WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 3, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n3.36
Journal homepage: https://wepub.org/index.php/IJCSIT/index

A Review of the Lightweight Technology of Object Detection Algorithms

Zexi Tan

Nanjing University of Information Science and Technology, Nanjing, Jiangsu, 210044, China
3329804566@qq.com

ABSTRACT

Deep learning has made significant progress in the field of object detection, especially convolutional neural networks have performed well in image classification, object detection, and segmentation tasks. However, with the increasing complexity of models and the demand for computing resources, traditional deep learning models face challenges in the deployment of resource-constrained mobile and embedded devices. In order to solve this problem, model compression and acceleration techniques have become a research hotspot, including pruning, quantification and knowledge distillation. The purpose of this paper is to review various algorithms in the field of object detection and their advantages and disadvantages, and to discuss the best optimization scheme based on the application and optimization effect of lightweight technology in various algorithms. The research objectives include: systematically summarizing and analyzing the main lightweight technologies currently used for object detection algorithms, evaluating their practical effects in object detection tasks, proposing improvement schemes suitable for specific application scenarios, and looking forward to the future development direction, and discussing potential research directions and technological breakthroughs.

KEYWORDS

Deep Learning; Object detection algorithms; Lightweight technology

1. INTRODUCTION

Object detection is of great significance in multiple application scenarios, including autonomous driving, intelligent surveillance, facial recognition, medical image analysis, etc. Object detection algorithms have made significant progress based on deep learning.

While deep learning models excel in a variety of vision tasks, their high computational costs and storage requirements limit their deployment on resource-constrained devices. Therefore, model compression and acceleration technology has become a research hotspot. These techniques, including Pruning, Quantization, and Knowledge Distillation, make it possible to run efficient deep learning models on mobile devices and embedded systems by reducing model complexity and resource requirements.

In July 2022, Li Kequan et al. detailed the origin and development of deep learning based object detection algorithms [1]. In addition, Ning Xin et al. introduced common lightweight optimization methods in a review of joint optimization methods for neural network compression published in January 2024 [2]. These studies provide a solid foundation for further exploration of object detection algorithms and lightweight technologies.

In this context, this paper aims to review the development status of object detection algorithms and their lightweight technologies, and evaluate the practical effects of lightweight technologies in object detection tasks. This promotes research and application in this field. For example, by studying lightweight technology, the computing and storage requirements of object detection algorithms on mobile and embedded devices can be significantly reduced. In addition, the application of lightweight technology can help improve the inference speed of object detection algorithms to meet the needs of real-time applications, such as autonomous driving and real-time video surveillance. At the same time, model lightweight can significantly reduce energy consumption and cost, which is of great significance for large-scale deployment and application.

The research scope of this paper covers common object detection algorithms and their lightweight technologies. First of all, the current common and popular object detection algorithms and lightweight technologies will be introduced, so that readers can have a clear understanding of these basic concepts. Then, according to the literature and experimental data, the influence of different lightweight technologies on various object detection algorithms and their advantages and disadvantages are discussed. Finally, this paper will summarize the lightweight technologies suitable for different object detection algorithms, so as to help scholars who are new to the field of deep learning to understand and apply these technologies more effectively, so as to improve the research efficiency and model effect.

Through a comprehensive review of object detection algorithms and their lightweight technologies, this paper hopes to provide a valuable reference for research in this field and promote the efficient application of deep learning technology in resource-constrained environments.

2. SURVEY OF OBJECT DETECTION ALGORITHM AND RESEARCH ON LIGHTWEIGHT TECHNOLOGY

2.1. Object Detection Algorithms

An object detection algorithm is a technology used in computer vision that aims to identify and locate multiple objects in an image or video. This requires an algorithm to not only identify the kind of object in the image, but also determine the specific position of each object in the image, usually represented by a bounding box. Unlike traditional object detection algorithms, which rely on manual feature selection, deep learning-based object detection algorithms improve recognition performance through automatic feature learning. The two-stage object detection algorithm first generates region proposals, and then uses convolutional neural networks (CNNs) to classify these regions, includig R-CNN [3], Mask R-CNN [5], SPPNet [6], Fast R-CNN [7], and Faster R-CNN [8]. This type of algorithm usually has excellent accuracy, but high computational complexity.

With higher efficiency, the single-stage object detection algorithm directly locates and classifies through deep convolutional neural network (DCNN), includig the YOLO [4] series, SSD [9], DSSD [10], and FSSD [11].

2.2. Lightweight Technology

At present, the mainstream lightweight technology is divided into two ways: redesigning the neural network part and compressing the deep learning model. This article mainly discusses model compression techniques. Common model compression algorithms include pruning, quantization, and knowledge distillation.

(1) Pruning: By removing redundant neurons or connections, thereby improving computational efficiency and reducing storage requirements.

(2) Quantization: Reduce model parameters and calculations to reduce the consumption of model volume and computing resources.

(3) Knowledge Distillation: Training a compact "student" model to emulate the performance of a larger, more complex "teacher" model.

These techniques are designed to make deep learning models more efficient and adaptable to resource-constrained environments without significantly degrading model performance.

2.3. Dual-stage Object Detection Algorithm

Deep CNNs excel in computer vision tasks, but their high compute and storage requirements limit their application on resource-constrained devices. Existing model compression methods, such as weight pruning and knowledge distillation, are effective in some cases, but face challenges when dealing with complex architectures such as ResNet. Ref. [12] proposes a network compression method combining weighted pruning and knowledge distillation, which effectively addresses this problem.

In this method, the ResNet model is pruned based on activation analysis, and only operates at specific layers to avoid destroying the integrity of the network structure. Specifically, neurons and connections that contribute less to the predicted output are removed by calculating the zero activation rate (APoZ) of each neuron. The pruned model serves as a network of teachers, and then the knowledge is transferred to a smaller network of students through knowledge distillation.

A new distillation loss function optimizes models by minimizing cosine similarity between teacher and student networks in deep feature and prediction layers. Applied to CIFAR-10, the method achieved: ResNet-110's parameters reduced from 1.74M to 0.37M with accuracy dropping from 94.27% to 93.0%, and ResNet-164's parameters reduced from 2.62M to 0.72M with accuracy decreasing from 94.52% to 93.7% (Ref. [12]).

This network compression method combining weighted pruning and knowledge distillation not only significantly reduces the number of model parameters, but also maintains high performance in terms of accuracy.

Table 1. Compression results on Cifar10

Model	Accuracy	#Params	CR Rate
ResNet-110 (Baseline)	94.27	1.74M	-
ResNet-110 (Teacher)	94.04	0.75M	2.32
ResNet-110 (Student-Distilled)	93.0	0.37M	4.7
ResNet-110 (student-Scratch)	90.0	0.37M	4.7
ResNet-164 (Baseline)	94.52	2.62M	
ResNet-164 (Teacher)	94.30	1.44M	1.81
ResNet-164 (Student-Distilled)	93.7	0.72M	3.63
ResNet-164 (Student-Scratch)	89.6	0.72M	3.63

In the application of quantization and pruning strategies, Mask R-CNN, Fast R-CNN, Faster R-CNN and R-CNN can use the same basic strategy, but due to the different architecture and characteristics of each algorithm, the specific implementation details and effects may be different.

1) R-CNN

R-CNN uses selective search to generate candidate regions, extracts feature with CNNs, and classifies them with SVMs. To cut computational and storage costs, quantization and pruning can be applied to the convolutional and fully connected layers.

2) Fast R-CNN

Fast R-CNN advances R-CNN by using a single network for both candidate region generation and classification. It applies CNNs to the entire image to create feature maps and uses an RoI pooling

layer to extract fixed-size features. Quantization and pruning, especially in fully connected layers post-RoI pooling, effectively reduce computational needs.

3) Faster R-CNN

Based on Fast R-CNN, Faster R-CNN introduces the Region Proposal Network (RPN) to generate candidate regions and share convolutional feature maps with Fast R-CNN to achieve end-to-end object detection. Quantization and pruning can be applied to convolutional layers, RPNs, and subsequent classification regression layers to reduce computational complexity and storage requirements. Since RPN and Fast R-CNN share feature maps in the end-to-end architecture of Faster R-CNN, quantization and pruning can be applied to both parts at the same time, improving overall performance and efficiency.

4) Mask R-CNN

Mask R-CNN adds a branch on the basis of Faster R-CNN to predict the pixel-level mask of the target and implement instance segmentation. Quantization and pruning can be applied to the entire network, including the RPN, Fast R-CNN sections, and the newly added mask prediction branch. Specifically, the convolutional layer, RPN, classification regression layer, and mask prediction layer can be pruned. However, due to the addition of mask prediction branches, special attention needs to be paid to the pruning and quantization of this part to ensure that the accuracy of instance segmentation does not decrease significantly.

5) SPPNet

Since SPPNet mainly performs spatial pooling operations in the last layer of convolutional neural networks, its convolutional layer is similar to that of traditional CNNs, so similar quantization and pruning strategies can be employed

2.4. Single-stage Object Detection Algorithm

2.4.1. YOLO series

The YOLO (You Only Look Once) series is a deep learning -based object detection with real-time object detection capabilities. Its main advantages include high detection speed and high detection accuracy, The YOLO model significantly reduces the computational overhead by predicting the class and location of the target simultaneously in a forward propagation. The disadvantage is that the robustness in complex backgrounds is poor. In addition, the YOLO series models may experience performance bottlenecks when processing high-resolution images. Overall, the YOLO series, with its balanced speed and accuracy, is the first choice for many real-time object detection tasks.

The pruning and quantification methods of the YOLOv5 model were evaluated and summarized in Ref. [13], and the practical application effect of the YOLOv5 model was analyzed.

Among them, structured pruning includes channel pruning, filter pruning, and kernel pruning, among others, to preserve the structural integrity of the model. Unstructured pruning removes individual weight parameters. (The experimental results of table2 are from Ref. [13])

1) Channel pruning: There are many methods to apply channel pruning on YOLOv5, and the results show that the model parameters and computational requirements can be significantly reduced through pruning. For example, pruning with the Batch Normalization Scaling Factor (BNSF) method can reduce parameters by more than 50% on different datasets, while resulting in only a small drop in accuracy.

2) Filter pruning: Using the filter pruning method, the computational complexity can be further reduced by pruning the filters that contribute less to the output.

3) Nuclear pruning: The nuclear pruning method reduces the computational burden by pruning the convolution kernel and maintains the overall performance of the model.

Table 2. Experimental results of pruning technology on YOLOv5

Paper	V	Task	Dataset	Saliency	Granul	↻/⊙	ΔAcc	ΔParam	ΔSize	ΔFLOPs	ΔFPS/Δ⏲
Unstructured Pruning											
[51]	5s	Object Detection	—	SPDY	unstructured	↻	-0.5	—	—	—	50/—
[51]	5m	Object Detection	—	SPDY	unstructured	↻	-1.7	—	—	—	75/—
Channel-Based Pruning											
[52]	5s	Position Detection	Manual	BNSF	channel	↻	-2.3	-45	-48	-55	—/-40
[53]	5n	Fault Detection	Manual	BNSF	channel	↻	-4.8	-72.2	—	—	—/-29.4
[54]	5	Target Detection	Military Aircraft Detection	BNSF	channel	↻	-4.5	—	—	—	560/—
[55]	5s	Fruitlet Detection	Manual	BNSF	channel	↻	0(F1)	-92	-90.5	—	—/-13
[56]	5	Outdoor Obstacles Detection	OBSTACLE	BNSF	channel	↻	-0.4	—	-59.1	—	—/-43.6
[57]	5s	Garlic Detection	Manual	BNSF	channel	↻	-0.1	—	-17	—	—/-1
[58]	5s	Wheat Grain Quality Detection	Manual	—	channel	↻	1	—	—	-86.7	—/—
[59]	5x	Ship Detection	Manual	BNSF	channel	↻	2.3	—	-48.9	—	61.4/—
[60]	5s	Flame Detection	Manual	BNSF	channel	↻	0	-37.8	-37.5	-54.5	10/—
[61]	5s	Satellite Components Recognition	Manual	BNSF	channel	↻	-1.2	—	-66.2	—	—/—
[62]	5s	Helmet-Wearing Detection	—	BNSF	channel	↻	-0.9	-87.3	-86.2	-87.4	53.5/—
[63]	5l	Object Detection	COCO	CDSC-BNSF	channel	↻	-0.9	—	-63.8	-37.4	—/—
[64]	5s	Sewer Defect Detection	Manual	BNSF	channel	↻	-0.5	-81	-79.3	-48.8	—/-34.2
[65]	5s	Tracking and Counting Grape Clusters	Manual	BNSF	channel	↻	-0.2	-73.3	-76.4	-57.6	—/-10.3
[66]	5	Fiber Defect Detection	PASCAL VOC	BNSF	channel	↻	-0.3	—	-29.5	—	—/-18.7
[67]	5s	Pedestrian Detection	VisDrone	BNSF	channel	↻	-0.4	—	-26.3	-11.9	—/-9.3
[68]	5	Blade Defect Detection	Manual	BNSF	channel	↻	7.8	—	-19.6	—	28.3/—
[69]	5l	Object Detection	PASCAL VOC	CRLST	channel	↻	-2.9	-49.2	—	-46.2	—/-35.5
[69]	5l	Object Detection	COCO	CRLST	channel	↻	-3.8	-48.8	—	-46.5	—/-29.4
[69]	5l	Object Detection	VisDrone	CRLST	channel	↻	-1.1	-51.2	—	-46	—/-35.4
[70]	5s	Pedestrian Detection	Manual	BNSF	channel	↻	3.78	-52.3	-51.8	-40.8	57.8/—
[71]	5s	Aerial Image Object Detection	DOTA	BNSF	channel	⊙	-6.46	-58.8	-57.6	-37	—/-17.3
[72]	5l	Object Detection	Manual	NSGA-II	channel	↻	-0.9	-73.9	-73.5	-62.7	59/—
[50]	5m	Object Detection	VisDrone	BNSF & ASR	channel	↻	0.4	-91.2	—	-65.8	—/-65.7
[50]	5s	Object Detection	VisDrone	BNSF & ASR	channel	↻	-0.5	-93.4	—	-74.3	—/-53.2
[73]	5	Tomato Maturity Detection	Manual	—	channel	—	-1.5	-78	-76.4	-84.1	183/-64.9
[74]	5s	Railway Defect Detection	Manual	FPGM	channel	↻	2.19	—	—	—	34.7/—
Hybrid Pruning											
[75]	5l	Object Detection	Manual	—	kernel/layer	—	1.27	—	-86.8	—	—/-58.4
[76]	5s	Fruit Detection	Manual	BNSF	channel/layer	↻	-1.57	—	-72.5	—	32.5/—

For quantification techniques, the following three quantification methods are proposed (the experimental results of Table 3 are from Ref. [13]):

1) Integer-only Quantization: This method can significantly improve the hardware acceleration performance by using low-precision integers instead of floating-point calculations. For example, on the TITAN RTX, 23 times faster computational speeds can be achieved using the INT4 data type than FP32.

2) Post-Training Quantization (PTQ): Quantization is performed after training, and the model does not need to be retrained, but the accuracy is usually reduced.

3) Quantization-aware training (QAT): Simulate the quantization operation during the training process to reduce the quantization error and improve the accuracy of the model after quantization. The results show that the weights and activations of the YOLOv5 model can be quantized to 4 bits or even lower by the QAT method, while maintaining high accuracy.

Table 3. Experimental results of quantization technology on YOLOv5

Paper	V	Task	Dataset	Symmetry	Interval	St/Dyn	Fake/IntOnly	Precision	ΔAcc	Δsize	ΔFLOPs	ΔFPS/Δ⏲
PTQ												
[73]	5s	Tomato Detection	Manual	—	—	—	—	FP16	-0.9	-51.1	—	18.8/—
[73]	5s	Tomato Detection	Manual	—	—	—	—	Int8	-5.75	-73.7	—	-5/—
[98]	5	Human Pose Estimation	COCO	—	—	—	—	8/16 bits	-1.3	—	—	—
[91]	5	Infrared Object Detection	FLIR ADAS22	—	—	static	—	Int8	-1.1	-75	—	40/—
QAT												
[93]	5s	Infrared Ship Detection	Manual	Asym	uniform	static	fake	Int8	-5	-27	-33	—
[85]	5s	Object Detection	MS-COCO	Asym act/ Sym w	uniform	dynamic	IntOnly	Int4	-3.1	—	—	—
[86]	5s	Object Detection	PASCAL VOC	Asym	uniform	dynamic	—	5W/5A	-1.2	—	—	—
[86]	5s	Object Detection	PASCAL VOC	Asym	uniform	dynamic	—	4W/4A	-2.5	—	—	—
[86]	5s	Object Detection	PASCAL VOC	Asym	uniform	dynamic	—	3W/3A	-5.8	—	—	—
[87]	5	Crowd Counting	Manual	Asym act/ Sym w	—	—	—	Int8	-14	-87.5	—	—/-92.7
[90]	5n	Object Detection	COCO-8class	Asym	uniform	—	bitserial	<Int4	-1	—	—	—/-60.6
Unspecified												
[94]	5	—	—	Asym	nonuniform	—	fake	Int8	-0.9	—	-89.2	—
[99]	5	Object Detection	COCO	—	—	—	—	FP16	—	—	—	—/-60

In general, channel pruning and filter pruning are the most commonly used pruning methods on the YOLOv5 model, which can significantly reduce the requirements of model parameters and computing resources while maintaining high accuracy. Compared with other quantization methods, the Quantized Perception Training (QAT) method can compress the model to a very low precision (such as 4 bits) without significantly affecting the model performance.

2.4.2. SSD

Single Shot MultiBox Detector (SSD) realizes target classification and localization through single forward propagation. The main advantage of SSDs is their fast detection speed, their ability to perform well in real-time applications, and their ability to perform well when dealing with targets at different scales. The model enhances the detection ability of targets of various sizes by predicting feature maps at multiple scales. Disadvantages include lower accuracy on complex backgrounds and small object detection, and may require higher computing resources when processing high-resolution images. Overall, SSDs can provide satisfactory detection accuracy while maintaining high detection speeds, making them an effective choice for many real-time inspection tasks.

In Ref. [13], a Field Programmable Gate Array (FPGA) SSD-MobileNet-v1 acceleration method. Specifically, FPGM pruning selects the filters that need to be pruned based on the Geometric Median, calculates the geometric median for the filters in each layer, and removes the filter furthest from the geometric median. This ensures that important traits are retained during the pruning process and performance losses are reduced. In addition, sensitivity analysis is required in order to ensure the effectiveness of pruning during pruning. Sensitivity analysis is used to determine the pruning rate for each layer to find the most suitable pruning strategy. With this approach, it is possible to avoid over-pruning of certain critical layers, which can affect the performance of the model. The paper also proposes a regularization-based pruning strategy to ensure that the pruning model can better adapt to the FPGA platform by adjusting the pruning rate of each layer.

For the quantitative method, the QAT (Quantization Aware Training) method was adopted. Unlike post-training quantization (PTQ), QAT introduces quantization operations during the training process. This allows quantization errors to be simulated during training and corrected by backpropagation, reducing the impact of quantization on model accuracy. This method realizes the full quantization of the entire network model, and converts the weights and activation functions of all layers into low-precision representations. With this approach, the model size and computational requirements can be significantly reduced. On the basis of the QAT method, an improved QAT method was proposed in Ref. [13] to further reduce the impact of quantization on the accuracy of the model by adjusting the quantization range and step size.

Experimental results show the accuracy is reduced by less than 6% with about 10 times of efficiency improved and the compressed model performs well in the actual test.

3. CONCLUSIONS AND PROSPECTS

When performing quantitative pruning operations on the model, only at specific layers. In addition, the accuracy of model detection can be improved by introducing new loss functions and optimizing the structure of neural networks, and then quantitative pruning operations can be superimposed on this basis to reduce the number of model parameters and computing resource requirements on the basis of keeping the detection accuracy unchanged.

When the single-stage object detection algorithm is optimized for lightweight, it is necessary to use filters and other methods to retain important features and reduce the performance loss caused by the quantization pruning operation of the model. Specifically, the YOLO model is more suitable for channel pruning and filter pruning methods and quantization-aware training (QAT) methods. In the SSD model, the pruning rate of each layer can be determined through sensitivity analysis, so as to find the most suitable pruning strategy. and simulating the quantization error during the training process and correcting it through backpropagation, so as to reduce the impact of quantization on the accuracy of the model.

At present, the mainstream model compression technology is quantization and pruning in parallel. In addition, some algorithm models will use model compression techniques such as knowledge

distillation. In the future, it is hoped that the optimization strategies of object detection algorithms at different stages can be studied in depth and more abundant lightweight technologies can be covered.

REFERENCES

[1] Li Kequan, Chen Yan, Liu Jiachen, Mou Xiangwei. Survey of Object Detection Algorithms Based on Deep Learning [J]. Computer Engineering, 2022, 48(7): 1-12.

[2] Ning Xin, Zhao Wenyao, Zong Yixin, Zhang Yugui, Chen Hao, Zhou Qi, Ma Junxiao. A Review of Joint Optimization Methods for Neural Network Compression [J]. Chinese Journal of Intelligent Systems, 2024, 19(1): 36-57.

[3] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE Con ference on Computer Vision and Pattern Recognition, 2014: 580-587.

[4] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//Proceed ings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016:779-788.

[5] HE K, GKIOXARI G, DOLLAR P, et al. Mask R-CNN[C]// Proceedings of the IEEE International Conference on Computer Vision, 2017:2980-2988.

[6] HE K M, ZHANG X Y, REN S Q, et al. Spatial pyramid pooling in deep convolutional networks for visual rec ognition [J]. European Conference on Computer Vision, 2014: 346-361.

[7] GIRSHICK R. Fast R-CNN[C]//Proceedings of the IEEE International Conference on Computer Vision, 2015:1440-1448.

[8] REN S Q, HE K M, GIRSHICK R, et al.Faster R-CNN: towards real-time object detection with region proposal networks[C]//Advances in Neural Information Processing Systems, 2015:91-99.

[9] LIU W, ANGUELOV D, ERHAN D, et al. SSD:single shot multibox detector[C]//European Conference on Computer Vision, 2016:21-37.

[10] FU C Y, LIU W, RANGA A, et al.DSSD:deconvolutional single shot detector [J]. arXiv:1701.06659, 2017.

[11] LI Z, ZHOU F.FSSD:feature fusion single shot multibox detector [J]. arXiv:1712.00960, 2017.

[12] Aghli, N., & Ribeiro, E. (2021). Combining Weight Pruning and Knowledge Distillation for CNN Compression. CVPRW 2021.

[13] Jani, M., Fayyad, J., Al-Younes, Y., & Najjaran, H. (2023). An SSD-MobileNet Acceleration Strategy for FPGAs Based on Network Compression and Subgraph Fusion. *Forests*. Retrieved from [MDPI] (https://www.mdpi.com/).