A Lightweight Target Detection Algorithm Based on Improved MobileNetv3-YOLOv3

Tong Fang, Baoshuai Du, Yunjia Xue, Guang Yang, and Jingbo Zhao(✉)

School of Information and Control Engineering, Qingdao University of Technology, Qingdao, Shandong, China
zhaoyancheng2021@163.com

Abstract. To solve the problems of complex model structure, large number of parameters, and high resource consumption that make it difficult to meet the real-time requirements of embedded target detection tasks, this paper proposed a lightweight target detection algorithm based on improved MobileNetv3-YOLOv3. This algorithm uses MobileNetv3 network to replace the backbone of the original YOLOv3 network, and the reduction of network parameters greatly improves the detection speed of the algorithm; the loss function is modified to CIoU to improve the accuracy and detection speed of the network. The experimental results showed that the improved lightweight detection algorithm on the VOC07 + 12 dataset has a 1.55% improvement in mAP and a 2.47 times improvement in FPS on CPU compared to the original YOLOv3 algorithm. This improved algorithm ensures the detection accuracy based on a significant increase in detection speed, which reflects the theoretical and application value of the research.

Keywords: MobileNetv3 · Object detection · YOLOv3 · Lightweight target detection algorithm · CIoU

1 Introduction

Currently SLAM (simultaneous localization and map building) algorithms have an important position in robot motion estimation and map building applications. Semantic perception of unknown environments by mobile robots is a frontier of current research in robotics and computer vision, and target detection can be used to achieve the perception of semantic information in the environment.

With the development of deep learning and the improvement of GPU computing power, deep learning based target detection algorithms have become mainstream [1]. The current target detection algorithms can be broadly divided into two categories with the development of deep learning [2]: Two-stage target detection algorithms and one-stage target detection algorithms [3, 4]. Two-stage target detection algorithms are far superior to traditional detection algorithms in terms of accuracy, but it is difficult to apply to mobile devices such as mobile robots with poor computing power due to their large computing power. The regression-based One-stage target detection algorithm, on the other hand, can achieve real-time operation with little loss of accuracy, and thus is

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13370, pp. 438–448, 2022.
https://doi.org/10.1007/978-3-031-10989-8_35

widely used in mobile. Also deep learning based target detection is widely used in urban traffic flow monitoring [5], intelligent fault diagnosis [6] and other fields.

The classical YOLOv3 has a better detection effect [8, 9], but its complex model and the number of ten million parameters have great drawbacks in both debugging, training and deployment stages. Considering the limited computing power of mobile platforms, this paper replaces the feature extraction network Darknet53 of YOLOv3 with MobileNetv3 to reduce the number of parameters of the network and modifies the regression loss function of the grasping frame with the CIoU method. In this way, the speed of the algorithm in mobile platform detection is improved on the basis of ensuring the detection accuracy. Finally, the effectiveness of the proposed model is verified on the VOC07 + 12 dataset.

The paper is organized as follows. Section 2 describes the innovative nature of the algorithms in this paper and introduces the improvement of the backbone network part and the improvement of the loss function based on CIoU. Section 3 presents the ablation experiments, which demonstrate the effectiveness of the proposed algorithm and understand the contributions of different elements. Section 4 contains conclusions and future work.

2 Algorithm of this Paper

2.1 Lightweight Feature Extraction Network Improvement Based on MobileNetv3

To be able to achieve low latency high frequency real-time target detection on an embedded platform with limited computing power, this paper proposes a backbone feature extraction network using the lightweight network MobileNetv3 to replace YOLOv3 [10].

In this paper, MobileNetv3-large is used. This version combines MobileNetv1's Deep Separable Convolution, MobileNetv2's Inverted Residuals and Linear Bottleneck, and SE modules [11], uses neural structure search to search the configuration and parameters of the network. Although the number of parameters is increased in the large version compared to MobileNetv3-small, the small increase in the number of parameters can be exchanged for an increase in the detection accuracy, which guarantees the detection accuracy for the lightweight detection algorithm proposed in this paper (Fig. 1).

In MobileNetv3, the number of parameters can be greatly reduced by using depth-separable convolution instead of normal convolution.

In the model where the input feature map size is $D_I \times D_I$ and the number of channels is M. When a convolution kernel of size $D_K \times D_K$ is used to output the feature map of size $D_I \times D_I$ and the number of channels is N, the number of parameters P_C and the amount of computation C_C required for one ordinary convolution are shown in Eqs. (1) and (2).

$$P_C = D_K \times D_K \times M \times N \tag{1}$$

$$C_C = D_K \times D_K \times M \times N \times D_I \times D_I \tag{2}$$

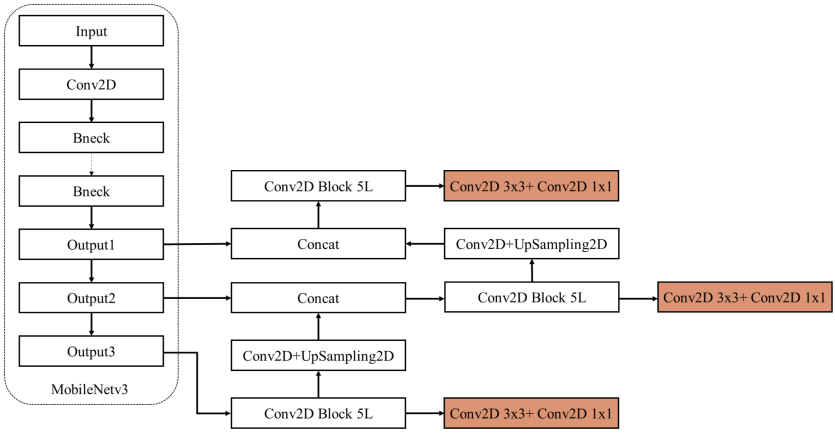

Fig. 1. Network structure diagram after replacing the backbone network.

And in the depth-separable convolution, the number of parameters P_W and the amount of computation C_W required are shown in Eqs. (3) and (4).

$$P_W = D_K \times D_K \times M + M \times N \tag{3}$$

$$C_W = D_K \times D_K \times M \times D_I \times D_I + M \times N \times D_I \times D_I \tag{4}$$

So for the same feature map and convolution kernel, the ratio of the number of parameters to the amount of computation required for a depth-separable convolution versus an ordinary convolution is.

$$P = \frac{D_K \times D_K \times M + M \times N}{D_K \times D_K \times M \times N} = \frac{1}{N} + \frac{1}{D_K^2} \tag{5}$$

$$C = \frac{D_K \times D_K \times M \times D_I \times D_I + M \times N \times D_I \times D_I}{D_K \times D_K \times M \times N \times D_I \times D_I} = \frac{1}{N} + \frac{1}{D_K^2} \tag{6}$$

It can be seen that the use of depth-separable convolution instead of normal convolution can reduce a significant portion of the number of parameters and computation, which lays the foundation for the implementation of porting the target detection algorithm to the mobile platform side where the arithmetic power is much less.

The special block of MobileNetv3 introduces the inverse residual structure with linear bottleneck. The residual structure can significantly improve the training effect of the network without adding additional parameters and with only less computation, and the inverse residual mechanism is to first use 1×1 convolution in the residual block to boost the number of channels before subsequent operations with residual edges; and the weight of each channel is adjusted by introducing a lightweight attention mechanism (Fig. 2).

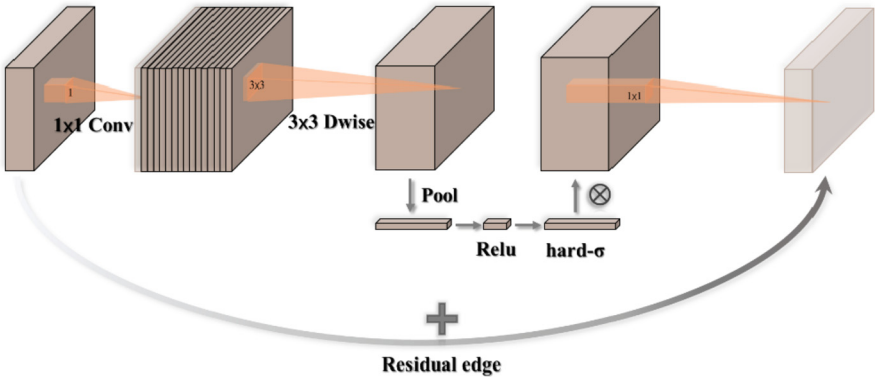

Fig. 2. MobileNetv3 block.

In the use of activation function, swish function can effectively improve the network accuracy, but the computation is too large and time consuming, especially in the mobile end of the time consuming embodiment will be more obvious, which is not conducive to the algorithm to reduce the detection time.

$$swish\, x = x \cdot \sigma(x) \tag{7}$$

where $\sigma(x)$ is the Sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{8}$$

So ReLU6(x + 3)/6 is used to approximate the replacement of the sigmoid function in order to achieve a fast computation that can be performed on any hardware or software platform. The algorithm uses the h-swish activation function instead of the original swish function.

$$h - swish[x] = s\frac{ReLU6(x+6)}{6} \tag{9}$$

2.2 CIoU-Based Loss Function Improvement

In YOLOv3, the performance of target detection is evaluated by IoU. IoU is defined as the intersection and merging ratio between the true frame and the predicted frame, as shown in Eq. (10). Where P is the prediction frame and R is the real frame.

$$IoU = \frac{P \cap R}{P \cup R} \tag{10}$$

However, there is a problem with using IoU, when there is no intersection between the prediction frame and the real frame, IoU is 0, and there is no gradient nor can the parameters be updated. To solve this problem, this algorithm uses the CIoU function

to replace IoU as the loss function of the enclosing frame [12]. The CIoU function is calculated as follows.

$$CIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{11}$$

where:

$$\alpha = \frac{v}{1 - IoU + v} \tag{12}$$

$$v = \frac{4}{\rho^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{13}$$

where $\rho^2(b, b^{gt})$ represents the Euclidean distance d between the center points of the prediction frame and the real frame, c represents the diagonal distance of the smallest closed region that can contain both the prediction frame and the real frame, α and v are penalty factors, and w^{gt}, h^{gt} and w, h are the width and height of the real frame and the prediction frame, respectively (Fig. 3).

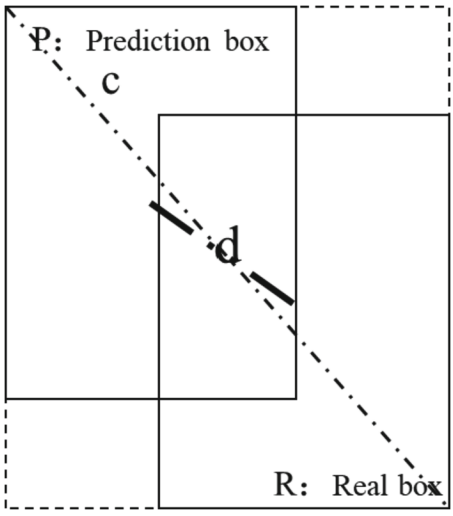

Fig. 3. Geometric relationship between the prediction box and the real box.

3 Experimental Results and Analysis

3.1 Experimental Environment and Model Training

The experiments in this paper are based on the Pytorch 1.7.1 framework, the programming language is Python 3.8, the experimental OS is Windows 10, the processor is

Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz, the GPU model is NVIDIA GeForce RTX 2060, the CUDA version is 11.0, and Cudnn version 8.0.5.39.

In the selection of dataset, this paper uses the classical dataset of target detection, VOC07 + 12. The VOC07 + 12 dataset is divided into a total of 4 major categories and 20 minor categories, with 21504 labeled images. The dataset is divided according to the ratio of 9:1, with 19354 images as the training set and 2150 images as the test set. The experiments use the average precision (AP) to respond to the detection results of targets in each category, use the mean average precision (mAP) as a measure of detection accuracy, the higher the mAP, the better the comprehensive performance of the model in all categories, and use the time consumed to detect each image as a measure of detection speed.

In order to verify the effect of the proposed algorithm in this paper, influenced by the arithmetic power, during the training process, only the pre-training weights of the backbone network are loaded, the optimizer for training is chosen as Adam, the initial learning rate is 1e−3, and the learning rate decay strategy is cosine annealing. In order to improve the robustness of the proposed algorithm, Mosaic data augmentation [13] was performed on the first 70% of the training set during the algorithm training process, and the specific implementation idea is as follows: firstly, four images are read at a time; then these four images are scaled, rotated, and color-field transformed, respectively, and placed well according to the four directions; finally, the combination of images and the combination of frames is performed.

3.2 MobileNetv3 Detection Effect After Replacing Backbone

The original YOLOv3 algorithm is set to A, and the algorithm after replacing backbone with MobileNetv3 is set to B. The experimental results of the two algorithms in terms of accuracy and speed are shown in Table 1.

Table 1. Comparison of detection effect after replacing backbone.

Detection algorithm	Training set	mAP/%	FPS (GPU)	FPS (CPU)
A	VOC07 + 12	67.56	27.24	2.10
B	VOC07 + 12	63.78	22.78	4.97

Compared with the original YOLOv3 network, the detection accuracy decreases after using MobileNetv3 to replace backbone, and the detection speed on CPU is greatly improved because the number of parameters is greatly reduced after replacing the backbone feature extraction network with MobileNetv3, which can reduce the time needed for algorithm detection, but the reduction in the number of parameters also makes the network less effective and the detection accuracy decreases.

3.3 Improved Detection Effect of CIoU-Based Loss Function

The original YOLOv3 algorithm is set as A, and the improved algorithm based on the loss function of CIoU is set as B. The experimental results of the two algorithms in terms of accuracy and speed are shown in Table 2.

Table 2. Comparison of detection results after using CIoU loss function.

Detection algorithm	Training set	mAP/%	FPS (GPU)	FPS (CPU)
A	VOC07 + 12	67.56	27.24	2.10
B	VOC07 + 12	69.50	27.67	2.35

Compared with the original YOLOv3 network, the detection effect of the network is improved after modifying the loss function, and the detection time is also slightly accelerated. The experiment proves that by improving the loss function of the original YOLOv3 algorithm, it can be more beneficial for the model to achieve better results.

3.4 Analysis of Experimental Results of Light-Weight Target Detection Algorithm

In this paper, we compare the detection effect of the improved YOLOv3 algorithm with the original YOLOv3 algorithm. It is shown in Fig. 4.

(a) YOLOv3 detection effect

(b) Improved YOLOv3 detection effect

Fig. 4. Comparison of detection effects on VOC07 + 12 dataset.

The improved algorithm in this paper was compared with the original YOLOv3 algorithm and the current mainstream Two-stage detection algorithm Faster-RCNN in terms of performance. The results of mAP and AP comparison for various types of targets on the test set are shown in Fig. 5 and Table 3.

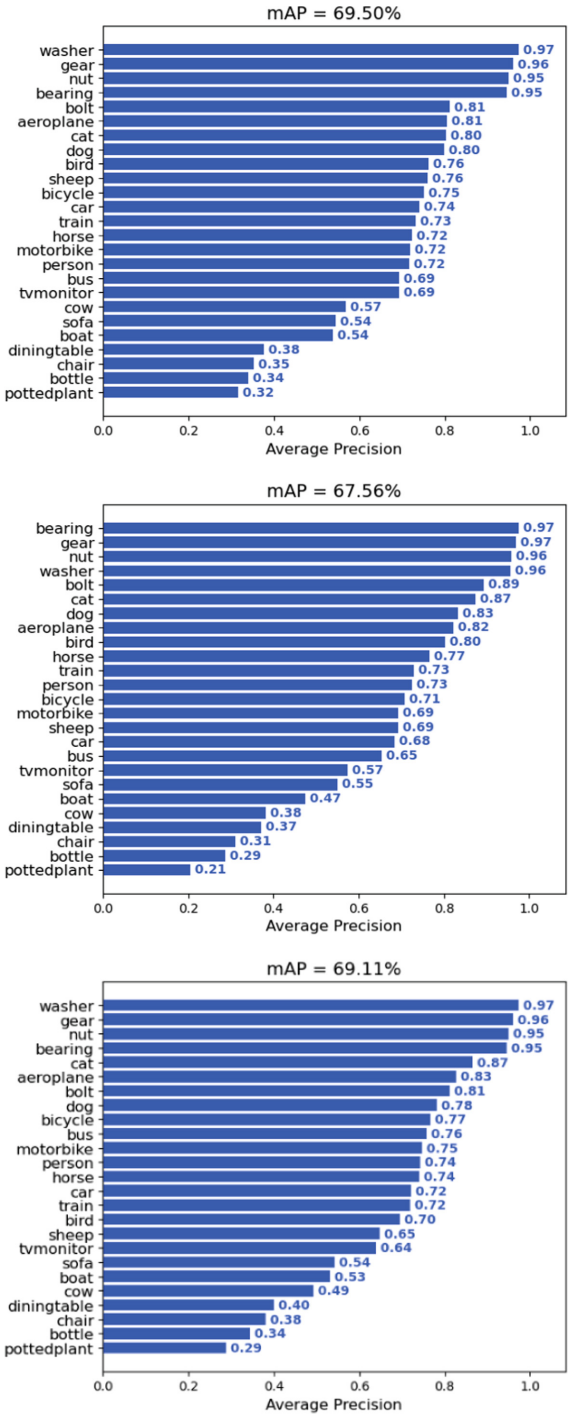

Fig. 5. Comparison of AP and mAP of different algorithms on VOC07 + 12 dataset.

Table 3. Comparison of detection results on the VOC07 + 12 dataset.

Detection algorithm	Training set	mAP/%	FPS (GPU)	FPS (CPU)
Faster-RCNN	VOC07 + 12	69.50	17.24	0.93
YOLOv3	VOC07 + 12	67.56	27.24	2.10
Algorithm of this paper	VOC07 + 12	69.11	23.93	5.24

The experimental results show that compared to the original YOLOv3 network, the detection speed of the algorithm in this paper is 2.47 times higher on the CPU than the original network, and the detection accuracy is slightly improved while the detection speed is guaranteed to increase, and the mAP is improved by 1.55%, compared with the Faster-RCNN network, the accuracy is slightly reduced, but the detection speed is improved by 5.63 times. The comparison also shows that the improvement of CIoU works better for the network after replacing backbone.

Table 4. Comparison of the number of parameters of the two algorithms.

Detection algorithm	Number of participants	Model file size/M
YOLOv3	61,949,149	236.32
Algorithm of this paper	23,608,237	90.06

As can be seen in Table 3, the improved network does not improve the detection speed on GPUs with sufficient arithmetic power, but has a slight decrease, which is due to the general optimization of the depth-separable convolution in the MobileNetv3 network on GPUs. Although the depth-separable convolution splits a standard convolution into two convolutions, reducing the number of parameters, as shown in Table 4, the number of parameters is reduced by nearly 2/3 compared to the original YOLOv3 network. The CPU tends to compute data serially. Therefore, if the GPU memory is large enough, because each layer can be processed in parallel at once, the total computing time is dominated by the number of layers of the network. For CPUs lacking parallelism, a significant reduction in the number of parameters, the dominant factor in computing time, results in a much higher detection speed.

The proposed algorithm is effective on CPU, which also verifies that this algorithm can achieve fast and accurate target detection on mobile platforms with poor arithmetic power.

4 Conclusion

In order to improve the detection speed of the target detection algorithm on embedded devices with poor arithmetic power while ensuring accuracy, this paper proposed a lightweight target detection algorithm based on improved MobileNetv3-YOLOv3, which

greatly reduced the number of parameters in the network by introducing MobileNetv3 as the backbone feature extraction network of the algorithm, and the loss function is also improved based on CIoU The loss function is also improved based on CIoU. Compared with the original YOLOv3 network, the mAP of the algorithm in this paper is improved by 1.55%, and the detection speed on CPU is improved by 2.47 times. The algorithm proposed in this paper is able to meet the task of real-time accurate target detection by embedded devices using limited on-board processor computing resources in terms of comprehensive evaluation of detection accuracy and detection speed.

In the subsequent research work, we will continue to compress the model, including channel pruning and other operations to continue to reduce the parameters, and further improve the detection speed of the model on the mobile device side with lower performance on the basis of ensuring the detection accuracy, the proposed algorithm will also be applied to more areas to prove the potential impact of the algorithm.

Acknowledgment. This work is supported in part by the National Natural Science Foundation of China under Grant 51475251, the Natural Science Foundation of Shandong Province under Grant ZR2013FM014 and in part by the Qingdao Municipality Livelihood Plan Project under Grant 22-3-7-xdny-18-nsh.

References

1. Lin, T.Y., Dollar, P., Girshick, R., et al.: Feature pyramid networks for object detection. In: IEEE Conference on Computer vision and Pattern Recognition (CVPR), pp. 936–944. Las Vegas, USA (2017)
2. Qin, P., Tang, C.M., Liu, Y.F., et al.: Infrared target detection method based on improved YOLOv3. Comput. Eng. **48**(3), 211–219 (2022)
3. Li, G.J., Hu, J., Ai, J.Y.: Vehicle detection based on improved SSD algorithm. Comput. Eng. **48**(1), 266–274 (2022)
4. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. In: IEEE Conference on Computer vision and Pattern Recognition (CVPR), pp. 779–788. Las Vegas, USA (2016)
5. Msahli, M., Qiu, H., Zheng, Q., et al.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE Trans. Intell. Transp. Syst. **22**(7), 4560–4569 (2021)
6. Li, Y., Song, Y., Jia, L., et al.: Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. IEEE Trans. Industr. Inf. **17**(4), 2833–2841 (2020)
7. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525. New York, USA (2017)
8. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. arXiv e-prints, (2018)
9. Zhao, Z.Q., Zheng, P., Xu, S.T., et al.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)
10. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
11. Howard, A., Sandler, M., Chen, B., et al.: Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314–1324. Seoul, Korea (2020)

12. Zheng, Z., Wang, P., Ren, D., et al.: Enhancing geometric factors in model learning and inference for object detection and instance segmentation, In: 9th International Proceedings on Proceedings, IEEE Transactions on Cybernetics, pp. 1–13 (2020)
13. Bochkovskiy, A., Wang, C.Y., Liao, H. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934 (2020)