Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Tomato maturity detection in natural scenes using an improved CenterNet

Guojie Chen

1825605756@qq.com

Wuhan Polytechnic University

Research Article

Keywords: tomato maturity, object detection, multiscale feature fusion, bounding-box regression, precision agriculture

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11772974/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Tomato maturity detection in natural scenes using an improved CenterNet

Guojie Chen[1*]

[1*]School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan, 430023, China.

Corresponding author(s). E-mail(s): 1825605756@qq.com;

Abstract

Natural-scene tomato maturity detection must localize individual fruits and distinguish the adjacent unripe, half-ripe, and ripe stages under occlusion, foliage interference, scale variation, uneven illumination, and gradual colour transitions. This study develops and evaluates an improved CenterNet detector that combines a ResNet-50 backbone with weighted skip fusion (WSF), pyramidal convolution (PyConv), a receptive field block (RFB), distribution focal loss (DFL), and a WIoU-style bounding-box term. Images collected at a tomato cultivation base in Guigang, China were divided at the image level into 2,828 training, 314 validation, and 784 test images. The model was trained with a 640 x 640 input for 200 epochs and evaluated together with ablations, class-level average precision, representative detector comparisons, and qualitative error analysis. On the reported test split, the final configuration achieved precision 0.9054, recall 0.8591, average precision at IoU 0.50 (AP50) 0.9226, and mean average precision averaged over IoU thresholds from 0.50 to 0.95 (mAP50:95) 0.8472. The results support the evaluated task-oriented integration of feature fusion, multiscale context modelling, and localization supervision, while the single-project dataset and single recorded seed limit conclusions about external generalisation.

Keywords: tomato maturity, object detection, multiscale feature fusion, bounding-box regression, precision agriculture

1 Introduction

Tomato maturity influences harvest timing, grading, storage, transportation, and market value. Manual assessment from peel colour and visible appearance can vary with operator experience, illumination, fatigue, and workload. Automated vision can support fruit localization and maturity classification as one stage in potential harvesting, grading, and field-monitoring systems.

Natural-scene maturity detection is more demanding than classifying isolated fruit crops. A detector must localize each visible fruit and assign one of the adjacent unripe, half-ripe, or ripe classes while handling leaf occlusion, fruit overlap, clutter, uneven illumination, and large differences in apparent scale. The transition between maturity stages is continuous; therefore, neighbouring classes can share colour and texture cues. Localization errors may also propagate to downstream picking or grading decisions.

Recent tomato-ripeness studies have pursued complementary detector designs. Wang et al. proposed a lightweight tomato-ripeness detector based on an improved RT-DETR, modifying the feature extractor, attention, neck, and localization

components to balance accuracy and computational cost [1]. Zhao et al. presented YOLO-DGS, a YOLOv10-based approach for tomato-ripeness detection in natural environments, with changes to feature representation and multiscale fusion [2]. These studies focus on transformer- and YOLO-family redesigns, respectively, whereas this work evaluates a ResNet50-based centre-point detector with weighted cross-level fusion, PyConv, RFB, and distributional width-height supervision. Since datasets, class definitions, and evaluation protocols differ, their published scores are not treated as direct benchmarks for the present results.

Recent tomato detectors have extended their objectives beyond maturity classification alone. Efficient-Tomato examines cherry tomato ripeness and disease detection [3], while TMAF-YOLO examines maturity and defective fruits in greenhouses [4]. Lightweight greenhouse and orchard designs include LGD-YOLO, a task-adaptive YOLO, YOLO-FLBM, and a real-time greenhouse ripeness model [5–8]. These studies provide relevant design contexts, but their published scores are not direct benchmarks for the present dataset.

Other related directions include a coupled Sequencer-MLP model for natural-environment ripeness detection [9] and RGB-D multimodal maturity detection in YOLO-TMD [10]. Occlusion and variable illumination are explicit concerns in the method of Fu et al. [11], whereas YOLO-RCM addresses tomato maturity detection in complex greenhouses [12]. A separate robotic study integrates YOLOv8n-BiFPN-WIoU with RGB-D perception and ROS-based control [13]. In contrast, the present work evaluates image-based fruit localization and three-stage maturity recognition; it does not evaluate robot control or an RGB-D pipeline.

CenterNet adaptations have also been investigated for tomato leaf disease, rail defects, and forest pest imagery [14–16], field peanut seedlings and banana leaf disease [17, 18], and multi-leaf disease and invasive-plant identification [19]. Related applications include low-light multimodal detection, wind-turbine surface inspection, and insulator defect recognition [20–22]. These applications differ from fruit maturity recognition in their targets and image conditions; their relevance here is methodological rather than evidence of transfer to the Guigang dataset.

CenterNet represents objects by their centre points and predicts a class heatmap, centre offsets, and box dimensions [23]. FCOS provides another established anchor-free single-stage formulation [24]. The centre-based CenterNet formulation provides a useful basis for examining feature fusion and localization, but a single deep feature path can underuse shallow boundary and colour details, while large scale variation and partial occlusion complicate feature representation and box regression. The present study examines whether a task-oriented combination of cross-level fusion, multiscale context, and localization losses is useful for three-stage tomato detection on the available natural-scene dataset. It does not claim that these established components are new algorithms or that one dataset establishes generalization to other farms or seasons.

The contributions are as follows:

1. A three-class, centre-based CenterNet workflow is evaluated for simultaneous localization and maturity recognition of unripe, half-ripe, and ripe tomatoes in natural scenes.
2. WSF, PyConv, and RFB are integrated to address, respectively, cross-level detail retention, scale variation, and contextual ambiguity from occlusion; DFL and a WIoU-style term are used as complementary localization supervision. These are established methods, and the contribution is their task-oriented integration and evaluation rather than their invention.
3. The design is examined using the reported module ablation, a comparison with representative detectors under the stated common test protocol, category-level AP, and qualitative error analysis. The analyses are descriptive and are not presented as statistical significance tests.
4. The detector combines fruit localization and maturity-stage recognition in one prediction pipeline. This formulation may be relevant to adjacent-category detection and could inform agricultural monitoring, grading, or robotic harvesting, subject to independent validation in those settings.



Fig. 1 Representative images from the tomato maturity dataset. The samples contain different fruit scales, maturity stages, viewing directions, occlusion patterns, and background conditions.

2 Materials and methods

2.1 Dataset and task definition

Images were captured and annotated at a tomato cultivation base in Guigang, Guangxi, China. Each tomato was labelled as unripe, half-ripe, or ripe, and its bounding box was stored in COCO format. The 3,926 images were divided at the image level into training, validation, and test subsets containing 2,828 (72.0%), 314 (8.0%), and 784 (20.0%) images, respectively. No image was assigned to more than one subset. The test images include variation in acquisition date, illumination, and scene conditions. Because all images came from the same project dataset and cultivation base, this split evaluates within-project performance rather than external-site generalisation. Representative images are shown in Figure 1, and the split and category counts are reported in Table 1.

The category distribution is imbalanced, with ripe fruit accounting for most annotated objects. For this reason, overall detection metrics were interpreted together with class-level AP50 and the confusion matrix. Images were resized to 640 × 640 pixels. Random geometric transformation, horizontal flipping, scale variation, and colour perturbation were used during training, whereas validation and test images were processed without random augmentation.

2.2 CenterNet baseline and proposed architecture

2.2.1 Backbone and feature hierarchy

The CenterNet baseline predicts a three-class heatmap, a two-channel centre-offset map, and object width and height from a stride-4 feature map. ResNet-50 [25] produces four feature maps: C_2 = `layer1` (256 channels, stride 4), C_3 = `layer2` (512 channels, stride 8), C_4 = `layer3` (1,024 channels, stride 16), and C_5 = `layer4` (2,048 channels, stride 32). For a 640×640 input, these maps have spatial dimensions of 160×160, 80×80, 40×40, and 20×20, respectively. In M5, PyConv and RFB process C_5 in sequence before top-down decoding. WSF then fuses the decoder output with lateral features from C_4, C_3, and C_2. The final detector uses one stride-4 prediction head; it does not include SE attention, CIoU, or additional heads at strides 8 and 16.

Figure 2 summarizes the feature flow and distinguishes training from inference: DFL and WIoU contribute to the training objective, whereas inference decodes the predicted centres and size distributions into fruit boxes and maturity labels.

2.2.2 PyConv and RFB

The branches adapt established pyramidal-convolution and receptive-field-block designs [26, 27]. PyConv receives the 2,048-channel C_5 map and applies four parallel branches, each producing 512 channels. The branches use a 1×1 convolution, a 3×3 convolution with 4 groups, a 5×5 convolution with 8 groups, and a 3×3 convolution with dilation 3 and 16 groups. Each branch includes batch normalisation and ReLU. The branch outputs are concatenated, projected by a 1×1 convolution and batch normalisation, added to the input, and passed through ReLU. RFB follows PyConv at the same spatial resolution. It has four 512-channel branches: one 1×1 convolution and three 3×3 convolutions with dilation rates of 1, 3, and 5. Each 3×3 branch uses 4 groups. The concatenated features are projected by a 1×1 convolution and batch normalisation, added to the input, and activated by ReLU. These branches were included to represent different apparent fruit scales and surrounding context under occlusion. Their individual causal effects were not measured.

Table 1 Dataset split and category distribution.

Subset	Images	Split	Objects	Unripe	Half-ripe	Ripe
Training	2828	72.0%	8137	1049	817	6271
Validation	314	8.0%	903	110	89	704
Test	784	20.0%	2250	280	213	1757

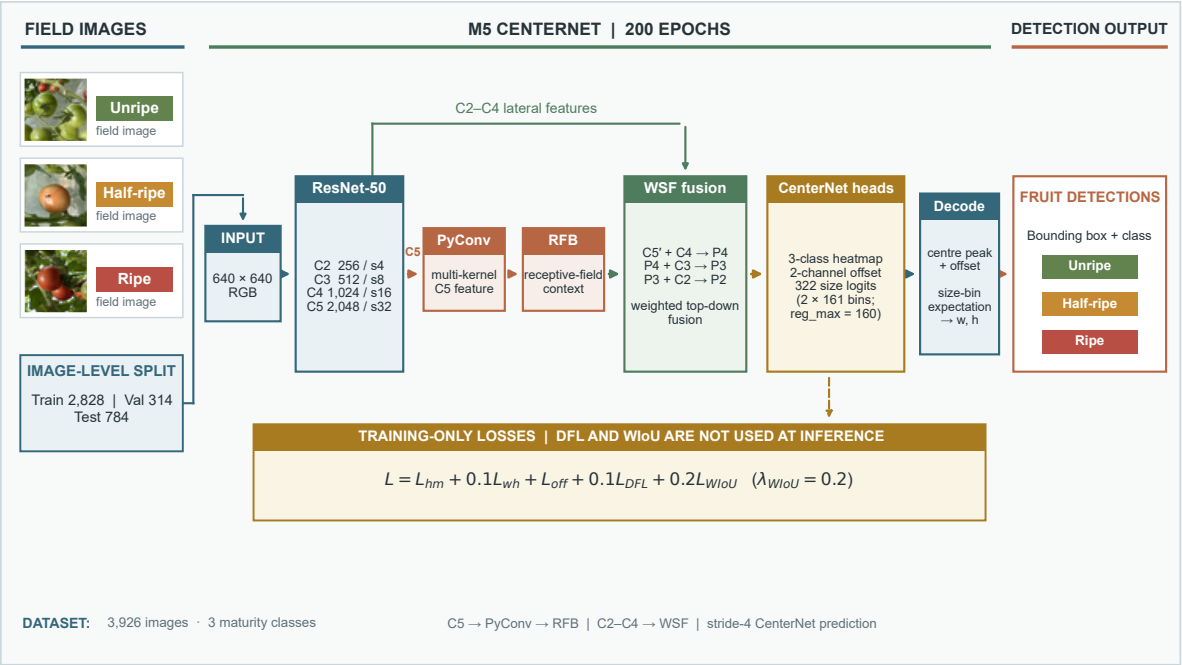


Fig. 2 Architecture and workflow of M5 for tomato maturity detection. ResNet-50 features C_2–C_5 feed the C5 PyConv–RFB path and WSF top-down fusion, followed by stride-4 CenterNet heatmap, offset, and size-distribution heads. The Guigang dataset was split at the image level into 2,828 training, 314 validation, and 784 test images; training used 200 epochs.

2.2.3 Weighted skip fusion

The decoder upsamples the processed C_5 feature using three 4×4 transposed convolutions with stride 2 and padding 1. These stages produce 256, 128, and 64 channels at strides 16, 8, and 4, respectively. Lateral 1×1 convolutions with batch normalisation map C_4, C_3, and C_2 to 256, 128, and 64 channels. At each level, two learned scalar weights are transformed by softplus, offset by 10^{-4}, and normalised to sum to one. The weighted sum of the upsampled and lateral features is then refined by a 3×3 convolution, batch normalisation, and ReLU. WSF therefore performs scalar-weighted, element-wise fusion rather than spatial attention. The final C_2 fusion combines fine colour and boundary cues with deeper semantic features.

2.2.4 CenterNet prediction heads

Three separate heads process the final 64-channel stride-4 feature. Each head contains a 3×3 convolution with 64 output channels, ReLU, and a 1×1 output convolution. The heatmap head predicts three class probabilities, and the offset head predicts two sub-pixel centre residuals. The width-height head outputs 322 logits, or $2(160 + 1)$ channels, for DFL. For an object centre (x, y) and stride $s = 4$, the integer grid coordinate indexes

the heatmap, while the fractional residual supervises the offset head. Width and height targets are expressed in output-grid units.

2.2.5 DFL, WIoU-style regression, and total loss

Following the DFL principle [28], width and height are represented separately over bins $0, \ldots, 160$ (`reg_max` $= 160$). Softmax converts each set of 161 logits to probabilities, and the decoded width or height is the probability-weighted bin expectation, $\hat{d} = \sum_{j=0}^{160} j \,\mathrm{softmax}(z)_j$. For a target d between bins $k = \lfloor d \rfloor$ and $k+1$, the distributional loss is the interpolated cross-entropy $(1-\alpha)\,\mathrm{CE}(z,k) + \alpha\,\mathrm{CE}(z,k+1)$, where $\alpha = d-k$. The implementation also applies a masked L1 term to the decoded width and height.

The code constructs predicted and target boxes from the centre offsets and decoded width-height values, computes their IoU, and applies the following implementation-specific WIoU-style term over valid objects $\mathcal{V}$:

$$L_{\mathrm{WIoU}} = \mathrm{mean}_{i\in\mathcal{V}}\big[\mathrm{stopgrad}(\exp(1-\mathrm{IoU}_i)) \times (1-\mathrm{IoU}_i)\big]. \tag{1}$$

This explicit formula describes the implementation-specific monotonic IoU-dependent weighting and is not presented as a new formulation of Wise-IoU [29]. The coefficient used for M5 was $\lambda_{\mathrm{WIoU}} = 0.2$.

The total single-scale objective is

$$L_{\mathrm{total}} = L_{\mathrm{hm}} + 0.1L_{\mathrm{wh}} + L_{\mathrm{off}} + 0.1L_{\mathrm{DFL}} + 0.2L_{\mathrm{WIoU}}. \tag{2}$$

Here, L_{hm} is the modified CenterNet focal loss; L_{wh} and L_{off} are masked L1 losses for decoded width-height and centre offsets; L_{DFL} is the adjacent-bin interpolated cross-entropy; and L_{WIoU} is the overlap term defined above. Focal loss was introduced for dense detection [30]. For heatmap probability p_{xyc} and Gaussian target Y_{xyc}, the implemented focal loss is

$$L_{\mathrm{hm}} = -\frac{1}{N_+}\sum_{xyc}\Big[\mathbf{1}(Y_{xyc}=1)(1-p_{xyc})^2\log p_{xyc} + \mathbf{1}(Y_{xyc}<1)(1-Y_{xyc})^4 p_{xyc}^2\log(1-p_{xyc})\Big], \tag{3}$$

where N_+ is the number of positive centres; when no positive centre is present, the code uses the negative-term sum without positive-count normalisation.

2.3 Training and evaluation protocol

Experiments were conducted in PyTorch on an NVIDIA GeForce RTX 4090 D. The recorded CenterNet configuration used 200 epochs, a batch size of 32, automatic mixed precision, and random seed 42. We used AdamW with an initial learning rate of 1×10^{-4} and weight decay of 1×10^{-4}. A cosine schedule reduced the learning rate to 1×10^{-6}, with five warm-up epochs. Training augmentation comprised horizontal flipping (probability 0.5) and geometric transformations (probability 0.5), with rotation from $-5°$ to $5°$, scaling from 0.85 to 1.15, and translation up to 5%. Brightness/contrast perturbation (0.3), HSV perturbation (0.25), and Gaussian blur (0.1) were also applied. Validation and test images received no random augmentation. We evaluated checkpoints every five epochs and selected the final checkpoint by validation mAP50:95.

Each comparison detector was trained on this dataset; no off-the-shelf detector checkpoint was used directly for test inference. All models used the same split, input size (640×640), pretraining setup, and training strategy. Evaluation used the same three classes, prediction settings (scan confidence 0.001, Top-K=100, operating confidence 0.35, and NMS IoU 0.70), and evaluation script. We computed AP50 and mAP50:95 using COCO evaluation; mAP50:95 averages AP across IoU thresholds from 0.50 to 0.95. Precision, recall, and F1 were calculated at the stated operating point. Params, GFLOPs, and FPS cover each complete detector, including its backbone, neck/decoder, and prediction heads, at 640×640 and batch size 1 on the same NVIDIA GeForce RTX 4090 D. FPS profiling used identical inference precision, preprocessing, postprocessing, 50 warm-up passes, and 200 timed passes. These speed measurements depend on the implementation and hardware.

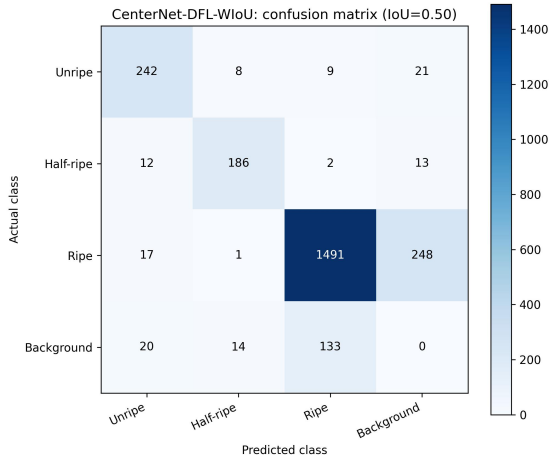


Fig. 3 Confusion matrix on the image-level test split at IoU 0.50. Rows show actual classes and columns show predicted classes. The background row and column represent unmatched predictions and missed objects, respectively. Values are raw counts.

3 Results

3.1 Ablation study

Table 2 summarizes the tested configurations. M5 achieved the highest mAP50:95 (0.8472). M2 and M3 form an alternative SE/CIoU branch; DFL and the WIoU-style term were added jointly, so their separate effects are not isolated.

3.2 Category-level performance

Table 3 reports test-set AP50 by maturity class. Unripe fruit had the lowest AP50, whereas half-ripe fruit had the highest. Similar colour between unripe fruit and foliage may contribute to this pattern, but the present analysis did not test that explanation.

The confusion matrix in Figure 3 contains 242, 186, and 1,491 correct detections for unripe, half-ripe, and ripe fruit, respectively. Ripe fruit accounted for the largest number of missed detections (248), while 133 background detections were classified as ripe. These counts describe this test split. Occlusion, foliage interference, and class imbalance are possible contributing factors, but their effects were not analysed separately.

3.3 Comparison with representative detectors

Table 4 compares detectors on the common test split. M5 had the highest precision and mAP50:95, while its AP50 exceeded RT-DETR-L by 0.0002; without repeated runs or uncertainty estimates, this difference is descriptive, not evidence of statistical superiority. The results do not establish superiority across detector families or datasets [31–33].

3.4 Qualitative results and error analysis

Figure 4 shows detections under fruit contact, leaf occlusion, and complex vegetation. Remaining errors include missed fruit when substantial portions of the boundary are occluded, confusion between green unripe fruit and foliage, and false positives on high-contrast leaf texture. These examples identify failure modes in the evaluated images; they do not quantify performance under external conditions.

4 Discussion

Main finding

M5 combined fruit localisation and three-stage maturity recognition in one centre-based detector. It produced the highest mAP50:95 among the detectors in the common comparison. This is a descriptive result for the specified test split and recorded run, not evidence of population-level or cross-site superiority.

Interpretation

Class-wise AP was lower for unripe fruit than for the other two maturity stages. The qualitative examples show misses under substantial occlusion and confusion with foliage. These patterns are consistent with the visual challenges addressed by the model design, but their causes were not isolated experimentally. The 0.0002 AP50 difference from RT-DETR-L is descriptive because uncertainty estimates and repeated-seed comparisons are unavailable.

Related methods examine multiscale fusion and contextual representation in other detection settings: LAF-DETR targets small and occluded

Table 2 Ablation study of module configurations on the test split.

Configuration	Precision	Recall	AP50	mAP50:95	Params (M)	GFLOPs	FPS
M0 Baseline	0.9035	0.8573	0.9118	0.7657	32.66	112.68	268.00
M1 +WSF	0.8938	0.8600	0.9154	0.7779	33.78	120.86	226.60
M2 +WSF+SE	0.8949	0.8627	0.9032	0.7659	33.78	120.86	221.65
M3 +WSF+SE+CIoU	0.8916	0.8662	0.9076	0.7835	33.78	120.86	204.47
M4 +PyConv+RFB	0.8929	0.8600	0.9211	0.7903	57.59	139.89	170.61
M5 +DFL+WIoU	0.9054	0.8591	0.9226	0.8472	57.61	140.94	173.08

Table 3 Class-wise AP50 on the test split.

Maturity class	Test objects	AP50
Unripe	280	0.8991
Half-ripe	213	0.9376
Ripe	1,757	0.9310

aerial objects [34], a dual-branch network combines global-local representation and multiscale fusion for occluded targets [35], and a fusion-aware framework studies multimodal autonomous-driving detection [36]. PyConv has also been used with CISBA in YOLO-based surface-defect detection [37]. These studies motivate comparison at the level of design choices; they are not tested baselines on the present tomato data. The current evidence supports the evaluated CenterNet configuration, not a new general convolution, fusion, or regression algorithm.

Contribution and scope

The contribution is a task-oriented integration of established components for fruit localization and three-stage maturity recognition. Related greenhouse robotic integration has been studied using RGB-D perception and ROS control [13]. In the present work, harvesting, field monitoring, automatic grading, and transfer to other crops remain potential uses rather than validated outcomes. Different datasets, classes, and protocols in the cited work preclude a direct ranking of published scores.

The reported detector has 57.61 million parameters and requires 140.94 GFLOPs. These measurements should be considered alongside its detection results. FPS depends on the hardware and implementation and does not establish real-time performance on edge devices. The evidence supports conclusions about the tested split and configurations only; it does not establish external robustness or consistent superiority over other detector families.

5 Conclusion

We evaluated a ResNet-50 CenterNet detector for recognising unripe, half-ripe, and ripe tomatoes in natural scenes. M5 combines WSF, PyConv, RFB, DFL, and an implementation-specific WIoU-style term ($\lambda_{\mathrm{WIoU}} = 0.2$) to address feature fusion, scale variation, context, and box regression. On the reported test split, it achieved precision 0.9054, recall 0.8591, AP50 0.9226, and mAP50:95 0.8472. The ablation study, detector comparison, class-wise metrics, and error analysis describe performance within the tested protocol, but the single-dataset, single-seed evidence does not establish external generalisation or statistical significance. The unified output may be relevant to agricultural automation, subject to validation in those applications. Evaluation across independent sites, repeated seeds, isolated component experiments, and target hardware would clarify the scope and reproducibility of these findings.

Table 4 Test performance and efficiency comparison under the common training and evaluation protocol.

Model	Precision	Recall	AP50	mAP50:95	Params (M)	GFLOPs	FPS
YOLOv5m	0.8166	0.8301	0.8706	0.6689	20.86	47.88	231.62
YOLOv7	0.8625	0.8354	0.9061	0.7869	36.49	103.19	197.10
YOLOv8m	0.8936	0.8343	0.9050	0.8371	23.22	67.43	146.25
RT-DETR-L	0.9008	0.8322	0.9224	0.8330	32.81	107.32	13.00
Faster R-CNN	0.8325	0.8301	0.9011	0.7461	40.83	115.44	91.17
M5 (proposed CenterNet)	0.9054	0.8591	0.9226	0.8472	57.61	140.94	173.08

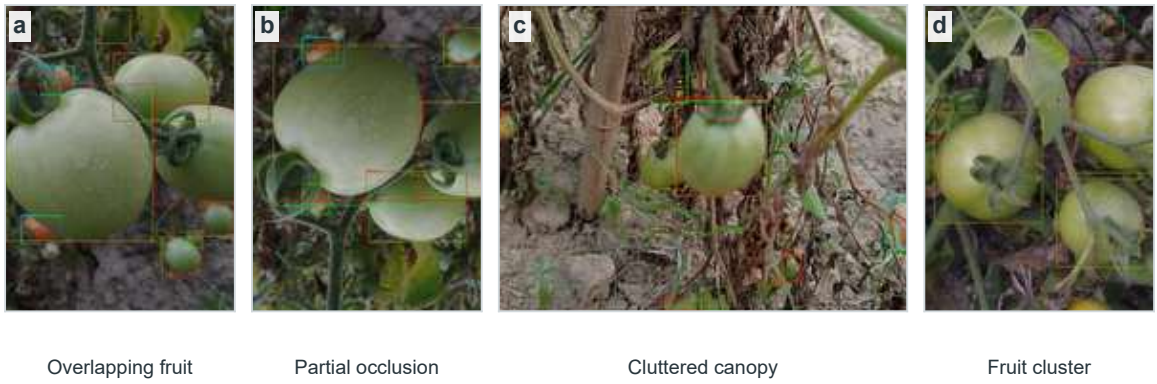


Fig. 4 Representative M5 detections in natural scenes: (a) overlapping fruit, (b) partial occlusion, (c) a cluttered canopy, and (d) a fruit cluster. Bounding boxes and predicted maturity labels are shown.

Statements and Declarations

Acknowledgments

No external funding or non-author assistance was received for this study.

Funding

No external funding was received for this research.

Competing interests

The author declares no competing interests.

Author contributions

Guojie Chen: conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization, writing–original draft, and writing–review and editing.

Data availability

The tomato maturity dataset, annotations, source code, trained weights, and supporting evaluation files are available at Zenodo: https://doi.org/10.5281/zenodo.22887607.

Declaration of generative AI and AI-assisted technologies

OpenAI Codex desktop application (model version not displayed in the user interface; accessed 20 October 2026) was used for English-language polishing, manuscript organization, consistency checking, and formatting assistance. It was not used to generate experimental data, alter reported results, or make scientific decisions. The author reviewed and edited all AI-assisted content and assumes full responsibility for the final manuscript.

Ethics approval and consent to participate

Not applicable. The study used images of tomato plants and did not involve human participants or identifiable personal data.

References

[1] Wang S, Jiang H, Yang J, Ma X, Chen J, Li Z, Tang X. Lightweight tomato ripeness detection algorithm based on the improved RT-DETR. Frontiers in Plant Science, 2024, 15: 1415297. https://doi.org/10.3389/fpls.2024.1415297.

[2] Zhao M, Cui B, Yu Y, Zhang X, Xu J, Shi F, Zhao L. Intelligent detection of tomato ripening in natural environments using YOLO-DGS. Sensors, 2025, 25(9): 2664. https://doi.org/10.3390/s25092664.

[3] Zheng Y, Wang M, Sun J, Han L, Ye Y. Efficient-tomato: a high-precision and lightweight method for cherry tomato ripeness and disease detection. Frontiers in Plant Science 2026; 17: 1880973. https://doi.org/10.3389/fpls.2026.1880973

[4] Huang C, He L, Huang W, Zhang X. TMAF-YOLO: A Lightweight Model for In Situ Detection of Tomato Maturity and Defective Fruits in Greenhouses. Sensors 2026; 26 (15): 4950. https://doi.org/10.3390/s26154950

[5] Xu X, Wu A, Zhao Y, Wu N, Yang D, et al. LGD-YOLO: An Asymmetric Lightweight Network with Dynamic Feature Alignment for Greenhouse Tomato Maturity Detection. Agriculture 2026; 16 (14): 1517. https://doi.org/10.3390/agriculture16141517

[6] Ding J, Zou Y, Wang Y, Han L, Xiao Y, et al. A Lightweight Task-Adaptive YOLO for Tomato Ripeness Detection in Complex Orchard Environments. Horticulturae 2026; 12 (7): 805. https://doi.org/10.3390/horticulturae12070805

[7] Su Y, Zhang P, Li H, Kang F, Zhao P, et al. YOLO-FLBM: a lightweight and high-performance model for tomato ripeness detection in complex greenhouse environments. Frontiers in Plant Science 2026; 17: 1824558. https://doi.org/10.3389/fpls.2026.1824558

[8] Cheng B, Wu X, Du H, Qi X, Wu J, et al. A real-time ripeness detection model for tomatoes in complex greenhouse environments. Frontiers in Plant Science 2026; 17: 1853690. https://doi.org/10.3389/fpls.2026.1853690

[9] Wu F, Li M, Zhou C, Zhang F. Automated detection of tomato ripeness in natural environment using a coupled deep learning model of Sequencer and MLP. Applied Soft Computing 2026; 203: 116038. https://doi.org/10.1016/j.asoc.2026.116038

[10] Gai Y, Zhu J, Wu R. YOLO-TMD: an RGB-D multi-modal network for tomato maturity detection. Journal of Food Measurement and Characterization 2026; 20 (8): 15874-15895. https://doi.org/10.1007/s11694-026-04672-5

[11] Fu B, Li W, Wang D, Zhao H, Li X, et al. A tomato maturity detection method against occlusion and variable illumination. Computers and Electronics in Agriculture 2027; 256: 112423. https://doi.org/10.1016/j.compag.2026.112423

[12] Chen D, Teng H, Lu Y, Zhang Y, Wu H. YOLO-RCM: An Improved Tomato Maturity Detection Model for Complex Greenhouse Environments. Agronomy 2026; 16 (12): 1146. https://doi.org/10.3390/agronomy16121146

[13] Meng F, Wang Z, Wang H, Li X, Xu X, et al. Edge-deployable greenhouse tomato cluster harvesting robot integrating YOLOv8n-BiFPN-WIoU and ROS-based autonomous control. Frontiers in Plant Science 2026; 17: 1926482. https://doi.org/10.3389/fpls.2026.1926482

[14] Li Y, Jiang C, Wang H, et al. Tomato leaf disease detection based on an improved CenterNet algorithm. Journal of Zhejiang University (Engineering Science), 2025, 59(11):

2370-2378.

[15] Yi C, Xu S, Hao Z, Xiao H. Rail surface defect detection algorithm based on CenterNet-EGS [J/OL]. Computer Measurement & Control, 2026-05-29. Available at: https://link.cnki.net/urlid/11.4762.tp.20260528.1648.002.

[16] Liang D, Liu W, Zhao Y, Zong S, Luo Y. UAV-based forest pest image object detection using improved CenterNet [J]. Journal of Beijing Forestry University, 2026, 48(3): 140-151. https://doi.org/10.12171/j.1000-1522.20240018.

[17] Wang Z, Zhao H, Yang J, et al. Identification and counting of field peanut seedlings using improved CenterNet from UAV imagery. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2025; X-1/W2-2025: 173-178. https://doi.org/10.5194/isprs-annals-X-1-W2-2025-173-2025

[18] Thai TH, Le HK, Nguyen TLN. EF-CenterNet: an efficient anchor-free model for UAV-based banana leaf disease detection. Computers and Electronics in Agriculture 2025; 231: 109927. https://doi.org/10.1016/j.compag.2025.109927

[19] Divyakanti R, Rao GS, Aruna S. BMA-CenterNet based multi-leaf multi-disease classification and invasive plant identification framework using cdeboar-50. Computers and Electrical Engineering 2025; 128: 110732. https://doi.org/10.1016/j.compeleceng.2025.110732

[20] Yao Z, Xu H, Zhang H, Tu X, Yin J. Improved CenterNet-Based Multimodal Object Detection for Low-Light and Complex Environments. Sensors 2026; 26 (12): 3735. https://doi.org/10.3390/s26123735

[21] Ji L, Cheng J, Wu S. Enhanced CenterNet model for accurate surface damage detection on wind turbine blades: addressing small and multi-scale targets challenges. Electric Power Systems Research 2026; 258: 113064. https://doi.org/10.1016/j.epsr.2026.113064

[22] Meng B. Researching on insulator defect recognition based on context cluster CenterNet++. Scientific Reports 2025; 15 (1): 2352. https://doi.org/10.1038/s41598-025-85630-x

[23] Zhou X, Wang D, Krähenbühl P. Objects as points. arXiv:1904.07850, 2019.

[24] Tian Z, Shen C, Chen H, et al. FCOS: fully convolutional one-stage object detection. ICCV, 2019: 9627-9636.

[25] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. CVPR, 2016: 770-778.

[26] Duta I, Liu L, Zhu F, et al. Pyramidal convolution: rethinking convolutional neural networks for visual recognition. arXiv:2006.11538, 2020.

[27] Liu S, Huang D, Wang Y. Receptive field block net for accurate and fast object detection. ECCV, 2018: 385-400.

[28] Li X, Wang W, Hu X, et al. Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. NeurIPS, 2020.

[29] Tong Z, Chen Y, Xu Z, et al. Wise-IoU: bounding box regression loss with dynamic focusing mechanism. arXiv:2301.10051, 2023.

[30] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. ICCV, 2017: 2980-2988.

[31] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. NeurIPS, 2015.

[32] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection. CVPR, 2016: 779-788.

[33] Zhao Y, Lv W, Xu S, et al. DETRs beat YOLOs on real-time object detection. CVPR, 2024: 16965-16974.

[34] Zhou M, Wang B, Li T, Wang Y, Zhu C. LAF-DETR: lightweight attentive fusion

DETR for small and occluded object detection in aerial images. Measurement Science and Technology 2026; 37 (27): 275408. https://doi.org/10.1088/1361-6501/ae7621

[35] Wu L, Wang Z, Zheng Z, Ke X, Zhou Z, et al. A dual-branch detection network with global–local backbone and multi-scale fusion for occluded targets. Expert Systems with Applications 2026; 300: 130468. https://doi.org/10.1016/j.eswa.2025.130468

[36] Li Z, Singh B. Robust Occluded Object Detection in Multimodal Autonomous Driving: A Fusion-Aware Learning Framework. Electronics 2026; 15 (1): 245. https://doi.org/10.3390/electronics15010245

[37] Ruan S, Zhan C, Liu B, Wan Q, Song K. A high precision YOLO model for surface defect detection based on PyConv and CISBA. Scientific Reports 2025; 15 (1): 15841. https://doi.org/10.1038/s41598-025-91930-z