YOLOv8-Based Visual Detection System for Taro Peeling State Assessment

Jianxin Dong[1]

[1]College of Mechanical & Electrical Engineering, Inner Mongolia Agricultural University, Hohhot 010018, Inner Mongolia, P. R. China

*Corresponding author: dongjianxin@imau.edu.cn

Abstract: Taro peeling quality affects the consistency of downstream washing, cutting, sorting, and prepared-food processing. Manual inspection is simple but difficult to scale because taro roots have irregular shapes, mottled brown skin, moist surfaces, and variable illumination during handling. This paper presents a YOLOv8-based visual detection system for assessing taro peeling state. A custom dataset of 1246 images was collected from web images and local photography, annotated into three states: completely unpeeled, partial peeling, and full peel. The dataset was divided into 873 training images, 187 validation images, and 186 test images. A YOLOv8n detector was trained for 100 epochs on a workstation equipped with an NVIDIA GeForce RTX 4050 Laptop GPU. Experimental results from the project records show strong convergence and high precision-recall performance, with validation mAP@0.5 of 0.992 and combined test/validation mAP@0.5 of 0.988. The partial-peeling category is the most challenging class, which is consistent with its boundary ambiguity between remaining skin and exposed flesh. A PySide6/PyQt6 graphical interface and a conceptual mechanical inspection line are also described to connect the detector with practical image and video inspection workflows. The study demonstrates the feasibility of lightweight deep-learning detection for taro peeling assessment while identifying dataset diversity, timing measurement, and production-line validation as necessary next steps.

Keywords: taro peeling; YOLOv8; object detection; agricultural product inspection; machine vision; graphical user interface

I Introduction

Taro is a widely consumed root crop and an important raw material for fresh retail, catering, and prepared-food processing. The local market research material used in this study reports increasing demand for taro products and highlights supply-chain standardization, quality inspection, and intelligent processing as important development directions [1]. Although those market values are treated here as project background rather than independently verified statistics, they motivate a practical technical need: taro products must be inspected quickly and consistently as processing volume grows.

Peeling state is one of the visible quality cues in taro processing. Completely unpeeled

roots may require additional peeling, fully peeled roots can move to later processing, and partially peeled roots require special attention because they may contain both edible exposed flesh and residual skin. Traditional visual inspection depends on workers' experience and can be affected by fatigue, lighting, water, and taro surface texture. An automated system can reduce repetitive labor and provide more consistent decisions for sorting or rework.

Deep learning has become a common approach for agricultural visual inspection because convolutional detectors can learn features directly from field and processing images [2], [3].

YOLO-series detectors are especially attractive for inspection lines because they combine localization and classification in a single-stage inference pipeline. This work therefore investigates a lightweight YOLOv8n model for three-class taro peeling-state detection and places the trained detector inside a graphical inspection interface and a conceptual mechanical workflow.

The contributions of this paper are as follows:

• A taro peeling-state detection task is formulated with three practical classes: completely unpeeled, partial peeling, and full peel.

• A custom 1246-image dataset is summarized, including data sources, annotation, augmentation strategy, and train-validation-test split.

• YOLOv8n training and evaluation results are reported from the project records, including class-level PR behavior and observed partial-peeling difficulty.

• A software interface and mechanical inspection concept are described to connect the detector with image, video, and production-line use.

II Related Work

Machine vision has long been used for agricultural product grading, but deep learning has reduced the need for hand-designed color, texture, or shape descriptors. Survey studies show that deep neural networks are effective for crop monitoring, plant phenotyping, disease recognition, and product quality assessment when adequate annotated data are available [2], [3]. For food and agricultural inspection, the key engineering challenge is not only accuracy on a test set but also robustness to surface variation, occlusion, background clutter, and lighting shifts.

The YOLO family is widely used for real-time object detection. YOLOv7 improved detector training through trainable bag-of-freebies and achieved strong speed-accuracy tradeoffs [4]. YOLOv6 was designed for industrial deployment scenarios and emphasized efficient single-stage detection [5]. YOLOv8, released by Ultralytics, provides models for detection, segmentation, classification, pose estimation, and tracking, and its detector uses an anchor-free design with decoupled prediction heads [6]. These characteristics make YOLOv8 suitable for small agricultural inspection systems where a lightweight model and easy deployment are valuable.

Compared with general object detection, taro peeling-state assessment is visually subtle. The object category remains the same, while the relevant state depends on how much of the skin has been removed. The partial-peeling class is naturally ambiguous because the boundary between residual peel, exposed flesh, moisture, and shadows is gradual. This study focuses on detecting practical peeling-state regions rather than estimating a continuous peeling ratio.

III Materials and Methods

A. Dataset and Labels

The dataset was built from Baidu image search, Google image search, and local photography, as recorded in the project technical report [7]. Images were annotated with X-AnyLabeling [8]. Three target classes were used:

• Completely-unpeeled: taro roots whose outer skin remains substantially intact.

• Partial-peeling: taro roots or regions where both residual skin and exposed flesh are visible.

• Full-peel: taro roots or pieces whose skin has been removed.

The dataset contains 1246 images. The class distribution and data split are shown in Table I. Basic image transformations, including blur, rotation, and related augmentation operations, were used to improve generalization to acquisition variation.

Table I. Dataset Summary

Item	Count	Share
Completely-unpeeled images	394	31.6%
Partial-peeling images	382	30.7%
Full-peel images	470	37.7%
Training split	873	70.1%
Validation split	187	15.0%
Test split	186	14.9%

B. Detector

YOLOv8n was selected as the detector because it is the smallest standard YOLOv8 variant and is suitable for lightweight deployment. YOLOv8 adopts a modern one-stage detection pipeline with an efficient backbone, decoupled detection head, and anchor-free prediction structure [6]. The model was trained to predict bounding boxes and peeling-state class labels for each taro instance or visible region.

Training used Python 3.8.20, PyTorch 1.10.0 [9], CUDA 12.1, and an NVIDIA GeForce RTX 4050 Laptop GPU with 6140 MiB memory. The project record reports 100 training epochs. The available materials do not include a measured end-to-end frames-per-second value, so this paper does not claim a specific real-time throughput.

C. System Workflow

The proposed workflow combines visual acquisition, object detection, and user-facing inspection. Images or video frames are provided to the trained YOLOv8n model. The detector outputs bounding boxes, class labels, confidence scores, and object counts. These outputs are displayed in a PySide6/PyQt6 interface [10] and can guide sorting or rework decisions. The mechanical concept in Fig. 1 shows a conveyor-oriented inspection line with feeding, handling, visual inspection, and downstream processing modules [11]. In the current project stage, this mechanical model is a design concept rather than a fully validated production deployment.

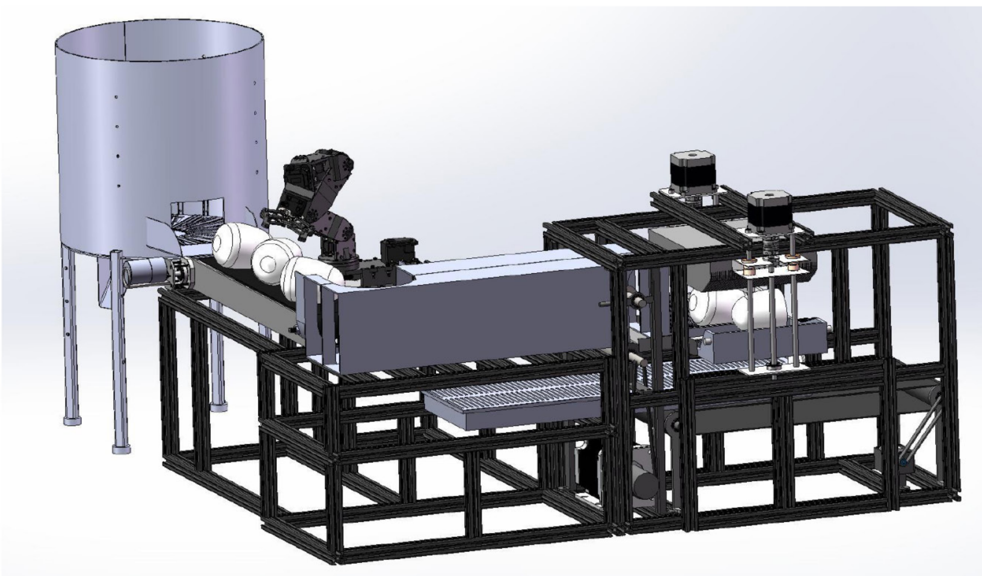

Figure 1. Conceptual mechanical model of the taro peeling inspection line, including feeding, conveying, handling, and

inspection modules.

D. Training Convergence

The training curves in Fig. 2 show that box loss, classification loss, and distribution focal loss decrease over the 100 epochs. Validation losses follow a similar decreasing trend, which suggests that the model learned transferable features rather than only memorizing the training split. Precision, recall, mAP@0.5, and mAP@0.5:0.95 rise quickly in the early epochs and then stabilize.

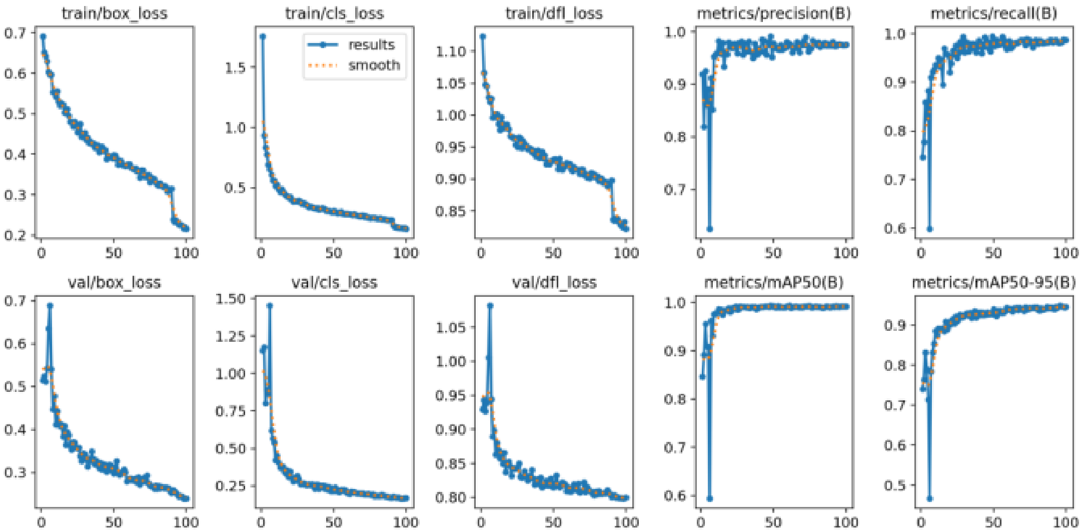

Figure 2. YOLOv8n training and validation curves over 100 epochs.

E. Precision-Recall Analysis

The validation PR curve in Fig. 3 reports class AP@0.5 values of 0.994 for completely unpeeled, 0.986 for partial peeling, and 0.995 for full peel, with an all-class mAP@0.5 of 0.992. The combined test/validation PR curve in Fig. 4 reports 0.992, 0.977, and 0.995 AP@0.5 for the same classes, with an all-class mAP@0.5 of 0.988.

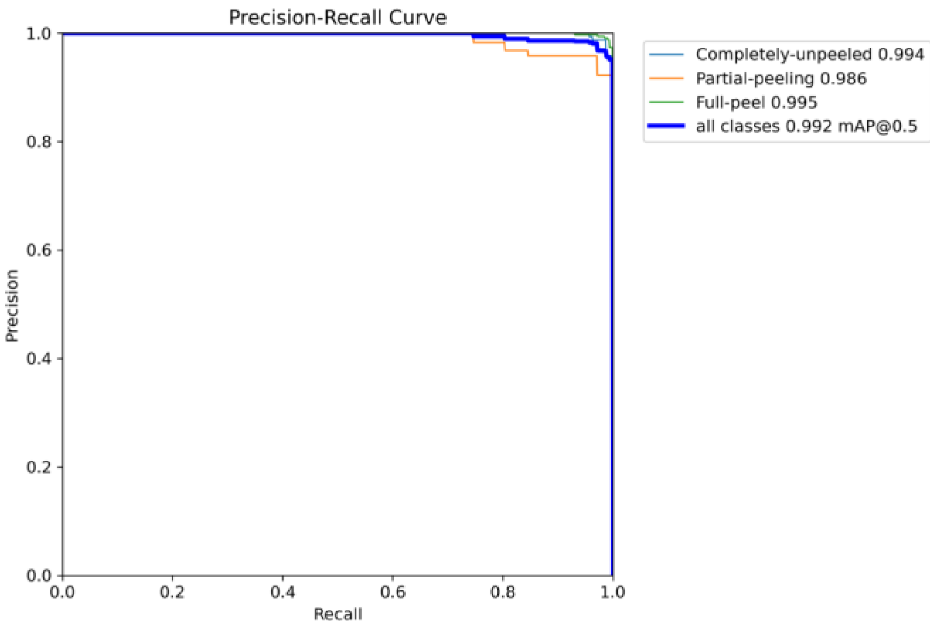

Figure 3. Validation precision-recall curve.

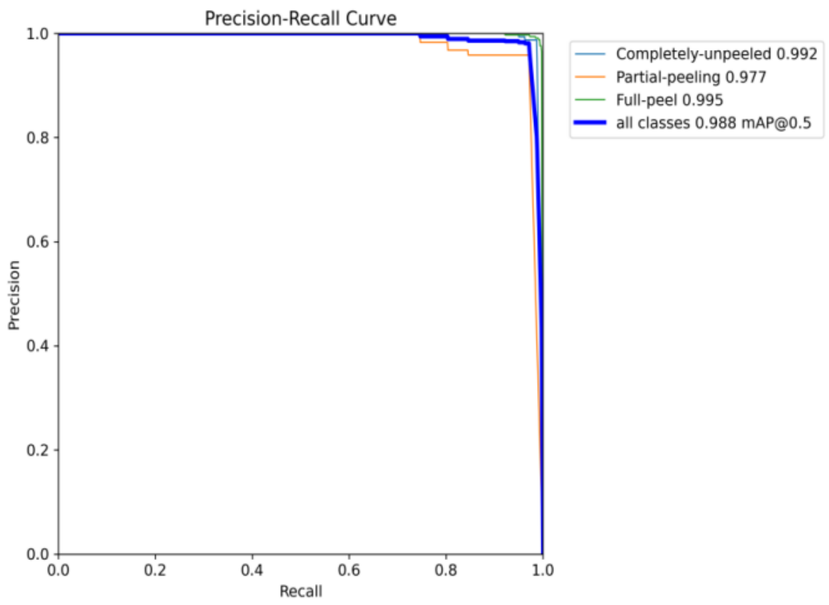

Figure 4. Combined test/validation precision-recall curve.

Table II summarizes the AP@0.5 values visible in the project PR figures. The partial-peeling class has the lowest AP in both evaluations. This is expected because partial peeling is a transitional state; a small exposed patch can overlap visually with either an unpeeled object or a fully peeled piece depending on crop, camera angle, wetness, and background.

Table II. AP@0.5 Values reported by the source pr curves

Class	Validation	Combined Test/Validation
Completely-unpeeled	0.994	0.992
Partial-peeling	0.986	0.977
Full-peel	0.995	0.995
All classes mAP@0.5	**0.992**	**0.988**

F. Detection Examples

Fig. 5 shows representative detection outputs. The detector correctly identifies multiple unpeeled and fully peeled samples in cluttered grid examples, and it also recognizes partial-peeling regions. Some outputs contain overlapping boxes or multiple detections for one visible taro region. These cases suggest that threshold tuning, non-maximum suppression settings, and additional annotation consistency checks would be useful before deployment.

Figure 5. Representative taro peeling-state detections from the project validation and test materials.

IV Experimental Results

A. Observed Error Modes

The qualitative results reveal several error modes that should be considered in later experiments. The first is duplicate localization, where a single taro root or visible region receives more than one bounding box. This does not always change the state-level decision, but it can distort object counts and therefore matters for automatic sorting or throughput statistics. The second is class competition between completely unpeeled and partial-peeling regions. A taro root with a small exposed area may be detected as unpeeled when the exposed flesh is partly occluded, while a peeled surface with shadows or residual fibers may be interpreted as partial peeling. The third is scale and background variation. Images collected from online sources, kitchen scenes, and local photography include different camera distances and backgrounds, whereas a conveyor system will require more controlled geometry.

These error modes suggest that future evaluation should report both detection accuracy and decision-level sorting accuracy. For example, a production system may only need to decide whether a root should be reworked, passed, or reviewed manually. In that setting, duplicate boxes and small false positives can be converted into higher-level rules, such as majority voting over frames or confidence-weighted class aggregation for each tracked object. The current dataset and project records are sufficient for demonstrating feasibility, but they do not yet establish those production decision rules.

A desktop graphical interface was developed with PySide6/PyQt6 to support model switching, image detection, video detection, and result logging. Fig. 6 shows image and video inspection modes. The interface displays the original input, the annotated output, class counts, and status messages. This makes the model usable by non-specialist operators and provides a bridge between training results and practical inspection tasks.

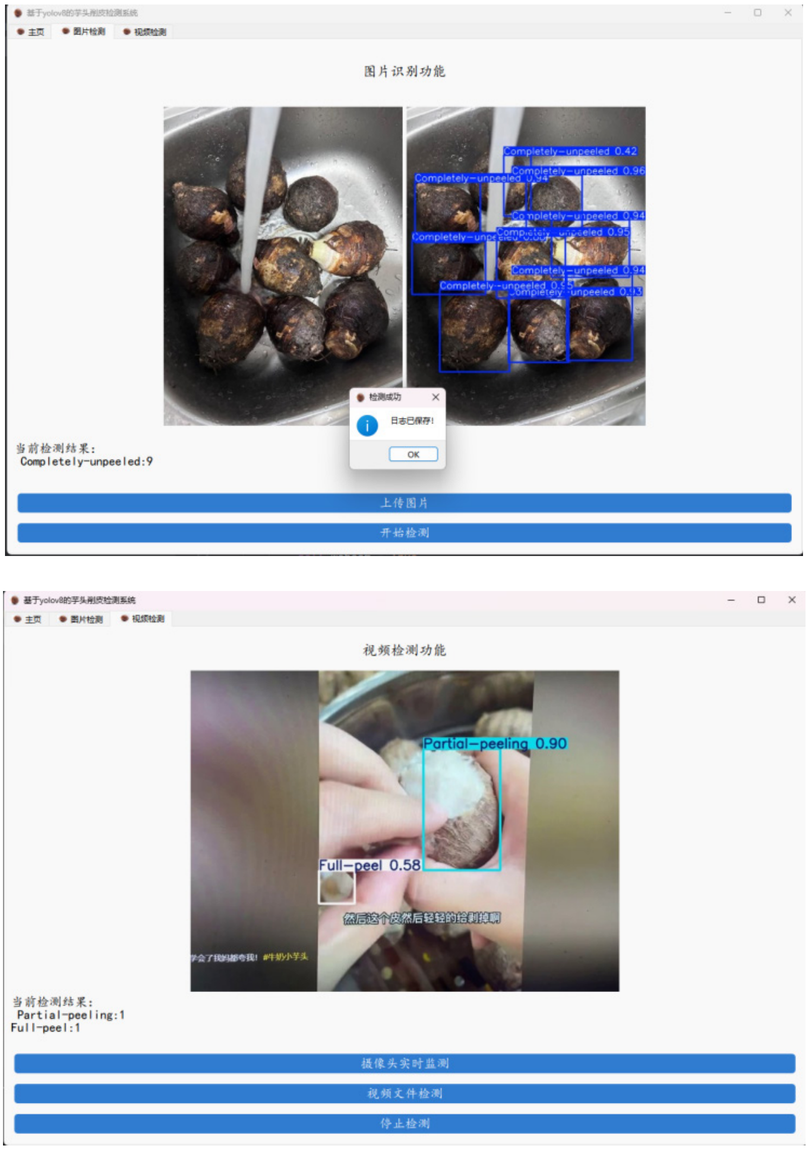

Figure 6. Graphical interface examples for image detection and video detection.

V Graphical Interface and System Integration

The conceptual inspection system can be summarized as follows. First, taro roots enter the system through a feeding and conveying structure. Second, images or frames are acquired at inspection positions. Third, the YOLOv8n model detects peeling states and outputs object-level labels and confidence values. Finally, the interface records results and can provide information for manual review or future automatic sorting. This architecture keeps the detector modular: the same model can be used for still-image testing, video analysis, and later integration with production hardware.

The additional mechanical renderings supplied with the project show the inspection zone in more detail. Fig. 7 illustrates a conveyor segment with a handling module and a guided passage where taro roots can be presented to the vision module. This layout suggests two practical requirements for later physical validation. First, the camera should observe each root with minimal self-occlusion, which may require controlled spacing or rotation before detection. Second, the lighting and background around the inspection zone should be standardized so that wet skin, exposed white flesh, and shadows do not create unstable class boundaries. These requirements are consistent with the partial-peeling errors observed in the PR analysis.

For engineering deployment, the detector should be treated as one module in a larger inspection pipeline. The image-acquisition module should control exposure, motion blur, field of view, and trigger timing. The inference module should provide class labels, confidence scores, and bounding-box geometry. The decision module should translate detections into operational actions, such as pass, re-peel, reject, or manual review. Finally, the logging module should preserve input frames, annotated outputs, timestamps, and model versions so that later failures can be audited. The current GUI already implements the first step toward this workflow by exposing image/video modes and saving detection records. A production version would also need hardware synchronization, safety interlocks, and a measured latency budget.

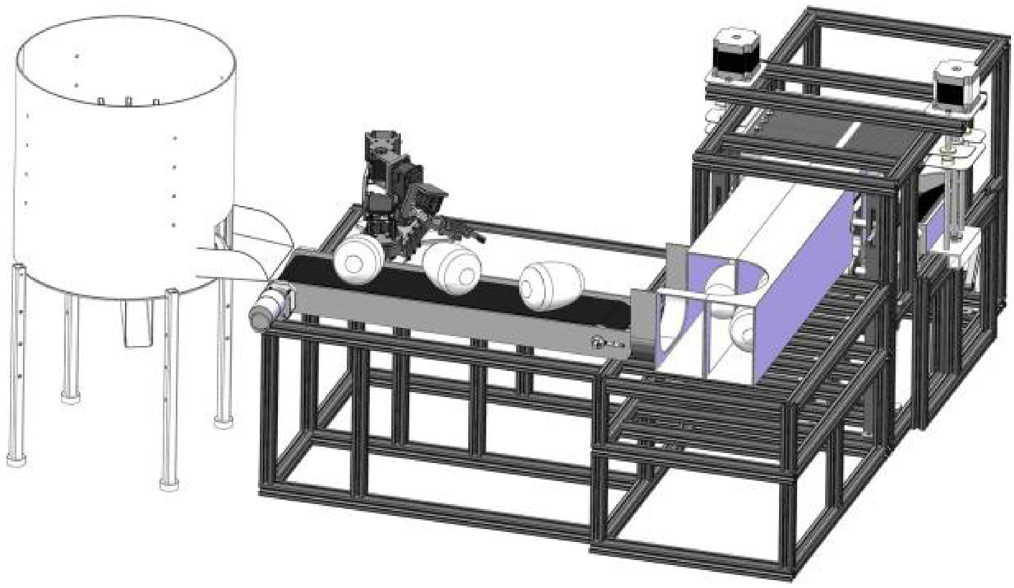

Figure 7. Detailed mechanical rendering of the conveyor and handling region used to contextualize production-line integration.

VI Discussion

The results indicate that a lightweight object detector can learn useful features for taro peeling-state recognition from a relatively small custom dataset. The high validation and combined test/validation mAP@0.5 values show that the detector is promising for proof-of-concept inspection. However, several limitations remain.

First, the dataset combines web images and local photographs. This improves visual variety but may introduce distribution differences between curated online images and production-line images. Second, the partial-peeling state is semantically ambiguous. A future dataset should define stricter labeling rules, possibly including peeling ratio intervals or segmentation masks. Third, the project records do not provide measured inference latency, camera frame rate, lighting conditions, or continuous conveyor tests. These measurements are required before making claims about real-time industrial performance. Finally, the GUI demonstrates usability, but integration with actuators, reject mechanisms, and production safety controls remains future work.

VII Conclusion

This paper presented a YOLOv8-based visual detection system for taro peeling-state assessment. Using a 1246-image custom dataset with three peeling-state classes, a YOLOv8n model was trained for 100 epochs and achieved mAP@0.5 values of 0.992 on the validation PR curve and 0.988 on the combined test/validation PR curve. The detector was integrated into a PySide6/PyQt6 graphical interface for image and video inspection, and a mechanical inspection-line concept was introduced to show how the method could support downstream automation. Future work will focus on larger production-line datasets, clearer partial-peeling annotations, segmentation-based peel-area estimation, latency measurement, and closed-loop sorting validation.

References

[1] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70-90, 2018.

[2] D. I. Patricio and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," Computers and Electronics in Agriculture, vol. 153, pp. 69-81, 2018.

[3] 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.

[4] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," arXiv preprint arXiv:2209.02976, 2022.

[5] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[6] L. Yu, X. Feng, Y. Zeng, and Y. Guo, "MDB-YOLO: A Lightweight, Multi-Dimensional Bionic YOLO for Real-Time Detection of Incomplete Taro Peeling," Electronics, vol. 15, no. 1, p. 97, 2025.

[7] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-Time End-to-End Object Detection," arXiv preprint arXiv:2405.14458, 2024.

[8] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," arXiv preprint arXiv:2402.13616, 2024.

[9] F. Xiao, Z. Liu, and Y. Zhang, "Fruit Detection and Recognition Based on Deep Learning for Automatic Harvesting: A Review," Agronomy, vol. 13, no. 6, p. 1625, 2023.

[10] P. Pugazhendi, et al., "Advances in agricultural fruit detection using you only look once (YOLO) algorithms: A review," Engineering Applications of Artificial Intelligence, vol. 137, p. 109123, 2024.

[11] D. G. Ayalde, et al., "AI-powered detection and quantification of post-harvest diseases in root crops using YOLO-driven deep learning," Computers and Electronics in Agriculture, vol. 220, p. 108890, 2024.

[12] A. Osipov, V. Shumaev, and S. Suvorov, "Identification and Classification of Mechanical Damage During Continuous Harvesting of Root Crops Using Computer Vision Methods," IEEE Access, vol. 10, pp. 28885-28894, 2022.

[13] H. Wu, R. Zhu, H. Wang, X. Wang, J. Huang, and S. Liu, "Flaw-YOLOv5s: A Lightweight Potato Surface

Defect Detection Algorithm Based on Multi-Scale Feature Fusion," Agriculture, vol. 15, no. 4, p. 875, 2025.

[14] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3-19.

[15] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proceedings of the International Conference on Machine Learning (ICML), 2019, pp. 6105-6114.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 8024-8035.

[17] C. Yang, et al., "Hyperspectral Imaging and Deep Learning for Quality and Defect Detection in Agricultural Products: A Review," Trends in Food Science & Technology, vol. 145, p. 104789, 2024.