Overview of Lightweight Technology for Pedestrian Detection Based on YOLOv8

Tan Li
Electronic Information Engineering, Hubei University, Hubei, China

Abstract: Pedestrian detection, as one of the core tasks in current computer vision, is a foundational technology for achieving goals such as autonomous driving and intelligent surveillance. With the increase in edge computing and real-time requirements, various detection models must meet stringent demands for low latency and low power consumption while maintaining high accuracy. YOLOv8 has become the mainstream framework for object detection, especially pedestrian detection tasks, due to its excellent accuracy-speed balance and reasonable modular design. This paper aims to systematically review the progress of lightweight research on pedestrian detection based on YOLOv8. First, it outlines the architectural features of YOLOv8 and its advantages as a choice for lightweight models; second, it categorizes and summarizes representative work in recent years on algorithm improvements and model compression, comparing their performance through analysis; then, it delves into core lightweight techniques such as pruning, quantization, and knowledge distillation, and their applicability in pedestrian detection; finally, it discusses current challenges like handling small targets and occlusions, balancing accuracy and speed, and looks forward to future directions including multi-modal fusion, dynamic inference, and new versions of the YOLO series. This paper provides a reference for researchers to fully understand the landscape of this field and choose technical pathways.

Keywords: YOLOv8; Lightweight; Pedestrian Detection

1. Introduction

1.1 Research Background

In autonomous driving, real-time and robust pedestrian perception technology forms the cornerstone of ensuring system reliability. Facing the demand for large-scale and complex real-time parsing, computationally efficient detection algorithms are key technical supports for achieving intelligent monitoring and emergency response. However, existing algorithms often struggle to achieve efficient and stable deployment on embedded and edge computing platforms with limitations in computing power, storage, and energy consumption. Therefore, developing high-precision, low-complexity lightweight pedestrian detection models for resource-constrained scenarios is not only an urgent practical need but also holds significant theoretical value and application prospects.

1.2 Research Significance

As an iterative version of the YOLO series, YOLOv8 has achieved significant improvements in both accuracy and speed. The multiple pre-trained models it provides (n/s/m/l/x) offer an ideal platform for lightweight model research. Conducting lightweight improvements on YOLOv8 for pedestrian detection tasks can effectively promote the practical application of related technologies in industrial fields, achieving a balance between algorithm performance and deployment cost.

1.3 Domestic and International Research Overview

Current domestic and international scholars' research on YOLOv8 lightweight primarily follows three technical routes: improving network architecture, model compression, and hybrid optimization strategies. Currently, domestic research on YOLOv8 lightweight tends to focus on scenario-specific or object-specific lightweight of the YOLOv8 model, while foreign scholars are more inclined to optimize and improve YOLOv8's architecture and algorithms, emphasizing end-to-end framework design. Scholars from both sides are committed to lightweight research on YOLO models in

Copyright @ STEMM Institute Press

different directions.

2. YOLOv8 Model Overview

2.1 Basic Architecture and Innovations of YOLOv8

(1) Advanced Backbone and Neck Architecture: YOLOv8 adopts the most advanced backbone and neck architecture, thereby improving feature extraction and target detection performance.

(2) Anchor-Free Decoupled Ultralytics Head: YOLOv8 uses an anchor-free decoupled Ultralytics head. Compared to anchor-based methods, this helps improve accuracy and detection efficiency.

(3) Optimized Accuracy-Speed Balance: YOLOv8 focuses on maintaining the best balance between accuracy and speed, suitable for real-time object detection tasks in various application domains.

(4) Rich Pre-trained Models: YOLOv8 provides a series of pre-trained models to meet various task and performance requirements, making it easier for users to find a suitable model for specific use cases. [1]

2.2 Performance of YOLOv8

Taking the coco training sets as an example (see Figure 1, Table 1)

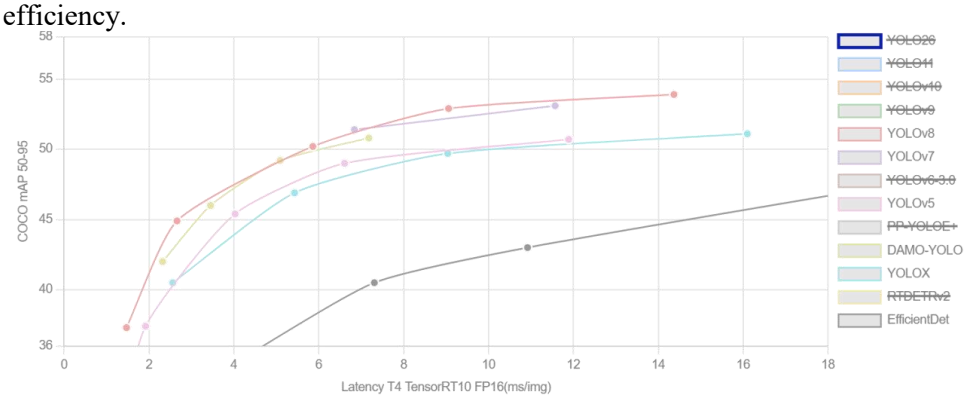

Figure 1. Comparison of Performance of different YOLO Models[1]

Table 1. Different Performances of YOLOv8's different Pre-trained Categories[1]

Model	size (pixels)	mAPval50–95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

On benchmark datasets like COCO, YOLOv8 shows improvements in both accuracy and efficiency compared to previous generations. Its clear modular structure and the spectrum of models officially provided, from extremely lightweight (YOLOv8n) to ultra-high precision (YOLOv8x), make it an ideal baseline model for lightweight pedestrian detection research.

3. Current Application Status of YOLOv8 in Pedestrian Detection

3.1 Research on Pedestrian Detection Based on YOLOv8

Main directions can be summarized into three categories: Algorithm structure improvement, Deep model compression, and Hybrid optimization strategies.

(1) Algorithm Structure Improvement: This type of research focuses on enhancing the model's feature extraction and fusion capabilities. For example, based on the YOLOv8n backbone architecture, adding a C2f-Ghost module reduces the number of floating-point operations during feature channel fusion, strengthens feature expression ability, and simultaneously introduces the Inner-CI0U loss function into YOLOv8n to enhance positioning capability for features of different scales. Experimental results show that compared to YOLOv8n, YOLOv8-GDI reduces the parameter count by 50% and computational complexity by 25% [2] while maintaining the same precision.

(2) Deep Model Compression: This type of research focuses on directly reducing the model's parameter count and computational load. For example, a proposed gradual pruning strategy based on Batch Normalization (BN) layer

 Copyright @ STEMM Institute Press

scaling factors. This strategy, through iterative pruning and fine-tuning, reduces parameters by 20.5% and computation by 21.7%, while preserving key underlying features. Simultaneously, a feature-perception-based knowledge distillation method is designed to compensate for the accuracy loss caused by pruning. [3]

(3) Hybrid Optimization is the current mainstream and forward-looking strategy for deploying models like YOLOv8 on resource-constrained devices (e.g., mobile terminals, embedded platforms). It is not simply stacking techniques like pruning, quantization, and knowledge distillation, but rather involves a meticulously designed workflow where steps work collaboratively and compensate for each other, ultimately achieving better results and a more optimal balance between model size, inference speed, and detection accuracy (See Table 2).

Table 2. Representative Methods in Different Directions

Technical Route	Representative Method	Core Strategy	mAP@0.5	Parameters (M)	Core Advantage	Main Limitations
Baseline	YOLOv8s [1]	Standard CSPDarknet	88.6% (KITTI)[1]	11.1 (Baseline)[1]	Performance Baseline	High computational requirements, difficult to deploy on edge devices
Algorithm Improvement	SE-YOLOv8 [4]	SENetV2+ Dynamic Head	87.1%[4] (BDD100K)	8.3(↓26%)[4]	Wins in both accuracy and lightweight design	Large computational overhead, memory bandwidth pressure
	YOLOv8-L [9]	GhostShuffle +C3Ghost	~37.0% (COCO)[9]	~2.2 (↓30%)[9]	Hardware-friendly, short development cycle	Long-distance target generalization needs verification
	YOLOv8-FN [7]	FasterNet +GN Head	37.2% (↓0.1%) [7]	~1.76 (↓45%) [7]	Extreme lightweight, lossless precision	GN optimization on some NPU is immature
Model compression	PCKD-YOLOv8 [5]	BN Pruning+ Feature distillation	74.8%[5]	~8.9 (↓20%)[5]	High precision maintained	Three-stage training, complex process, long cycle
Hybrid optimization	CMF-YOLO	CSCB+MFFP +LAMP	~80.6% (↑5.3%)	~1.4 (↓88%)	Precision and efficiency both improved	Hyper parameter tuning is complex, speed is moderate
	SSL-YOLO [6]	C2f-Star +LSDECD+ LAMP	88.1% (↓0.5%)[6]	2.0 (↑82%)[6]	High frame rate adaptation for in-vehicle video streaming	Rapid pruning accuracy decay

4. Overview of YOLOv8 Lightweight Technologies

4.1 Pruning Techniques

The goal of model pruning is to significantly reduce parameter count, computational load, and memory usage while almost maintaining detection accuracy, enabling YOLOv8 to run efficiently on edge devices. Its core idea is to identify and remove redundant channels or structures in the network that contribute minimally to the final output.

Current pruning techniques are mainly divided into structured pruning and unstructured pruning (See Table 3).

Table 3. Comparison of Structured and Unstructured Pruning Techniques

Type	Pruning target	Acceleration effect	Applicable scenarios
Structured pruning	Entire convolutional channel/module	Good	Edge devices, mobile terminals, industrial deployment
Unstructured pruning	Individual weights	Poor	Research exploration

Structured pruning is the preferred solution implemented in current YOLOv8 projects due to its strong compatibility and stable acceleration effects, far superior to unstructured pruning.

The following are common structured pruning methods

Copyright @ STEMM Institute Press

(1) Layer-Adaptive Magnitude-based Pruning (LAMP): The core of LAMP pruning technology lies in evaluating the importance of BN layer scaling factors for channels and applying differentiated pruning thresholds to different layers. This is an advanced global importance scoring method that can more evenly distribute pruning rates across layers, reducing the loss of accuracy while completing model lightweight.

(2) Progressive Channel Pruning: This is a method that systematically compacts model volume through multiple rounds of iteration, gradual refinement, and fine-tuning recovery, rather than removing redundant channels all at once. Each round prunes only 10% to 20% of redundant channels, followed by short-cycle fine-tuning, giving the model sufficient time to adapt to structural changes, thereby avoiding a cliff-like drop in accuracy. Therefore, progressive channel pruning has stable and effective characteristics in YOLOv8 model lightweight.

4.2 Quantization Techniques (INT8, binary, Quantization-Aware Training, etc.)

In YOLOv8 models, the core role of quantization technology is to compress model size, increase inference speed, and reduce power consumption by lowering numerical precision, while retaining model accuracy as much as possible.

It includes two key aspects:

(1) Numerical Representation Compression: Converting FP32 (32-bit) → INT8 (8-bit) → binary (1-bit), etc., to reduce storage and computational overhead.

(2) Computational Process Simulation: During the inference stage, using integer multiplication instead of floating-point operations, leveraging hardware's native support for integer operations (such as TensorRT Tensor Core) to achieve acceleration.

Table 4. Comparison of Mainstream Quantization Techniques

Technology	Precision	Compression rate	Acceleration ratio	Map loss (YOLOv8)	Feasibility	Recommended scenarios
INT8 + QAT	8-bit integer	4×	High	Minimal	Very high	In-vehicle, industrial vision, smart cameras
INT8 + PTQ	8-bit integer	4×	Moderate	Small to moderate	Relatively high	Prototype verification
Binary (1-bit)	±1	32×	Very high (theoretical)	Significant	Very low	Ultra-low power consumption sensor

From Table 4, it can be concluded that for lightweight deployment of YOLOv8 models used in pedestrian recognition, Quantization-Aware Training (QAT) is more suitable than Post-Training Quantization (PTQ), because it can significantly reduce accuracy loss while maintaining the advantage in inference speed.

Simultaneously, although binary (1-bit) has a high compression rate, YOLOv8 suffers from severe prediction distortion after binary and is not suitable as a quantization technique.

4.3 Knowledge Distillation

Knowledge distillation is the optimal supplementary method for improving small model accuracy and compensating for performance loss after quantization and pruning. By transferring additional supervisory information from large teacher networks, lightweight student networks can be trained. It examines mainstream methods including output distillation and feature distillation. (See Table 5)

Table 5. Comparison of Output Distillation and Feature Distillation Techniques

Method	Output distillation	Feature distillation
Distillation position	Detection head final output: category probability, confidence, regression coordinates	Intermediate layer: FPN feature map (P3/P4/P5)
Knowledge form	Soft labels, predictive distribution	Feature map spatial structure, channel activation pattern
Computational overhead	Low	High
Information granularity	Loss of spatial details	Preserves spatial relationships, beneficial for dense scenarios

Limitations of Output Distillation:

Loss of dense crowd information: Output distillation focuses only on the final prediction boxes, resulting in weak modeling capability for

 Copyright @ STEMM Institute Press

the spatial relationships of overlapping pedestrians. When the teacher model predicts two overlapping pedestrians, the student model struggles to learn occlusion boundary information from the 'softened' coordinate labels.
Advantages of Feature Distillation:
Preservation of multi-scale features: When pedestrian scale varies significantly, feature distillation (e.g., at P3/P4/P5 layers) can inherit the teacher model's multi-scale representation capability.
Therefore, for dense pedestrian detection tasks, feature distillation is more suitable than pure output distillation for YOLOv8 lightweight models.

4.4 Neural Architecture Search (NAS) and Lightweight Module Design

This type of method reduces complexity from the source by automatically searching or manually designing cheap operations (e.g., GhostConv) to replace standard convolutions.
(1) C2f-Star Module: SSL-YOLO introduces StarNet's StarBlock to replace the C2f module. It effectively reduces computational complexity through cross-channel feature reuse while enhancing multi-scale discrimination capability. This module uses a channel competition mechanism, achieving compact expression through feature reuse, reserving redundant space for subsequent LAMP pruning.
(2) Partial Convolution: YOLOv8-FN uses partial convolution to replace standard convolution, performing spatial feature extraction only on a part of the input channels, while the remaining channels are directly passed through, reducing redundant computation and memory access costs. Combined with the Group Normalization detection head, it achieves parameter reduction and computational complexity reduction, with a precision loss of 0.1%.
(3) Ghost Module: YOLOv8-L uses GhostShuffle and C3Ghost modules, generating 'ghost' feature maps through cheap linear transformations to reduce redundant computation. The regular structure of the Ghost module is friendly to acceleration libraries like TensorRT.

5. Challenges and Future Research Directions

Indeed, current research on large model lightweight has made tremendous progress, but still faces severe challenges.

5.1 Existing Challenges

(1) Speed-Accuracy Trade-off: This remains a major challenge for current models. Current lightweight techniques (e.g., LAMP pruning, Ghost modules) reduce computational load by decreasing the number of channels or reusing features, but directly weaken the feature representation capability of shallow networks, and some precision loss cannot be recovered through distillation methods.
(2) Insufficient Cross-Domain Generalization: Compared to large models, lightweight models suffer more severe accuracy degradation in harsh environments (rainy, foggy days) or under sudden illumination changes (e.g., tunnel entrances/exits). Current lightweight methods are mostly optimized for single domains (pruning/distilling on a single data set), lacking a compression framework with cross-domain perception.
(3) Reduced Computational Efficiency in Occlusion Scenarios: Lightweight models struggle to simultaneously maintain explicit occlusion relationship modeling and low-latency inference. In large cities, pedestrians often appear in occluded scenarios, so models need to perform pre-detection using other human features, such as shadows, leg features, etc.
(4) Hardware Heterogeneity and Optimization Fragmentation: The diversity of automotive-grade chips leads to significant differences in computing power support among different hardware, often making large model lightweight a 'customized' process.

5.2 Future Directions

(1) Multi-modal Fusion: Fusing more information like light, infrared, etc., to build more robust pedestrian detection systems adaptable to different weather and scenarios.
(2) Dynamic Detection: Dynamically allocating appropriate resources and computing power for different scenarios to meet model inference requirements. For complex scenarios, more computing power is invoked to speed up inference, while simpler scenarios use fewer resources to save computing power.
(3) Self-supervised and Semi-supervised Learning: Leveraging large amounts of unlabeled or weakly labeled pedestrian data to reduce the model's dependence on precisely annotated data and enhance model generalization capability.

Copyright @ STEMM Institute Press

6. Summary

This article systematically reviews the research progress of lightweight pedestrian detection technologies based on YOLOv8. The article first elaborates on the critical role of pedestrian detection in fields such as autonomous driving and intelligent surveillance, and points out that in the context of increasing demands for edge computing and real-time performance, model lightweight holds significant theoretical value and application prospects. YOLOv8, with its excellent precision-speed balance and modular design, has become an ideal baseline model for lightweight research.

The article organizes current mainstream technical routes from three aspects: algorithmic structure improvement, model compression and hybrid optimization, focusing on the adaptability and effectiveness of core lightweight techniques like pruning, quantization, and knowledge distillation on YOLOv8. Comparative experiments show that various methods achieve significant optimization in parameter count, computational complexity, and inference speed, while striving to maintain detection accuracy. Furthermore, the article explores forward-looking directions such as Neural Architecture Search (NAS) and lightweight module design.

Despite significant progress, lightweight models still face challenges such as the speed-accuracy trade-off, insufficient cross-domain generalization, low efficiency in handling occluded scenes, and hardware heterogeneity. Future research is expected to delve deeper into directions like multi-modal fusion, dynamic inference, and self-supervised learning to enhance model robustness and practicality in complex real-world scenarios.

Overall, this article provides a systematic overview of the technological landscape of YOLOv8 in the field of lightweight pedestrian detection, offering valuable references for subsequent research and engineering practice.

References

[1] Ultralytics, "YOLOv8 - Supported tasks and modes," Ultralytics Docs, 2023. [Online]. Available: https://docs.ultralytics.com/zh/models/yolov8/#supported-tasks-and-modes. [Accessed: Jan. 29, 2026].

[2] J. Sang et al., "YOLOv8-GDI: A lightweight YOLOv8 for real-time pedestrian detection," in *2024 7th International Conference on Robotics, Control and Automation Engineering (RCAE)*, 2024, pp. 422-428.

[3] Zhu G L, Yuan C X, Jiang F, et al. Lightweight YOLOv8 real-time object detection via progressive pruning and feature-aware knowledge distillation[C]//2025 IEEE 8th Advanced Information Technology and Electronic Information Engineering Conference. IEEE, 2025: 1-6.

[4] Y. Gu, Y. Zheng, T. Ding, X. Song and X. Zhang, "Urban Road Autonomous Driving Vehicle Target Detection Algorithm Based on Improved YOLOv8," 2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP), Xian, China, 2024, pp. 167-173,

[5] G. Zhu, C. Yuan and F. Jiang, "Lightweight YOLOv8 Real-Time Object Detection via Progressive Pruning and Feature-Aware Knowledge Distillation," 2025 IEEE 8th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Guiyang, China, 2025, pp. 925-933.

[6] K. Li, S. Wang and Z. Zhang, "Research on Road Object Detection Method Based on Lightweight YOLOv8s," 2025 7th International Conference on Information Science, Electrical and Automation Engineering (ISEAE), Harbin, China, 2025, pp. 1205-1209.

[7] C. Hu, Y. Wei and X. Tao, "Lightweight YOLOv8 Pedestrian Detection Model Based on FasterNet," 2024 4th International Symposium on Artificial Intelligence and Intelligent Manufacturing (AIIM), Chengdu, China, 2024, pp. 787-790.

[8] J. Cui, H. Zheng, X. Huang and Y. Zhang, "A Lightweight Detection Algorithm YOLOv8-L Based on YOLOv8n," 2024 6th International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 2024, pp. 853-857.

[9] W. Zhao, X. Yang, X. Ma and Y. Wang, "A Lightweight Multi Object Detection Algorithm for Complex Road Scenes Based on CMF-YOLO," 2024 IEEE 22nd International Conference on Industrial Informatics (INDIN), Beijing, China, 2024, pp. 1-6.

 Copyright @ STEMM Institute Press