ICCK Transactions on Intelligent Cyber-Physical Systems
http://dx.doi.org/10.62762/TICPS.2026.107157

REVIEW ARTICLE

Lightweight YOLO Object Detection Algorithms for Vision-Based Robotic Arms: A Review

Fajun Guo[1,*]

[1] College of Engineering, Lishui University, Lishui 323000, China

Abstract

To address the requirements of vision-based robotic manipulators for object detection accuracy, real-time performance, and lightweight deployment, this paper reviews lightweight You Only Look Once (YOLO) object detection techniques and their applications in robotic vision. First, the main techniques, including structural lightweighting, model compression, and edge deployment, are summarized. Representative algorithms for vision-based robotic manipulators are then categorized into three groups: backbone lightweighting, local structural optimization, and model compression, with their detection performance and lightweighting characteristics analyzed. On this basis, the limitations of existing methods are discussed in terms of the trade-off between accuracy and efficiency, cross-scenario generalization, coordination between visual perception and grasp execution, and comprehensive evaluation. Finally, future research directions are outlined, including multi-strategy collaborative optimization, cross-domain adaptation, and system-level evaluation for robotic manipulators.

Submitted: 10 May 2026
Revised: 08 September 2026
Accepted: 15 September 2026
Published: 20 September 2026

Vol. 1, **No.** 3, 2026.
10.62762/TICPS.2026.107157

***Corresponding author:**
✉ Fajun Guo
guofj020@163.com

Keywords: YOLO, object detection, lightweight model, vision-based robotic manipulator.

1 Introduction

With the rapid development of intelligent robotics, vision-based perception has become a key enabler of robotic manipulation [1], and You Only Look Once (YOLO) detectors have been widely used in such perception tasks [2, 3]. YOLO integrates object classification and bounding-box regression into an end-to-end single-stage framework and is characterized by a simple architecture and fast inference [2]. In recent years, the YOLO series has continuously evolved through network architecture optimization [3–5], improvements in training mechanisms [4], and end-to-end inference design [4], achieving a better balance between detection accuracy and inference efficiency. However, with the continuous expansion of network depth, channel scale, and feature fusion structures, the number of parameters and computational cost of some high-performance models has consequently increased. Vision-based robotic manipulators need to identify and locate objects to be grasped in real time, and their operating scenarios often involve small objects, occlusion, stacking, and pose variations. In such scenarios, excessive lightweighting may lead to missed detections and localization errors. This paper reviews the structural lightweighting, model compression, and deployment techniques of lightweight YOLO models, summarizes their applications in robotic vision, and analyzes the

Citation
Guo, F. (2026). Lightweight YOLO Object Detection Algorithms for Vision-Based Robotic Arms: A Review. *ICCK Transactions on Intelligent Cyber-Physical Systems*, 1(3), 97–103.

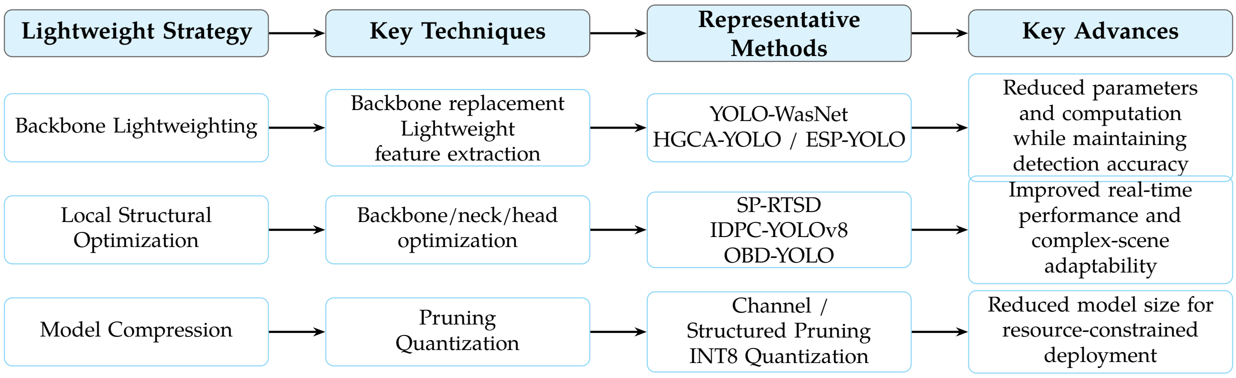

Figure 1. Technical routes and research advances of lightweight YOLO for vision-based robotic manipulators.

applicability of different lightweighting approaches.

2 Major Techniques for Lightweight YOLO

2.1 Structural Lightweighting

Structural lightweighting mainly reduces redundant computation by optimizing the backbone, feature fusion network (neck), and detection head, thereby reducing the complexity of YOLO models [6]. Lightweight networks such as MobileNetV2 [7], ShuffleNet [8], and GhostNet [9] are commonly used to replace the original backbone, while network width adjustment and channel scaling can further reduce model complexity to meet the throughput requirements of edge hardware [10]. Lightweight structures such as GSConv [11] can be applied to the feature fusion network to reduce the computational cost of multi-scale feature fusion, while lightweight detection heads such as the ES-Head proposed in PV-YOLO [12] can reduce the computation required for classification and regression. Such methods can reduce model complexity and facilitate deployment on resource-constrained platforms; however, lightweight structural design still requires a trade-off between detection accuracy and computational cost [13].

2.2 Model Compression

Model compression can reduce the storage and computational overhead of YOLO models by removing redundant parameters, lowering numerical representation precision, or transferring knowledge [14]. Common approaches include network pruning [15, 16], model quantization [17], and knowledge distillation [18]. These compression strategies have been applied in robotic manipulation contexts, including augmented reality-assisted arm control [19] and real-time grasp detection [20]. Specifically, pruning compresses the network by removing redundant weights or structures, quantization reduces storage and computational requirements by lowering parameter bit-width, and knowledge distillation uses a teacher model to guide the training of a lightweight student model [14]. Model compression requires a trade-off between complexity reduction and accuracy degradation; when multiple compression strategies are jointly applied, error accumulation and interactions among different strategies should also be considered [21].

2.3 Edge Deployment

Edge deployment refers to deploying AI models on end devices, embedded systems, or edge computing platforms to perform local processing and real-time inference close to the data source [22]. YOLO models have been deployed on various resource-constrained hardware platforms, including mobile terminals such as Android smartphones [23], embedded devices such as microcontrollers [24], and low-compute edge AI systems-on-chip (SoCs) [25]. Edge deployment can reduce data-processing latency and support local real-time inference; however, practical deployment performance is jointly affected by factors such as hardware platforms, software frameworks, resource constraints, model architectures, and optimization strategies [26].

3 Typical Lightweight YOLO Object Detection Algorithms for Vision-Based Robotic Manipulators

Building on the structural lightweighting, model compression, and edge deployment techniques discussed in Section 2, we classify representative lightweight YOLO algorithms for robotic manipulators into three categories based on the primary source of model complexity reduction: backbone lightweighting, local structural optimization, and model compression. When a method employs multiple strategies, it is classified according to the strategy that contributes most to reducing model complexity. The main technical routes, representative methods, and research progress of the above three types of approaches are shown in Figure 1.

3.1 YOLO Algorithms Based on Backbone Lightweighting

Backbone lightweighting mainly reduces front-end computational cost by replacing the original backbone or reconstructing feature extraction modules. Existing studies generally employ lightweight convolutional structures to reduce the number of parameters and floating-point operations (FLOPs), while incorporating attention or feature enhancement modules to mitigate the degradation in representational capability. Wang et al. [27] proposed YOLO-WasNet, which replaces the YOLOv5 backbone with MobileNetV3-Small [28] and introduces the SCBA module (a combination of spatial pyramid pooling-fast and a convolutional block attention module) and depthwise parallel triple-attention convolution (DPT-Conv) to enhance feature extraction. On their self-constructed waste classification dataset, the model achieved a mean average precision at an intersection-over-union (IoU) threshold of 0.5 (mAP50) of 96.8% with 4.89 M parameters and 8.6 GFLOPs. The model was subsequently integrated into a robotic waste sorting system. In addition to replacing the entire backbone, some studies have further focused on lightweighting local feature extraction structures. Zhang et al. [29] proposed HGCA-YOLO, which reduces computational cost through Ghost-based lightweight feature extraction modules and introduces coordinate attention to enhance object feature representation. Building on this local restructuring approach, Chen et al. [30] proposed ESP-YOLO, in which the Efficient Layer Shuffle Aggregation Network (ELSAN) replaces the C3 and CBS modules in the backbone to reduce computational complexity while maintaining detection accuracy. The model was further validated on a UR5-based grape harvesting platform. Overall, such methods integrate full backbone replacement with local feature reconstruction and compensate for the representational loss caused by lightweighting through feature enhancement.

3.2 YOLO Algorithms Based on Local Structural Optimization

Compared with backbone lightweighting, local structural optimization focuses more on selectively reducing redundant computation in the backbone, neck, and head, while accounting for occlusion, small targets, and pose variations in robotic manipulation tasks. To address computational redundancy in the feature extraction and prediction stages, Chen et al. [31] proposed SP-RTSD, which employs C2f-Faster and a parameter-sharing detection head to reduce model computational overhead. On this basis, some studies have further combined lightweighting with pose perception requirements. Jiao et al. [32] proposed IDPC-YOLOv8, which introduces rotated bounding-box prediction based on efficient feature extraction and detection-head pruning to meet the orientation localization requirements of precision grasping. For occluded and densely distributed object scenarios, Cao et al. [33] proposed SRF-YOLO, which enhances occluded-object perception through C2PSA-SHSA and strengthens multi-scale feature fusion using C3k2-RSCM. Dong et al. [34] proposed OBD-YOLO, which collaboratively optimizes the backbone and neck, reducing FLOPs from 22.5 G to 19.3 G while achieving a mAP50 of 84.0%. Overall, such methods reduce computational overhead through module reconstruction, detection-head optimization, and feature enhancement while improving adaptability to robotic manipulation scenarios.

3.3 YOLO Algorithms Based on Model Compression

Model compression mainly reduces redundant parameters or lowers numerical representation precision on the basis of existing YOLO networks, making the compressed models more suitable for rapid deployment in resource-constrained robotic vision applications. Among these methods, pruning is a commonly used compression approach. Chiu et al. [35] reduced the parameter count of YOLOv5s from 7.02 M to 2.83 M through sparse training and channel pruning. Compared with channel pruning based on batch normalization (BN) scaling factors [15], Han et al. [36] adopted structured pruning based on the dependency graph (DepGraph) [16] to compress YOLOv8, reducing detection overhead while maintaining the overall network structure. In addition to pruning, quantization reduces model storage and computational requirements at the numerical representation level. Almaliki et al. [24] adopted INT8 quantization to reduce the YOLOv7-tiny model size by approximately 75%, thereby enabling the integration of object detection and six-degree-of-freedom (6-DoF) robotic manipulator control on a single microcontroller.

Table 1. Comparison of representative lightweight YOLO algorithms for vision-based robotic manipulators.

Category	Representative Algorithm	Main Lightweighting Method	Detection Metric	Lightweighting Metric
Backbone lightweighting	YOLO-WasNet [27]	MobileNetV3-Small backbone replacement	mAP50 96.8%	4.89 M; 8.6 GFLOPs
	HGCA-YOLO [29]	Ghost-based feature extraction	mAP50 95.2%	Params ↓46.2%; GFLOPs ↓48.4%
Local structural optimization	SP-RTSD [31]	C2f-Faster; shared detection head	mAP50 91.7%	Model size ↓40.3%
	IDPC-YOLOv8 [32]	Dual-backbone feature extraction; head pruning	mAP50 97.8%	5.84 MB; 126.59 FPS
	OBD-YOLO [34]	MDC backbone; FasterBlock-based neck	mAP50 84.0%	19.3 GFLOPs
Model compression	Chiu et al. [35] (YOLOv5s)	Channel pruning	AP 99.4%	Params: 7.02 M → 2.83 M
	Almaliki et al. [24] (YOLOv7-tiny)	INT8 quantization	mAP50 88.2%	Model size ↓≈75%

Note: ↓ denotes a relative reduction with respect to the respective baseline model; AP, average precision; MDC, modulated deformable convolution; FPS, frames per second.

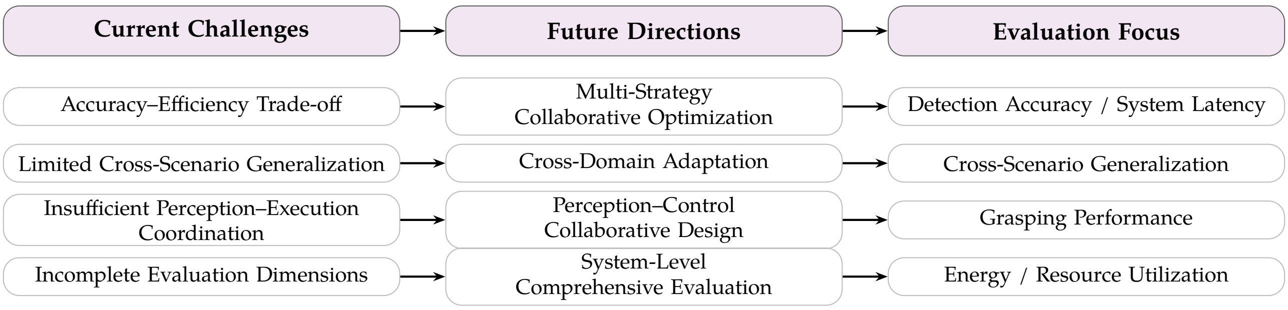

Figure 2. Main challenges, future directions, and evaluation focuses of lightweight YOLO for vision-based robotic manipulators.

Compared with network reconstruction, model compression generally does not require redesigning the entire network; however, the compression intensity still needs to be determined according to detection accuracy, hardware support, and actual inference efficiency.

3.4 Comparison and Analysis of Representative Algorithms

From the perspective of lightweighting mechanisms, the three categories have different focuses. Backbone lightweighting targets front-end computation, and its key challenge is to balance model complexity against feature representation capability. Local structural optimization offers stronger task adaptability because it can further account for robotic vision requirements such as occlusion, small objects, and orientation variations. Model compression is applied on the basis of existing networks and therefore mainly reduces storage and inference overhead, which facilitates deployment on resource-constrained platforms. Overall, practical applications require comprehensive trade-offs among detection accuracy, computational complexity, inference latency, object localization capability, and robotic execution performance. The main characteristics of representative algorithms are summarized in Table 1. Because different studies use different datasets, detection objects, and experimental platforms, and report detection metrics (mAP50 or AP) and lightweighting metrics on inconsistent bases, the performance figures in the table are intended to reflect the lightweighting characteristics of individual algorithms rather than to support direct cross-study comparison.

4 Existing Problems and Future Trends

Currently, lightweight YOLO methods for vision-based robotic manipulators still face trade-offs among accuracy, efficiency, and deployment performance. Excessive lightweighting may lead to decreased detection accuracy and weakened feature extraction capability [37], while cross-scenario generalization in practical grasping tasks still needs to be improved [38]. In addition, coordination among visual perception, grasp localization, and execution control still requires further enhancement [39]. Existing related studies continue to focus largely on model computational complexity and inference performance, whereas comprehensive evaluation of power consumption, hardware resource utilization, and system-level energy efficiency remains insufficient [40]. Future research should coordinate model lightweighting, cross-domain adaptation, and perception–control collaborative design, while establishing a comprehensive evaluation system for practical robotic platforms that considers detection accuracy, system latency, energy consumption, and grasping performance. The main challenges, future directions, and evaluation focuses of lightweight YOLO for vision-based robotic manipulators are summarized in Figure 2.

5 Conclusion

This paper reviews efficient YOLO object detection techniques, including lightweight network architectures, model compression, and edge deployment, and further summarizes representative lightweight YOLO algorithms for vision-based robotic manipulators. Existing studies have made certain progress in reducing model complexity, maintaining detection accuracy, and adapting to resource-constrained platforms. However, there remains room for improvement in cross-scenario generalization, coordination between visual perception and grasping execution, and evaluation of practical deployment performance. For practical robotic manipulator applications, lightweight YOLO still needs to balance detection accuracy, inference efficiency, and overall system responsiveness.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

The author declares no conflicts of interest.

AI Use Statement

The author declares that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Du, G., Wang, K., Lian, S., & Zhao, K. (2021). Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review, 54*(3), 1677-1734. [CrossRef]

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779–788). [CrossRef]

[3] Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction, 5*(4), 1680–1716. [CrossRef]

[4] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). YOLOv10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems, 37*, 107984-108011. [CrossRef]

[5] Tian, Y., Ye, Q., & Doermann, D. (2026). YOLOv12: Attention-centric real-time object detectors. *Advances in Neural Information Processing Systems, 38*, 78433-78457. [CrossRef]

[6] Wang, L., Wang, H., Letchmunan, S., Xiao, R., Ahmed, O. H., & Liu, Z. (2025). A systematic literature review of lightweight YOLO models for object detection. *PeerJ Computer Science, 11*, e3357. [CrossRef]

[7] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018, June). Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 4510-4520). IEEE. [CrossRef]

[8] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018, June). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 6848-6856). IEEE. [CrossRef]

[9] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020, June). Ghostnet: More features from cheap operations. In 2020 IEEE/CVF conference on

computer vision and pattern recognition (CVPR) (pp. 1577-1586). IEEE. [CrossRef]

[10] Ganesh, P., Chen, Y., Yang, Y., Chen, D., & Winslett, M. (2022, January). YOLO-ReT: Towards high accuracy real-time object detection on edge GPUs. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 1311-1321). IEEE. [CrossRef]

[11] Li, H., Li, J., Wei, H., Liu, Z., Zhan, Z., & Ren, Q. (2024). Slim-neck by GSConv: A lightweight-design for real-time detector architectures. *Journal of real-time image processing, 21*(3), 62. [CrossRef]

[12] Liu, Y., Huang, Z., Song, Q., & Bai, K. (2025). PV-YOLO: A lightweight pedestrian and vehicle detection model based on improved YOLOv8. *Digital Signal Processing, 156*, 104857. [CrossRef]

[13] Bacea, D. S., & Oniga, F. (2026). Boosting lightweight object detection with enhanced feature fusion and optimized receptive field. *Neurocomputing, 132728*. [CrossRef]

[14] Baicheng, Y. E., Youpan, Z. H. U., Yongkang, Z. H. O. U., Chenhao, D. U. A. N., Yudong, Z. H. A. N. G., Zhigang, T. A. O., & Zhiyu, F. U. (2025). Review of Lightweight Target Detection Algorithms. *Infrared Technology, 47*(3), 289-298. http://hwjs.nvir.cn/en/article/pdf/preview/df48da80-f152-4563-95fb-801ceebcbded.pdf

[15] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017, October). Learning efficient convolutional networks through network slimming. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 2755-2763). IEEE. [CrossRef]

[16] Fang, G., Ma, X., Song, M., Mi, M. B., & Wang, X. (2023, June). Depgraph: Towards any structural pruning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16091-16101). IEEE. [CrossRef]

[17] Salah, O. B. H., Messaoud, S., Hajjaji, M. A., Atri, M., & Liouane, N. (2026). Low-latency QYOLOv10-based FPGA implementation for real-time object detection. *Integration, 106*, 102591. [CrossRef]

[18] Tamura, K., & Todo, Y. (2026). Dual-head self-distillation via an auxiliary lightweight neck for YOLO-based object detection. *Neurocomputing, 686*, 133750. [CrossRef]

[19] Hyung, J. W., Na, W., Won, K., & Kim, D. J. (2025). Robotic arm control by augmented reality-assisted object detection. *Scientific reports, 15*(1), 35678. [CrossRef]

[20] Han, T., & Yu, D. (2024). Jensen–Shannon divergence you only look once: a real-time robotic grasp detection network. *Advanced Intelligent Systems, 6*(5), 2300497. [CrossRef]

[21] Fan, C., Li, J., Zhang, Z., Li, F., & Wang, B. (2026). Two-phase collaborative model compression training for joint pruning and quantization. *Neural Networks, 197*, 108506. [CrossRef]

[22] Wang, X., Tang, Z., Guo, J., Meng, T., Wang, C., Wang, T., & Jia, W. (2025). Empowering edge intelligence: A comprehensive survey on on-device AI models. *ACM Computing Surveys, 57*(9), 1-39. [CrossRef]

[23] Song, Q., Li, S., Bai, Q., Yang, J., Zhang, X., Li, Z., & Duan, Z. (2021). Object detection method for grasping robot based on improved YOLOv5. *Micromachines, 12*(11), 1273. [CrossRef]

[24] Almaliki, H. H., Mazinan, A. H., & Modaresi, S. M. (2026). Design and implementation of a 6-DoF robot arm control with object detection based on machine learning using mini microcontroller. *Scientific Reports, 16*(1), 6842. [CrossRef]

[25] Kong, C., Li, F., Yan, X., Yang, J., Mo, P., Luo, Q., & Mao, R. (2026). Object detection on low-compute edge SoCs: A reproducible benchmark and deployment guidelines. *Scientific Reports, 16*(1), 5875. [CrossRef]

[26] Khan, T. M., Ul Haq, Q. E., Iqbal, S., & Soomro, T. A. (2026). Edge-based artificial intelligence: Understanding the evolution of hardware and software and future trends. *Engineering Applications of Artificial Intelligence, 174*, 114526. [CrossRef]

[27] Wang, A., Jiao, H., Chen, Z., & Yang, J. (2025). An improved YOLO-based waste detection model and its integration to robotic gripping systems. *Computers, Materials & Continua, 84*(3), 5773–5790. [CrossRef]

[28] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 1314-1324). IEEE. [CrossRef]

[29] Zhang, P., Dai, N., Liu, X., Yuan, J., & Xin, Z. (2024). A novel lightweight model HGCA-YOLO: Application to recognition of invisible spears for white asparagus robotic harvesting. *Computers and Electronics in Agriculture, 220*, 108852. [CrossRef]

[30] Chen, J., Chen, H., Xu, F., Lin, M., Zhang, D., & Zhang, L. (2024). Real-time detection of mature table grapes using ESP-YOLO network on embedded platforms. *Biosystems Engineering, 246*, 122–134. [CrossRef]

[31] Chen, Y., Shu, A., Liu, Z., Chen, Y., Lee, W. S., & Zhang, Y. (2025). SP-RTSD: A lightweight real-time strawberry detection on edge devices for onboard robotic harvesting. *Journal of Field Robotics, 42*(7), 3361–3379. [CrossRef]

[32] Jiao, J., Li, Z., Xia, G., Wang, G., Chen, Y., & Gao, R. (2026). A lightweight object detection approach for precision gripping in multiple peg-in-hole assembly tasks. *Robotics and Computer-Integrated Manufacturing, 98*, 103185. [CrossRef]

[33] Cao, H., Yang, S., Zhang, Y., Zhang, H., & Ye, L. (2026). SRF-YOLO: A lightweight detection model and robotic system integration for protected tomato harvesting. *Smart Agricultural Technology, 14*, 102036. [CrossRef]

[34] Dong, J., Deng, L., Wan, D., Liu, C., Yin, J., Guo, M., Zhang, H., Lin, S., Liu, H., & Liu, L. (2026).

Oriented bounding box detection algorithm for dense scenarios of robotic arm operation. *Expert Systems with Applications*, *298*, 129678. [CrossRef]

[35] Chiu, Y. J., Yuan, Y. Y., & Jian, S. R. (2024). Design of and research on the robot arm recovery grasping system based on machine vision. *Journal of King Saud University Computer and Information Sciences*, *36*(4), 102014. [CrossRef]

[36] Han, D., Li, H., Li, Y., & Chen, S. (2026). Real-Time Target-Oriented Grasping Framework for Resource-Constrained Robots. *Sensors*, *26*(2), 645. [CrossRef]

[37] He, J., Jiang, J., & Zhang, C. (2026). A survey of lightweight methods for object detection networks. *Array*, *29*, 100589. [CrossRef]

[38] Song, X., Li, Y., Zhang, Y., Liu, Y., & Jiang, L. (2025). An overview of learning-based dexterous grasping: recent advances and future directions. *Artificial Intelligence Review*, *58*(10), 300. [CrossRef]

[39] Wang, Q., Tu, Y., Xu, W., Zhang, J., Knoll, A., Zhou, M., & Ying, Y. (2025). Towards Damage-Less Robotic Fragile Fruit Grasping: A Systematic Review on System Design, End Effector, and Visual and Tactile Feedback. *Journal of Field Robotics*, *42*(8), 4521-4543. [CrossRef]

[40] Le, Q. N. N., Chembakasseril, M. T., & Hartanto, R. (2026). Benchmark analysis of deep learning algorithms for edge-based robotic applications. *Array*, 100818. [CrossRef]

Fajun Guo obtained his bachelor's degree from the University of Science and Technology Liaoning. He is currently pursuing his master's degree in Mechanical Engineering at the Graduate School of Lishui University. His research interests focus on the development of core algorithms for modern industrial robotic arms, with a particular dedication to improving the motion accuracy and intelligent grasping performance of desktop-scale manipulators. (Email: guofj020@163.com)