Proceedings of Machine Learning Research vol 245:1–9, 2024 2024 International Conference on Machine Learning and Intelligent Computing

Traffic Sign Detection Algorithm Based on Improved YOLOv5

Xing Wei
Shanghai Maritime University, Shanghai, 201306, China

Hongqiong Huang*　　1355010153@QQ.COM
Shanghai Maritime University, Shanghai, 201306, China

Editors: Nianyin Zeng and Ram Bilas Pachori

Abstract

Due to the phenomenon of small size, complex background or high density of traffic signs, there is a certain degree of missing or false detection, which ultimately leads to the problem of reduced detection accuracy. To solve this problem, a real-time traffic sign detection algorithm based on YOLOv5s is proposed. Firstly, feature upsampling is carried out through ContentAware ReAssembly of Features upsampling operator in the neck network, which can aggregate information in the large receptive field, so that the network can get a more accurate feature map. Secondly, the normalized Gaussian Wasserstein distance is used as the similarity measure to construct the NIOU regression bounding box loss function to improve the overall performance of the model. Finally, the FasterNet module is used instead of the C3 module, which is lighter and faster. Experiments were carried out on TT100K data set. Compared with YOLOv5s, CNF-YOLO algorithm reduced Parameters by 1/5, the computing load decreased by 3GFLOPs, the detection speed increased by 18.4 frames/SEC and the weight file decreased by 2.1MB. All models were lighter. And its mAP@0.5 has also been increased by 0.5% to enable rapid detection of traffic signs.

Keywords: Traffic sign detection, YOLOv5, Loss function, TT100K

1. Introduction

In recent years, convolutional network-based traffic sign detection methods (Shyan et al., 2022; Yao et al., 2022; Zaki et al., 2020) have become the mainstream in the field of traffic sign detection.

Peng Jin (2023) improved YOLOv5s by designing C3CBAM convolutional module to increase the attention of traffic sign feature map. The accuracy reached 80.1%, but the detection speed decreased by 1/5. Yang and Zhang (2020) verified that the detection accuracy of YOLOv4 was higher than that of YOLOv3 through the self-labeled data set, while Aggar et al. (2021) found that YOLOv5 was superior to previous algorithms in the field of traffic sign detection. Mijić et al. (2023) created different scenes in the CARLA simulator to test, and the detector based on YOLOv4 reached 90% detection accuracy on real data and synthetic data, but it did not specify the size and detection speed of the model. Liu et al. (2021) improved the YOLOv5s model with MobileNetV2 as the main backbone network. The model parameters decreased by 60%, the weight became lighter, and the map increased by 0.0129%. YOLOv5 was officially released in June 2020, and its detection accuracy and speed have been improved because of the use of lightweight CSPNet-Lite module for feature extraction. YOLOv5 also uses reinforcement learning strategies to achieve the goal of adapting the model to different scenarios. However, for the detection of targets with small size, complex background or dense background, it is easy to miss or misdetect to a certain extent.

Therefore, an improved YOLOv5s network is proposed to improve the accuracy under the premise of satisfying real-time performance. Firstly, the content-aware feature Recombination

(CARAFE) upsampling operator is used to replace feature upsampling in the Neck network (Wang et al., 2019). The normalized Gauss Wasserstein distance (NWD) is used as the similarity measure (Wang et al., 2022), and the IOU is integrated to construct the NIOU boundary frame loss function. Finally, the FasterNet module is used to replace the C3 module (Chen et al., 2023), which is called the CNF-YOLO algorithm.

2. Improved overall network –CNF-YOLO network

By replacing the C3 module and the traditional upsampling, the bounding frame loss function is improved, and the CNF-YOLO model is proposed as shown in Figure 1. On the basis of YOLOv5, the IOU bounding box loss function is replaced by NIOU bounding box loss function. In Neck's feature pyramid, the content-aware Feature Recombination upsampling operator is used for each upsampling. Use FasterNet modules instead of C3 modules throughout the network. For the above improvements, the purpose is to improve the detection accuracy under the premise of ensuring real-time performance.

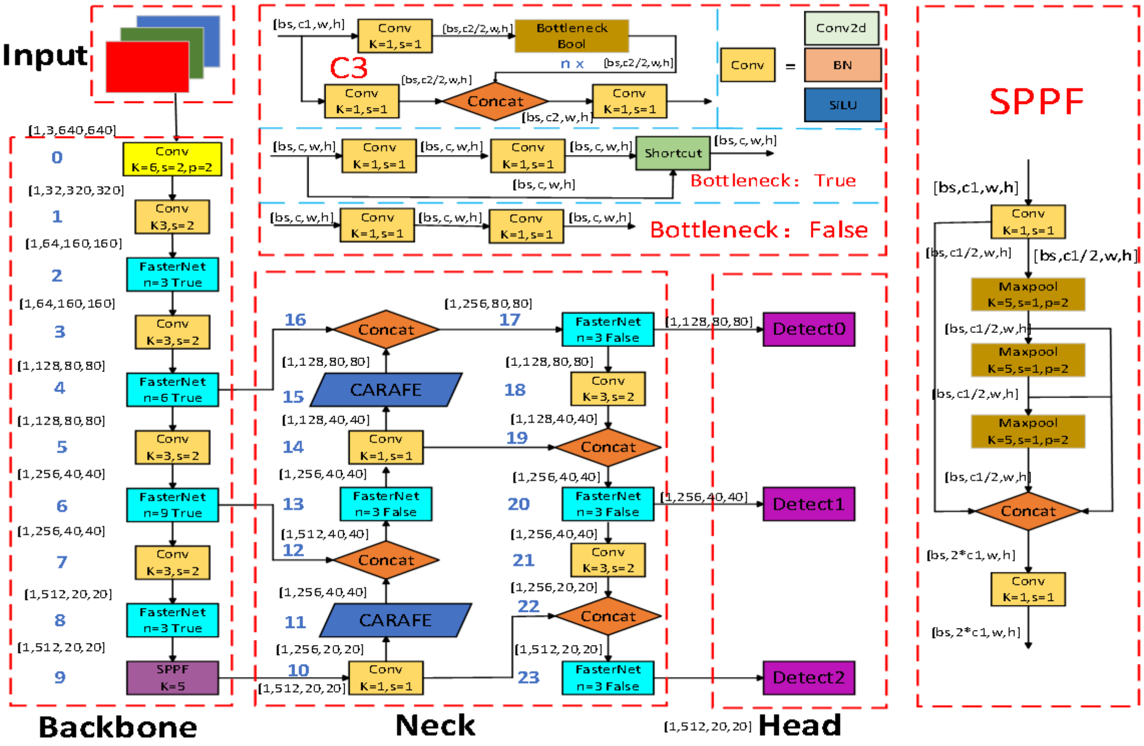


Figure 1: CNF-YOLO model network structure diagram.

3. Improvement method

3.1. Content-aware feature recombination upsampling operator

Before the YOLO detection head, the feature extraction and processing phase needs to ensure that the extracted features are sensitive enough to the position and shape of the target to improve the positioning accuracy of the target. In order to further improve the sensitivity of the model to the location and shape of the target, by using the content-aware Feature Recombination (CARAFE) upsampling operator to carry out the feature recombination, the semantic of the reconstructed feature map can be stronger than the original feature map, and more attention can be paid to the information from the relevant points in the local region, so as to further improve the sensitivity of the model to the location and shape of the target, as shown in Figure 3.

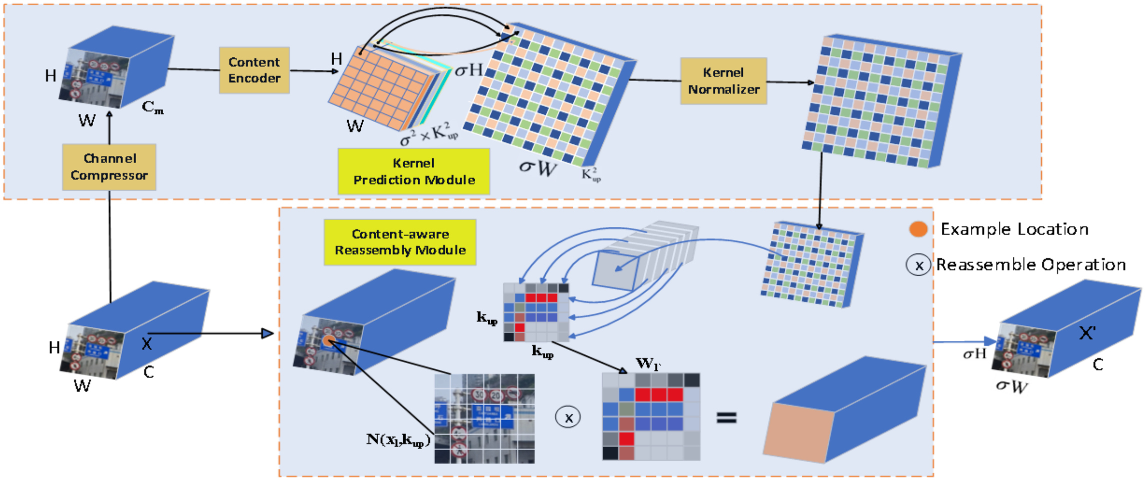


Figure 2: Detailed diagram of content-aware feature recombination upsampling operator module.

As can be seen from Figure 2, the upsampling operator module of content awareness feature reorganization is composed of two parts: kernel prediction module and content awareness reorganization module. The first step is the kernel prediction module to predict a reassembled kernel according to the content of each target location, as shown in Figure 2. In the second step, the content-aware reorganization module reorganizes the features of the kernel predicted in the first part, as shown in the second half of Figure 2. The kernel prediction module ψ predicts the appropriate kernel $\mathrm{W}_{l'}$ for l' location for each location based on x_l's neighbors, as shown in Equation 1. The formula of the reorganization step is formula 2, where ϕ is the content-aware reorganization module, which reorganizes x_l's neighbor and kernel $\mathrm{W}_{\mathrm{l}'}$:

$$\mathrm{W}_{l'} = \psi(\mathrm{N}(\mathrm{x}_\mathrm{l}, \mathrm{K}_{\mathrm{encoder}})) \tag{1}$$

$$\mathrm{X}_{l'} = \phi(\mathrm{N}(\mathrm{x}_\mathrm{l}, \mathrm{K}_{\mathrm{up}}), \mathrm{W}_{\mathrm{l}'}) \tag{2}$$

where $N(x_1, K_{up})$ is the corresponding square area for the target location l' and $l = (i, j)$ center. K_{up} is the size of the reassembled kernel.

The channel compressor uses a 1×1 convolution layer to compress the input feature channel from C to C_m, The content encoder uses a convolutional layer of kernel size $k_{encoder}$ to generate a reassembled kernel based on the content of the input features, each $K_{up} \times K_{up}$ recombination kernel is spatially normalized using the softmax function before being applied to the input feature map.

3.2. FasterNet module

The C3 module in YOLOv5 is a convolution module consisting of multiple Conv layers to extract features and increase the expressive power of the network. While the C3 module performs well in YOLOv5, there are some drawbacks:

1. High computational complexity: C3 module contains multiple convolutional layers, resulting in increased computational complexity. This can cause the model to run slowly with limited resources.
2. Excessive redundant information may be introduced: Multiple convolution of C3 modules may lead to redundant information, thus affecting the generalization ability of the model.

In order to solve the problems faced by the C3 module above, the C3 module in the YOLOv5 model is replaced by the FasterNet module, as shown in Figure 3.

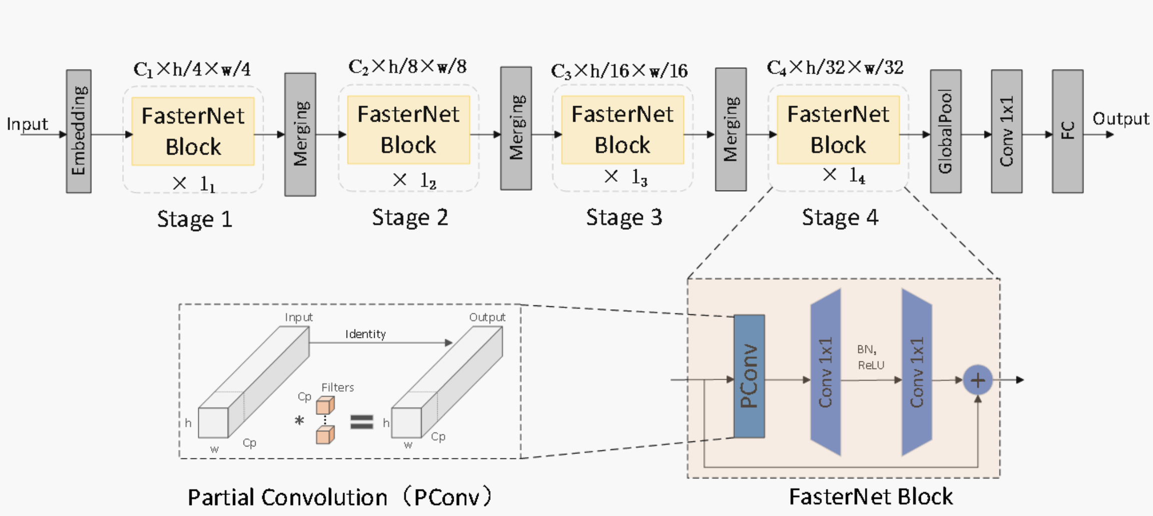


Figure 3: FasterNet overall architecture diagram.

As can be seen from the overall architecture diagram of FasterNet in Figure 3, the FasterNet module is composed of the Embedding layer, FasterNet Block module, Merging layer, GlobalPool layer, 1×1 convolution layer and full connection layer.

3.3. Loss function improvement

In the center and border of some bounding boxes, foreground and background pixels are usually centrally distributed. In order to better describe the weights of different pixels in the bounding box, the normalized Gaussian Wasserstein distance is to construct the bounding box through two two-dimensional Gaussian distributions, the center pixel in the bounding box has the highest weight, and then the importance of the center to the boundary pixel decreases, that is, the weight decreases. Wasserstein distance is derived from optimal transport theory and is used to calculate the distance between two distributions.

For bounding boxes $\mathrm{R} = (cx, cy, w, h)$, where $(cx, cy), w, h$ represents the center point, width, and height of the bounding box, respectively. For the two Gaussian distributions N_a and N_b, the second-order Wasserstein distance between N_a and N_b:

$$W_2^2\left(N_a, N_b\right) = \left\| \left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2} \right]^T, \left[cx_b, cy_b, \frac{w_b}{2}, \frac{h_b}{2} \right]^T \right) \right\|_2^2 \tag{3}$$

where $||.||^2$ indicates the Frobenius norm.

The normalization results in a normalized Gaussian Wasserstein distance:

$$\mathrm{NWD}(N_a, N_b) = \exp\left(-\frac{\sqrt{{W_2}^2\left(N_a, N_b\right)}}{C} \right) \tag{4}$$

IOU fuses NWD as a bounding box loss function NIOU:

$$\mathrm{L}_{\text{box}} = (1.0 - R) * (1.0 - NWD(N_a, N_b)) + R(1.0 - IoU) \tag{5}$$

In formula 5, N_a represents the center point coordinates, width and height of the real boundary box a, N_b represents the center point coordinates, width, and height of the predicted boundary box a. as well. R is a hyperparameter, and when $R = 1$, it means that NWD is not used, and only IOU is used as the bounding box loss function. When $R = 0$, it means that IOU is not used, and only NWD is used as the bounding box loss function.

4. Experiment and analysis

4.1. Experimental setting and data set introduction

The operating system version of the experimental machine in this paper is Windows11, the GPU model is NVIDIA GeForce RTX 4060, the video memory is 8G, the virtual memory size is 200G, the processor is Intel(R) Core(TM) i7-12650H CPU, the main frequency is 2.70GH. All models are based on Pytorch1.13.1 and use CUDA 11.6 and CUDNN 8.0 for GPU acceleration, with Python3.9 as the interpreter.

The data set used in this paper is a traffic sign open data set called Tsinghua-TENcent100K (TT100K for short) jointly produced by Tsinghua University and Tencent Lab based on real traffic scenes. In this data set, there are 45 categories with more than 100 instances, excluding images without sign files, a total of 9166 traffic sign pictures are screened out, among which 7208 are used for training and 1958 are used for verification.

The improved YOLOv5s model was then used for training. The optimizer is SGD, the initial learning rate is 0.07, a total of 400 training cycles, the weight attenuation coefficient is 0.0005, the number of samples input to the model at one time in the training process is 24, the number of multithreads is 8, and other hyperparameters are default values. Through iterative training, the model will learn how to accurately identify different categories of traffic signs. After the training, the validation set is used to evaluate the performance of the model by calculating various evaluation indexes.

4.2. Evaluation index

The evaluation indicators used are general indicators. In the field of object detection images, the evaluation indicators mainly include Recall, Precision, mAP, which is used to measure the detection accuracy of the model, and FPS, which is used to measure the detection speed of the model.

4.3. Ablation experiment

In order to verify the validity of the CNF-YOLO model in this paper, ablation experiments were conducted on the added or improved modules, including NIOU boundary frame loss function, CARAFE module and FasterNet module, in the same environment as the comparison experiment.

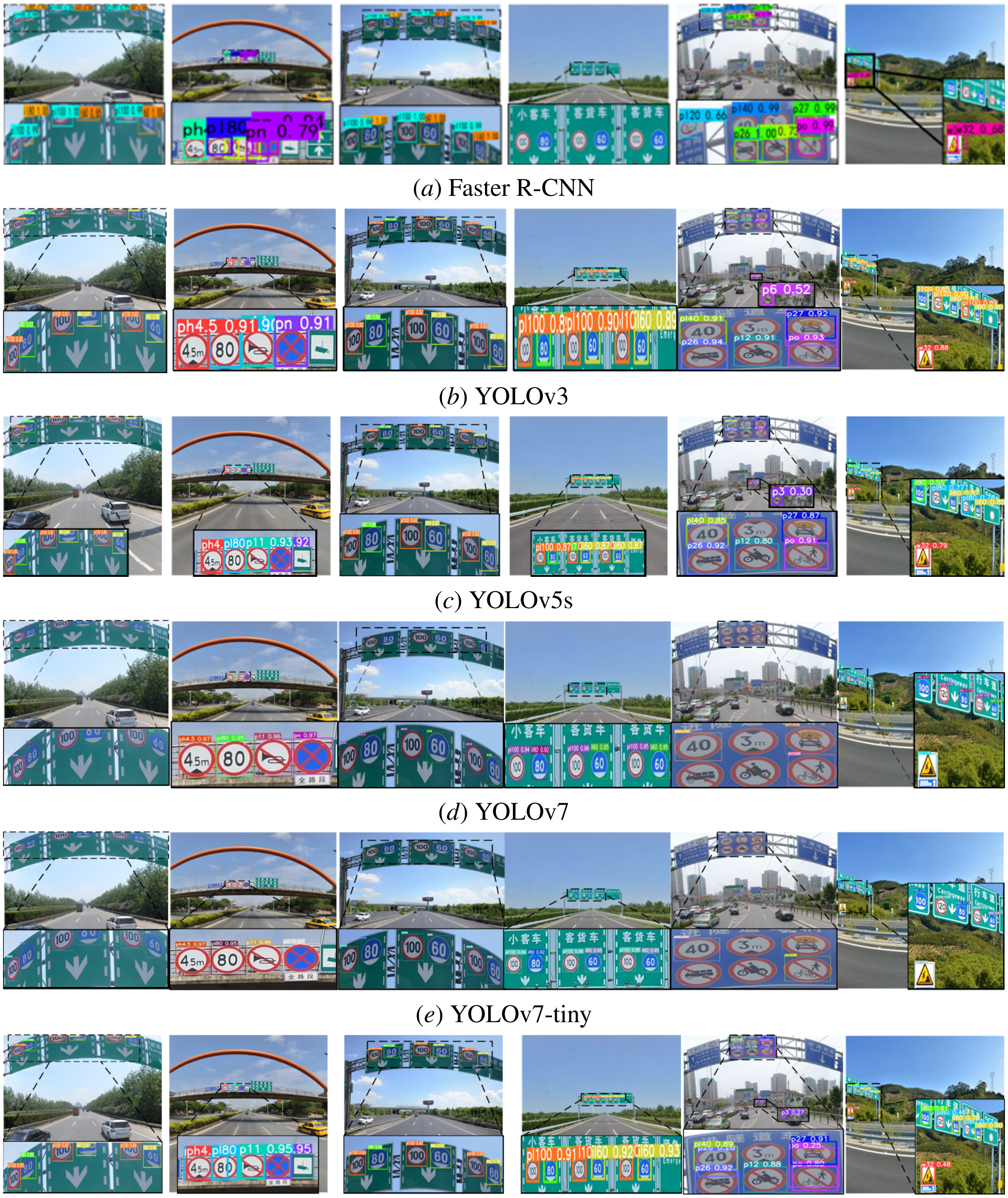
As can be seen from Table 1, compared with YOLOv5s, model a replaces the IOU boundary frame loss function with the NIOU boundary frame loss function integrating NWD on the basis of YOLOv5s to more accurately calculate the loss between the predicted frame and the real frame, mAP@0.5 increased by 0.6%. Based on YOLOv5s, Model b uses CARAFE module for upsampling on the neck network, and mAP@0.5 has increased by 0.3%. In model c, based on YOLOv5s, FasterNet module is used to replace the traditional C3 module. mAP@0.5 decreases by 0.6%, but the number of parameters, computational load and weight also decrease significantly. mAP@0.5 of model d increased by 0.9%, mAP@0.5 of model e and f decreased by 0.1% and 0.2% respectively, but the parameter number and GFLOPs both decreased significantly. On the other hand, the CNF-YOLO model in this paper, mAP@0.5, is increased by 0.5%, its parameter number is decreased by 1.15*106, and the computation amount is decreased by 3 GFLOPs.

Table 1: Ablation results table.

Models	NIOU	CARAFE	FasterNet	Parameters/10^6	GFLOPs	mAP@0.5/%	FPS
YOLOv5s				7.13	16.1	84.1	147.1
a	√			7.14	16.3	84.7	128.0
b		√		72.1	16.2	84.4	90.9
c			√	5.9	13.0	83.5	92.6
d	√	√		7.21	16.2	85.0	109.9
e	√		√	5.9	13.0	84.0	112.4
f		√	√	5.98	13.1	83.9	107.5
CNF-YOLO	√	√	√	5.98	13.1	84.6	168.7

4.4. Comparison of different models

As can be seen from Table 2, compared with Faster R-CNN, SSD, YOLOv7 and Yolov7-TINY, CNF-YOLO model has not only higher mAP@0.5, which is 28.5%, 46.8%, 1.7% and 8.1% higher respectively. In addition, the number of floating point operations and the number of parameters are less, and the calculation amount is reduced by 256.3GFLOPs, 49.6GFLOPs, 90.8GFLOPs and 0.5GFLOPs respectively, which indicates that the structure of Faster R-CNN, YOLOv3 and YOLOv7 networks is complex.



(*a*) Faster R-CNN

(*b*) YOLOv3

(*c*) YOLOv5s

(*d*) YOLOv7

(*e*) YOLOv7-tiny

(*f*) CNF-YOLO (our)

Figure 4: Comparison of each model detection effect.

Although the mAP@0.5 of YOLOv3 model is higher, its floating point operation times, parameter number and weight are almost 11 times that of the model in this paper respectively, and its detection speed is only 41.9 frames per second, which is only 1/4 of that of the model in this paper, failing to meet the requirements of reliable detection and real-time detection of automatic driving.

The mAP@0.5:0.95 of CNF-YOLO model is 8.3% higher than that of YOLOv7 and 4.4% higher than that of YOLOV7-TINY, which indicates that CNF-YOLO model has higher accuracy and better target detection performance under different IOU thresholds. Compared with YOLOv5s, mAP@0.5 of CNF-YOLO model has increased by 0.5%, the computing load has decreased by 3GFLOPs, and the detection speed FPS has increased by 18.4 frames/SEC.

Table 2: Comparative experimental results.

Models	**GFLOPs**	**Parameters /10^6**	**Precision /%**	**Recall /%**	**mAP @0.5/%**	**mAP @0.5:0.9/%**	**FPS**
Faster R-CNN	269.4	41.6	—-	—-	55.1	—-	12
YOLOv3	155.3	61.7	90.1	87.1	91.0	71.8	41.9
SSD	62.7	26.3	—-	—-	37.8	—-	50.3
YOLOv5s	16.1	7.13	85.0	78.1	84.1	64.1	47.1
YOLOv7	103.9	36.7	84.6	76.7	82.9	54.7	47.8
YOLOv7-tiny	13.6	6.1	75.9	70.7	76.5	58.6	128
CNF-YOLO	13.1	5.98	83.6	76.0	84.6	63.0	168.7

As can be seen from Figure 4, both YOLOv7 and YOLOV7-Tiny have missed detection phenomena during the detection of the fifth image, and the missed detection phenomenon is more serious for YOLOV7-TINY, while the model in this paper has not missed detection phenomenon, indicating that the model in this paper has good performance.

5. Summary

This paper presents a real-time CNF-YOLO detection algorithm. Firstly, FasterNet module is used to replace C3 module and NIOU loss function is used to replace IOU loss function to calculate boundary loss more accurately. Upsampling is done by CARAFE upsampling operator. The experimental results show that the CNF-YOLO algorithm improves the average accuracy, is lighter than the original algorithm, and the detection speed meets the real-time requirements of automatic driving. In the future, we will consider the scene characteristics of traffic signs to design the network structure and build a sign detection algorithm with stronger generalization ability.

References

Ammar Aggar, Abd Alrazak Rahem, and Mohammed Zaiter. Iraqi traffic signs detection based on yolov5. In *2021 International Conference on Advanced Computer Applications (ACA)*, pages 5–9, 2021. doi: 10.1109/ACA52198.2021.9626821.

Jierun Chen, Shiu hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S. H. Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks, 2023.

Xun Liu, Xiangkui Jiang, Haochang Hu, Rui Ding, Hong Li, and Chunlin Da. Traffic sign recognition algorithm based on improved yolov5s. In *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 980–985, 2021.

David Mijić, Mario Vranješ, Ratko Grbić, and Borna Jelić. Autonomous driving solution based on traffic sign detection. *IEEE Consumer Electronics Magazine*, 12(5):39–44, 2023. doi: 10.1109/MCE.2021.3090950.

Li Muyi Peng Jin, Sang Zhengxiao. A traffic sign detection algorithm based on yolov5s. *Automation Technology and Application*, 42(9):53–57, 2023.

Lai Jia Shyan, T H Lim, and Dk Norhafizah Pg Hj Muhammad. Real time road traffic sign detection and recognition systems using convolution neural network on a gpu platform. In *2022 International Conference on Digital Transformation and Intelligence (ICDI)*, pages 236–240, 2022. doi: 10.1109/ICDI57181.2022.10007277.

Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features, 2019.

Jinwang Wang, Chang Xu, Wen Yang, and Lei Yu. A normalized gaussian wasserstein distance for tiny object detection, 2022.

Wenkao Yang and Wei Zhang. Real-time traffic signs detection based on yolo network model. In *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 354–357, 2020. doi: 10.1109/CyberC49757.2020.00066.

Yingbiao Yao, Li Han, Chenjie Du, Xin Xu, and Xianyang Jiang. Traffic sign detection algorithm based on improved yolov4-tiny. *Signal Processing: Image Communication*, 107:116783, 2022. ISSN 0923-5965. doi: 10.1016/j.image.2022.116783.

Pavly Salah Zaki, Marco Magdy William, Bolis Karam Soliman, Kerolos Gamal Alexsan, Keroles Khalil, and Magdy El-Moursy. Traffic signs detection and recognition system using deep learning, 2020.