ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

Research on Lightweight Target Detection Algorithm based on YOLOv5s

Wei Shi, Zhaoli Liu, and Haonan Qi

School of Mechanical Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China

Abstract

Aiming at the problems of high model complexity and high consumption of computational resources faced by target detection algorithms when deployed on mobile and embedded devices, this study proposes a lightweight and improved algorithm based on YOLOv5s. By replacing the backbone network of YOLOv5s with MobileNetV3, the multi-scale feature expression capability is retained while reducing the number of parameters. Experiments on the COCO2017 dataset show that the improved model's number of parameters is reduced to 7.9 MB and the detection accuracy (mAP@0.5) is maintained at 84.75%. The algorithm significantly improves the computational efficiency while maintaining the detection accuracy, providing an effective technical solution for real-time target detection in resource-constrained environments.

Keywords

Lightweight Network; Target Detection; YOLOv5s; Deep Learning.

1. Introduction

Target detection is an important topic in the field of computer vision, which aims to find the objects of interest from images and make a judgment on the category and location of the objects, which is beneficial for people to analyze and make decisions in application scenarios[1]. Early target detection algorithms are mainly composed of region selection, manual feature extraction, classifiers and other parts, which have the problems of large computation and poor robustness[2]. With the development of deep learning, target detection algorithms based on convolutional neural networks have replaced the traditional target detection algorithms, and there is a significant improvement in generalization ability and detection effect. Nowadays, target detection technology has been successfully applied in many fields such as urban security, military, automatic driving, medical diagnosis, industrial production, etc[3].

Deep learning based target detection algorithms can be categorized into two-stage target detection algorithms and single-stage target detection algorithms according to the algorithmic process. Two-stage target detection algorithms integrate feature extraction, region suggestion, and bounding box classification and regression into a single network with high detection accuracy, and the typical representative model is the R-CNN[4] series. Single-stage target detection algorithms simplify the processing flow into an end-to-end regression problem, and can generate category probability and location information through a single operation, which greatly improves the detection rate, and at the same time avoids the loss of spatial information caused by region candidates, typical representatives include YOLO (You Only Look Once) and SSD (Single Shot MultiBox. Detector). Therefore, for scenarios requiring fast target detection, one-stage target detection algorithms are usually used.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

2. Algorithm Design

2.1 Convolutional Neural Network

The basic structure of convolutional neural network consists of input layer, convolutional layer, pooling layer, activation function and fully connected layer, and each layer plays a different role in the process of feature extraction and classification, and its structure is shown in Figure 1. The input layer serves as the data entrance and assumes the responsibility of data standardization and format conversion. After receiving the original image, it is converted into a three-dimensional tensor (height × width × number of channels). The convolution layer slides on the input data through a learnable convolution kernel, calculates the dot product summation of the local region and the convolution kernel window by window, combines the padding and step size to control the output size, processes the multi-channel inputs in parallel and superimposes the results, and finally introduces the nonlinearities through the activation function to generate the feature map, which is shown in Fig. 2. The activation function is the core component of the model to realize intelligent feature extraction, and its role is to introduce nonlinear transformations for the network, enabling the neural network to learn and fit complex data patterns[5-7]. By introducing a nonlinear function, the activation function breaks the limitations of linear transformations and enables the network to construct hierarchical feature representations from simple edges to complex semantics layer by layer. The convolutional layers are followed by a pooling layer, which reduces the size of the feature map and reduces the computational effort of the subsequent layers[8]. And the pooling operation plays the role of data enhancement to some extent, because it aggregates the features in the local region, which makes the model more robust to small changes in the input data and reduces the risk of overfitting. The main function of the fully connected layer is to globally synthesize and correlate the local features extracted from the previous convolutional and pooling layers.

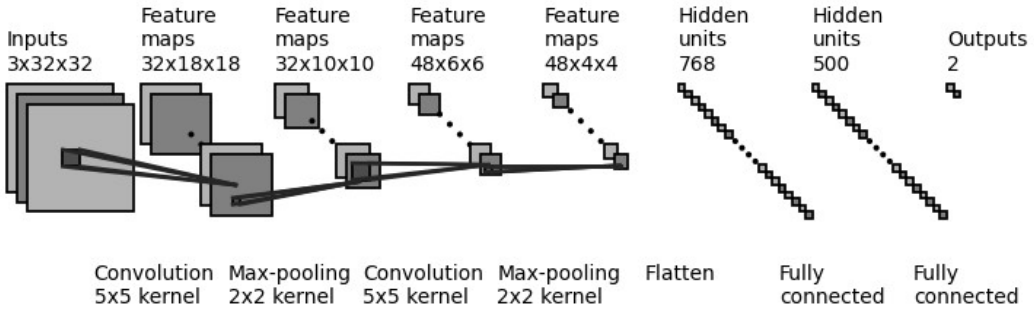

Fig. 1 CNN structure diagram

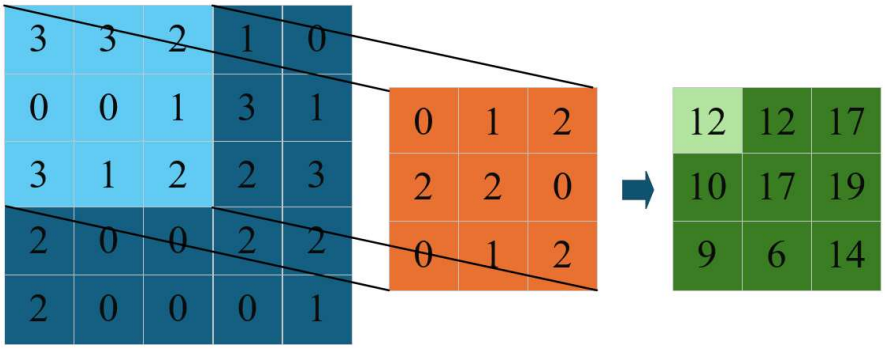

Fig. 2 Convolutional layer computation process

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

2.2 YOLOv5s Model Structure

YOLOv5 is a classic algorithm in the YOLO series, which is a kind of single-stage target detection model and is widely used in target detection instances.Its main features are high efficiency, accuracy, and the ability to realize high-speed inference, which is suitable for various devices with limited computational resources, such as mobile devices, embedded systems, etc.YOLOv5 is subdivided into four different network architectures, which are YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, among which YOLOv5s has a moderate model size among the several versions of YOLOv5, is easy to deploy, and has the most balanced detection speed and accuracy, which is suitable for most real-time scenarios.

YOLOv5s adopts a three-stage detection architecture consisting of Backbone, Neck and Head networks, which is shown in Fig. 3. This design combines data enhancement strategies with loss function optimization mechanisms to effectively improve the detection efficiency of the model. The core component of the architecture, the CSPDarknet53 backbone network, converts the input image into a multi-scale feature map with rich semantic information through hierarchical convolutional operations, laying the foundation for subsequent detection tasks. This backbone network is innovatively improved from the classical Darknet architecture by introducing a cross-stage local connection (CSP) structure, which strengthens the feature reuse capability while reducing computational redundancy. In the feature fusion stage, the neck network adopts a bidirectional feature pyramid architecture to organically integrate the multi-scale features output from the backbone network. This pyramidal feature aggregation approach enables the model to simultaneously capture detailed features and contextual information of targets of different sizes. By adaptive weighted fusion of multiple different scale feature layers, the feature scale conflict problem existing in traditional methods is effectively solved. The final detection head part adopts a multi-task learning mechanism to simultaneously perform target classification, confidence prediction and bounding box coordinate regression based on the optimized fused features.

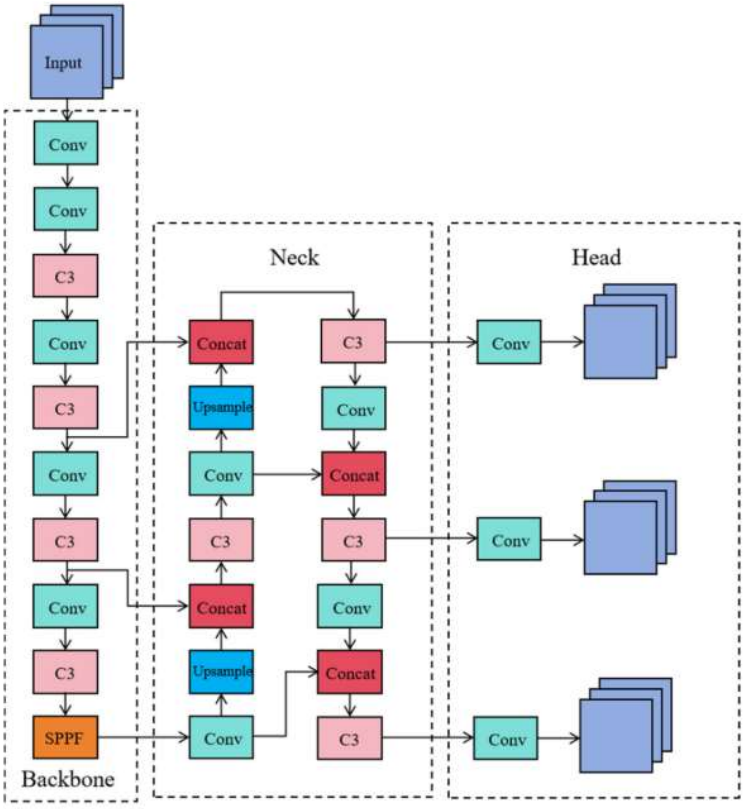

Fig. 3 YOLOv5s network structure diagram

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

2.3 Backbone Network Optimizatio

YOLOv5s, as a typical representative of lightweight detection networks, strikes a good balance between model efficiency and accuracy. The model adopts a streamlined backbone network structure, effectively reduces the number of parameters by controlling the network depth and width, and has faster inference speed compared to other versions. This compact architectural design makes it particularly suitable for deployment on mobile devices with limited computational resources. In this study, YOLOv5s is used as the base architecture for optimization and improvement. While maintaining the original detection accuracy, the model operation efficiency is enhanced by further network lightweighting to ensure that the system can perform real-time target detection. This optimization direction takes into account both the arithmetic limitations of the embedded platform and the practical application requirements in working scenarios.

Under the premise of ensuring that the model detection accuracy will not be significantly reduced, the YOLOv5s algorithm is lightweighted and improved, and the MobileNetV3-small structure is chosen to replace the backbone network of YOLOv5s to complete the feature extraction.MobileNetV3 is a Google-developed lightweight convolutional neural network architecture, optimized for mobile devices, which achieves an excellent balance between computational efficiency and model accuracy. It achieves an excellent balance between computational efficiency and model accuracy. The network incorporates a number of highly efficient designs: the use of depth-separable convolution as the basic unit of operation, by decomposing the standard convolution operation to significantly reduce the computational complexity; the introduction of improved inverted residual structure to enhance the feature expression ability, with cross-layer connections to maintain the flow of information; the integration of Squeeze-and-Excitation mechanism to dynamically adjust the weight of the feature channel; and the use of computationally efficient Hard-switch neural network architecture. The computationally efficient Hard-swish activation function is used to enhance the nonlinear expression capability. These technological innovations enable MobileNetV3 to significantly improve its feature extraction capability while maintaining its lightweight characteristics, making it one of the preferred backbone networks for mobile vision tasks. Its modular design idea also provides an important reference for the development of subsequent lightweight networks.

The core module of MobileNetV3 is shown in Fig. 4, which employs an efficient feature processing flow: firstly, the feature dimensions are extended by 1×1 convolution, followed by spatial feature extraction by applying 3×3 depth-separable convolution, which is a decomposed convolution operation that significantly reduces the parameter size. The features then enter the channel attention module, which first obtains channel-level statistics via global average pooling, and then generates channel weight vectors via a fully connected layer activated by ReLU and Hard-swish. These learned weights are channel multiplied with the original features to achieve feature recalibration. Eventually, feature compression is accomplished by 1×1 convolution, resulting in a streamlined and adequate feature representation. The modular structure of this design strikes a good balance between computational efficiency and feature discriminability.

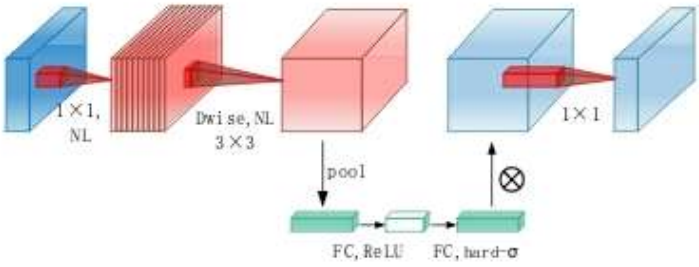

Fig. 4 Block structure diagram of MobileNetV3

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

3. Experimentation and Analysis

3.1 Model Training

3.1.1 Experimental Environment and Datasets

The experimental environment for model training is shown in Table 1. The experiments use the official recommended parameters of YOLOv5, enable adaptive anchor as well as mosaic data enhancement, the input network image size is set to 640×640, the batchsize is set to 32, the test sample batch size is set to 1, the number of model iterations epoch is set to 100, the initialization learning rate is 0.01, and the momentum of the learning rate is 0.937, and the target is optimized using the Stochastic Gradient Descent (SGD) optimization objective. The dataset for model training is selected from the COCO2017 dataset, which contains more than 330,000 images covering 80 classes of common objects, with an average of 7.7 target instances labeled per image.

Table 1. Model training experimental environment

Component Category	Specific Configuration
Operating System	Windows 64-bit
CPU	Intel Core i5-12400
GPU	NVIDIA GeForce RTX 3060
Memory	32GB
Programming Language	Python 3.8
Deep Learning Framework	PyTorch 1.8.2

3.1.2 Evaluation Metrics

Evaluation metrics such as Precision (P, %), Recall (R, %), F1 value, AP (Average Precision, %) value, and mAP (Mean Average Precision, %) value are generally used to evaluate the goodness of the model in target detection, as in Eq. 1 to Eq. 5:

$$P = \frac{TP}{TP + FP} \times 100\% \tag{1}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{2}$$

$$F1 = \frac{2PR}{P + R} \times 100\% \tag{3}$$

$$AP = \int_0^1 P(R)\mathrm{dR} \times 100\% \tag{4}$$

$$\mathrm{m}AP = \frac{\sum_{\mathrm{n}=1}^{\mathrm{n}} AP(\mathrm{n})}{\mathrm{n}} \times 100\% \tag{5}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

Where TP (True Positive) is the target that is detected correctly and FP (False Positive) is the non-target region that is mistakenly detected as a target.FN (False Negative) is the target that is missed. Accuracy indicates the proportion of samples detected as positive cases that are actually positive cases. Recall indicates the proportion of samples that are actually positive cases that are correctly detected as targets. f1 is the reconciled mean of precision and recall, which is used to measure the performance of the model in a combined manner. ap (Average Precision) is the average of the precision calculated based on different levels of recall. It measures the overall performance of the target detection model by considering the area under the precision-recall curve. mAP (Mean Average Precision) is the result of averaging the AP values over all categories. For a multi-category target detection task, mAP is the average of the AP values of all categories.

3.2 Analysis of Results

The backbone network of YOLOv5s model uses convolution for feature extraction and PANet for feature fusion in the neck network, and finally at the HEAD for prediction of the results on three scales: 80× 80, 40× 40, and 20× 20. After training, the training results of the lightweight YOLOv5s model are as follows:

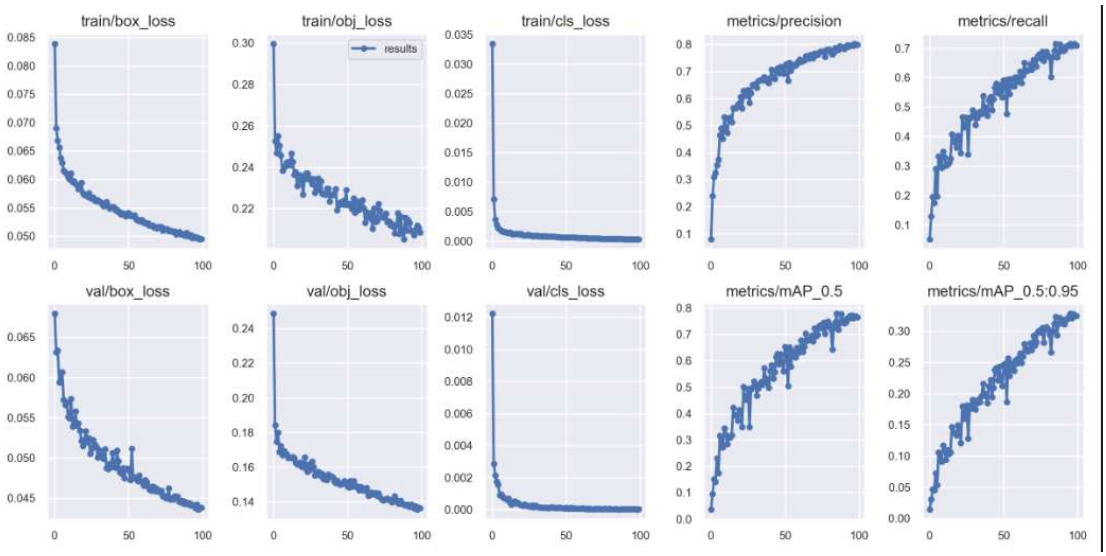

Fig. 5 Lightweight YOLOv5s model training result plot

As analyzed in Fig. 5, in the model training, the bounding box of training and validation, target existence, and classification loss all show a decreasing trend, indicating that the model continues to learn and is not seriously overfitted; the evaluation indexes such as precision rate, recall rate, mAP_0.5, and mAP_0.5:0.95 are steadily improving, indicating that the model detection ability is continuously increasing, and that the training is effective, the convergence is normal, and the generalization ability is better, which can be used in practical target Detection Scenario. The weight files obtained from the training of the lightweight detection model are tested on the test set images, and the following results are obtained and compared with the base yolov5s and the traditional detection models SSD and Faster R-CNN model.

From the analysis in Table 2, it can be seen that the mAP value of the lightweight YOLOv5s model is 84.75%, which is a slight decrease in the detection accuracy compared with the original YOLOv5s. However, due to the smaller computational volume of MobileNetV3, the detection speed reaches 92.3FPS, and the reasoning speed of the model is effectively improved, and the latency can be significantly reduced for the real-time detection task. The model size is reduced by 42%, and the lightweight model is easier to deploy on mobile or embedded devices.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

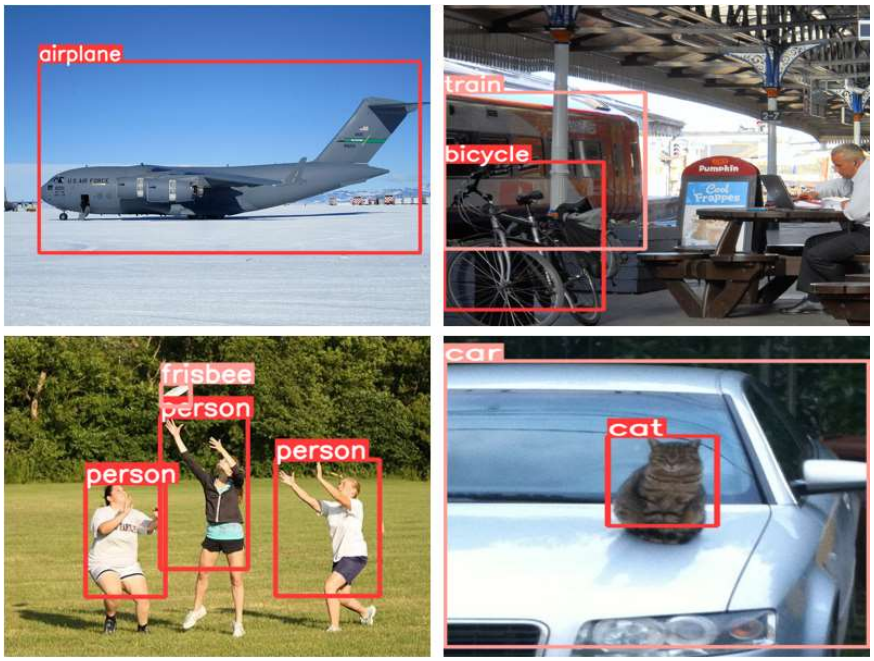

Fig. 6 Target detection results

Table 2. Performance comparison of target detection network models

Model	Model volume/MB	mAP@0.5%	Rate/(FPS)
SSD	91.1	71.2	66.2
Faster R-CNN	108.5	72.9	11.2
YOLOv5s	13.79	85.99	85.5
Lightweight YOLOv5s	7.9	84.75	92.3

4. Conclusion

In order to be better deployed on the host computer and get rid of the problems such as resource constraints, this study has lightweighted the target detection model by replacing the original CSPDarknet53 backbone network structure of YOLOv5 with the lightweight MobileNetv3 network structure. Experiments show that the deep separable convolution of the MobileNetV3 backbone with neural network architecture search characteristics significantly improves the feature extraction efficiency.

References

[1] Zheng X ,Fang X ,Lan K , et al.Improved Intelligent Learning Filter in Deep Learning Systems and Its Application in Traffic Object Detection[J].Traitement du Signal,2024,41(6):

[2] Dahui L ,Qi F ,Jianzhao C , et al.Research on Target Detection Algorithm of Radar and Visible Image Fusion Based on Wavelet Transform[J].Tehnički vjesnik,2020,27(5):1563-1570.

[3] Zhongqin B ,Lina J ,Chao S , et al.Transmission line abnormal target detection algorithm based on improved YOLOX[J].Multimedia Tools and Applications,2023,83(18):53263-53278.

[4] Limei S ,Jiawei K ,Qile Z , et al.A weld feature points detection method based on improved YOLO for welding robots in strong noise environment[J].Signal, Image and Video Processing,2022,17(5):1801-1809.

[5] Chenchen J ,Huazhong R ,Xin Y , et al.Object detection from UAV thermal infrared images and videos using YOLO models[J].International Journal of Applied Earth Observation and Geoinformation,2022,112

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0026

[6] Tao T ,Wei X .STBNA-YOLOv5: An Improved YOLOv5 Network for Weed Detection in Rapeseed Field[J].Agriculture,2024,15(1):22-22.

[7] Keumsun P ,Minah C ,Hyuk J C .Image Pre-Processing Method of Machine Learning for Edge Detection with Image Signal Processor Enhancement[J].Micromachines,2021,12(1):73-73.

[8] YiJia Z ,FuSu X ,ZheMing L .Helmet Wearing State Detection Based on Improved Yolov5s[J].Sensors, 2022,22(24):9843-9843.