Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 21, No. 2, 2026

Research and Implementation of an Agricultural Pest and Disease Detection Algorithm Based on Deep Learning

Xuesong Sha*

College of Computer Science and Artificial Intelligence, Southwest Minzu University, Chengdu, China

*** Corresponding author**

Abstract: Food security is increasingly threatened by crop pests and diseases, while conventional field inspection remains labor intensive, experience dependent, and prone to missed detection during early disease stages. To support accurate and real-time pest and disease monitoring in precision agriculture, this study proposes RSC-YOLOv11-EMA, an improved lightweight detection framework for soybean leaf pest and disease images. First, YOLOv11 is selected as the baseline after comparison with YOLOv5 and YOLOv8 on the same dataset. Second, the backbone network is reconstructed with RepVGG so that multi-branch training can be converted into a single-path inference structure, reducing computational cost for edge deployment. Third, SENet channel attention is embedded into multi-scale feature outputs to strengthen discriminative lesion channels and suppress background interference. Fourth, K-means++ is used to analyze the true scale distribution of lesions and guide scale-aware augmentation for small targets. On this basis, an Efficient Multi-scale Attention module, warm-up cosine annealing, and exponential moving average weight updating are introduced to improve spatial sensitivity, convergence stability, and generalization on a small-sample dataset. Experiments show that the final RSC-YOLOv11-EMA model achieves 93.6% precision, 90.4% recall, 91.2% mAP@0.5, and 31.2 FPS, improving mAP@0.5 by 6.5 percentage points over the YOLOv11 baseline while preserving real-time inference. A PyQt5-based detection system is further implemented, with an end-to-end response time of 145 ms on edge equipment. The results demonstrate that the proposed method provides an effective and deployable solution for early agricultural pest and disease detection.

Keywords: Deep learning; Agricultural pest and disease detection; YOLOv11; RSC-YOLOv11-EMA; Small-object detection; Precision agriculture.

1. Introduction

Crop pests and diseases are major constraints on yield stability, product quality, and sustainable agricultural production. Manual scouting by plant-protection specialists is still widely used, but the method is slow, costly, and strongly influenced by observer experience. In early disease stages, tiny lesions and low-density pests are easily overlooked, which may lead to excessive pesticide use and delayed control decisions [1].

Computer vision and deep learning provide a practical route for automated monitoring in agricultural production [2]. Compared with threshold segmentation, morphology, and handcrafted classifiers, YOLO-style convolutional detectors can learn hierarchical texture, color, edge, and semantic features directly from field images [3]. Their effectiveness has also been demonstrated in real-time plant disease and pest recognition scenarios [4]. However, agricultural deployment remains difficult because pest and disease targets are often small, visually similar, and embedded in complex backgrounds with leaf overlap, illumination variation, soil interference, and motion blur.

To address these constraints, this work focuses on soybean leaf pest and disease detection and improves YOLOv11 from three dimensions: lightweight inference, feature discrimination, and small-target scale adaptation. The resulting RSC-YOLOv11 is further optimized with an Efficient Multi-scale Attention module and stable training strategy to produce RSC-YOLOv11-EMA. The study also implements a detection system to validate the engineering feasibility of the model [3].

2. Dataset and Baseline Model

The experimental dataset contains five soybean leaf categories: Health, Late blight, Rust, Frog eye, and Aphid infestation. Each category initially contains 3,100 images. Data augmentation expands the number of images in each class to 9,300, improving diversity for small-sample training and reducing overfitting risk.

Table 1. Dataset distribution before and after augmentation

Class	Original images	Images after augmentation
Health	3,100	9,300
Late blight	3,100	9,300
Rust	3,100	9,300
Frog eye	3,100	9,300
Aphid infestation	3,100	9,300

Training and evaluation are conducted under a unified environment: Windows 11, Intel Core i9-13900H CPU, NVIDIA GeForce RTX 4060 Laptop GPU with 8 GB memory, CUDA 12.1, Python 3.10, PyTorch 2.3.1, and Ultralytics 8.2.0. The models are trained for 200 epochs with an input resolution of 640 x 640 and a batch size of 8.

Table 2. Baseline comparison of YOLO models

Model	P (%)	R (%)	mAP@0.5 (%)	FPS
YOLOv5	84.2	84.9	80.4	32.7
YOLOv8	85.1	85.8	82.6	32.1
YOLOv11	90.8	87.6	84.7	31.4

YOLOv11 achieves the best accuracy among the baseline models, with mAP@0.5 reaching 84.7%. Although its frame rate is slightly lower than YOLOv5 and YOLOv8, it still satisfies real-time monitoring requirements above 30 FPS. Therefore, YOLOv11 is adopted as the base architecture for subsequent optimization.

3. RSC-YOLOv11 Method

3.1. RepVGG Backbone Reconstruction

The original YOLOv11 backbone is replaced with RepVGG blocks to reduce inference complexity. During training, RepVGG uses parallel 3 x 3 convolution, 1 x 1 convolution, and identity branches to improve feature learning and gradient flow. During deployment, these branches are re-parameterized into a single VGG-style path, reducing latency and computational burden while preserving representational capacity [5].

3.2. SENet Channel Attention

SENet is inserted after multi-scale backbone outputs before neck fusion. It performs global average pooling, nonlinear channel interaction, and channel-wise recalibration, thereby strengthening lesion-related responses while suppressing background channels. Related attention designs such as CBAM show that spatial-channel recalibration can improve feature selectivity [6]. Efficient channel attention further demonstrates that lightweight attention can preserve efficiency while improving representation [7]. This design targets confusion among visually similar symptoms such as early Frog eye and Late blight spots.

3.3. K-means++ Scale-Prior Optimization

Because the dataset contains many tiny lesions, default scale assumptions may cause early disease targets to shrink into weak or even invalid pixel patterns during augmentation. K-means++ is used to cluster annotated box widths and heights at the 640 x 640 input scale. The best balance is obtained at $k = 9$, where the average best IoU reaches 0.689, providing useful scale priors for small, medium, and larger lesions.

Table 3. Scale-prior clustering comparison

Method	Average best IoU	IoU standard deviation	Remark
Traditional K-means	0.652	0.038	Random initialization is unstable
K-means++	0.689	0.012	D2 sampling improves stability and matching

4. RSC-YOLOv11-EMA Optimization

Although RSC-YOLOv11 improves feature extraction and channel selection, small-sample training still suffers from parameter fluctuation and insufficient spatial sensitivity. Therefore, the final model introduces an Efficient Multi-scale Attention module, warm-up cosine annealing, and exponential moving average weight updating.

The EMA module groups channels and models horizontal and vertical spatial context, allowing the model to capture lesion positions and fine-grained texture cues without large computational overhead [8]. Warm-up cosine annealing smooths the learning-rate trajectory and prevents abrupt learning-rate drops [9]. Exponential moving average weight updating filters batch-level noise and stabilizes the parameter path, improving generalization under limited and noisy annotations [10].

Table 4. Ablation results of optimization strategies

Model architecture	P (%)	R (%)	mAP@0.5 (%)	FPS
YOLOv11	90.8	87.6	84.7	31.4
RSC-YOLOv11	92.5	89.2	88.6	31.1
RSC-YOLOv11 + EMA module	93.2	89.9	90.5	31.1
RSC-YOLOv11 + cosine annealing	92.6	89.4	89.2	31.1
RSC-YOLOv11 + EMAW	92.8	89.6	89.4	31.1
RSC-YOLOv11 + cosine annealing + EMAW	93.0	89.8	89.8	31.1
RSC-YOLOv11-EMA	93.6	90.4	91.2	31.2

5. Results and Discussion

Table 5. Final comparison with mainstream and recent methods

Model architecture	P (%)	R (%)	mAP@0.5 (%)	FPS
YOLOv5	84.2	84.9	80.4	32.7
YOLOv8	85.1	85.8	82.6	32.1
YOLOv11	90.8	87.6	84.7	31.4
PEW-YOLO (Xue and Wang, 2025)	91.2	87.5	86.3	28.3
SP-YOLO (Tang et al., 2025)	90.5	86.8	85.9	30.2
RSC-YOLOv11	92.5	89.2	88.6	31.1
RSC-YOLOv11-EMA	93.6	90.4	91.2	31.2

The proposed RSC-YOLOv11-EMA achieves the strongest overall performance among all compared models. Recent pest and disease detectors provide useful comparison baselines, including SP-YOLO [11], PEW-YOLO [12], YOLO-PLNet [13], YOLO-RP [14], and EMA-YOLO [15]. Relative to YOLOv11, precision increases from 90.8% to 93.6%, recall increases from 87.6% to 90.4%, and mAP@0.5 increases from 84.7% to 91.2%. The frame rate remains essentially unchanged, confirming that the added attention and training strategies do not undermine real-time deployment.

The improvement is mainly attributed to complementary optimization. RepVGG lowers model complexity and improves deployment feasibility; SENet enhances discriminative channels for similar disease symptoms; K-means++ scale priors improve small-target matching; and EMA with stable training mechanisms improves spatial feature learning and convergence. Together, these components improve small-lesion recall and reduce background-induced false detections.

6. Detection System Implementation

A pest and disease detection system is implemented to convert the trained model into a practical tool. The system adopts a client/server architecture. The client is developed with PyQt5 for image loading, result display, and report generation, while the backend loads the trained RSC-YOLOv11-EMA model with PyTorch and communicates with a MySQL database for user and detection records.

Table 6. End-to-end system performance on edge equipment

Component	Latency
Image transmission	28 ms
Preprocessing	35 ms
Model inference	32 ms
Post-processing	25 ms
Visualization	25 ms
Total response time	145 ms
End-to-end throughput	6.9 FPS
Peak memory usage	890 MB

The pure model inference speed is 31.2 FPS under the GPU test setting, whereas the full system throughput is 6.9 FPS because image input, preprocessing, post-processing, visualization, I/O, and edge-device resource constraints are included in the end-to-end measurement. The measured response time still supports field-oriented pest and disease monitoring and early warning tasks.

7. Conclusion

This paper presents RSC-YOLOv11-EMA, a lightweight and accurate detection model for agricultural pest and disease monitoring. The method builds on the one-stage detection idea of YOLO [3], uses RepVGG backbone reconstruction for efficient inference [5], introduces Efficient Multi-scale Attention for cross-spatial feature enhancement [8], applies cosine annealing for smoother optimization [9], and uses exponential moving average weight updating to improve generalization [10].

Experimental results show that the final model reaches 93.6% precision, 90.4% recall, 91.2% mAP@0.5, and 31.2 FPS. Compared with the YOLOv11 baseline, mAP@0.5 improves by 6.5 percentage points while real-time performance is maintained. The PyQt5 detection system further confirms that the model can be integrated into practical agricultural monitoring workflows.

Future work should evaluate cross-region and multi-crop generalization, introduce multispectral or thermal information, explore semi-supervised and self-supervised training with unlabeled field images, and further compress the model for lower-power embedded hardware.

Acknowledgment

The author thanks the supervisors and colleagues who supported the research and system implementation work. The original thesis was completed at the College of Computer Science and Artificial Intelligence, Southwest Minzu University.

References

[1] Ngugi, L. C., Abelwahab, M., & Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition: A review. Information Processing in Agriculture, 8(1), 27-51. https://doi.org/10.1016/j.inpa.2020.04.003

[2] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70-90.

https://doi.org/10.1016/j.compag.2018.02.016

[3] 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). IEEE.

https://doi.org/10.1109/CVPR.2016.91

[4] Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17(9), 2022. https://doi.org/10.3390/s17092022

[5] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). RepVGG: Making VGG-style ConvNets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13733-13742). IEEE.

https://doi.org/10.1109/CVPR46437.2021.01352

[6] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (pp. 3-19). Springer. https://doi.org/10.1007/978-3-030-01234-2_1

[7] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11534-11542). IEEE.

https://doi.org/10.1109/CVPR42600.2020.01155

[8] Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., Zhan, J., & Huang, Z. (2023). Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 1-5). IEEE.

https://doi.org/10.1109/ICASSP49357.2023.10096512

[9] Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. arXiv, arXiv:1608.03983. https://doi.org/10.48550/arXiv.1608.03983

[10] Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., & Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. arXiv, arXiv:1803.05407. https://doi.org/10.48550/arXiv.1803.05407

[11] Tang, K., Qian, Y., Dong, H., Zhang, Y., Liu, S., & Wang, Z. (2025). SP-YOLO: A real-time and efficient multi-scale model for pest detection in sugar beet fields. Insects, 16(1), 102. https://doi.org/10.3390/insects16010102

[12] Xue, R., & Wang, L. (2025). Research on lightweight citrus leaf pest and disease detection based on PEW-YOLO. Processes, 13(5), 1365. https://doi.org/10.3390/pr13051365

[13] Sun, J., Feng, Z., Han, J., Liu, Y., Wang, X., & Li, H. (2025). YOLO-PLNet: A lightweight real-time detection model for peanut leaf diseases based on edge deployment. Frontiers in Plant Science, 16, 1707501.

https://doi.org/10.3389/fpls.2025.1707501

[14] Yang, X., He, Q., Xie, X., Liu, Y., & Zhang, W. (2025). YOLO-RP: A lightweight and efficient detection method for small rice pests in complex field environments. Symmetry, 17(10), 1598. https://doi.org/10.3390/sym17101598

[15] Xu, D., Xiong, H., Liao, Y., Zhao, Z., Liu, T., & Chen, J. (2024). EMA-YOLO: A novel target-detection algorithm for immature yellow peach based on YOLOv8. Sensors, 24(12), 3783. https://doi.org/10.3390/s24123783