IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 April 2024, accepted 24 April 2024, date of publication 29 April 2024, date of current version 16 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3394693

RESEARCH ARTICLE

TTD-YOLO: A Real-Time Traffic Target Detection Algorithm Based on YOLOV5

WENJUN XIA[1], PEIQING LI[1,3,4], HEYU HUANG[2], QIPENG LI[1,3], TAIPING YANG[1,3], AND ZHUORAN LI[5]

[1]School of Mechanical and Energy Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
[2]China Industrial Research Institute, Zhejiang University of Finance & Economics, Hangzhou 310018, China
[3]Zhejiang Southwest Research Institute, Zhejiang University of Science and Technology, Hangzhou 310058, China
[4]School of Mechanical Engineering, Zhejiang University, Hangzhou 310058, China
[5]Faculty of Information Technology, City University Malaysia, Petaling Jaya 46100, Malaysia

Corresponding authors: Peiqing Li (lpqing@163.com) and Heyu Huang (heyu_hg@163.com)

This work was supported in part by Zhejiang Lingyan Project 2024C04037, in part by the Natural Science Foundation of Zhejiang Province under Grant LGG20F020008, and in part by the Key (Team) Project of Zhejiang University of Science and Technology under Grant 2021JLZD004.

ABSTRACT To solve the problems of limited computing power resources, low accuracy of small target detection, high miss rate, and poor real-time detection of mobile vehicle platforms in the automatic driving environment, The present study introduced a one-stage target detection algorithm TTD-YOLO (Traffic Target Detection YOLO) that improved YOLOV5-S, which is enhanced in four aspects: Enhanced the network's multi-scale feature extraction performance through the utilization of the improved M-ELAN architecture; added 3D attention mechanism SimAM to the network structure to enable the network to learn important feature information and enhance the efficiency of detecting accuracy; the parameter ratio of backbone and neck is adjusted to close to 1:1 by adjusting the number of output channels and stacking times of CSPLayer modules in the backbone and neck, while maintaining the model complexity, experiments show that improving the neck's parameter ratio helps enhance the efficiency of detecting accuracy without changing the network structure; used EIoU loss instead of the bounding box loss function accelerates network convergence and improved detection accuracy. Under the condition of avoiding significantly changing the network structure, our TTD-YOLO outperforms the baseline model and other mainstream object detection algorithms such as Faster RCNN, SSD, and YOLOX-S on the autopilot dataset SODA10M with fewer parameters, higher detection accuracy, and faster inference speed. Compared to the baseline model, the model parameters decreased by 8.6%, average precision(mAP@.5:.95)increased by 2.5%, and the inference speed under the same experimental platform increased by 4.8%.

INDEX TERMS YOLO, traffic target detection, real-time detection.

I. INTRODUCTION

The study of autonomous driving has consistently been a significant area of focus in the field of artificial intelligence and holds promising potential for advancement. Automated driving technology mainly includes an environment awareness system, positioning and navigation system, decision planning system, and control execution system.

An environmental awareness system provides necessary scene environment information for decision planning and control of autonomous vehicles and is critical to the technical support for automatic driving [1]. Target detection is an integral part of the environmental perception of an autonomous vehicle. As an essential part of perception technology, detecting traffic targets in the road environment can collect the surrounding road environment information and participate in the decision-making activities in the panoramic perception system of autonomous vehicles [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

However, the road environment in the automatic driving scene is often very complex. The detection tasks, including traffic targets and traffic signs, may be interfered with by many factors, especially in urban areas where there are many vehicles and pedestrians when the cars share the road with other traffic participants, safety accidents are likely to occur [3]. To avoid such mishaps, collecting the location information of other traffic participants or obstacles is necessary. In general target detection tasks, Large objects in the image occupy many pixels. They are often easier to detect, while small objects are usually easily ignored because they occupy fewer pixels and carry limited information [4]. Other factors, such as lighting conditions, photography angle, object deformation, motion blur, distance, etc., will affect the detection accuracy [5]. At the same time, as an essential part of the automatic driving perception system, we should enhance the precision of detecting and ensure real-time detection so that the collected environmental data can be handed over to the decision-making system for processing the first time planning for the next step. Some unexpected situations in the driving process should be promptly addressed to ensure traffic participants' safety. At present, the problem of striking a balance between the accuracy and speed of detecting targets is a significant issue in the field of target detection. It is also a severe difficulty in the critical technologies of automatic driving perception systems.

To address the aforementioned issues, we employ a deep learning-based approach, thoroughly evaluate the precision of detection and the ability to detect in real-time, and use the single-stage target detection algorithm YOLOV5-S as the research and development benchmark to propose a lightweight traffic target detection algorithm TTD-YOLO exhibiting exceptional detection precision and rapid detection velocity.

The subsequent sections of this manuscript are outlined as follows. The second chapter primarily presents an overview of the existing literature on automatic driving object detection, explicitly focusing on deep learning techniques. The third section describes the specific implementation methods, working principles, and innovations. The fourth chapter is about the experiment and result analysis. Finally, the fifth chapter puts forward the conclusion.

II. RELATED WORKS

Within this particular section, we mainly introduce some work in relation to traffic target detection.

Traditional target detection methods usually include three steps:

(1) Preprocess the picture data that was entered. For instance, image correction, camera calibration, cropping and scaling of images.

(2) Extract candidate regions that may contain detection targets. Usually, sliding windows with different proportions are used to stroke the whole picture to extract the area of interest.

(3) Use the classifier to classify the target. Process the filtered candidate regions and classify the targets in the candidate regions.

Although this method can accurately identify the target objects, the time complexity of region selection technology based on sliding windows is high, and it will generate several redundant windows. And the features designed by hand could be more robust to diversity changes.

In recent times, the approach relies on deep learning techniques has made a significant breakthrough in target detection. Training the depth convolution neural network through labeled datasets can enable the network to quickly learn the required feature information to detect the target object end-to-end. Currently, the techniques employed for target detection can be categorized into two main types: two-stage and single-stage. Two-stage algorithms are represented by the R-CNN [5] series, which first generate regions of interest in the image using region recommendation methods and then classify these regions of interest. After classification, use post-processing methods to eliminate redundant borders. This kind of algorithm is characterized by high detection accuracy. Still, the model is more intricate and the speed of inference is reduced, which is unsuitable for completing the real-time target detection task in high-speed scenes and requires high computing power on the deployment platform. Hence, deploying to a mobile platform with limited computing power is complicated. The algorithm in a single-stage, as denoted by SSD [6] and YOLO [7] series, the user can input an image and generate a direct output that includes the category name and score of the box, as well as the objects contained within the box. Only one network can complete the detection frame and classification tasks. It has high real-time, fast inference speed, small model parameters, and low computational complexity. However, the precision of the suggested approach is marginally inferior to that of the two-stage approach. At present, the research of single-stage algorithms is relatively hot. With the deepening of research, some single-stage algorithms can surpass two-stage algorithms in accuracy while ensuring detection speed and parameters.

At the same time, some excellent network structures and improvement strategies have been put forward. Backbone networks such as MobileNet [8] and ShuffleNet [9] adopt methods such as deep separable convolution [10] and channel shuffle, which not only have excellent feature extraction performance but also have fewer parameters compared with some traditional networks and are easier to deploy to mobile devices; The Feature Pyramid Network (FPN) [11] structure combines advanced semantic characteristics with detailed geographical characteristics, strengthens the information exchange among feature layers of various scales, effectively enhances the detection accuracy of small targets. However, these methods are relatively limited in optimizing the target detection model, and the task requirements in complex traffic scenarios cannot be met.

Yang et al. [2] introduced a novel object detection algorithm to tackle the challenges associated with low detection accuracy and high miss rate of minor traffic signs. And signals in the road environment. This technique incorporates a YOLOV3-based multi-scale attention mechanism module. This can enhance the network's focus on crucial feature information and strengthen the network's capacity to represent features. The inclusion of low-level prediction heads enhances the detection accuracy of small targets. Although adding additional layers of detection is beneficial to small targets, it also leads to an escalation in computational expenses and impacts the model's inference speed. Dewi et al. [12], [13] combined SPP (Spatial Pyramid Pooling) with YOLOV3 and YOLOV4 to enhance the model's capability to extract features from a global perspective and achieved better results in the traffic sign detection task. Cai et al. [14] Suggested a YOLOV4-based technique for real-time traffic target recognition and performed model pruning, effectively enhancing the inference speed. At the same time, Wang et al. [15] suggested a streamlined traffic target detection method utilizing enhanced YOLOv4 Tiny to address the issue of a high rate of undetected dense targets in intricate traffic situations. The algorithm uses a K-means clustering algorithm to produce an a priori anchor box that is appropriate for the training datasets. It proposes a feature map optimization strategy, enriching the network feature level through the low-level feature map's low-level information and enhancing the precision of detecting diminutive targets. The NMS algorithm in the post-processing stage is improved. Based on soft NMS [16], the prediction box's confidence score, which exceeds the predetermined threshold, is no longer assigned a value of 0. This method is very effective for improving the recall rate under dense targets. Wenjie et al. [17]captured fine-grained spatial features and improved detection by introducing a deformable convolutional coordinate attention mechanism in YOLOV8. Cao et al. [18] proposed the MCS-YOLO algorithm based on YOLOV5-S, added the swin transformer structure and coordinate attention module to the backbone network to improve the feature extraction performance of the model, and a multiscale structure was also designed for detecting small targets. The algorithm enhanced the precision to detect small targets in the traffic road scenario.

III. TRAFFIC TARGET DETECTION-YOLO

This section first briefly introduces the principle of YOLO series algorithms and the overall network structure of TTD-YOLO (Traffic Target Detection YOLO), then introduces the improvement module in detail.

A. OVERALL INTRODUCTION TO NETWORK ARCHITECTURE

YOLO series algorithms are representative of single-stage real-time target detectors. Their basic idea is to redefine target detection as a single regression problem. After end-to-end training, the image input network will directly output predicted bounding box coordinates and category probabilities [19]. Compared with the general two-stage algorithm, it does not need to generate the region of interest. Therefore, compared with the two-stage algorithm, it has a more significant improvement in detection speed, excellent detection accuracy, and better robustness.

Currently, the most popular YOLO series algorithm, YOLOV5, uses CSPDarknet as the backbone network. The network consists of a multilayer residual module CSPLayer with strong feature representation capability. Meanwhile, the SPPF (Spatial Pyramid Pooling Fast) structure is added after the backbone network. The SPPF structure enhanced the multiscale features by serially passing the input feature layer through multiple maximum pooling layers with different sizes. It performs the feature fusion, thus solving the target multiscale problem to a certain extent. Compared with parallel connections, serial connections have less memory and faster computation speeds. After the input image passing through the backbone network, three feature layers of different scales are generated to detect objects of different scales. These feature layers are bi-directionally semantically fused in the FPN+PAN structure of the neck to promote fusion and communication amidst the low-level and high-level feature information and enhance the model's multi-scale feature representation capability. Finally, a coupled detector simultaneously outputs the network's predicted bounding box coordinate and category probability information.

However, when this series of algorithms are used in some specific scenes, numerous issues persist (such as target occlusion, low accuracy of small target detection, high rate of missed detection, intricate models that pose challenges in implementation, and slow inference speed). Therefore, based on YOLOV5-S, we propose a traffic target detection algorithm TTD-YOLO (Traffic Target Detection YOLO) in the automatic driving scene. This model reduces the model's parameters and achieves the overall improvement of detection accuracy and speed without the initial model undergoing a substantial alteration in its network structure. Fig 1 illustrates the network architecture of TTD-YOLO. Its main innovation points are as follows:

(1) The enhanced M-ELAN module substitutes the partial convolution in the initial network nick. The M-ELAN module's parameters are reduced by 81.4% compared to the ELAN module, and the module utilizes a multi-size convolution kernel to enhance its feature extraction capability. The experiment demonstrates a notable reduction in model parameters, a decrease in computing complexity, and an enhancement in detection speed while maintaining accuracy.

(2) SimAM is added in the feature fusion stage of the network. SimAM can calculate the channel and space weights of the output feature layer without adding any parameters, providing 3D weights for the network.

(3) Modify the ratio of parameters. Under the condition of ensuring the original parameter quantity, the proportion of parameters in the neck can be improved by adjusting the output channels and stacking times of the CSPLayer

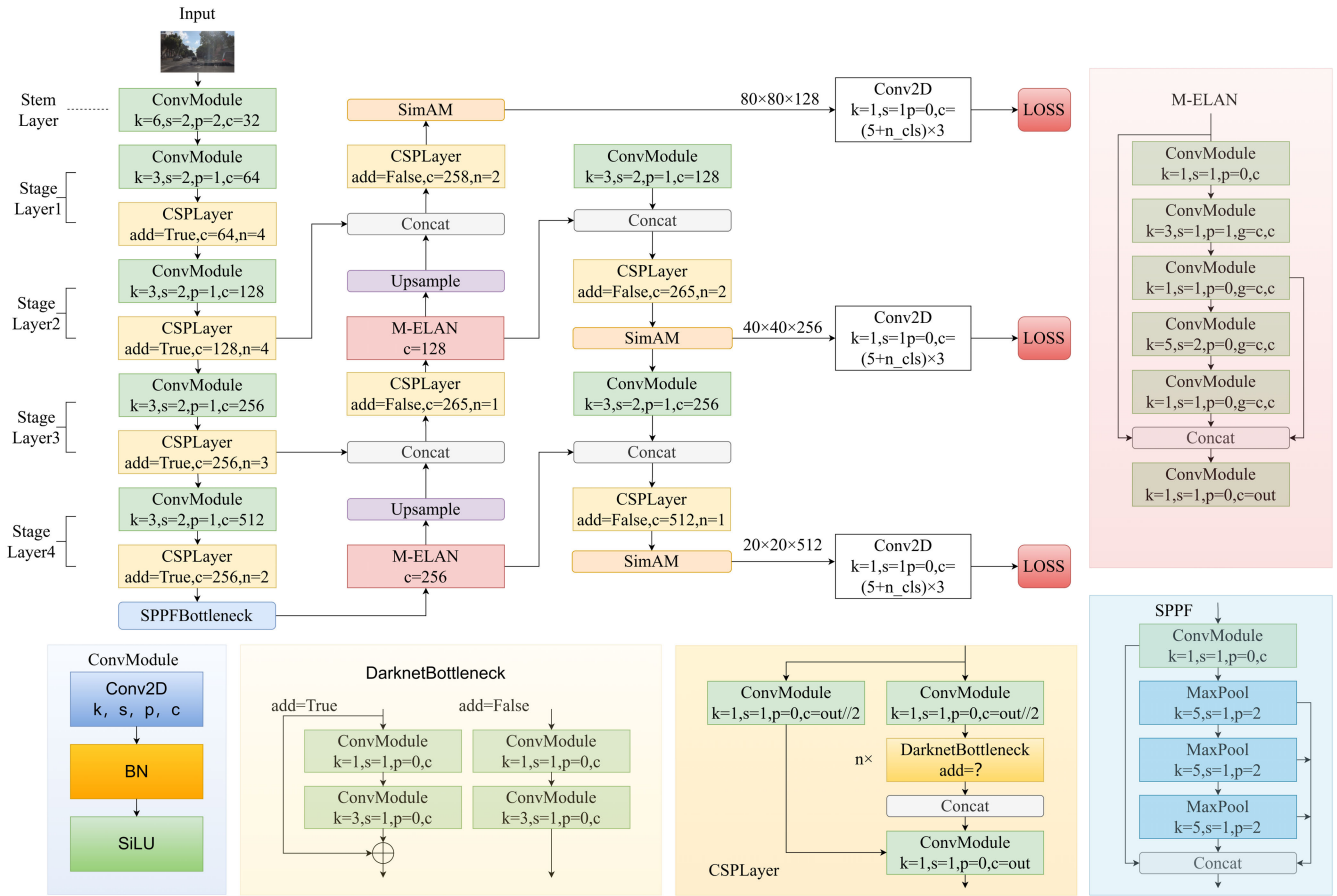


FIGURE 1. Proposed TTD-YOLO network.

in the backbone and neck. Experiments show that under the condition that the total parameters remain unchanged, the enhancement of detection accuracy can be efficiently achieved by increasing the parameter proportion of the neck part within a specific range.

(4) Substitute the CIoU loss employed in the first model with the EIoU loss. In the context of bounding box regression, the EIoU loss quantifies the disparity among the three geometric components: overlapping area, center point distance, and aspect ratio. Simultaneously, the Focal loss is introduced as a solution to address the issue of sample imbalance. It can potentially enhance the detection accuracy and recall rate of the model while also expediting the convergence process.

B. IMPROVED M-ELAN MODULE

The ELAN (Efficient Layer Aggregation Networks) module proposed in YOLOV7 [20] enables more efficient learning and convergence of deeper networks by regulating the shortest and longest paths of the gradient. This cross-layer feature intermingling combining shallow abstract information and deep fine-grained information allows the model to ensure efficient recognition capabilities while reducing the over-reliance on the depth and width levels of the network. It mainly comprises three convolution layers of size 1 × 1, four convolution layers of size 3 × 3, and one concat layer (all convolution operations will perform the corresponding padding operation without altering the width and height dimensions of the feature layer). There are four branches in total, as shown in Fig 2 (a). The top two convolution layers of size 1 × 1 are used to adjust the quantity of channels, and four convolution layers of size 3 × 3 are used to extract features. The concat layer splice feature layers are generated from four branches according to channel dimensions. Finally, use one convolution layer of size 1 × 1 to integrate feature information and adjust the quantity of channels to the specified output. Besides the three convolution layers of size 1 × 1 used to adjust the quantity of channels, the quantity of input channels and output channels of each branch are consistent. This design minimizes memory loss, assists in improving the inference speed, and reduces the hardware cost, as demonstrated in ShufflenetV2 [21]. At the same time, this multi-branch

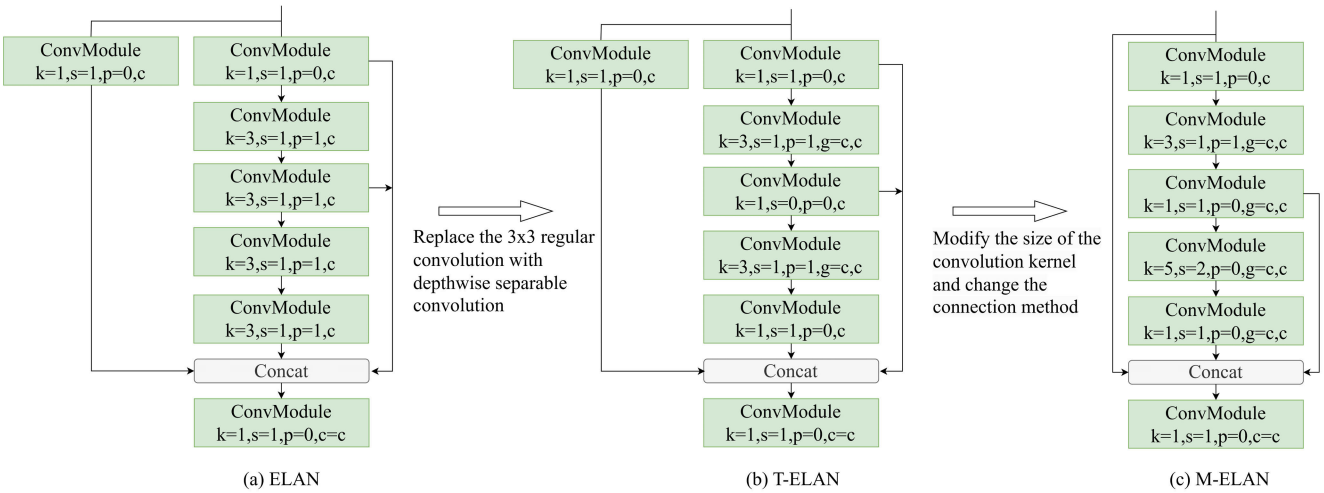


FIGURE 2. ELAN module and its improved version.

structure enhances the integration of semantic data, hence promoting the improvement of detection accuracy. However, it also increases network parameters, increases computational complexity, and significantly impacts the model's detection speed. It is unfriendly to deploy mobile terminals, and applying them to some scenarios requiring high real-time performance is challenging. Therefore, we first carried out a lightweight transformation to the last two branches. Convolution layers of size 3×3 replaced by depthwise separable convolution layers of size 3×3, and then the second convolution layer of size 3×3 on the branch is replaced by a convolution layer of size 1×1 is used to integrate the information on the channel. This can significantly reduce the parameters of the module. Here, we name the lightweight ELAN structure T-ELAN (Tiny ELAN), as seen in Fig 2 (b).

Through experimental comparison, the parameters are reduced by about 77.0%. However, the computational complexity is still relatively large due to many branches, even if the parameters are reduced. The computational cost for mobile devices is high, so we further reduced the module branches. Firstly, we removed the convolution operation on the leftmost branch, which not only reduces the amount of computation but also preserves the initial information from the input feature layer. To avoid the module repeatedly extracting redundant feature information and improve the speed of inference, we removed the first branch on the right. The depthwise separable convolution also reduced the model's parameters. In addition, convolution kernels with different sizes are used to obtain multi-scale semantic information to compensate for the loss of accuracy caused by branch reduction. We named it M-ELAN (Multi-scale ELAN), and the precise configuration is depicted in Fig 2 (c). M-ELAN can significantly minimize the parameters and computational complexity, enhance the inference speed, and reduce 81.4% compared with the original ELAN module parameters, meeting the requirements of mobile terminal deployment.

C. SIMAM

The current attention mechanisms are limited to calculating the weight either over the channel or spatial dimension. This approach disregards the association between the channel and spatial dimension and lacks flexibility. Examples of such mechanisms include SE [22], CBAM [23], CA [24], and ECA [25]. In comparison to the current channel and spatial attention mechanisms, SimAM [26] has designed an energy function that can simultaneously calculate the weights of channel and spatial dimensions without adding additional parameters, providing 3D attention weights for feature layers, and improving the network's ability to extract critical information while avoiding excessive adjustment of network structure. In this paper, we enhanced the information extraction capability of the network by adding a SimAM mechanism after three feature layers of neck structure output. By quantifying the similarity between feature maps and using this to adjust the weight allocation of each feature map, SimAM prompts the model to focus more sharply on regions within the image that share similar features, which in turn improves the recognition of and attention to the target object and surrounding associated structures. The Fig 3 compares the existing attention mechanism and the working principle of the SimAM.

The energy function associated with each neuron in SimAM is determined by quantifying the linear divergence among the target neuron and other neurons. Specifically as shown in Formula 1:

$$e_t(w_t, b_t, y_t, x_i) = \left(y_t - \widehat{t}\right)^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}\left(y_o - \widehat{x_i}\right)^2, \quad (1)$$

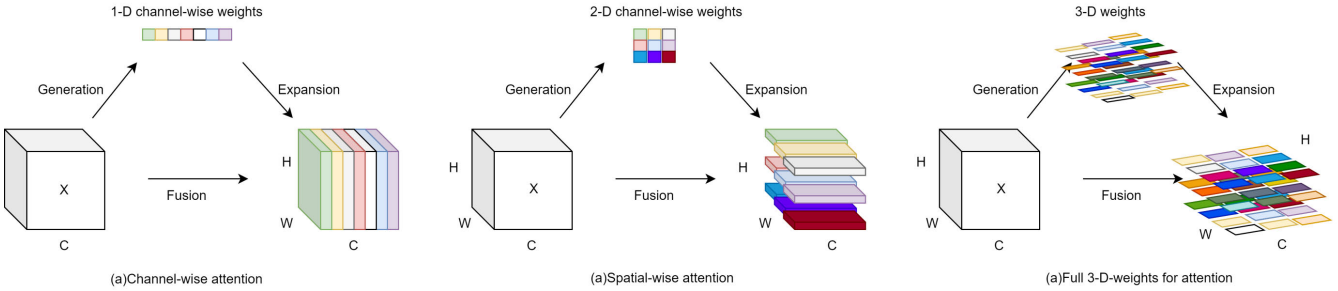


FIGURE 3. Comparisons of different attention steps.

where t and x_i represent the focal neuron and additional neurons in the single input feature's channel;w_t is the weight, b_t is the offset; $\widehat{x_i}$ and $\widehat{t}$ are linear transformations of x_i and t; y_t and y_o are two different variables used to regulate the final output. Output is minimized when $\widehat{t}$ is equal to y_o. M is the aggregate count of all neurons present on the channel. Minimizing the aforementioned formula is equivalent to training the linear independence among a selected neuron and other neurons in the identical channel. The ultimate energy function is derived by utilizing binary labels and incorporating standard terms. The specific definition is shown in Formula 2:

$$e_t(w_t, b_t, y_t, x_i) = \frac{1}{M-1}\sum_{i=1}^{M-1}(-1-(w_t x_i + b_t))^2 + (1-(w_t t + b_t))^2 + \lambda w_t^2, \quad (2)$$

after deriving the neuron energy function, the author uses the scaling operator instead of adding it to obtain better thinning features. As shown in Formula 3:

$$\widetilde{X} = sigmod(\frac{1}{E}) \odot X, \quad (3)$$

where E represents grouping all neurons in channel and spatial dimensions and finally using the sigmoid function to ensure that the value is between 0 and 1 and multiplying with the input result to obtain the final output result.

D. PARAMETER BALANCE STRATEGY BETWEEN BACKBONE AND NECK

Some previous improvements of the neck, such as NASFPN [27], BiFPN [28], and ASFF [29], often focus on how to modify the feature fusion method and strengthen the feature fusion through the multi-branch and multi-connection structure. Although it has some effect on improving the detection accuracy, introducing too many connections will increase the delay of the detector and the memory overhead, which is unsuitable for lightweight algorithms. In RTMdet [30], the strategy of not introducing additional connections but changing the ratio of parameter quantities between backbones and necks is selected. By upwardly adjusting the expansion ratio of the neck base module, a portion of the parameter count and computational load in the backbone network is shifted to the neck structure, ensuring that the parameter ratio of the two is close to 1:1 and avoiding the excessive tilting of resources among different parts of the model, thus realizing an ideal balanced deployment of computational accuracy. Besides, experiments show that when necks account for a higher proportion of parameter quantities in the whole model, the delay is lower, and the impact on accuracy is small. Using this idea for reference, we adjusted the stacking times and the number of channels of the CSPLayer modules in the YOLOV5-S backbone and neck structures so that the parameter ratio of the backbone and neck is close to 1:1. According to the experimental findings, the accuracy, average precision, and inference speed of this algorithm have undergone substantial enhancements in comparison to the initial network architecture.

E. IMPROVED LOSS FUNCTION

The loss function is a technique utilized to quantify the predictive accuracy of a model. During the training process, the model's optimization is guided by identifying the discrepancy between the predicted and actual values. The selection of an appropriate loss function has the potential to expedite the convergence of the model and enhance its quality.

The BCEWithLogits loss function is utilized in YOLOV5 for both object loss and classification loss, and CIoU is used as bounding box loss. The bounding box loss is primarily used to locate the prediction target in the image [31]. The traditional IoU loss [32] only works when the bounding boxes intersect. In the nonoverlapping scenario, it does not generate any dynamic gradient. It is not feasible to ascertain the most suitable intersection method when the prediction box and the real box exhibit identical intersection and union ratios. The CIoU loss [33] considers three important geometric factors: overlapping area, center point distance, and aspect ratio. The gradient needs to be updated when the borders are not coincidental, thorough attention is paid to the data concerning the distance between the center points of the bounding box and the scale information related to the width-height ratio of the bounding box. Prediction box regression exhibits superior speed and accuracy. Formula 4 provides the exact definition

of the cost term in the loss.

$$\Re_{CIoU} = \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + av, \tag{4}$$

where a denotes a favorable trade parameter; v employed for quantifying the uniformity of aspect ratio.The definitions of a and v are shown in Formula 5 and Formula 6:

$$\alpha = \frac{v}{(1 - IoU) + v}, \tag{5}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2, \tag{6}$$

the final loss function is is shown in Formula 7:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v, \tag{7}$$

where IoU denotes the extent to which the forecast box and the actual box intersect and blend, b and b^{gt} indicate the central points of the forecast box and the actual box, $\rho(\bullet)$ is the Euclidean distance, and c represents the lateral length of the minimal bounding box that contains both boxes.

When evaluating the bounding box regression, the CIoU loss incorporates the overlapping area, distance between center points, and aspect ratio, the observed discrepancy solely pertains to the difference in aspect ratio rather than the difference between the projected width and height and the actual width and height. Occasionally, it impedes the efficient optimization of the model. EIoU loss [34] took apart the aspect ratio based on CIoU loss, clearly measured the difference of three geometric factors, and introduced Fcoal loss [35] in order to address the issue of an imbalance between challenging and straightforward samples. Through experiments, It has been observed that employing the EIoU loss can expedite the convergence rate during the training process. Additionally, this approach enhances the detection accuracy and recall rate of the model. The specific function is shown in Formula 8:

$$\begin{aligned} L_{EIoU} &= L_{IoU} + L_{dis} + L_{asp} \\ &= 1 - IoU + \frac{\left(b, b^{gt}\right)}{(w^c)^2 + (h^c)^2} + \frac{\rho^2\left(w, w^{gt}\right)}{(w^c)^2} \\ &\quad + \frac{\rho^2\left(h, h^{gt}\right)}{(h^c)^2}, \end{aligned} \tag{8}$$

the w^c and h^c, as well as w^{gt} and h^{gt} depict the dimensions of the tiniest box that encompasses both the forecast box and the actual box. The L_{IoU} is IoU loss; the L_{dis} is distance loss; the L_{asp} is phase loss.

IV. EXPERIMENTS

This section introduces the autopilot dataset SODA10M [36], experimental settings, and evaluation indicators. Following this, the group tests were carried out in order to validate the efficacy of certain approaches that had been previously enhanced, and all the improvements were applied to the original model. Then, ablation experiments were carried out to test our proposed new model. Finally, the suggested framework is being compared against the current mainstream works to assess the feasibility.

TABLE 1. Experimental platform.

Parameters	Configuration
Operating System	Ubuntu 20.04.3
CPU	Intel(R)Core(TM)i9- 12900k,CPU/3.20GHz
GPU	NVIDIA GeForce RTX 3090Ti/24GB
Experimental Environmental	CUDA11.3, PyTorch1.11,Python3.8
Visual Studio Framework	Visual Studio 2022

A. DATASET

The model was trained and validated using the publicly available dataset SODA10M. SODA10M is a comprehensive dataset for automated driving that focuses on 2D self/semi-supervised object detection, including 10 million unmarked automatic driving environment images and 10000 images marked with six representative traffic targets. The photographs encompass a diverse range of weather conditions, temporal periods, and geographical locations across 32 distinct urban centers. Experiments show that the dataset has an excellent performance in training and fine-tuning models in the field of automatic driving. We use 10000 tagged images as the dataset and divide them into the training set and verification group according to the proportion of 9:1. The model's training procedure employs a mosaic data enhancement method to augment the quantity and variety of training samples, hence enhancing the model's resilience.

B. EXPERIMENTAL SETUP

The tests undertaken in this study were carried out within the specified experimental setting. It mainly includes the Linux operating system, Intel (R) - i9 processor, NVIDIA GeForce GTX 3090Ti/24G, Python 3.8, PyTorch 1.11, CUDA11.3, etc. See Table 1 for details.

Experiments were conducted using YOLOV5's default optimizer SGD to train the models. Each model is trained for 200 epochs iteratively, and the dimensions of the input images are uniformly scaled to 640 × 640, each batch of training data contains 32 images, and a mosaic enhancement strategy is used to improve the diversity of the training samples. The initial learning rate is set to 0.01, in order to ensure that the model can be trained stably, the learning rate is gradually increased in the earliest three epochs using the warm-up strategy, and the cosine annealing strategy is used to decay the learning rate, so that the trajectory of the learning rate is associated with the cosine function, to prevent the occurrence of local minima, and to improve the overall training efficiency. The main parameter configuration during training is shown in Table 2.

C. INDICATORS OF EVALUATION

To effectively evaluate the model, AP (Average Precision), Precision, recall rate, mAP@.5:. 95 (mAP represents the average AP for all categories, mAP@.5:. 95 is the average

TABLE 2. Main training parameters.

Parameters	Values
weight decay	0.0005
learning rate	0.01
epochs	200
warmup epochs	3
batch size	64
Input image size	(640,640)

TABLE 3. The comparative experiment of the ELAN module and its improved version.

Algorithm	mAP	Precision	Recall	Parameters(M)
YOLOV5-S(base)	44	77.4	57.2	7.0
Base + ELAN	44.6	78.5	61.7	10.5
Base + T-ELAN	44.3	73.8	60.1	7.7
Base + M-ELAN	44.9	76	60.3	7.5

value of mAP calculated at the IoU threshold of every 0.05 units between 0.5 and 0.95 to validate the precision of the model's detection), and FPS (Frames Per Second) were used to quantify the detection accuracy and the speed of model inference. Among them, the deployment mode of the model has a substantial influence on the FPS of the algorithm. Therefore, for the fairness of the experiment, All models were optimized and accelerated using TensorRT and deployed to an experimental platform (Table 1) for testing.

D. M-ELAN MODULE

We replaced only the two convolutional layers of size 1×1 in the neck structure with the ELAN module and the improved T-ELAN and M-ELAN modules without changing the rest of the structure of the network. Under identical conditions, the convolution is learned and subsequently compared to the original model. The experimental findings demonstrate that the network's detection accuracy is enhanced by using ELAN in comparison to the original model, but the network parameters and floating point computation are significantly increased; the T-ELAN module is a lightweight version of the ELAN module we proposed. Although it reduces the parameters of the module, it also causes a decline in accuracy. Neither of these two modules can balance the detection accuracy and detection speed well. In comparison to the preceding two enhancements, the M-ELAN module that we have ultimately put forth not only significantly decreases the number of parameters and floating point computations but also enhances the precision. See Table 3 for specific experimental data. Compared with the original model, with only a few parameters and floating point calculations added, the detection accuracy and recall rate have been comprehensively improved. Moreover, the parallel structure of the module is designed to be highly compatible with the computing characteristics of the GPU, and the computational efficiency is greatly improved. Even though a small number of parameters are added, the inference speed of the model is not reduced, which is proved in the subsequent ablation experiments(Table 8).

TABLE 4. The comparative experiment of different attention mechanisms.

Algorithm	mAP	Precision	Recall	Parameters
YOLOV5-S(base)	44	77.4	57.2	7035811
Base + CBAM	42.6	71.3	59.6	7113479
Base + CA	42.9	71.8	59.8	7088403
Base + ECA	44.2	76.8	58.5	7035823
Base + SimAM	44.3	79.4	59.3	7035811

E. SIMAM

We have selected several mainstream attention mechanisms and added them behind the three feature layers of the neck structure outputs while ensuring that 200 epochs are trained under identical training circumstances. Table 4 shows that SimAM achieves the best average precision and accuracy results without adding parameters. At the same time, in comparison to the initial model, there has been a notable enhancement in the recall rate.

F. PARAMETER BALANCE STRATEGY BETWEEN BACKBONE AND NECK

Under the condition of guaranteeing the original parameter amount and calculation complexity amount, we adjusted the stacking times and channel numbers of CSPLayer modules in the YOLOV5-S backbone and neck to change the parameter ratio of backbone and neck to close to 1:1. Table 5 provides precise information regarding the specific adjustments. Compared with the original model experiment, Table 6 displays the average precision and recall rate after adjusting the parameter ratio of backbone and neck, which have significantly improved; the effectiveness of the improvement is verified. At the same time, as the proportion of parameters in the feature fusion part is strengthened, the information fusion between feature layers of different scales is more abundant, and the low-level feature layer responsible for detecting small target objects also obtains more feature information, which dramatically improves the precision when detecting small targets.

G. LOSS FUNCTION

We used EIoU [34], SIoU [37], and GIoU [38] loss as the model's bounding box loss function. Under the condition that other conditions remain unchanged, the model employs the EIoU loss achieved the best balance in the average precision and recall results, and its average precision and recall are significantly improved compared with the original model. See Table 7 for specific experimental data.

H. ABLATION EXPERIMENT

Based on the above experimental results, we added the improved modules to YOLOV5-S to analyze their contributions to the model performance and propose TTD-YOLO based on these improvements. As shown in Table 8, the M-ELAN module enhances the network feature extraction performance with its multi-scale efficient aggregation network structure, enriches the semantic information fusion

TABLE 5. Parameter adjustment details.

Algorithm	Structure	Stacking times	Channel	Parameters	Proportion
YOLOV5-S(base)	Backbone	1,2,3,1	64,128,256,512	4171456	1.47:1
	Neck	1,1,1,1	256,128,256,512	2834688	
Parameter balance	Backbone	1,2,4,1	64,128,128,512	3164736	1:1.12
	Neck	1,2,1,2	256,128,256,512	3532288	

TABLE 6. Comparison of the initial model and the enhanced model.

Algorithm	mAP	Precision	Recall	FPS
YOLOV5-S(base)	44	77.4	57.2	270
Base + Parameter balance	45.1	77.5	61.4	285

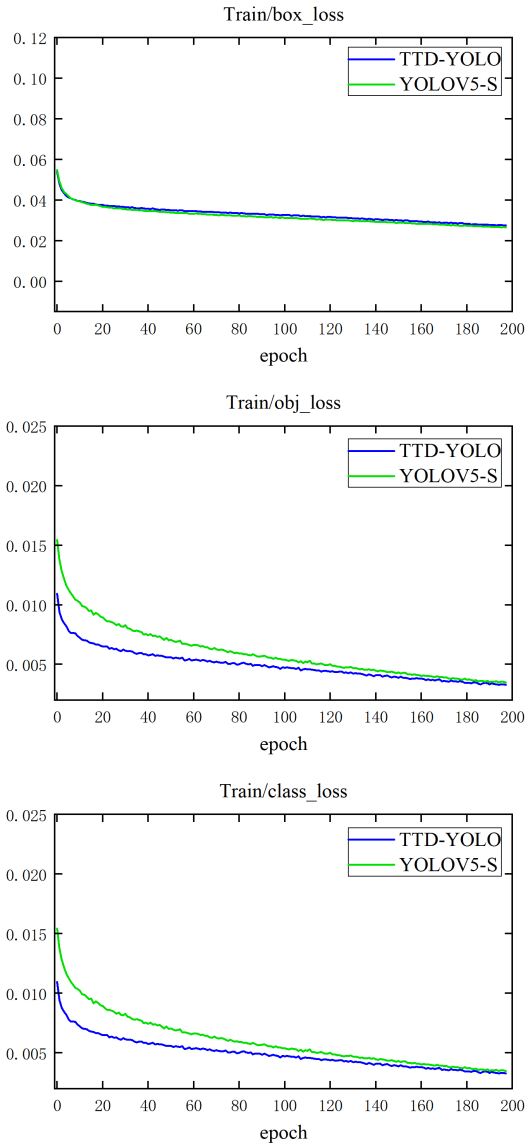


FIGURE 4. Comparison of training loss between the initial model and the improved model.

between different feature layers, and effectively improves the detection accuracy; SimAM enhances the network's ability to acquire valuable feature information from the channel and

TABLE 7. Comparative analysis of the initial model and the enhanced model.

Algorithm	mAP	Precision	Recall	FPS
YOLOV5-S(base)	44	77.4	57.2	270
Base + GIoU	43.5	73.8	58.7	270
Base + SIoU	44.1	73.7	60.9	238
Base + EIoU	44.8	76.4	60.6	244

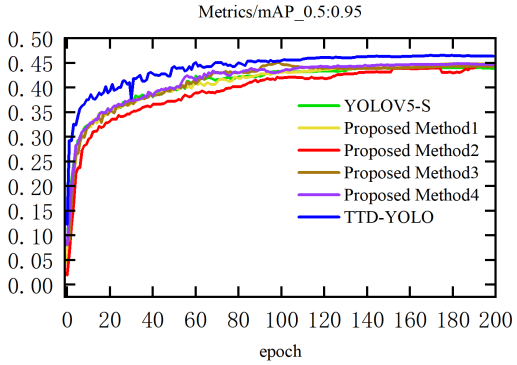
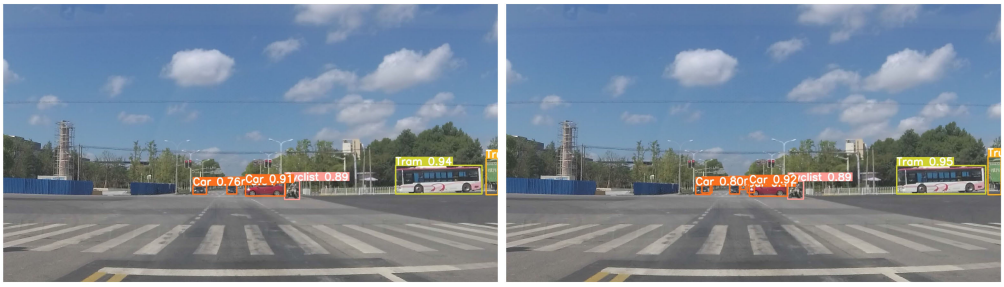


FIGURE 5. The average precision comparison curve between the proposed method and the initial model.

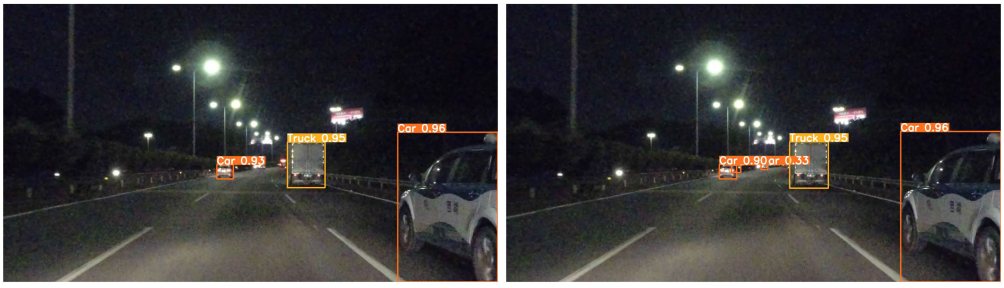
spatial dimensions; the parameter balance strategy between backbone and neck enhance the detection accuracy with the most straightforward idea without adding any additional connections; using EIoU loss as the loss function can accelerate the convergence speed during training. In the end, our proposed model obtains the highest detection accuracy but lags slightly behind the inference speed compared to Method 1 and Method 3, due to the fact that the EIoU loss function and SimAM attention mechanism increase the computational effort while improving the model performance. Ignoring these minor delays, their contribution to the detection accuracy is quite substantial, and the final model exhibits superior inference speed compared to the baseline model. Considering the balance between detection accuracy and inference speed, we adopt these two strategies and finally propose TTD-YOLO that outperforms the baseline model in terms of detection accuracy and inference speed.

The training loss curves of the original model and the modified one are compared in Fig 4. A comparison curve depicting the average precision between the suggested method and the initial model is presented in Fig 5. Fig 6 presents a comparative analysis of the detection outcomes obtained from the original model and the enhanced model across various traffic conditions, Table 9 records the number of targets detected in each scenario in detail.

(a) Comparison of detection results of densely occluded targets



(b) Comparison of detection results of small long-range targets in the road environment



(c) Comparison of detection results under the influence of weak light

(d) Comparison of detection results under the influence of solid light

(e) Comparison of detection results in dense pedestrian scenes

FIGURE 6. **Comparison of detection results of YOLOV5-S(left) and TTD-YOLO(right) in different traffic scenarios.**

TABLE 8. Ablation experiment.

Algorithm	MELAN	SimAM	Parameter balance	EIoU	mAP	Recall	FPS
Base					44	57.2	270
Proposed Method1	√				44.9	60.3	287
Proposed Method2		√			44.4	59.3	260
Proposed Method3			√		45.1	61.4	303
Proposed Method4				√	44.8	60.6	244
TTD-YOLO	√	√	√	√	46.5	62	283

TABLE 9. Comparison of the number of targets detected in different scenarios.

Scene	YOLOV5-S(base)	TTD-YOLO
a	2 Pedestrians,4 Cyclists,7 Cars,1 Truck	2 Pedestrians,5 Cyclists,8 Cars,1 Truck
b	1 Cyclist,3 Cars,1 Truck,1 Tram	1 Cyclist,6 Cars,1 Truck,1 Tram
c	2 Cars,1 Truck	4 Cars,1 Truck
d	1 Car,5 Trucks	1 Car,7 Trucks
e	15 Pedestrians	16 Pedestrians,1 Cyclist

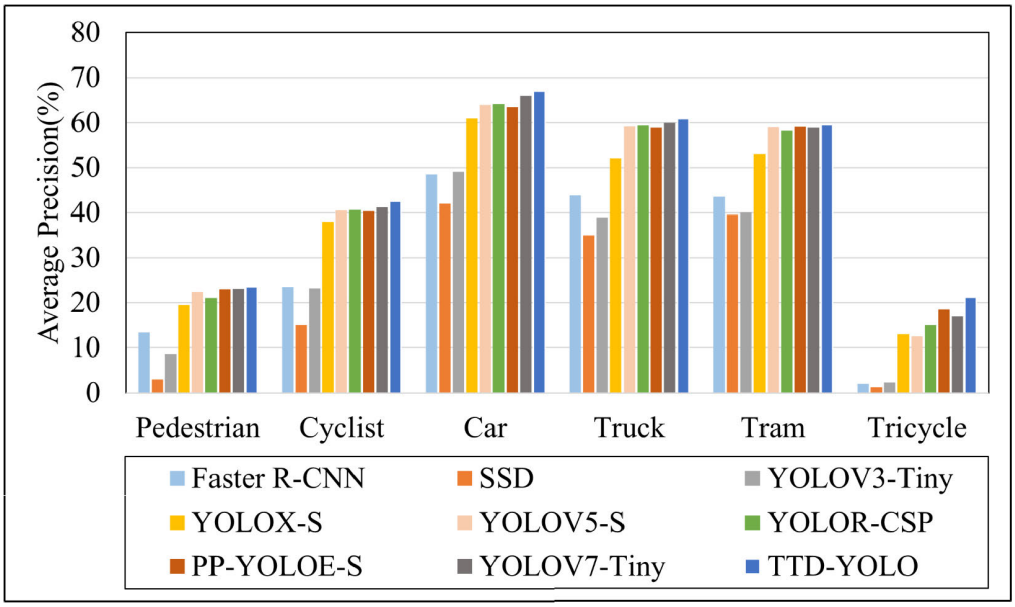


FIGURE 7. Comparison of average precision calculated by different algorithms for each category.

TABLE 10. Comparison of detection results of different algorithms on the SODA10M dataset.

Algorithm	mAP	Recall	FPS
Faster R-CNN	29.4	39.0	60
SSD	23.3	33.1	67
YOLOV3-Tiny	27.4	41.8	281
YOLOX-S	40.3	50.8	145
YOLOV5-S	44	57.2	270
YOLOR-CSP	44.6	58.0	266
PP-YOLOE-S	45.2	59.6	275
YOLOV7-Tiny	45.9	60.3	253
TTD-YOLO	46.5	62	283

We experimentally compared the proposed TTD-YOLO with the current mainstream target detection algorithm. According to the data presented in Table 10, TTD-YOLO demonstrates superior performance compared to other models of similar size regarding both the accuracy of detection and the speed of inference. Fig 7 shows the comparison chart of average precision calculated by all models for each category, it can be seen that our proposed model is significantly superior to other models in detecting small targets (bicycles and tricycles), mainly due to the addition of the M-ELAN module, which improves the model's ability to extract multi-scale feature information. At the same time, the parameter balancing strategy increases the proportion of feature fusion in the neck structure, allowing the shallow feature layer to obtain rich semantic information, greatly improving the model's ability to detect small targets.

V. CONCLUSION

Utilizing the YOLOV5-S algorithm framework, this paper presents a lightweight real-time traffic target detection algorithm TTD-YOLO for complex traffic scenarios. To solve the problems of limited computing power resources, low

accuracy of small target detection, and poor real-time detection of the mobile vehicle platform in the automatic driving environment, the algorithm mainly proposes four feasible improvement strategies: the utilization of the enhanced M-ELAN module is proposed as a substitute for the partial convolution within the initial network neck, compared with the ELAN module, the M-ELAN module has 81.4% fewer parameters and has a more vital ability to extract multi-scale feature information, it can improve the detection speed of the model while ensuring accuracy; in the feature fusion stage of the network, SimAM is incorporated to compute 3D weight, enabling the network to acquire more valuable feature information without the need for supplementary parameters; under the condition that the original parameter quantity is guaranteed, the proportion of parameters in the neck can be improved only by adjusting the number of output channels and stacking times of CSPLayer modules in backbone and neck, the experiment proves that under the condition that the total parameter quantity is unchanged, enhancing the parameter fraction of the neck within a specified range can significantly enhance the detection accuracy; the loss function in the initial model is substituted with the EIoU loss, which can accelerate the convergence and enhance the detection accuracy. The conclusive experiment demonstrates that TTD-YOLOV5-S surpasses its baseline model and other mainstream algorithms, such as SSD, Faster RCNN, YOLOV3-Tiny, and YOLOX-S, on the SODA10M dataset with fewer parameters, higher detection accuracy, and faster inference speed. In comparison to the YOLOV5-S baseline model, the parameter decreased by 8.6%, average precision(mAP@.5:. 95)increased by 2.5%, and inference speed increased by 4.8%.

REFERENCES

[1] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos, and A. F. De Souza, "Self-drivingr cas: A survey," *Expert Syst. Appl.*, vol. 165, Jul. 2021, Art. no. 113816.

[2] T. Yang and C. Tong, "Real-time detection network for tiny traffic sign using multi-scale attention module," *Sci. China Technol. Sci.*, vol. 65, no. 2, pp. 396–406, Feb. 2022.

[3] R.-C. Chen, C. Dewi, Y.-C. Zhuang, and J.-K. Chen, "Contrast limited adaptive histogram equalization for recognizing road marking at night based on YOLO models," *IEEE Access*, vol. 11, pp. 92926–92942, 2023.

[4] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Found. Trends Comput. Graph. Vis.*, vol. 12, nos. 1–3, pp. 1–308, 2020.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Mali, Jun. 2014, pp. 580–587.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV*, vol. 9905. Cham, Switzerland: Springer, Sep. 2016, pp. 21–37.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[9] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[12] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Deep convolutional neural network for enhancing traffic sign recognition developed on YOLOV4," *Multimedia Tools Appl.*, vol. 81, no. 26, pp. 37821–37845, Nov. 2022.

[13] C. Dewi, R.-C. Chen, H. Yu, and X. Jiang, "Robust detection method for improving small traffic sign recognition based on spatial pyramid pooling," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 7, pp. 8135–8152, Jul. 2023.

[14] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "YOLOV4-5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[15] L. Wang, K. Zhou, A. Chu, G. Wang, and L. Wang, "An improved light-weight traffic sign recognition algorithm based on YOLOV4-tiny," *IEEE Access*, vol. 9, pp. 124963–124971, 2021.

[16] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5562–5570.

[17] W. Yang, J. Wu, J. Zhang, K. Gao, R. Du, Z. Wu, E. Firkat, and D. Li, "Deformable convolution and coordinate attention for fast cattle detection," *Comput. Electron. Agricult.*, vol. 211, Aug. 2023, Art. no. 108006.

[18] Y. Cao, C. Li, Y. Peng, and H. Ru, "MCS-YOLO: A multiscale object detection method for autonomous driving road environment recognition," *IEEE Access*, vol. 11, pp. 22342–22354, 2023.

[19] J. Redmon and A. Farhadi, "YOLOV3: An incremental improvement," 2018, *arXiv:1804.02767*.

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOV7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[21] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[23] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[24] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.

[25] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.

[26] L. Yang, R. Zhang, L. Li, and X. Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 11863–11874.

[27] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7029–7038.

[28] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[29] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," 2019, *arXiv:1911.09516*.

[30] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang, and K. Chen, "RTMDet: An empirical study of designing real-time object detectors," 2022, *arXiv:2212.07784*.

[31] C. Baoyuan, L. Yitong, and S. Kun, "Research on object detection method based on FF-YOLO for complex scenes," *IEEE Access*, vol. 9, pp. 127950–127960, 2021.

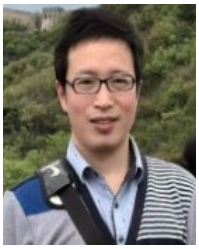
[32] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 516–520.

[33] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.

[34] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022.
[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
[36] J. Han, X. Liang, H. Xu, K. Chen, L. Hong, J. Mao, C. Ye, W. Zhang, Z. Li, X. Liang, and C. Xu, "SODA10M: A large-scale 2D self/semi-supervised object detection dataset for autonomous driving," 2021, *arXiv:2106.11118*.
[37] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
[38] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

WENJUN XIA was born in Chuzhou, Anhui, China. He received the bachelor's degree in automation from the Chengxian College, Southeast University. He is currently pursuing the master's degree with the School of Mechanical and Energy Engineering, Zhejiang University of Science and Technology, China.

His research interests include machine vision and vehicle control in autonomous driving scenarios.



PEIQING LI received the Ph.D. degree in engineering from Southeast University.

He was a Postdoctoral Fellow with Zhejiang University. He is currently a master tutor. Mainly engaged in vehicle dynamics, intelligent transportation, unmanned driving, vehicle-road coordination, road traffic safety, and other interdisciplinary teaching and research work. In recent five years, more than 20 papers have been published in international academic journals and important academic conferences, of which more than ten papers have been retrieved by SCI/EI and participated in the compilation of a monograph. He has presided over and participated in more than ten scientific research projects, applied for 15 national invention patents, and authorized three projects. He is in charge of one National Natural Science Foundation, one Zhejiang Natural Science Foundation, one Postdoctoral Fund, and several horizontal projects. He is a special reviewer of several international SCI/EI journals.

HEYU HUANG received the Ph.D. degree from Yunnan University.

She was a Researcher with China Industry Research Institute, Zhejiang University of Finance & Economics. She is currently a Lecturer. Mainly engaged in research in social sciences, led and participated in more than ten scientific research projects, published papers, and participated in the writing of more than ten monographs.

QIPENG LI received the Ph.D. degree in engineering from Zhejiang University, in 2005.

He went to the Fraunhofer Association, Germany, Italian Energy Economic Research Center, and Ukrainian National Technical University, for academic exchanges and visits. He is currently a Professor and the Dean of the School of Mechanical and Energy Engineering (formerly School of Mechanical and Automotive Engineering). His research interests include intelligent special electromechanical equipment, intelligent assembly and detection systems, and electro-hydraulic servo/proportional control technology.

TAIPING YANG was born in Fuyang, Anhui, China. He received the bachelor's degree in vehicle engineering from Ludong University. He is currently pursuing the master's degree with the School of Mechanical and Energy Engineering, Zhejiang University of Science and Technology, China.

His research interests include vehicle dynamics, intelligent control, modeling, and simulation of vehicle road collaborative safety systems.

ZHUORAN LI received the B.E. degree from Shandong Agriculture and Engineering University, Jinan, China, in 2020. He is currently pursuing the master's degree with the Faculty of Information Technology, City University Malaysia, Petaling Jaya, Malaysia.

His research interests include intelligent driving and the Internet of Things. His research interests include reinforcement learning, decision making, and path planning technology of autonomous vehicle.