Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 14, No. 2, 2024

Improved Road Target Detection Algorithm for YOLOv7-Tiny

Shuyan Chen [1], Dongmei Ma [1,*], Xiaoyun Luo [2]

[1] School of Physics and Electronic Engineering, Northwest Normal University, Lanzhou 730070, China
[2] Gansu Intelligent Information Technology and Application Engineering Research Center, Lanzhou 730070, China
[*] **Corresponding author**: Dongmei Ma (Email: madongmei@nwnu.edu.cn)

Abstract: In complex road scenes, we propose an enhanced road target detection algorithm for YOLOv7-Tiny to address issues such as large model size, misclassification, and low localization accuracy. Our approach involves several key modifi-cations. Firstly, we replace the LeakyReLU activation function in YOLOv7-Tiny with H-swish. This replacement not only reduces the number of parameters in the model but also enhances its feature extraction capabilities. Additionally, we replace the ELAN module in the Neck with the DPCH-ELAN module and introduce the darknetblock module along with the pcconv convolution. These modifications improve the network's ability to comprehend complex patterns and semantics, thereby enabling it to capture features at different levels of input data. Moreover, we introduce the pcconv convolution building block at the output side to handle heterogeneous information in complex road scenes, thereby enhancing the network's performance in detecting road targets and abnormalities. In our experiments using a mul-ti-source dataset, the improved model exhibits a reduction in GFLOPs by 20.90% and a decrease in the number of pa-rameters by 24.49%. Furthermore, the mean average precision scores (map) at thresholds of 0.5 and 0.5~0.9 are im-proved to 77.3% and 51.8%, respectively, compared to the original YOLOv7-Tiny model. These experimental results demonstrate that our enhanced model achieves a reduction in model size while simultaneously enhancing detection accuracy, thereby meeting the requirements for real-time detection. To assess the generalizability of our approach, we conducted comparison experiments on the VOC2012 dataset. The results indicate that the improved algorithm exhibits robust generalization capabilities across different datasets.

Keywords: Road target detection; YOLOv7-Tiny; H-swish; Darknetblock; Pcconv; Multi-source dataset.

1. Introduction

With the continuous development of urban transportation and the widespread adoption of various means of transportation, there is a growing demand for the monitoring and management of road transportation systems. Concurrently, road-related issues such as potholes and traffic cones pose significant challenges to urban transportation and necessitate effective monitoring and management strategies. Road target detection technology, a pivotal area in computer vision research, endeavors to offer crucial support for urban traffic management, intelligent transportation systems, and driver assistance systems by autonomously identifying and tracking targets on the road [1-3].

Currently, target detection algorithms in deep learning can be categorized into two main groups. The first group consists of two-stage detection algorithms, which typically involve two main stages. Initially, candidate target

regions are generated using Region Proposal Network (RPN). Subsequently, classification and location regression are performed on these candidate regions. Representative algorithms in this category include R-CNN, Fast R-CNN, and Faster R-CNN (Region-based Convolutional Neural Network). The second group comprises single-stage detection algorithms, which predict the location and class of the target directly from the image without explicitly generating candidate regions. Such algorithms often exhibit faster inference speeds. Representative single-stage detection algorithms include the YOLO (You Only Look Once) series [7-11], SSD (Single Shot MultiBox Detector), and EfficientDet.

In addressing the challenges posed by complex road scenes, Qiu M et al. proposed an algorithm for extracting traffic element information from UAV remote sensing images, based on ASFF-YOLOv5. This algorithm employs an adaptive spatial feature fusion method utilizing the sensing field module, which effectively leverages diverse scale information, enhances feature scale invariance, and improves the detection accuracy of small targets. When detecting multiple road traffic elements, the proposed method demonstrates a 2% improvement in mAP compared to the original YOLOv5 network, albeit with a high model complexity. Jiang T et al. introduced CBAM into the YOLOv5s backbone network to optimize its structure, and adopted CIoU loss as the loss function for object bounding box regression to expedite the regression process. Nonetheless, some detection errors and leakage issues persist in target-dense complex images. Wang C Y et al. proposed a novel network design strategy based on gradient path analysis, which served as the foundational concept for the YOLO series of models.

Qi Linglong et al. combined the concept of feature discrete merging with enhancements to the MPConv module in the YOLOv7 network model to mitigate feature loss during network feature processing. They employed the ACmix attention module to enhance the network's sensitivity to small-scale targets while minimizing noise impact. Additionally, they replaced CIoU with SIoU in the original YOLOv7 network model to optimize the loss function, reducing its degree of freedom and enhancing network robustness. The improved YOLOv7 network model achieved an mAP of 71.1%, outperforming the original network and other classical target detection networks, although its detection speed still falls short of real-time requirements.

When applying the aforementioned target detection methods on mobile devices, they often encounter challenges such as oversized models, high computational parameters, and difficulty in striking a balance between speed and accuracy. To effectively address these issues, particular emphasis is placed on the computational requirements of deploying target detection algorithms in assisted driving systems on mobile devices. Given the current challenges, this paper focuses on optimizing and enhancing the lightweight YOLOv7-Tiny model to create the HDP-YOLOv7 model, which aims to achieve a balance between detection accuracy and speed. The following summarizes the important contributions of this paper.

Replacing all CBL convolution blocks in YOLOv7-Tiny with CBH, and substituting the LeakyReLU activation function in these CBL convolution blocks with H-swish, enables the model to achieve a better balance between performance and computational overhead, particularly in scenarios requiring lightweight design and constrained computational resources.

To enhance the feature fusion capability of the algorithm, modifications are made to the ELAN layer in the neck network. The 2nd CBL module is replaced with a pcconv convolutional block, while the 3rd CBL module is replaced with n darknetblock modules in series. The outputs of all these darknetblock modules are directly linked to the final concatenation operation, thereby improving the feature fusion network's fusion capability.

In order to enhance the network's ability to detect road targets and diseases, pcconv convolutional building blocks are introduced in the detection head section. This facilitates more accurate capturing of features related to road targets and road diseases.

2. Overall structure of the Yolov7-tiny algorithm

The YOLO family is renowned for its end-to-end design and real-time performance across a diverse range of target detection tasks. YOLOv7-Tiny, a lighter iteration within the YOLO (You Only Look Once) target detection family, is specifically designed to offer faster reasoning speeds by simplifying the network structure while maintaining high efficiency and performance. This model leverages its end-to-end design and real-time capabilities to address the efficiency and accuracy limitations commonly associated with traditional algorithms.

The YOLOv7-Tiny model is particularly well-suited for resource-constrained embedded systems and real-time applications, providing faster inference times through network simplification. This enhancement compensates for the inefficiencies and accuracy drawbacks typically encountered with traditional algorithms. In the context of this paper, which focuses on complex road target detection, including the identification of road diseases, both speed and accuracy requirements are paramount. Hence, the YOLOv7-Tiny model has been chosen for optimization and improvement. Comprising four main components input, backbone network, feature fusion network, and output, YOLOv7-Tiny's structure is illustrated in Figure 1 [18-20]

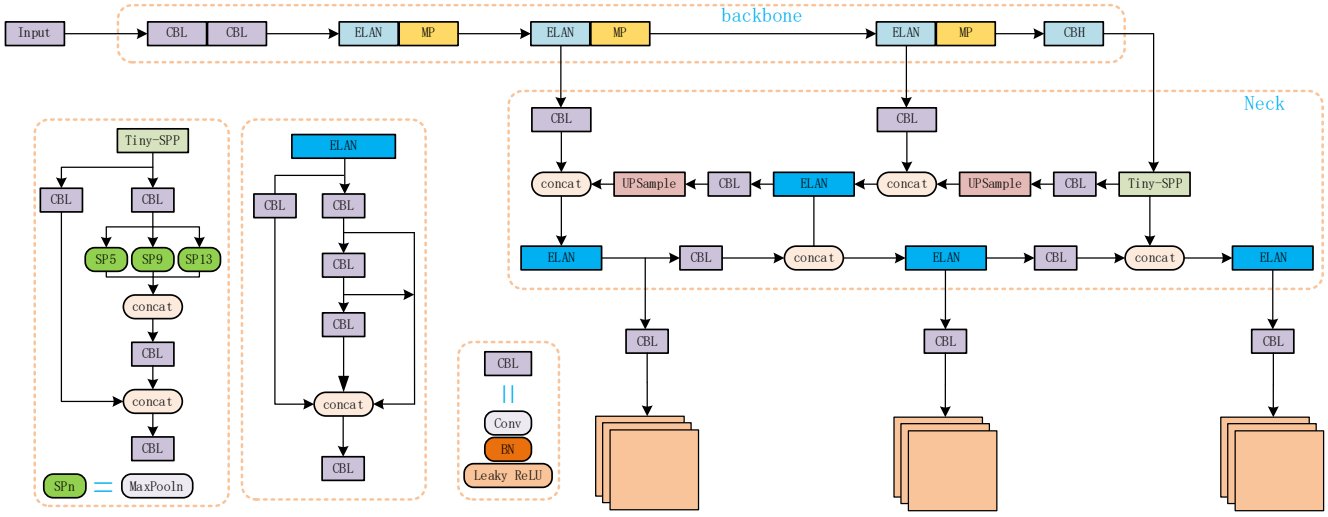


Figure 1. Network structure of Yolov7-tiny algorithm

In this paper, we enhance the YOLOv7-Tiny algorithm to bolster its feature extraction network capability, enhance the performance of the feature fusion network, and improve the integration of global information. These enhancements enable the model to autonomously learn and comprehend complex features, thereby enhancing detection performance across various traffic scenarios. Additionally, the improved model is adept at effectively monitoring and identifying roadway diseases, thereby injecting more possibilities into future intelligent transportation systems.

3. Improved Road Target Detection Algorithm for YOLOv7-Tiny

While YOLOv7-Tiny has been simplified compared to YOLOv7, it still entails numerous parameters and a complex model structure. To enable its usage across diverse traffic scenarios and enhance its capability to effectively monitor and identify roadway distress, several modifications have been proposed. By refining the lightweight YOLOv7-Tiny model, we introduce the HDP-YOLOv7 network, as depicted in Figure 2.

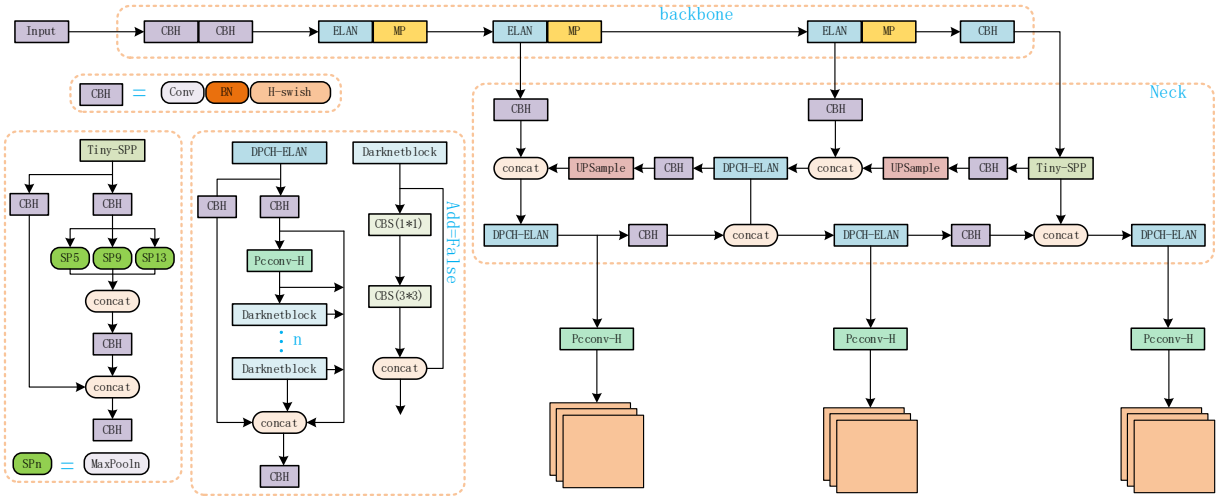


Figure 2. HDP-YOLOv7 network model

3.1. H-swish activation function improves computational efficiency and increases feature extraction capability

To better capture complex features in complex road target images and enhance the performance of the feature extraction network, the original activation function has been replaced with H-Swish. Given that road target detection often operates in resource-limited environments such as mobile devices, the H-Swish activation function offers advantages in designing lightweight models compared to the LeakyReLU activation function. It provides a choice of activation function that improves computational efficiency. Moreover, the H-Swish activation function is characterized by its smoothness and continuous derivatives within its domain of definition. This smoothness aids in stable gradient propagation, mitigates the gradient vanishing problem, and ultimately enhances the performance of the feature extraction network.

The H-Swish (Hard Swish) activation function was initially proposed by Howard A et al. It serves as an updated activation function in MobileNet-V3. While MobileNet-V2 employs the ReLU6 activation function, MobileNet-V3 adopts the H-Swish activation function, an improvement over the Swish function.

Swish is a nonlinear activation function originally proposed by Google, which has been found to outperform ReLU in certain deep networks. For instance, its performance in fully connected layers comprising more than 40 layers is notably superior to that of other activation functions. The formula for the Swish activation function is shown in Eq. (1):

$$\text{swish}\, x = x \cdot \sigma(x) \tag{1}$$

Swish possesses properties such as being unbounded from above and bounded from below, smoothness, and non-monotonicity. However, due to the impracticality of implementing sigmoid functions on hardware devices, the sigmoid component in Swish is replaced with a similar computation based on ReLU6, resulting in H-Swish.

H-Swish computation is faster and simpler to quantify compared to Swish. The formula for H-Swish is shown in Eq. (2):

$$\text{h}-\text{swish}[x] = x\frac{\text{ReLU}\,6(x+3)}{6} \tag{2}$$

where ReLU6(x) denotes the ReLU function that trims the input x so that it ranges from 0 to 6.

From the above equation, it can be seen that the form of H-Swish contains linear terms, which makes the computation more lightweight on certain hardware, especially in lightweight model design, H-Swish can better balance the performance and computational overhead. Therefore, the use of H-Swish activation function in complex road scenarios can largely improve the computational efficiency.

A comparison of the smoothness of the curves for the activation function Swish and the activation function H-Swish is depicted in Figure 3.

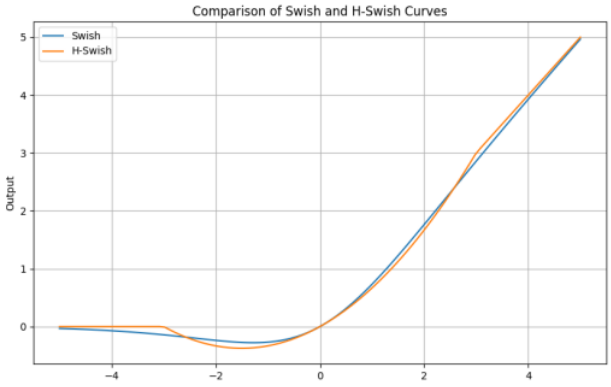


Figure 3. Comparison of curve smoothness between swish and H-Swish

The Leaky ReLU activation function used in YOLOv7-Tiny, while also a nonlinear activation function, is not entirely smooth around zero. This characteristic may result in gradient destabilization, particularly during the training of deep networks. The formula for Leaky ReLU is shown in Eq. (3).

$$\text{LeakeyReLU}\,(x) = \begin{cases} x, x > 0 \\ ax, x \le 0 \end{cases} \tag{3}$$

A comparison of the smoothness of the curves for the activation function LeakyReLU and the activation function H-Swish is illustrated in Figure 4.

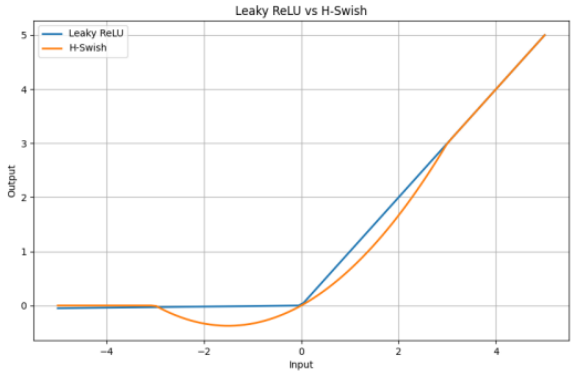


Figure 4. Comparison of curve smoothness between LeakeyReLU and H-Swish

As evident from the curve smoothness comparison graph, Leaky ReLU exhibits smaller gradients in the negative region, potentially leading to gradient vanishing, particularly during the training process in deep networks. On the other hand, the replacement activation function H-Swish demonstrates smaller derivatives in the saturated region, which aids in alleviating the issue of gradient explosion and enhances the stability of model training. Experimental results indicate that the H-Swish function facilitates the model in better capturing complex features in complex road target images and enhances the performance of the feature extraction network.

3.2. Improvement of ELAN Module to Enhance Feature Fusion Network Performance

The ELAN layer in the Neck segment of the YOLOv7-Tiny network is composed of multiple CBL modules, as depicted in Figure 5. Initially, the first CBL block generates a feature layer, followed by the second CBL module producing another feature layer, and subsequently, the third CBL module and the fourth CBL module, branching from the periphery, each outputting a feature layer. These feature layers from the various branches are then fused through a Concat layer connection. Finally, the fused features from the Concat layer undergo another CBL module before being outputted. This structure enhances fusion efficiency by allowing the stacking of more blocks while ensuring the shortest gradient path. The feature information extracted from the convolutional blocks is then fused through tensor splicing. However, tensor splicing alone may not adequately fuse feature information between the upper and lower layers of the network. The structural diagram is presented in Figure 5.

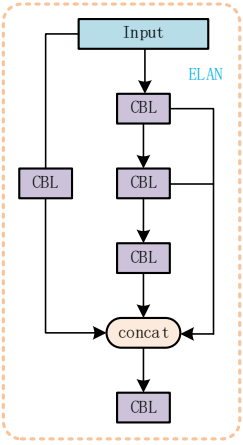


Figure 5. ELAN network structure

The improvement method involves modifying the ELAN structure as follows: replacing the 2nd CBL module with a pcconv convolutional block, changing the 3rd CBL module to a series of n darknetblocks, and routing the output to the final Concat fusion. This alteration enhances the fusion capability of the feature fusion network while reducing the overall network's parameter count through parameter sharing. The enhanced ELAN module is denoted as DPCH-ELAN. The structure of DPCH-ELAN is illustrated in Figure 6.

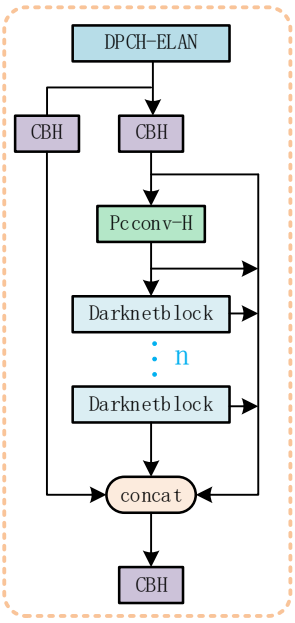


Figure 6. DPCH-ELAN network structure

Here, replacing the 2nd CBL module with the pcconv convolution block enables the network to more effectively capture the relationships between various entities in the road scene. The pc-conv convolution employs a dual filtering approach to unify homogeneity and heterogeneity, utilizing Poisson-Charlier polynomials for precise numerical approximation. This polynomial formulation facilitates accurate numerical computation in aggregating both heterogeneous and homogeneous information, thereby enhancing representation learning of graph data and improving the fusion capabilities of feature fusion networks. For a detailed description of pcconv, please refer to Section3.3.

The darknetblock module is introduced to enhance the nonlinear representation capability of the model. A darknetblock comprises two convolutional blocks (CBS) concatenated into a single "darknetblock" block, as illustrated in Figure 7. By employing two convolutional blocks, the network can learn more intricate and abstract feature representations. This augmentation enhances the network's capacity to comprehend complex patterns and semantics. Furthermore, the first convolutional block primarily focuses on capturing low-level details, while the second one emphasizes capturing high-level semantic information, thus facilitating multi-level feature extraction. Moreover, the parameter sharing between the two convolutional blocks helps reduce the total number of parameters in the network. Consequently, this aids in mitigating the computational burden of the model and lowers the risk of overfitting.

In complex road scenarios, multiple darknetblocks (n) are connected in series to bolster the model's generalization ability. This approach enables the model to better adapt to

intricate and evolving real-world scenarios.

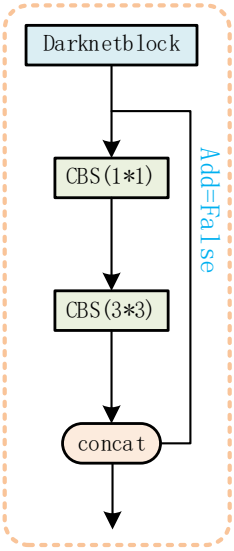


Figure 7. darknetblock structure

3.3. 3.3Introducing pcconv convolution to improve the detection performance at the output side

In this paper, we introduce an innovative graph convolution operation, PCConv convolution, to enhance adaptability to complex road scenes and improve target detection performance. PCConv convolution, serving as a graph convolution utilizing polynomial filters, exhibits a potent expressive capability for learning complex nonlinear features. This feature makes it particularly suitable for capturing the diversity and complexity present in road scenes. The structure of PCConv convolution is illustrated in Figure 8.

The PCConv convolution operation consists of three inputs, namely the input feature $f_{c,i}\left(\mathbf{x}_i\right)$, the input point set x and the output point set y. PCConv produces the output feature $h_{k,j}\left(\mathbf{y}_j\right)$ by combining the information from the input and output point sets through a filtering operation on the input feature. The key formula is Eq. (4):

$$h_{k,j}\left(\mathbf{y}_j\right)=\sum_{c=1}^{C}\sum_{i=1}^{N} f_{c,i}\left(\mathbf{x}_i\right) g_{c,k}(\mathbf{e};\boldsymbol{\theta}) \tag{4}$$

where C and K represent the dimensions of the input and output features, respectively, and N denotes the number of input points. PCConv convolution is categorized into two modes: heterogeneous aggregation and homogeneous aggregation. Feature fusion is executed through different energy functions to effectively accommodate the graph structure.

Experiments have demonstrated that the utilization of PCConv convolution at the output stage not only enhances detection performance but also better caters to the detection requirements of small targets. The superior performance and flexibility of its polynomial filters enable the model to adeptly adjust to the complex characteristics of various road scenes. PCConv convolution exhibits a stronger expressive capability compared to traditional low-pass and high-pass filters, enabling it to learn any polynomial filters and better adapt to complex road scenes. Consequently, PCConv convolution makes a significant contribution to the advancement of intelligent transportation systems and the safety of urban traffic.

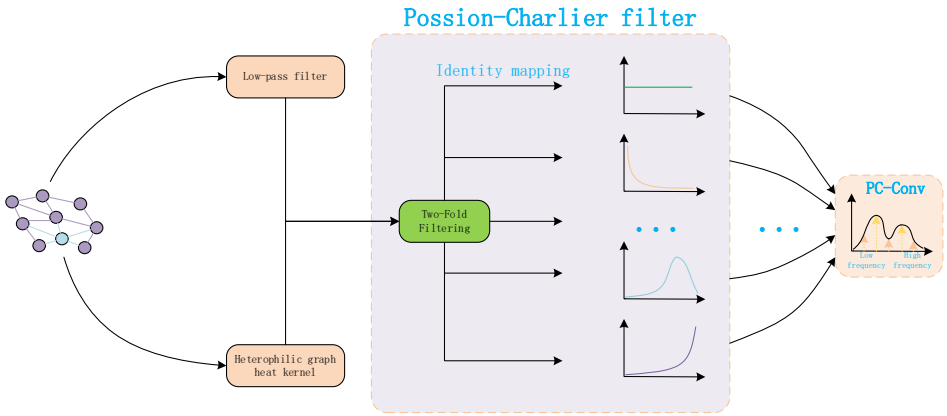


Figure 8. pcconv convolution

4. Experiment and Result Analysis

4.1. Experimental environment and parameter settings

The specific experimental environment is detailed in Table 1.

In this paper, the YOLOv7-Tiny network serves as the baseline model. The network is trained using the SGD optimizer with the following hyperparameters: an initial learning rate of 0.01, a momentum of 0.937, and a weight decay of 0.0005. Additionally, the number of preheat iterations is set to 3, with a preheat momentum of 0.8. The batch size is set to 16, and training is conducted for 300 epochs. The image inputs are resized to dimensions of 640 x 640.

Table 1. Experimental environment

Project	Content
CPU	Intel Core i5-10400@2.90GHz
GPUs	NVIDIA RTX 3070
Video memory	8GB
Running memory	16GB
Operating system	Ubuntu 22.04LTS
Deep Learning Framework	Torch1.12.1+cu113
Data processing	Python 3.8

4.2. Data sets

The improved algorithm model discussed in this paper is primarily studied and analyzed using multi-source datasets. Additionally, experimental validation is conducted on the

Pascalvoc2012 dataset to demonstrate the effectiveness of the proposed algorithm. This approach aims to verify the universality of the algorithm presented in this paper and provides empirical evidence of its performance across various datasets.

(1) Multi-source data sets

Based on common road target detection objects, this paper intentionally retrieves detection targets, including road potholes, road cones, and barrels, which pose risks to safe driving, from the platforms of Flying Paddle and Pole City. Additionally, road targets are selected from the Pascal VOC2012 dataset to create a new multi-source dataset. This dataset comprises 10 types of targets, such as road potholes, road cones, pedestrians, bicycles, etc., and contains a total of 1,4359 images.

(2) Pascalvoc2012 dataset

The PASCAL VOC 2012 dataset is widely utilized in various computer vision tasks, including target detection and semantic segmentation. This dataset comprises a total of 20 classes of objects, with 11,530 images in the TRAIN and VAL subsets, along with a total of 27,450 target detection labels. For the experiments, 90% of the images in the dataset will be selected for training purposes, while the remaining 10% will be utilized to validate the detection performance.

4.3. Evaluation indicators

The experiments utilize five evaluation metrics: mAP, Precision, Recall, Parameters, and GFLOPs, each described as follows:

(1) mAP (Mean Average Precision): This is a widely employed performance evaluation metric in target detection, representing the mean of the average precision across different categories of targets. mAP consolidates the accuracy and recall of the model over each category. It is calculated as follows:

$$mAP = \frac{1}{c}\sum_{i=1}^{c} AP_i \tag{5}$$

For each category, the Area Under the Precision-Recall curve (AUC) is initially computed, and then averaged across all categories. Here, C represents the number of categories, and APi denotes the average precision of the ith category.

(2) Precision: Precision reflects the accuracy of the model's predictions by indicating how many of the samples predicted as positive categories by the model are truly positive category samples. It is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Where TP represents the true positive cases (the number of samples correctly predicted as positive cases by the model), and FP signifies the false positive cases (the number of samples incorrectly predicted as positive cases by the model).

(3) Recall: Recall assesses how well the model covers positive examples by indicating how many of the true positive category samples are successfully predicted as positive by the model. It is calculated using the formula:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

Where TP represents the true positive cases, and FN represents the false negative cases (the number of samples that the model incorrectly predicts as negative).

(4) Parameters: This metric indicates the number of parameters in the model, encompassing weights and biases. A higher number of model parameters may lead to increased model complexity, but it also raises the risk of overfitting.

(5) GFLOPs (Giga Floating-point Operations Per Second): GFLOPs serve as a measure of a model's computational speed, representing the number of billion floating-point operations performed by the model per second. GFLOPs are correlated with the model's computational complexity and computational efficiency.

4.4. Ablation experiments

In this paper, we adopt the YOLOv7-Tiny benchmark model as our baseline and implement several improvements. Specifically, we replace the LeakyReLU activation function in YOLOv7-Tiny with H-swish, substitute the ELAN module in the Neck with the DPCH-ELAN module, and introduce the pcconv convolution to replace the CBL at the output end. These modifications result in the improved model, termed HDP-YOLOv7. To comprehensively evaluate the effectiveness of these enhancements, we conduct ablation experiments on multi-source datasets to assess the impact of each proposed improvement. Each enhancement is sequentially incorporated into the YOLOv7-Tiny model, and consistent training parameters and environmental conditions are maintained across all experiments. The experimental results are summarized in Table 2.

Table 2. Ablation experiments

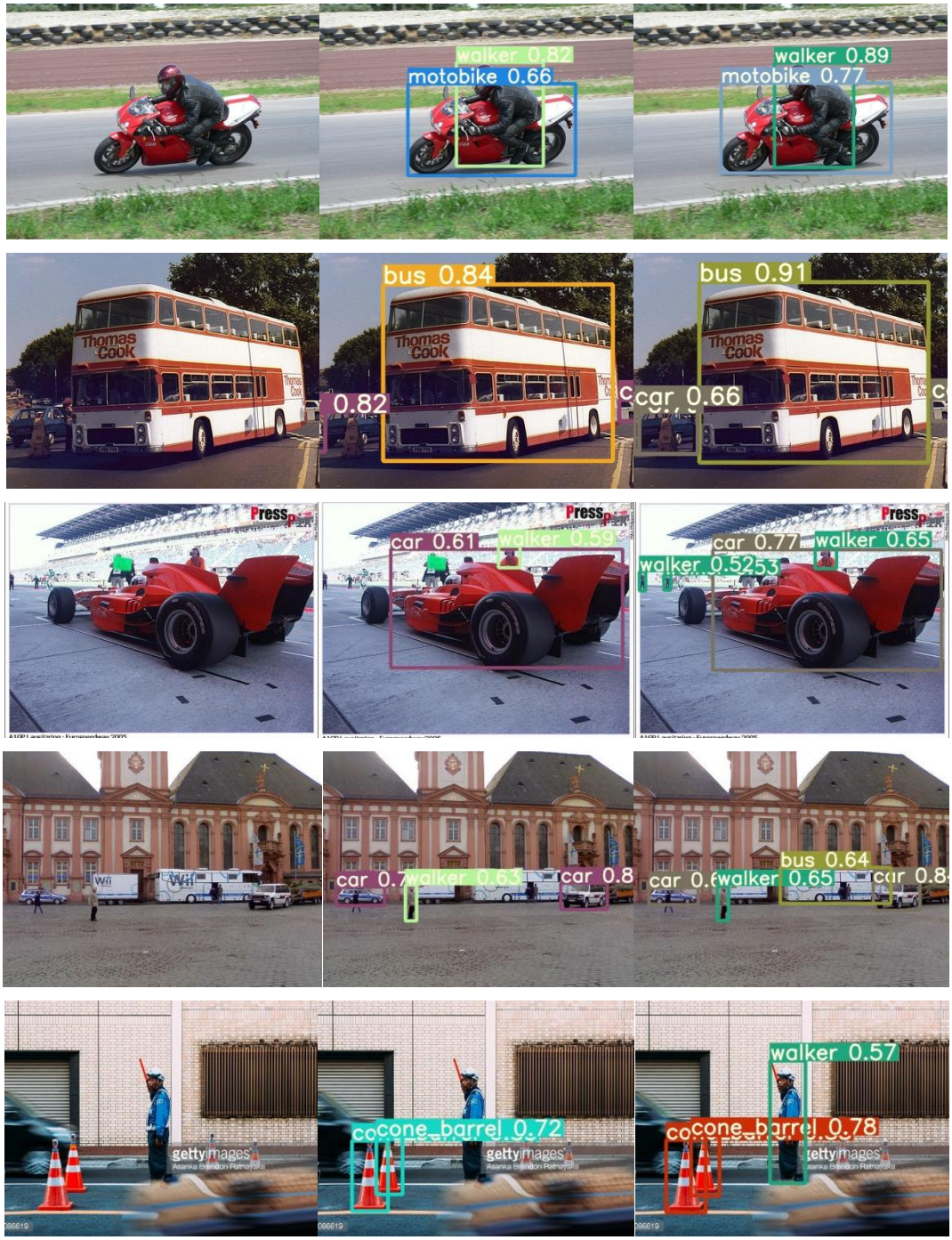
Model	Yolov7-tiny	h-swish	pcconv	DPCH-ELAN	GFLOPs	Parameters	map@0.5/%	map@0.5~0.95/%	Precision	recall
1	√				13.3G	6.03	75.8	50.1	75.6	71.7
2	√	√			**10.1G**	4.33	75.1	49.1	71.7	73.1
3	√		√		13.3G	6.03	76.9	**52.0**	76.5	71
4	√			√	13.3G	6.06	76.4	50.8	**76.9**	71.4
5	√	√	√		10.3G	**4.25**	75.9	50.3	74.7	72.9
6	√	√	√	√	10.5G	4.54	**77.3**	51.8	**76.9**	**73.3**

Analyzing the data in Table 2, Model 1 serves as the baseline model. Model 2 replaces all activation functions with h-swish. Although there is a slight decrease in mAP@0.5, the number of parameters is reduced from 6.03 to 4.33, a decrease of 28.24%. Additionally, GFLOPs are reduced from 13.3G to 10.1G, and Recall shows a slight improvement. This indicates that pcconv can effectively reduce computational complexity and the number of parameters. Model 3 replaces the output Head with pcconv convolution. While the GFLOPs and number of parameters remain unchanged compared to the original model, Precision improves to 76.5%, an increase of 4.8%. Furthermore, mAP@0.5 improves to 76.8%, and mAP@0.5~0.95 increases to 52.0%. This demonstrates that the h-swish activation function can enhance network precision without increasing the number of parameters or computational requirements. Model 4 enhances the ELAN module of the Neck section of the model, maintaining the number of parameters and computational load stable. Both

Precision and Recall experience slight improvements. While DPCH-ELAN has a limited impact on performance enhancement, it helps maintain the model's relatively high-performance level. Model 5 introduces pcconv convolution and h-swish activation function simultaneously. This improves performance relative to the original model, with a significant decrease in the number of parameters and computational load. This suggests that the simultaneous introduction of pcconv and h-swish can reduce model complexity and enhance detection speed. Model 6 implements improvements simultaneously and achieves more significant performance enhancements. Precision reaches 76.9%, while mAP@0.5 is 77.3% and mAP@0.5~0.95is 51.8%. In summary, the improved algorithm reduces the number of parameters by 24.49% and computational load by 20.9% compared to the original model. Furthermore, both mAP@0.5 and mAP@0.5~0.95 improve by 1.5% and 1.7%, respectively.

Analyzing the ablation experiments, from the basic algorithm YOLOv7-Tiny to different module combination experiments, the improved algorithm presented in this paper demonstrates superiority in terms of both model size and detection speed and accuracy. Figure 9 illustrates the comparison of the network's detection performance before and after the improvements.

As evident from the comparison chart in Figure 8, the enhanced algorithm presented in this paper effectively addresses issues such as missed detections and false detections. Moreover, there is an improvement in the confidence level of detections. This demonstrates the practicality and effectiveness of the algorithm proposed in this paper.



a. Original image **b.** Original model detection map **c.** HDP-YOLOv7 detection map

Figure 9. Comparison of detection effect of network before and after improvement

4.5. Comparative experiments

To further demonstrate the effectiveness of the proposed HDP-YOLOv7 algorithm, we trained and tested it alongside several mainstream algorithms on the PASCAL VOC 2012 dataset. The experimental results are summarized in Table 3.

Table 3. Comparative generalization test

Model	GFLOPs	Parameters	map@0.5/%
SSD-bs18	30.8G	24.9	0.760
Faster-RCNN-bs3	134G	41.3	0.778
Yolov5-5s	16.6G	7.11	0.595
Yolov7-tiny	13.3G	6.03	0.628
HDP-YOLOv7	10.6G	4.57	0.639

The experiments demonstrate that HDP-YOLOv7 achieves significant reductions in GFLOPs compared to SSD-bs18, Faster-RCNN-bs3, Yolov5-5s, and Yolov7-tiny, with decreases of 65.6%, 92.1%, 36.2%, and 20.3%, respectively. Similarly, the parameters are reduced by 81.6%, 88.9%, 35.7%, and 31.9%, showcasing substantial improvements in both parameter count and computational complexity. These reductions align more closely with lightweight requirements, further validating the generality and effectiveness of the proposed algorithm.

We attribute these improvements to the reduction of redundant information generated during the feature extraction process and the compression of model size. In realworld complex road scenarios, where numerous overlapping relationships exist, traditional models may suffer from decreased accuracy. However, our proposed HDP-YOLOv7 model effectively addresses the issue of cross-features. For instance, as depicted in Figure 7, HDP-YOLOv7 accurately detects people and bicycles separately, even in cases of overlapping targets such as vehicles and people.

a. Original image **b.** Original model detection map **c.** HDP-YOLOv7 detection map

Figure 10. Comparison of the effect of detecting overlapping targets

5. Conclusion

The HDP-YOLOv7 network model is proposed to address current challenges encountered in complex road scenarios. By introducing the H-swish activation function, we effectively balance performance and computational overhead in scenarios with lightweight design and constrained computational resources. Furthermore, to enhance the feature fusion capability of the algorithm, we improve the ELAN layer in the neck network, thereby enhancing the multilevel extraction of features and improving the generalization performance of the network. Additionally, a pcconv convolutional building block is introduced in the detection head to more accurately capture the features of road targets and road lesions. The improved model achieves significant progress in both accuracy and lightweighting while meeting the requirement for real-time detection, thus achieving an effective balance between accuracy, lightweighting, and real-time performance.

The experimental results demonstrate that the improved model has achieved a reduction of 24.49% in parameters and 20.90% in computation. Moreover, there is an improvement in accuracy, with mAP@0.5 and mAP@0.5~0.9 increasing by 1.3, 1.5, and 1.7, respectively. The performance of the enhanced network in real-time target detection tasks has shown significant enhancement. Its lightweight structure and efficient performance render it suitable for embedded systems and application scenarios with stringent real-time requirements. Future efforts will focus on further optimizing the model, designing an even more efficient and lightweight version, and deploying it on embedded platforms to address challenges in real detection tasks.

References

[1] YUAN Lei, TANG Hai, CHEN Yanrong et al. Improvement of YOLOv5 for road target detection in complex environments[J]. Computer Engineering and Applications, 2023, 59(16): 212-222.

[2] Zhu Youwei. Research on fast road target detection algorithm based on YOLO [D]. Yunnan University, 2022. doi:10.27456/d.cnki.gyndu.2022.002492.

[3] Fan Zhihan. Research and application of road target detection based on YOLO [D]. Sichuan University, 2021. DOI:10.27342/d.cnki.gscdu.2021.000109

[4] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[5] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[6] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[7] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition, June 26-July 1, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 779-788.

[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263 -7271.

[9] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[10] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[11] Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv:2209.02976, 2022.

[12] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision-ECCV 2016: 14th European Conference, Amsterdam , The Netherlands, October 11-14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[13] Tan M, Pang R, Le Q V. Efficientdet: scalable and efficient object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790.

[14] Qiu M, Huang L, Tang B H. ASFF-YOLOv5: Multielement detection method for road traffic in UAV images based on multiscale feature fusion[J]. Remote Sensing, 2022, 14(14): 3498.

[15] Jiang T, Li C, Yang M, et al. An improved YOLOv5s algorithm for object detection with an attention mechanism[J]. Electronics, 2022, 11(16): 2494.

[16] Wang C Y, Liao H Y M, Yeh I H. Designing network design strategies through gradient path analysis[J]. arXiv preprint arXiv:2211.04800, 2022.

[17] Qi Linglong, Gao Jiantai. Small target detection based on improved YOLOv7[J/OL]. (2022-12-08).

[18] QI Xiangming, DONG Xu. Improved Yolov7-tiny algorithm for steel surface defect detection[J]. Computer Engineering and Applications, 2023, 59(12): 176-183.

[19] LIU Haohan, FAN Yiming, HE Huaiqing et al. A lightweight model for target detection with improved YOLOv7-tiny[J]. Computer Engineering and Applications, 2023, 59(14): 166-175.

[20] ZHAO Min, YANG Guoliang, WANG Jixiang et al. Improved real-time helmet detection algorithm for YOLOv7-tiny[J]. Radio Engineering, 2023, 53(08): 1741-1749.

[21] Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv:2209.02976, 2022.

[22] Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. arXiv preprint arXiv:2107.08430, 2021.

[23] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314- 1324.

[24] Lyon M, Armitage P, Álvarez M A. Spatio-Angular Convolutions for Super-resolution in Diffusion MRI[J]. arXiv preprint arXiv:2306.00854, 2023.

[25] Li B, Pan E, Kang Z. PC-Conv: Unifying Homophily and Heterophily with Two-fold Filtering[J]. arXiv preprint arXiv:2312.14438, 2023.