly improves the mAP50, mAP75, and mAP50-90 metrics compared to other algorithms, indicating that the self-attention gating mechanism combined with the area ratio loss function can achieve better detection effects.

Figure 6 provides the PR curves of the mAP50 of the four algorithms. According to Figure 6, the mAP50 value of the 'car' category for the YOLO5s-CIoU algorithm is 0.75, for the 'bus' category it is 0.751, for the 'van' category it is 0.475, and for the 'others' category it is 0.124. Due to the significant differences in the proportion of each category in the training set, the results for the 'van' and 'others' categories are relatively poor because they comprise a smaller proportion of the training data. Except for the 'car' category, whose mAP50 value is slightly weaker than the YOLO5s-CIoU algorithm,

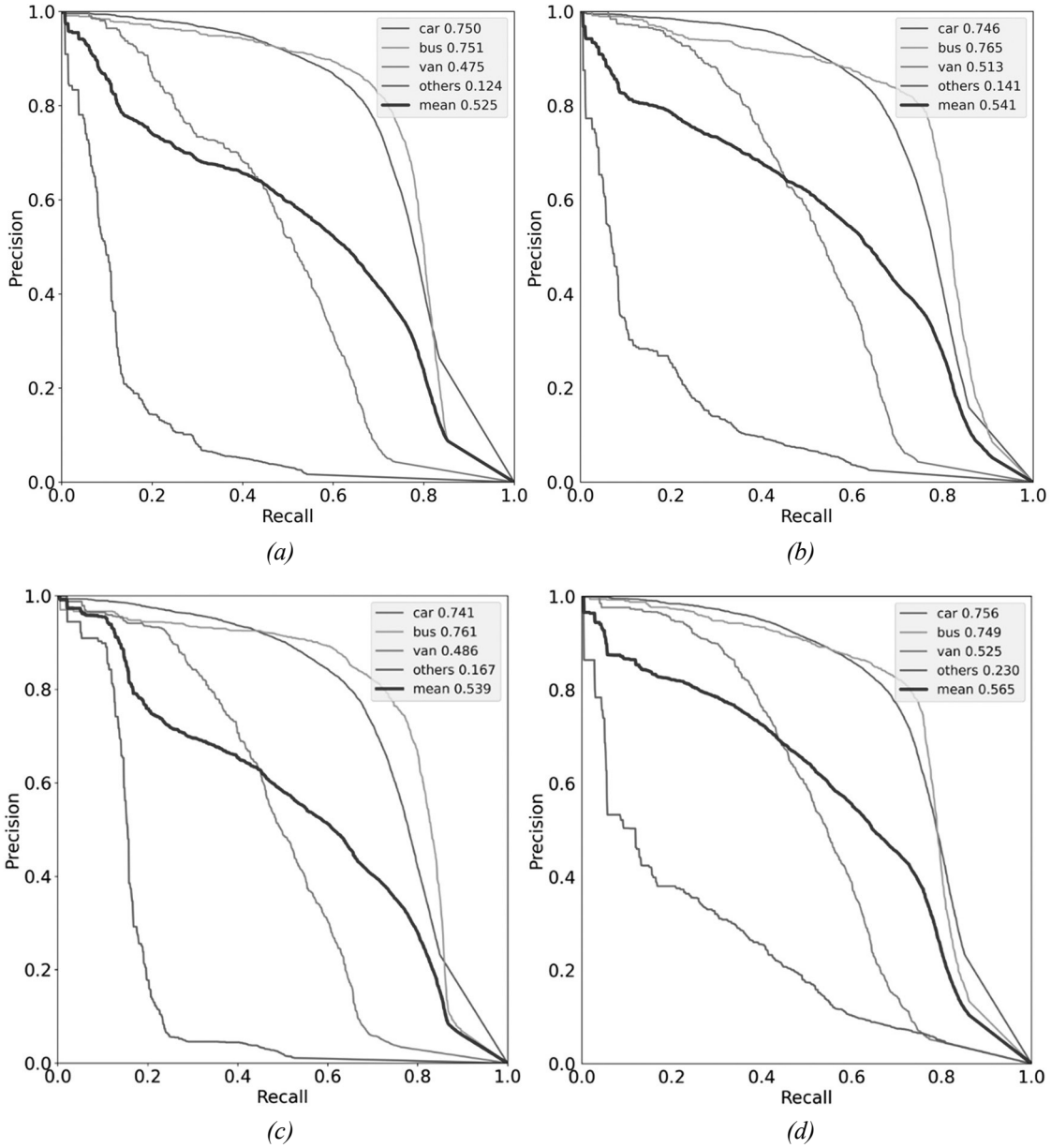

Figure 6. PR curves of mAP50 for four algorithms; (a) YOLO5s-CIoU; (b) YOLO5s-SAG-CIoU; (c) YOLO5s-AIoU; (d) YOLO5s-SAG-AioU.

the mAP50 values for all other categories in the YOLO5s-SAG-CIoU and YOLO5s-AIoU algorithms are superior to those of the YOLO5s-CIoU algorithm. This indicates that these two algorithms are more balanced and generally perform better compared to the YOLO5s-CIoU algorithm. Aside from performing slightly weaker in the 'bus' category compared to the YOLO5s-CIoU algorithm, the YOLO5s-SAG-AIoU algorithm outperforms the YOLO5s-CIoU algorithm in all other categories. Especially in the 'others' category, which has less training data, its mAP50 has significantly improved compared to the other three algorithms, indicating that the YOLO5s-SAG-AIoU algorithm can yield excellent detection results.

Figure 7 presents a comparison of the detection effects of YOLO5s-CIoU, YOLO5s-AIoU, and YOLO5s-SAG-AIoU under different scenarios. In the first row, which depicts a densely populated vehicular environment, YOLO5s-AIoU manages to detect more of the relatively smaller vehicles in the upper left corner compared to YOLO5s-CIoU. However, it fails to identify the obvious vehicles at the bottom right and on the left. In contrast, YOLO5s-SAG-AIoU not only detects the small vehicles in the upper left but also successfully identifies the vehicles on the right and left.

In the second-row scenario, YOLO5s-CIoU misidentifies non-vehicle targets on the left as vehicle targets. YOLO5s-AIoU avoids this situation but misses some vehicles occupying small areas compared to YOLO5s-CIoU. On the other hand, YOLO5s-SAG-AIoU effectively avoids both false positives and missing small targets.

In the third-row scenario, YOLO5s-CIoU exhibits a case where different categories of detection boxes appear for a single vehicle target. The small car below is marked as both 'car' and 'others' categories, while the bus on the right is marked as both 'bus' and 'car' categories. The position and size of the marked boxes are basically consistent, representing the aforementioned multi-category redundant box situation. YOLO5s-AIoU removes the redundant box of the bus but retains the redundant one for the small car. However, YOLO5s-SAG-AIoU deletes both the redundant boxes for the bus and the small car.

Figure 7. Detection effects of YOLO5s-CIoU (left), YOLO5s-AIoU (middle), and YOLO5s-SAG-AIoU (right).

The visualization results under these different scenarios reveal that the detection effectiveness of YOLO5s-AIoU is superior to that of YOLO5s-CIoU, while the performance of YOLO5s-SAG-AIoU significantly surpasses both YOLO5s-CIoU and YOLO5s-AIoU. This suggests that the YOLO5s-SAG network structure and AIoU loss function proposed in this study can effectively enhance the performance of the original YOLOv5s algorithm in vehicle object detection scenarios.

4.2.3. The Impact of Bounding Box Filtering Algorithm on Model Performance

This study presents the CCNMS algorithm based on the multi-class Non-Maximum Suppression algorithm (NMS) in YOLOv5s, aimed at avoiding situations where a single vehicle target has multiple redundant bounding boxes from different categories. The algorithm involves two thresholds: the same category Intersection over Union (IoU) threshold and the non-same category IoU threshold. The same category IoU threshold is generally set to 0.45, while the non-same category IoU threshold is determined experimentally. Figure 8 shows the mAP50 fluctuation curve for various algorithms under different non-same category IoU thresholds.

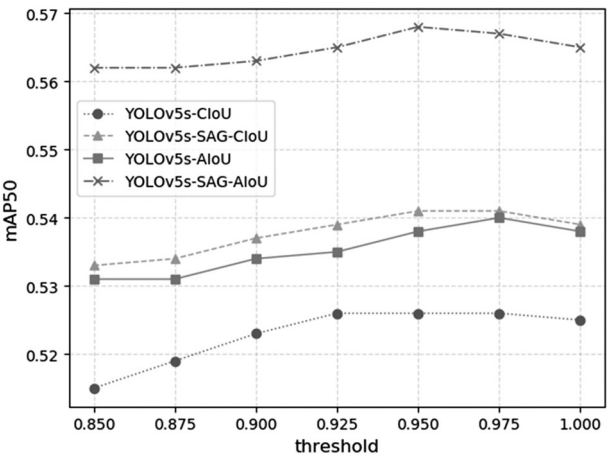

Figure 8. Detection effects of YOLO5s-CIoU (left),mAP50 fluctuation curve for various algorithms under different non-same category IoU thresholds.

According to Figure 8, we can select 0.95 as the non-same category IoU threshold of YOLOv5s-SAG-AIoU and YOLOv5s-SAG-CioU and 0.975 as the non-same category IoU threshold of YOLOv5s-AIoU and YOLOv5s-CIoU.

The results obtained by various algorithms under optimal IoU thresholds are shown in Table 3.

According to Table 3, compared to the NMS detection algorithm, the CCNMS detection algorithm improves the vehicle detection effect for different networks and different loss functions. Overall, CCNMS has a more noticeable improvement effect on algorithms with lower original mAP50 values, which also have more cases of single vehicle multi-category redundant boxes.

Figure 9 shows several scene detection images where YOLOv5s-CIoU-NMS produced redundant boxes for single vehicles of multiple categories and YOLOv5s-SAG-AIoU failed to handle redundant boxes. These redundant boxes have been circled in green, and under the action of the CCNMS detection algorithm, these redundant boxes were successfully eliminated.

While the CCNMS can directly eliminate the redundant boxes that appear under the YOLOv5s-CIoU-NMS algorithm, such as the simple scenario in the first row where applying CCNMS directly to the image on the left can immediately achieve the effect of YOLOv5s-SAG-AIoU-CCNMS on the right, it is not as effective in dense vehicle scenarios, such as those in the second and third rows. Here, YOLOv5s-SAG-AIoU-NMS detects many relatively smaller vehicle boxes, and although this algorithm produces multi-category redundant boxes, they can be directly deleted by CCNMS without causing the wrong deletion of detection boxes of the same category. This is because the CCNMS algorithm uses a higher threshold to filter non-same category detection boxes, making its impact on same-category detection boxes negligible.

Table 3. Comparison of bounding box filtering algorithm effects.

Model	Loss Function	Detection Algorithm	mAP50	mAP75	mAP50-95
YOLOv5s	CIoU	NMS	0.525	0.46	0.379
YOLOv5s	CIoU	CCNMS	0.526	0.461	0.38
YOLOv5s	AIoU	NMS	0.539	0.466	0.389
YOLOv5s	AIoU	CCNMS	0.541	0.467	0.391
YOLOv5s-SAG	CIoU	NMS	0.538	0.457	0.377
YOLOv5s-SAG	CIoU	CCNMS	0.54	0.459	0.378
YOLOv5s-SAG	AIoU	NMS	0.565	0.487	0.398
YOLOv5s-SAG	AIoU	CCNMS	0.568	0.489	0.399

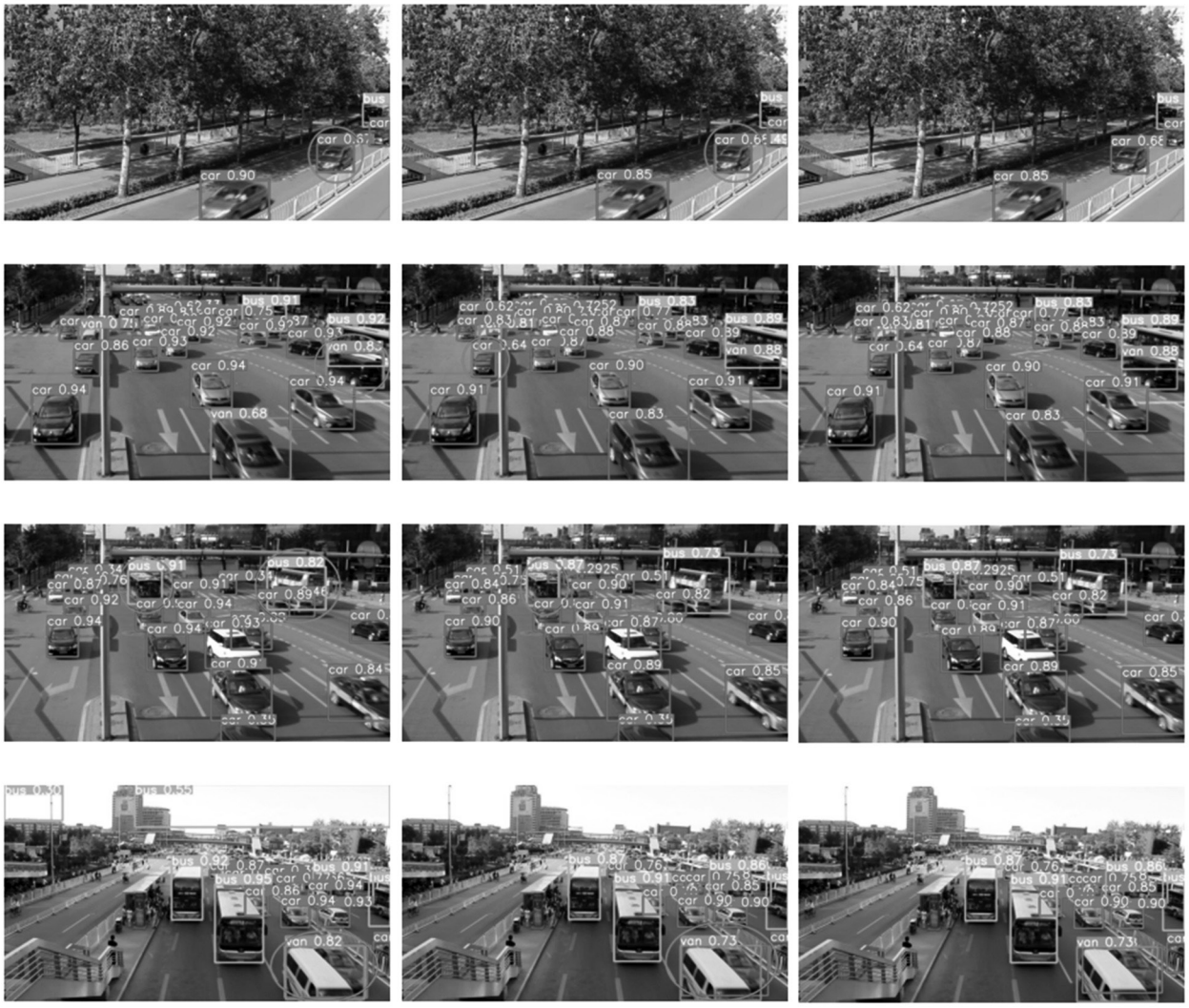

Figure 9. Comparison of detection effects between YOLO5s-CIoU-NMS (left), YOLO5s-SAG-AIoU (middle) and YOLO5s-SAG-AIoU-CCNMS (right).

On the other hand, the function of the CCNMS algorithm is not only to remove redundant boxes. In some special scenarios, it can also make minor adjustments to the detection boxes. For example, in the fourth-row scenario, YOLOv5s-CIoU-NMS algorithm has almost identical detection boxes for the van and the small car in the lower right corner, essentially redundant boxes. YOLOv5s-SAG-AIoU-NMS separates them, but the effect is not obvious. With the application of YOLOv5s-SAG-AIoU-CCNMS algorithm, a more apparent separation is achieved. This indicates that the CCNMS algorithm is not just about deleting multi-category detection boxes from the NMS algorithm. In some special scenarios, it can also adjust detection boxes slightly.

4.2.4. Comparison with Mainstream Object Detection Algorithms

In this section, the proposed YOLOv5s-SAG-AIOU-CCNMS algorithm is compared with the currently mainstream object detection algorithms in the application of vehicle object detection. The experimental dataset is consistent with the dataset in Section 4.1.

The comparison results are shown in Table 4. According to Table 4, the algorithm proposed in this paper performs better than faster R-CNN, SSD, R-FCN, YOLOv4, and YOLOv5s algorithms in vehicle object detection tasks. It slightly underperforms compared to RetinaNet, CenterNet, and YOLOv5x algorithms. The reason is that these algorithms use more network parameters, and the inference time of the network is longer than ours, which can also be seen from the FPS metrics.

For the real-time traffic video vehicle detection task, the network inference time measured by FPS is also an extremely important factor, especially in the application of small embedded devices, so our proposed algorithm can better balance the network detection effect and FPS. Meanwhile, AIoU loss function and CS-NMS algorithm we proposed can be compatible with excellent algorithms of the same type, such as YOLOv5x, but the inference time of the model is relatively long, which is not suitable for the scene of real-time traffic video vehicle target detection in this paper.

Table 4. Comparative Vehicle Object Detection Performance of Mainstream Algorithms.

Model	Backbone	FPS	mAP50	mAP75	mAP50-95
Faster R-CNN [15]	VGG-16	31	0.495	0.398	0.312
SSD [20]	VGG-16	115	0.508	0.449	0.382
RetinaNet [22]	X-101-FPN	35	0.575	0.494	0.402
R-FCN [14]	ResNet-101	93	0.515	0.452	0.38
CenterNet [27]	DLA-34	125	0.586	0.505	0.411
YOLOv4 [15]	CSPDarknet53	171	0.563	0.482	0.394
YOLOv5s [20]	CSPDarknet53	217	0.525	0.46	0.379
YOLOv5x [20]	CSPDarknet53	117	0.622	0.558	0.477
YOLOX-s [17]	CSPDarknet53	193	0.557	0.472	0.385
Ours	CSPDarknet53	208	0.568	0.489	0.399

4.2.5. Comparison of Detection Effects on General Datasets

In order to further validate the applicability of the proposed algorithm for general object detection tasks, the YOLO5s-SAG-AIoU algorithm was retrained based on the coco2017 general object detection dataset. The experiment parameters were basically consistent with those of vehicle object detection. The training results were compared with the results of mainstream object detection, and the evaluation index was the AP value provided by COCO. The results are shown in Table 5, where "-" indicates that the result of this item is not available. According to Table 5, compared with YOLOv5s, the detection effect of our proposed algorithm in general target detection tasks has been improved significantly. This indicates that our proposed method is still effective for general target detection tasks when it is applied to YOLOv5S. Although the overall result is slightly lower than RetinaNet, CenterNet and other algorithms, the algorithm requires fewer parameters, which is more suitable for real-time target detection tasks.

5. Conclusion

In order to satisfy the requirements of both effective detection and real-time processing in the task of vehicle target detection in traffic videos, this paper investigates multi-object vehicle detection in traffic videos. The study involves an analysis and improvement of the currently advanced lightweight object detection model, YOLOv5s.

A vehicle object detection model, based on YOLOv5s-SAG-AioU, is proposed. Additionally, an improved non-maximum suppression bounding box filtering algorithm is introduced to address the issue of overlapping redundant bounding boxes from multiple categories for a single target.

Experimental comparisons on the UA-DETRAC dataset show that, in comparison to the original YOLOv5s algorithm, the algorithm proposed in this paper effectively improves the vehicle object detection while satisfying the real-time constraints. To verify that the proposed algorithm can be applied to general object detection tasks, experiments were conducted on the coco2017 dataset. The results show that, compared to YOLOv5s, the proposed algorithm can also improve the detection effectiveness in general target detection tasks.

Table 5. Comparative General Object Detection Performance of Mainstream Algorithms.

Model	Backbone	AP	AP_50	AP_75	AP_S	AP_M	AP_L
Faster R-CNN [15]	VGG-16	0.219	0.427	-	-	-	-
SSD [20]	VGG-16	0.288	0.485	0.303	0.109	0.318	0.435
RetinaNet [22]	X-101-FPN	0.390	0.594	0.417	0.226	0.434	0.509
R-FCN [14]	ResNet-101	0.299	0.519	-	0.108	0.328	0.450
CenterNet [27]	DLA-34	0.421	0.611	0.459	0.241	0.455	0.528
YOLOv4 [15]	CSPDarknet53	0.435	0.657	0.473	0.267	0.467	0.533
YOLOv5s [20]	CSPDarknet53	0.355	0.55	-	-	-	-
YOLOv5x [20]	CSPDarknet53	0.472	0.666	-	-	-	-
Ours	CSPDarknet53	0.375	0.581	0.391	0.212	0.429	0.531

Moreover, the AIoU loss and CS-NMS algorithm prove to be universal for the target detection network framework. In the future, we will study the adaptability of these algorithms when applied to an updated and lightweight network framework to further improve the target detection effectiveness while meeting real-time requirements.

Acknowledgement

This work was supported by the Universities Joint Key Laboratory of photoelectric detection Science and Technology in Anhui Province (Grant No.: 2020GDTC01); Anhui Province University Collaborative Innovation Project (Grant No.: GXXT-2021-091, GXXT-2022-045); Horizontal project of Hefei Normal University (Grant No.: HXXM2022238); 2023 Anhui Province Scientific Research Compilation Plan Project (Grant No.: 2023AH051306).

References

[1] Q. H. Zhao *et al.*, "Review of Single-stage Vehicle Detection Algorithms Based on Deep Learning", *Computer Applications*, vol. 40, no. z2, pp. 30–36, 2020.

[2] J. R. Uijlings *et al.*, "Selective Search for Object Recognition", *International Journal of Computer Vision*, vol. 104, pp. 154–171, 2013. https://doi.org/10.1007/s11263-013-0620-5

[3] Y. H. Long *et al.*, "Prediction of Vegetation Change by Discrete Wavelet Decomposition Based on Remote Sensing Time Series Images", *Traitement du Signal*, vol. 40, no. 1, pp. 123–132, 2023. https://doi.org/10.18280/ts.400111

[4] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA*, 2001, pp. 511–518. https://doi.org/10.1109/CVPR.2001.990517

[5] L. H. Zhong *et al.*, "Integration Between Cascade Region-based Convolutional Neural Network and Bi-directional Feature Pyramid Network for Live Object Tracking and Detection", *Traitement du Signal*, vol. 38, no. 4, pp. 1253–1257, 2021. https://doi.org/10.18280/ts.380437

[6] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA*, 2005, pp. 886–893. https://doi.org/10.1109/CVPR.2005.177

[7] B. More and S. Bhosale, "A Comprehensive Survey on Object Detection using Deep Learning", *Revue d'Intelligence Artificielle*, vol. 37, no. 2, pp. 407–414, 2023. https://doi.org/10.18280/ria.370217

[8] I. H. Kartowisastro and J. Latupapua, "A Comparison of Adaptive Moment Estimation (Adam) and RMSProp Optimisation Techniques for Wildlife Animal Classification using Convolutional Neural Networks", *Revue d'Intelligence Artificielle*, vol. 37, no. 4, pp. 1023–1030, 2023. https://doi.org/10.18280/ria.370424

[9] E. N. Yildiz *et al.*, "Diagnosis of Chronic Kidney Disease Based on CNN and LSTM", *Acadlore Transactions on AI and Machine Learning*, vol. 2, no. 2, pp. 66–74, 2023. https://doi.org/10.56578/ataiml020202

[10] N. Sharma *et al.*, "Utilizing Mask R-CNN for Automated Evaluation of Diabetic Foot Ulcer Healing Trajectories: A Novel Approach", *Traitement du Signal*, vol. 40, no. 4, pp. 1601–1610, 2023. https://doi.org/10.18280/ts.400428

[11] W. Liu *et al.*, "SSD: Single Shot Multibox Detector", in *Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands*, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[12] B. G. Amira *et al.*, "Driver Drowsiness Detection and Tracking Based on Yolo with Haar Cascades and ERNN", *International Journal of Safety and Security Engineering*, vol. 11, no. 1, pp. 35–42, 2021. https://doi.org/10.18280/ijsse.110104

[13] J. C. Xie *et al.*, "Mask Wearing Detection Based on YOLOv5 Target Detection Algorithm under COVID-19", *Acadlore Transactions on AI and Machine Learning*, vol. 1, no. 1, pp. 40–51, 2022. https://doi.org/10.56578/ataiml010106

[14] J. Redmon and A. Farhadi, "Yolov3: An Incremental Improvement", arXiv preprint arXiv:1804.02767, 2018. https://doi.org/10.48550/arXiv.1804.02767

[15] A. Bochkovskiy *et al.*, "Yolov4: Optimal Speed and Accuracy of Object Detection", arXiv preprint arXiv:2004.10934, 2020. https://doi.org/10.48550/arXiv.2004.10934

[16] T. Y. Lin *et al.*, "Focal Loss for Dense Object Detection", in *Proceedings of the IEEE International*

Conference on Computer Vision, Venice, Italy, 2017, pp. 2980–2988.
https://doi.org/10.1109/ICCV.2017.324

[17] Z. Ge *et al.*, "YOLOX: Exceeding YOLO Series in 2021". arXiv preprint arXiv:2107.08430. 2021.
https://doi.org/10.48550/arXiv.2107.08430

[18] H. Rezatofighi *et al.*, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA*, 2019, pp. 658–666.
https://doi.org/10.1109/CVPR.2019.00075

[19] Z. Zheng *et al.*, "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation", *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8574–8586, 2021.
https://doi.org/10.1109/TCYB.2021.3095305

[20] G. Jocher *et al.*, "YOLOv5-Master [EB/OL]", 2022.
https://github.com/ultralytics/yolov5

[21] S. Chen *et al.*, "Dual-bottleneck Feature Pyramid Network for Multiscale Object Detection", *Journal of Electronic Imaging*, vol. 31, no. 1, pp. 013009–013009, 2022.
https://doi.org/10.1117/1.JEI.31.1.013009

[22] K. He *et al.*, "Deep Residual Learning for Image Recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA*, 2016, pp. 770–778.
https://doi.org/10.1109/CVPR.2016.90

[23] T. Y. Lin *et al.*, "Feature Pyramid Networks for Object Detection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA*, 2017, pp. 2117–2125.
https://doi.org/10.1109/CVPR.2017.106

[24] H. Hu *et al.*, "Relation Networks for Object Detection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, 2018, pp. 3588–3597.
https://doi.org/10.1109/cvpr.2018.00378

[25] F. Liu *et al.*, "Depth-wise Separable Convolution Attention Module for Garbage Image Classification", *Sustainability*, vol. 14, no. 5, p. 3099, 2022.
https://doi.org/10.3390/su14053099

[26] L. Wen *et al.*, "UA-DETRAC: A New Benchmark and Protocol for Multi-object Detection and Tracking", *Computer Vision and Image Understanding*, vol. 193, pp. 102907, 2020.
https://doi.org/10.1016/j.cviu.2020.102907

[27] X. Zhou *et al.*, "Objects as Points", arXiv preprint arXiv:1904.07850, 2019.
https://doi.org/10.48550/arXiv.1904.07850

Received: September 2023
Revised: October 2023
Accepted: October 2023

Contact addresses:
Hao Wu
School of Computer Science and Technology
Hefei Normal University
Hefei
China
School of Information and Control Engineering
China University of Mining and Technology
Xuzhou
China
Universities Joint Key Laboratory of Photoelectric Detection
Science and Technology in Anhui Province
Hefei
China
Key Laboratory of Philosophy and Social Science of Anhui
Province on Adolescent Mental Health and
Crisis Intelligence Intervention
Hefei
China
e-mail: wuhao@hfnu.edu.com

Wei Wu*
School of Economics and Trade
Anhui Business and Technology College
Hefei
China
e-mail: 2019010648@ahbvc.edu.cn
*Corresponding author

Xiaoyan Sun
School of Information and Control Engineering
China University of Mining and Technology
Xuzhou
China
e-mail: 3730@cumt.edu.cn

Jin Zhong
School of Computer Science and Technology
Hefei Normal University
Hefei
China
e-mail: zhongjin@hfnu.edu.cn

Fengyun Cao
School of Computer Science and Technology
Hefei Normal University
Hefei
China
e-mail: caofengyun@hfnu.edu.cn

Hao Wu is an associate professor at Anhui Normal University in Hefei, China. He graduated from Hefei University of Technology in 2010 with an MSc degree. He is currently pursuing a PhD in Control Theory and Control Engineering at China University of Mining and Technology in Xuzhou, China. He has presided over several projects in Anhui Province and participated in two research projects funded by the National Natural Science Foundation of China. His research interests include computer vision research related to intelligent transportation, image processing, and deep learning.

Wei Wu is a lecturer at Anhui Business College in Hefei, China. He graduated from Xi'an Jiaotong University in 2018 with an MSc degree. His research interests include multimedia and signal processing, pattern recognition, and computer vision.

Xiaoyan Sun is a professor and doctoral advisor at the School of Information and Control Engineering, China University of Mining and Technology. She obtained a PhD degree in 2009 from the China University of Mining and Technology. Her research interests include evolutionary optimization algorithms and applications, design of multi-objective evolutionary algorithms, and machine learning. She is an IEEE Senior Member.

Jin Zhong is a professor at Hefei Normal University. He obtained a PhD degree in 2008 from the Hefei University of Technology, China. His research interests include artificial intelligence, deep learning, and smart healthcare.

Fengyun Cao received a BSc degree in mechanical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2004 and an MSc degree in mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2006. He is currently pursuing his PhD degree in mechanical engineering at Texas A&M University, College Station, TX, USA. His research interest includes the development of surface processing and biological/medical treatment techniques using nonthermal atmospheric pressure plasmas, fundamental study of plasma sources, and fabrication of micro- or nanostructured surfaces.