Efficient Pest Detection via Low-Rank Additive Attention and Scale-Dynamic Loss for Deep Edge Intelligence in Smart Agriculture

Zhiqiang Chen[1, 2], Shuai Zhou[1, 2], Zhangjun Peng[1, 2], Guoqiang Zheng[1, 2], Chuanhao Chang[1, 2], and Li Li[1, 2, *]
[1]*Southwest University of Science and Technology, Mianyang, China*
[2]*Sichuan Engineering Technology Research Center of Industrial Self-Supporting and Artificial Intelligence, Mianyang, China*
*Corresponding author: llki2000@163.com

***Abstract*—Addressing the challenges of feature blurring in sub-pixel objects and limited computing power of edge devices in field pest detection, this paper proposes APLS-YOLO, a lightweight detection framework oriented towards Deep Edge Intelligence. First, targeting the slender geometric features of pests and the need for effective receptive fields, we design the Asymmetric-Priority Pinwheel Convolution (AP-PConv). By approximating the large kernel receptive field through orthogonal strip convolutions, it significantly reduces computational redundancy while suppressing background noise. Second, to resolve the high-entropy feature blurring problem of traditional Softmax attention, we propose Low-Rank Additive Attention (LR-AAtten). This module utilizes an Additive Token Mixing strategy to replace complex matrix multiplication, effectively preserving the texture polarity features of tiny pests with $O(N)$ complexity. Furthermore, a Scale-based Dynamic Loss (SD-Loss) is introduced to solve the gradient collapse problem of small targets during training via an adaptive re-weighting mechanism. Finally, combined with Global Channel Pruning based on BN layer scaling factors, extreme model compression is achieved. On the IP102 dataset, APLS-YOLO achieves an mAP@0.5 of 61.80%, a 2.45% improvement over the YOLOv11n baseline. On the more challenging FieldPest-25 dense tiny pest dataset, the mAP stabilizes at 60.31%, demonstrating strong robustness. On this dataset, the model parameters are only 2.18 M (17% reduction from baseline), and the model size is compressed to 3.88 MB. It achieves a real-time inference speed of 79 FPS on the RK3588 edge platform, showing exceptional potential for edge-side deployment.**

***Index Terms*—Deep Learning, Edge Intelligence, Object Detection, Pest Detection, Smart Agriculture**

I. Introduction

Agricultural production is the cornerstone of the global economy, yet pests and diseases remain major threats to crop yield and quality, often causing 20%-40% reductions in global staple food production [1]. Timely and accurate identification of pest species and their severity is crucial for implementing targeted control measures and reducing pesticide abuse [2]. However, the field environment is complex, and pests are diverse in morphology and extremely small (e.g., aphids and planthoppers are often less than 2mm in length). This poses significant challenges to traditional manual inspection, which is often time-consuming, subjective, and limited in coverage [3].

Although deep learning has made significant progress in general object detection, applying these algorithms directly to edge monitoring in smart agriculture still faces severe pain points: 1) Existing anchor-free assignment strategies often exhibit suboptimal performance on tiny targets due to inefficient sample assignment, as the limited number of positive samples (often just a single grid point) leads to insufficient training supervision for sub-pixel pests [4]. 2) Detection performance is often compromised by severe background interference, where drastic changes in field lighting, occlusion by branches, and complex soil textures trigger massive false alarms [5]. 3) The deployment is constrained by limited edge resources, as SoCs on farm monitoring stations and UAVs typically possess less than 10 TOPS of computing power, making them unable to support the massive redundant parameters and high energy costs of traditional models [6].

To break through these limitations, this paper moves beyond the conventional large model compression paradigm. Instead, we collaboratively redesign the model from three dimensions: operators, architecture, and loss functions, proposing a lightweight pest detection network named APLS for Edge Intelligence. The main contributions of this paper are as follows:

- To address the challenge of slender and variable directional features in pests, we design the AP-PConv. Inspired by the Gaussian distribution characteristics of tiny objects [7] and the geometric advantages of strip convolutions [8], this lightweight gated priority mechanism dynamically aggregates four-directional asymmetric features to achieve precise capture of key pest anatomical structures.
- To significantly enhance fine-grained texture recognition capabilities under restricted memory, we propose the LR-AAtten. Drawing on the polarity perception theory of PolarFormer [9] and the additive interaction paradigm of CAS-ViT [10], this module decouples spatial dependence into horizontal and vertical low-rank components, effectively reducing the computational complexity of self-attention from $O(N^2)$ to linear $O(N)$.
- To resolve gradient collapse for small targets, we introduce SD-Loss. Improving upon the method in [7],

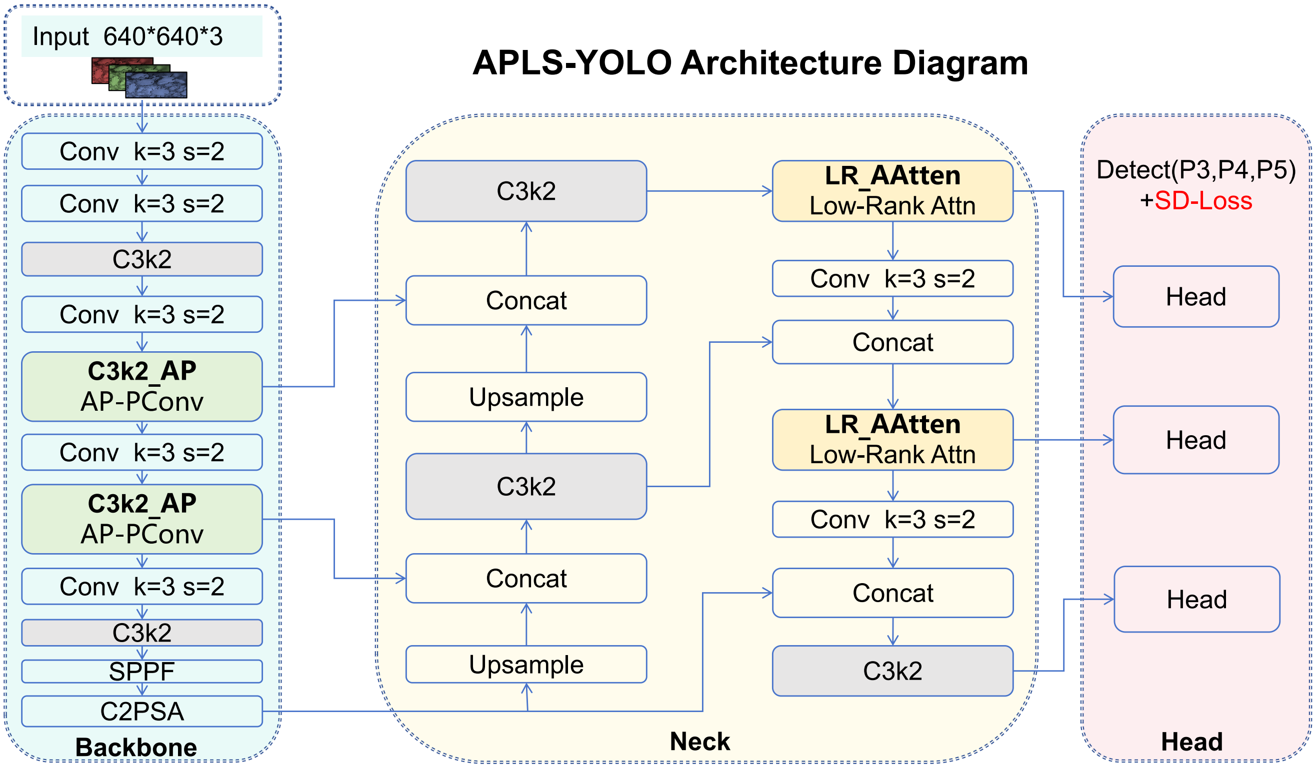


Fig. 1. The overall architecture of APLS-YOLO. It consists of a Backbone with AP-PConv for directional feature extraction, a Neck integrated with LR-AAtten for global context modeling, and a Head optimized with SD-Loss.

this mechanism optimizes training by dynamically re-weighting gradients based on the average normalized target area.

- To facilitate deployment on edge platforms like the Orange Pi 5 (RK3588), we apply structural compression via L1 regularization on BN layers [11]. This strategy reduces the model size to under 4MB, achieving efficient end-to-end deployment.

We conducted extensive experiments on the self-built FieldPest-25 dataset containing 25 core pests and the public IP102 dataset [12]. Results show that the APLS model achieves an mAP@0.5 of 61.80% while maintaining superior inference speed (79 FPS), a 2.45% performance improvement over YOLOv11n, providing a feasible edge intelligence solution for real-time, portable field pest monitoring.

II. Related Work

In recent years, deep learning-based object detection technology has become a core driver of smart agriculture. The YOLO series, due to its excellent balance between speed and accuracy, has been established as the mainstream paradigm for field pest detection. With rapid architectural iteration, YOLOv8 [13] and YOLOv10 [14] have been proven to have significant advantages in identification tasks in cotton fields and complex backgrounds [15]–[17]. Recent studies have further explored the potential of YOLO11; for example, Li et al. [18] effectively balanced feature extraction capability and inference efficiency in rice pest detection by introducing AP_BiFPN and re-parameterized ViT. However, existing general YOLO models are designed mainly for regular-scale objects. When facing sub-pixel pests in the field, they often struggle to achieve the best trade-off between detection accuracy and feature resolution.

As computing tasks shift from the cloud to the edge (Edge Intelligence), building low-latency, low-power detection models has become a key trend [19]. To reduce computational load, Ghost Convolution (GhostConv) is often used to replace standard convolution to compress model volume [16]. However, Chen et al. [20] pointed out that GhostConv still

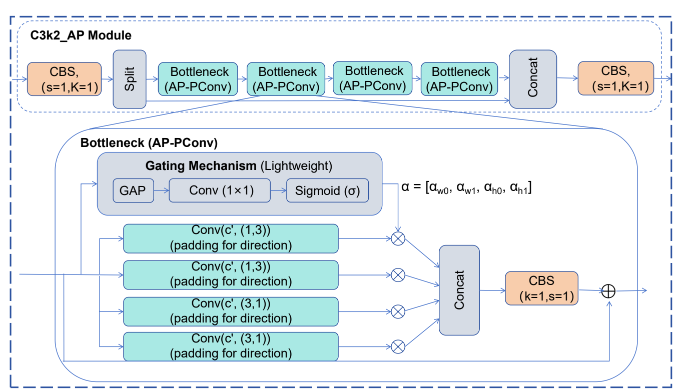


Fig. 2. Architecture of the C3k2_AP module integrated with AP-PConv.

contains a large number of 1×1 operations, limiting actual inference efficiency improvements on specific hardware (such as NPUs). In contrast, reconstruction schemes based on Partial Convolution (PConv) demonstrate better speed potential while maintaining accuracy by reducing redundant computation and memory access. Furthermore, model compression techniques are widely applied; Wang [21] and Zhang et al. [22] verified the effectiveness of structured channel pruning in removing redundant feature channels and reducing video memory occupancy.

The texture blurring and background interference of tiny field pests (especially aphids) are current detection difficulties [2]. To address the issue of small target feature loss, mainstream research focuses on multi-scale fusion [23], [24] and attention mechanisms [25], [26]. Although introducing Coordinate Attention (CA) or parameter-free SimAM attention [27] can enhance the model's focus on key features, existing attention mechanisms often fail to establish effective global dependencies with low computational complexity. Addressing this challenge, Tang et al. [28] proposed a fusion of receptive fields and global modeling. Inspired by this, this paper combines the Dynamic Asymmetric PConv operator with a global channel pruning strategy.

III. Materials and Methods

A. APLS Network Structure

The network proposed is APLS (Asymmetric-Priority Low-rank Scale-aware Network), built upon the lightweight YOLOv11n architecture known for its efficiency. As shown in Fig. 1, the structure comprises four main parts: Input, Backbone, Neck, and Detection Head. **Input:** Images are resized to 640×640 and normalized. **Backbone:** We replace original C3 modules with our proposed AP-PConv blocks to adaptively capture directional features. **Neck:** We integrate LR-AAtten modules in the feature fusion path to enhance spatial dependency modeling with linear complexity. **Detection Head:** We retain the efficient YOLOv11n head but train using our SD-Loss to improve small object detection capabilities.

B. C3k2_AP and AP-PConv

The C3k2 module in YOLOv11 transforms and extracts features through multiple residual bottlenecks. To adapt to the slender, variable morphology of pests and the resource constraints of edge devices, we propose the AP-PConv. Innspired by the need for large effective receptive fields [7] and and the advantages of strip convolutions in capturing slender geometric features [8], AP-PConv is designed to reduce background noise. We replace the Bottleneck in C3k2 with AP-PConv to form the C3k2_AP module, inheriting the efficiency of PConv from FasterNet [20]. The structure is shown in Fig. 2.

The core innovation of AP-PConv is a lightweight gated directional priority mechanism. Given input feature map $X \in \mathbb{R}^{B\times C\times H\times W}$, GAP generates a 4D weight vector via global average pooling and 1×1 convolution:

$$W = \sigma(\mathrm{Conv}_{1\times 1}(\mathrm{GAP}(X))) \in \mathbb{R}^{B\times 4\times 1\times 1}, \qquad (1)$$

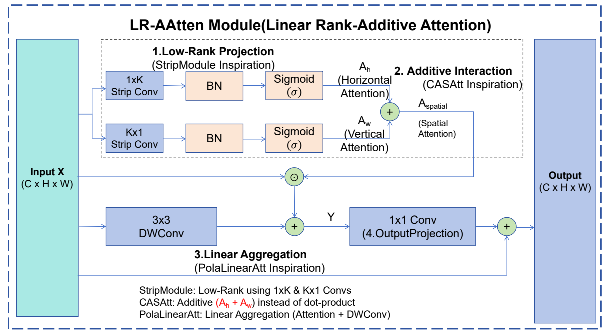


Fig. 3. Architecture of the decomposition-based LR-AAtten mechanism.

where σ is the Sigmoid function, and $W = [w_1, w_2, w_3, w_4]$ corresponds to the dynamic priority weights of four directional branches (left, right, up, down). The features are processed by asymmetric $(1 \times k)$ or $(k \times 1)$ kernels after zero-padding specific to each direction, and multiplied by their corresponding dynamic weights:

$$Y_i = \mathrm{Conv}_{dir_i}(\mathrm{ZeroPad}_i(X)) \cdot W_i. \qquad (2)$$

The outputs of all four branches are concatenated and fused via a 2×2 convolution. This design allows the network to adaptively allocate computational resources to the spatial directions most sensitive to the current pest morphology.

C. LR-AAtten Block

To overcome the $O(N^2)$ complexity of traditional Multi-Head Self-Attention, we propose the LR-AAtten module (Fig. 3). Inspired by the latest additive attention paradigm [10] and the theoretical analysis of feature entropy in PolarFormer [9], LR-AAtten abandons computationally intensive Matrix Multiplication and Softmax operations. Instead, it employs a convolutional additive token mixing strategy.

LR-AAtten decouples global spatial dependence modeling into the additive combination of horizontal and vertical low-

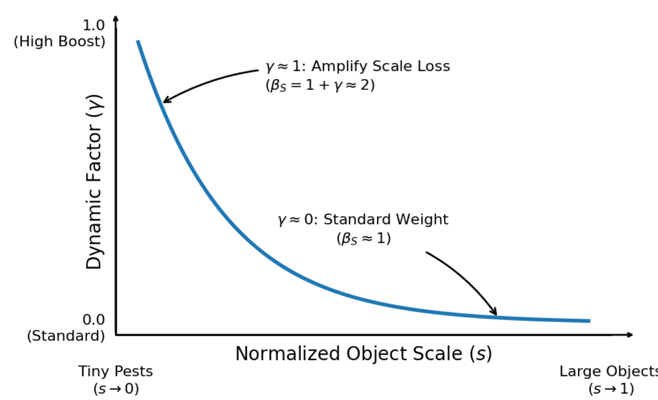


Fig. 4. The dynamic adjustment curve of factor γ. As the target scale decreases, γ increases exponentially to adaptively amplify the scale supervision.

rank components. For input X, directional context is extracted via two grouped depth-wise strip convolutions:

$$A_h = \sigma(\text{DWConv}_{1\times k}(X)), \quad (3)$$

$$A_w = \sigma(\text{DWConv}_{k\times 1}(X)), \quad (4)$$

where k is the strip kernel size. Subsequently, spatial attention is aggregated linearly ($A_{spatial} = A_h + A_w$). This additive operation simulates global interaction but reduces complexity to linear $O(N)$. The enhanced features modulate the original features via the attention map, supplemented by a local 3×3 depth-wise convolution for detail compensation.

D. Scale-Based Dynamic Loss (SD-Loss)

In field pest detection, targets often exhibit extreme scale variations. Traditional bounding box regression losses (e.g., CIoU or DIoU) usually employ static weight allocation. However, Yang et al. [7] pointed out that IoU metrics are highly sensitive to position offsets for tiny targets.

To address this, we adopt Scale-based Dynamic Loss (SD-Loss). SD-Loss introduces a dynamic adjustment factor γ based on the average normalized area s of targets in the current batch (Fig. 4):

$$\gamma = e^{-k\cdot s}. \quad (5)$$

Here, k is a tunable sensitivity hyperparameter (set to 0.5). As target size decreases ($s \to 0$), γ approaches 1, providing an adaptive signal to enhance the model's focus.

Based on this dynamic coefficient, we reformulate the traditional bounding box loss. Unlike previous methods that might suppress localization weights for small targets, we propose a **Scale-Augmented Strategy**:

$$\mathcal{L}_{SD} = \beta_{\mathcal{L}_S} \cdot \mathcal{L}_{scale} + \mathcal{L}_{location}. \quad (6)$$

TABLE I
DATASET SPLIT STATISTICS

Dataset	Total	Training	Validation	Test
IP102	18,974	15,182	1,898	1,899
FieldPest-25	14,714	11,262	2,300	1,152

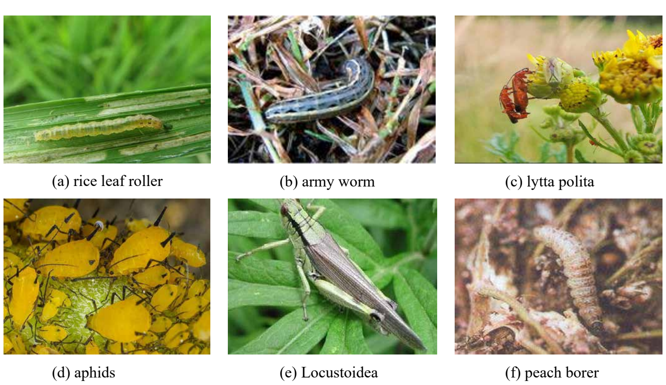


Fig. 5. Sample images from the datasets. (a)-(c) are from IP102, and (d)-(f) are complex field samples from FieldPest-25.

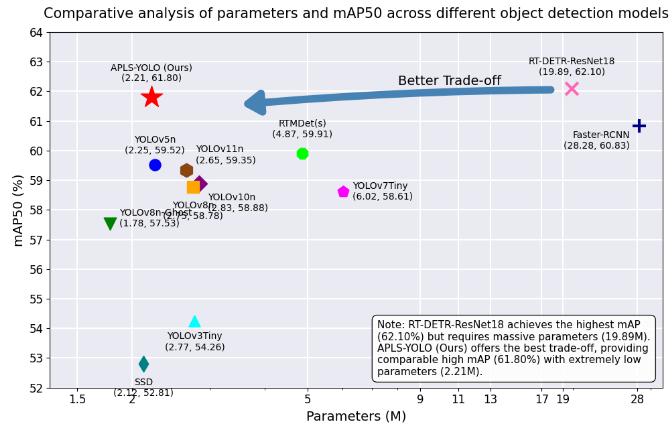


Fig. 6. Comparative analysis of parameters and mAP50. APLS-YOLO (Red Star) achieves the best trade-off.

TABLE II
PERFORMANCE COMPARISON SUMMARY

Dataset	Model	Prec.(%)	mAP50	Param(M)	Size(MB)
IP102	YOLOv11n	57.21	59.35	2.65	5.30
	APLS	**60.15**	**61.80**	**2.21**	**3.93**
FieldPest	YOLOv11n	56.63	58.77	2.62	5.20
	APLS	**58.22**	**60.31**	**2.18**	**3.88**

The scale weighting factor is assigned as $\beta_{\mathcal{L}_S} = 1 + \gamma$, while the location term retains a stable weight of 1. With this design, when the model processes tiny pest targets, the system automatically amplifies the gradient contribution of Scale Loss to prevent feature collapse, while maintaining consistent supervision for bounding box regression to ensure precise localization.

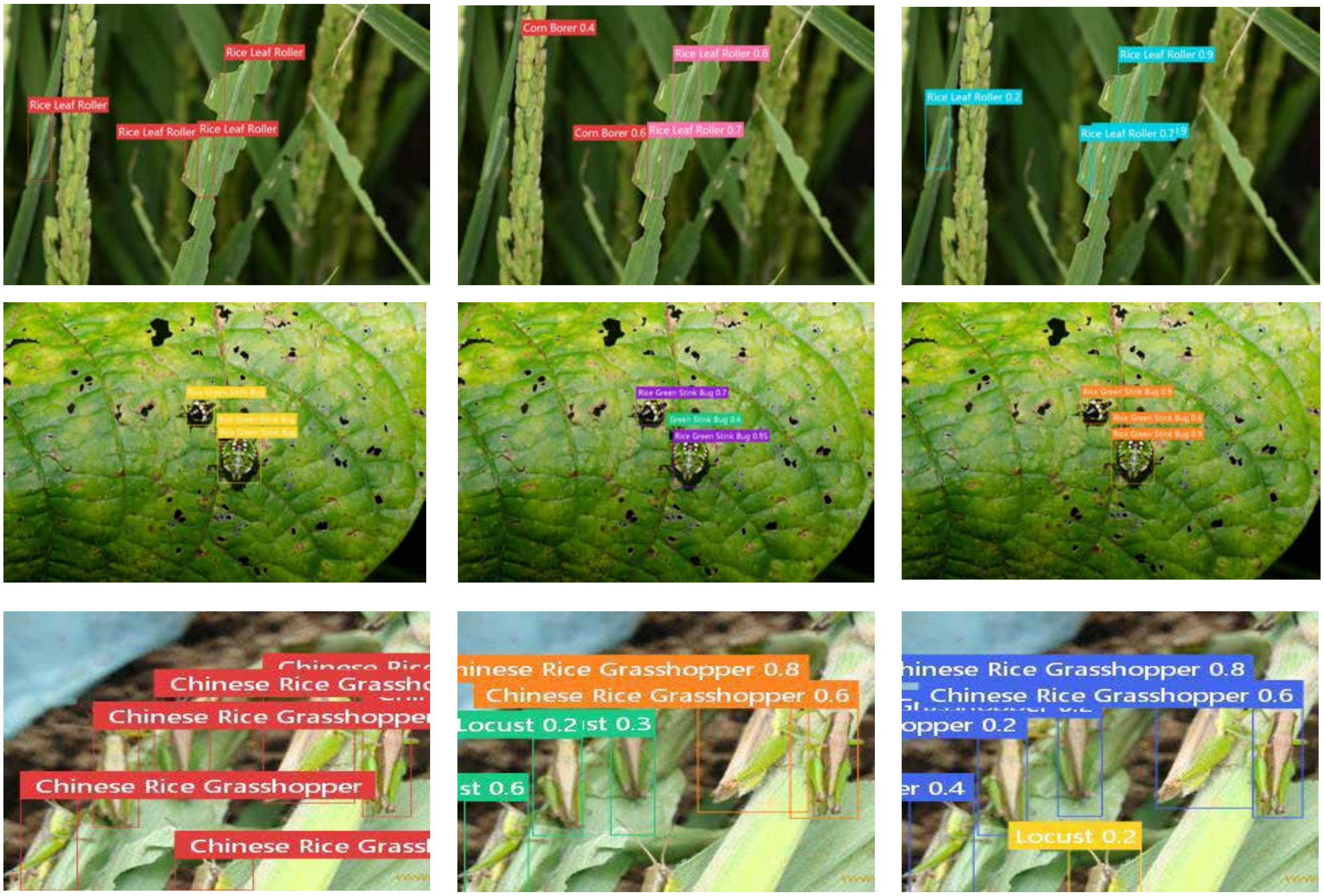
E. Global Channel Pruning

To compress the model for edge devices like the RK3588, we implement one-shot global channel pruning after training. This strategy utilizes the sparse training method proposed by Liu et al. [11]. Specifically, L1 regularization is applied to the scaling factors γ of BN layers during training. After training, channels are globally ranked based on importance scores $I_c = |\gamma_c|$, and those with $I_c < \tau$ are removed. The pruned model is fine-tuned briefly to recover accuracy, successfully compressing the model size to under 4MB.

IV. EXPERIMENTS AND ANALYSIS

A. Datasets

We utilize the **IP102** benchmark [12] and a self-built **FieldPest-25** dataset. IP102 contains approx. 75,000 images of 102 crop pest classes. For detection, it includes nearly 19,000 fully annotated images. FieldPest-25 focuses on 25 core pests of rice, corn, and wheat, containing 14,714 images (Fig. 5). It includes dense sub-pixel targets and highly similar crop backgrounds. Dataset splits are shown in Table I.



(a) Ground Truth (b) YOLOv11n (c) APLS-YOLO

Fig. 7. Visualization comparison of prediction results. (a) Ground Truth, (b) YOLOv11n, (c) APLS-YOLO. APLS-YOLO successfully detects dense and occluded pests.

TABLE III
ABLATION STUDY ON FIELDPEST-25

Base	C3k2_AP	LR-AAtten	SD-Loss	mAP50	Params(M)	FPS
✓				58.77	2.62	75.0
✓	✓			58.42	1.95	84.5
✓	✓	✓		59.85	2.18	79.0
✓	✓	✓	✓	**60.31**	2.18	79.0

B. Experimental Setup

Training was conducted on an NVIDIA RTX 4090 (24GB) using the Ultralytics framework [13],PyTorch, Python 3.10, and CUDA 12.2. Inference performance was tested on an Orange Pi 5 (RK3588). Images were resized to 640×640 with a batch size of 16. The initial learning rate was 0.01, momentum 0.937, and weight decay 0.0005.

C. Comparative Analysis

We evaluated APLS against state-of-the-art models including SSD [29], Faster R-CNN [30], YOLOv3 [31], YOLOv5 [32], YOLOv7 [33], RTMDet [34], and RT-DETR [35]on both datasets. As shown in Table II, on IP102, APLS-YOLO achieves 60.15% Precision, surpassing the YOLOv11n baseline by 2.94%. On the FieldPest-25 dataset, traditional models like YOLOv5n and YOLOv8n struggle with tiny pests. In contrast, APLS-YOLO demonstrates strong robustness, achieving 60.31% mAP thanks to the gradient balancing capability of SD-Loss. Fig. 6 illustrates the Parameters vs. mAP trade-off. APLS (red star) is in the top-left corner, indicating high accuracy with minimal parameters.

Detailed comparisons are provided in Table IV and Table V. APLS-YOLO achieves the best trade-off. Thanks to global channel pruning and AP-PConv, parameter count is reduced to 2.21M (16.7% reduction from YOLOv11n). On the RK3588, it reaches 79 FPS, a 6.9% speedup over the baseline. Fig. 7 visualizes detection results, showing reduced missed detections for dense small targets.

D. Ablation Study

To validate each component, we conducted ablation studies on FieldPest-25 (Table III). 1. Backbone Replacement (Exp II): Replacing C3k2 with C3k2_AP reduced parameters

TABLE IV
COMPARISON OF RESULTS ON IP102 DATASET

Model	Params(M)	Size(MB)	Prec.(%)	mAP50	mAP50:95	FLOPs(G)	FPS(RK3588)
YOLOv3Tiny	2.77	5.6	50.37	54.26	34.77	2.52	78
YOLOv5n	2.25	4.5	54.47	59.52	38.39	3.12	71
YOLOv7Tiny	6.02	12.1	53.82	58.61	37.15	3.41	68
YOLOv8n	2.75	5.5	53.50	58.78	38.78	3.62	67
YOLOv8n-Ghost	1.78	3.25	52.14	57.53	37.27	1.97	73
YOLOv10n	2.83	5.7	55.53	58.88	38.71	4.50	70
RTMDet(s)	4.87	9.8	56.52	59.91	39.20	4.17	64
Faster-RCNN	28.28	56.6	58.53	60.83	42.55	40.75	12
RT-DETR-R18	19.89	44.51	59.22	62.10	45.31	63.2	18
SSD	2.12	4.3	47.52	52.81	32.16	4.12	62
YOLOv11n	2.65	5.3	57.21	59.35	38.52	3.37	72
APLS-YOLO	**2.21**	**3.93**	**60.15**	**61.80**	**40.10**	**2.55**	**79**

TABLE V
COMPARISON OF RESULTS ON FIELDPEST-25 DATASET

Model	Params(M)	Size(MB)	Prec.(%)	mAP50	mAP50:95	FLOPs(G)	FPS(RK3588)
YOLOv3Tiny	2.74	5.5	49.17	53.16	33.57	2.48	81
YOLOv5n	2.22	4.4	53.20	58.35	37.14	3.08	74
YOLOv7Tiny	5.98	12.1	52.82	57.61	35.95	3.36	71
YOLOv8n	2.72	5.4	52.81	58.05	37.27	3.58	70
YOLOv8n-Ghost	1.75	3.2	51.50	57.20	36.13	1.94	75
YOLOv10n	2.80	5.6	54.82	58.42	37.61	4.45	73
RTMDet(s)	4.82	9.7	55.72	59.21	38.10	4.11	67
Faster-RCNN	28.20	56.4	58.13	60.33	41.54	40.65	13
RT-DETR-R18	19.85	44.4	59.97	61.84	44.31	63.1	20
SSD	2.09	4.2	46.22	51.61	30.96	4.08	65
YOLOv11n	2.62	5.2	56.63	58.77	37.46	3.32	75
APLS-YOLO	**2.18**	**3.88**	**58.22**	**60.31**	**38.52**	**2.51**	**79**

and increased FPS, with a slight expected mAP drop. 2. Neck Enhancement (Exp III): Adding LR-AAtten increased mAP to 59.85% with minimal parameter increase, proving its ability to capture global context. 3. Loss Optimization (Exp IV): Introducing SD-Loss further improved mAP to 60.31% without adding any inference cost. The results confirm the effectiveness of our pipeline.

V. CONCLUSION

In this paper, we propose the APLS-YOLO, a high-performance lightweight detector for agricultural pest detection. By introducing AP-PConv in the backbone, we reduced computational redundancy. The LR-AAtten in the neck enhances global context modeling via linear attention, while SD-Loss optimizes training for small targets. Validated on IP102 and FieldPest-25, APLS-YOLO achieves the best accuracy-efficiency trade-off. It enables real-time inference on edge devices like RK3588 while maintaining accuracy comparable to large Transformer models, offering a promising solution for automated precision agriculture.

ACKNOWLEDGEMENTS

This work was supported by the Central Guidance Fund for Local Science and Technology Development (No. 2023ZYDF078).

REFERENCES

[1] S. Savary, L. Willocquet, S. J. Pethybridge *et al.*, "The global burden of pathogens and pests on major food crops," *Nature Ecology & Evolution*, vol. 3, no. 3, pp. 430–439, 2019.

[2] B. Mo, Z. Zhang, C. Li *et al.*, "Field-oriented rice pest detection: Dataset construction and performance analysis," *Agronomy*, vol. 16, no. 1, p. 53, 2025.

[3] Y. Zhou, W. Huang, B. Liu *et al.*, "Lightweight GAFNet model for robust rice pest detection in complex agricultural environments," *AgriEngineering*, vol. 8, no. 1, p. 26, 2026.

[4] Q. Dong, L. Sun, T. Han *et al.*, "PestLite: A novel YOLO-based deep learning technique for crop pest detection," *Agriculture*, vol. 14, no. 2, p. 228, 2024.

[5] Z. Yan, Z. Chen, H. Li *et al.*, "Lightweight cotton field pest detection method based on improved YOLOv8s," *Jiangsu Agricultural Sciences*, 2025.

[6] Y. Huang, Z. Liu, H. Zhao *et al.*, "YOLO-YSTs: An improved YOLOv10n-based method for real-time field pest detection," *Agronomy*, vol. 15, no. 3, p. 575, 2025.

[7] J. Yang, S. Liu, J. Wu *et al.*, "Pinwheel-shaped convolution and scale-based dynamic loss for infrared small target detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 9, 2025, pp. 9202–9210.

[8] X. Yuan, Z. Zheng, Y. Li *et al.*, "Strip R-CNN: large strip convolution for remote sensing object detection," *arXiv preprint arXiv:2501.03775*, 2025.

[9] W. Meng, Y. Luo, X. Li *et al.*, "PolaFormer: Polarity-aware linear attention for vision transformers," *arXiv preprint arXiv:2501.15061*, 2025.

[10] T. Zhang, L. Li, Y. Zhou *et al.*, "CAS-ViT: Convolutional additive self-attention vision transformers for efficient mobile applications," *arXiv preprint arXiv:2408.03703*, 2024.

[11] Z. Liu, J. Li, Z. Shen *et al.*, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2736–2744.

[12] X. Wu, C. Zhan, Y. K. Lai *et al.*, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8787–8796.

[13] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," https://github.com/ultralytics/ultralytics, [Accessed: Jan. 28, 2026].

[14] A. Wang, H. Chen, L. Liu *et al.*, "YOLOv10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[15] J. Zhao, J. Li, L. Yang *et al.*, "Lightweight real-time detection algorithm for cotton seedlings under film based on PConv-CGLU and reparameterized detection head," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 41, no. 18, pp. 142–151, 2025.

[16] F. Li, Y. Lu, Q. Ma *et al.*, "GhostConv+ CA-YOLOv8n: a lightweight network for rice pest detection based on the aggregation of low-level features," *Frontiers in Plant Science*, vol. 16, p. 1620339, 2025.

[17] Y. Xia and S. Wang, "An object detection algorithm based on improved YOLOv10," *Journal of Hunan Institute of Science and Technology (Natural Sciences)*, vol. 38, no. 3, 2025.

[18] J. Li, Y. Xu, J. Lu *et al.*, "Lightweight multi-scale rice pest recognition model based on improved YOLOv11n," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 41, no. 17, pp. 175–184, 2025.

[19] M. H. Akhtar, I. Eksheir, and T. Shanableh, "Edge-optimized deep learning architectures for classification of agricultural insects with mobile deployment," *Information*, vol. 16, no. 4, p. 348, 2025.

[20] J. Chen, S. Kao, H. He *et al.*, "Run, don't walk: chasing higher FLOPS for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 021–12 031.

[21] Y. Wang, "Research on deep learning-based pest detection and model compression," Master's thesis, Central South University, Changsha, 2023.

[22] H. Zhang, X. Wu, X. Cai *et al.*, "Research on structured pruning method of neural network for visual recognition," *Science and Technology Innovation*, no. 19, pp. 135–138, 2025.

[23] P. Lv, H. Xu, Y. Zhang *et al.*, "An improved multi-scale feature extraction network for rice disease and pest recognition," *Insects*, vol. 15, no. 11, p. 827, 2024.

[24] H. Deng, Z. Wang, C. Yin *et al.*, "Lightweight small target fruit fly detection model based on KSGM-YOLO," *Jiangsu Agricultural Sciences*, vol. 53, no. 5, pp. 213–223, 2025.

[25] L. Jia, T. Wang, Y. Chen *et al.*, "MobileNet-CA-YOLO: An improved YOLOv7 based on the MobileNetV3 and attention mechanism for rice pests and diseases detection," *Agriculture*, vol. 13, no. 7, p. 1285, 2023.

[26] N. Kumar, Nagarathna, and F. Flammini, "YOLO-based light-weight deep learning models for insect detection system with field adaption," *Agriculture*, vol. 13, no. 3, p. 741, 2023.

[27] L. Yang, R. Y. Zhang, L. Li *et al.*, “SimAM: A simple, parameter-free attention module for convolutional neural networks,” in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021, pp. 11 863–11 874.

[28] Y. Tang, X. Hu, X. Luo *et al.*, “Dual-modal small target detection fusing receptive field and cross-scale global modeling,” *Infrared Technology*, 2025.

[29] W. Liu, D. Anguelov, D. Erhan *et al.*, “SSD: Single shot multibox detector,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[30] S. Ren, K. He, R. Girshick *et al.*, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[31] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[32] G. Jocher, “YOLOv5 by Ultralytics,” https://github.com/ultralytics/yolov5, [Accessed: Jan. 28, 2026].

[33] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475.

[34] C. Lyu, W. Zhang, H. Huang *et al.*, “RTMDet: An empirical study of designing real-time object detectors,” *arXiv preprint arXiv:2212.07784*, 2022.

[35] W. Lv, Y. Zhao, S. Xu *et al.*, “DETRs beat YOLOs on real-time object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16 965–16 974.