2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

LMCNet: A MobileNetV4-Enhanced YOLOv10 with Cross-Scale Fusion for Tomato Ripeness Detection

Jianying Chen[1] and Chuanying Yang[2(✉)]

[1] School of Information Engineering, Inner Mongolia University of Technology, Hohhot 01000, China
Cjy0011292016@163.com
[2] School of Information Engineering, Inner Mongolia University of Technology, Hohhot 01000, China
ycy@imut.edu.cn

Abstract. In order to quickly and accurately identify tomato fruit ripeness and implement automated tomato harvesting in agricultural environments, this study proposes a lightweight tomato ripeness detection model based on an improved YOLOv10. Firstly, a lightweight model based on the improved YOLOv10 is proposed by introducing the Universal Inverted Bottleneck module from the MobileNetV4 network and integrating it with the C2f module in YOLOv10. Then, a new feature fusion structure is designed, where the C2fUIB module replaces the original feature fusion module in the CCFM structure, and the GhostConv module is introduced to replace the standard Conv module. The improved model efficiently handles and fuses the different scale information, and at the same time enhances the model's detection accuracy and computational efficiency for tomato fruits. The results of this research model on tomato fruit ripeness detection show that the accuracy, recall and average precision are 88.2%, 86.2% and 90.2%, respectively, and the number of parameters of the network model is 4.62M, and the model memory occupancy is 9.7MB, which has a high detection precision and low number of parameters. It highlights the effect of the improved model on tomato fruit ripeness detection.

Keywords: YOLOv10, Ripeness Detection, Lightweight Model, Tomato, MobileNetV4, CCFM.

1 Introduction

At present, modern agriculture across the globe has increasingly adopted smart farming, establishing it as a significant trend in agricultural development [1]. Researchers around the world have incorporated computer vision technology into agriculture, applying it to areas such as crop pest detection, fruit recognition, grading, and automated harvesting. Deep learning-based fruit detection and picking robots have attracted growing interest. Employing detection robots for automated fruit ripeness assessment and mechanized

harvesting offers the potential to greatly enhance harvesting efficiency and lower costs [2]–[4]. Nonetheless, the development of fully mature mechanized harvesting technology continues to encounter significant challenges.

Tomatoes are valued for their rich vitamin content and high nutritional benefits. The fruits grow in clusters, and due to the diversity of tomato varieties and the complexity of growing environments, achieving objective and standardized ripeness detection remains challenging. This complexity poses significant challenges to the recognition process, impacting both accuracy and reliability. Consequently, the critical step in advancing automated harvesting technology lies in overcoming environmental complexities and ensuring precise multi-class target recognition in tomato imagery.

To tackle this issue, the study introduces a lightweight tomato ripeness detection method based on an improved YOLOv10, leveraging advancements from the latest YOLO series of object detection algorithms. By adopting the consistent dual allocation strategy proposed in YOLOv10, which eliminates the need for NMS during training, and incorporating an efficiency and accuracy-driven design, the model achieves optimization in both aspects. This approach significantly enhances the recognition accuracy of tomato ripeness. This research offers the following major contributions:

- The backbone network of the YOLOv10 model is im-proved by integrating the Universal Inverted Bottleneck (UIB) module from MobileNetV4 with the C2f module, creating a new modular structure, C2fUIB, which reduces both the number of model parameters and computational complexity.
- The cross-scale feature fusion model (CCFM) is employed to replace the original Neck structure, with C2fUIB substituting for RepC3, the feature fusion mod-ule in CCFM, to more effectively integrate feature in-formation from different scales and enhance the model's adaptability to scale variations and the detection of small-scale targets.
- The GhostConv module is introduced as a replacement for the standard Conv module in the CCFM architecture, aimed at reducing the number of model parameters and memory footprint to support lightweight deployment.

2 Related work

2.1 Traditional Methods of Fruit Detection

Traditional fruit identification and detection methods are mainly digital image processing, through the image extraction of the target fruit color, shape and texture features such as matching, and discernment of the type of disease and quality analysis [5]. Surya Prabha [6] developed two algorithms that utilize color and size features to classify banana ripeness by analyzing the color characteristics in banana images. Lin [7] employed Hough transformation, leveraging color and texture information, for contour-based detection of citrus, tomatoes, and other fruits.

While color-based fruit recognition methods have shown promising results for certain fruits, their reliance on color features limits their adaptability to variations in lighting and color conditions. On the other hand, texture-based recognition methods face

challenges due to filter constraints and feature description limitations, which reduce their efficiency. As a result, traditional detection methods are constrained to a single detection scenario and struggle with the accuracy and robustness needed for fruit recognition in complex environments. Furthermore, these methods often involve more complex and time-consuming feature extraction processes, making them less suitable for addressing the demands of modern agriculture.

2.2 Deep Learning-Based Tomato Fruit Detection Methods

Deep learning techniques not only overcome the robust-ness limitations of traditional methods that rely on color, shape, and texture features but also deliver significantly higher accuracy compared to conventional image processing techniques, demonstrating substantial potential for practical applications [8]. Deep learning-based target detection algorithms can be broadly categorized into two-stage and single-stage approaches. Two-stage detection methods utilize region proposal techniques to identify potential candidate regions, followed by classification and localization of these regions. Notable examples of such algorithms include R-CNN [9], Fast R-CNN [10], and Faster R-CNN [11]. Single-stage object detection methods, by contrast, bypass the need for pre-extracting candidate regions and directly perform classification and bounding box regression on feature maps. Examples include SSD [12] and the YOLO series [13].

Zheng [14] introduced the YOLOX-Dense-CT detection algorithm, which incorporates the DenseNet network and CBAM attention mechanism. These improvements enhance the network's suitability for cherry tomato detection, significantly improving the model's recognition performance. Cai [15] developed an improved YOLOv7-tiny method for cherry tomato detection, leveraging multi-modal RGB-D images and an optimized network. When tested on an AGV-based robot, this method achieved a harvesting success rate exceeding 80%.

3 Approaches

3.1 LMCNet: Integrating MobileNetV4 and CCFM into YOLOv10

This study focuses on an enhanced YOLOv10 model designed for detecting tomato ripeness in complex environments. To minimize model parameters and computational complexity, the original C2f module in the backbone network is replaced with the Universal Inverted Bottleneck (UIB) module from the MobileNetV4 [16] network, forming the C2fUIB module. Additionally, the neck feature fusion module is optimized by replacing the original PANet (Path Aggregation Network) with a lightweight Cross-Scale Feature Fusion Module (CCFM). Within the CCFM structure, the RepC3 feature fusion module is substituted with C2fUIB, enhancing adaptability to scale variations and small-scale object detection. Furthermore, the standard Conv convolution module in the CCFM structure is replaced with a GhostConv module, reducing the parameter count and memory usage. This improved model retains the advanced end-to-end real-time object detection capabilities of the original design while achieving a lightweight

architecture, significantly lowering computational resource requirements. The architecture of the refined network model is shown in Fig. 1.

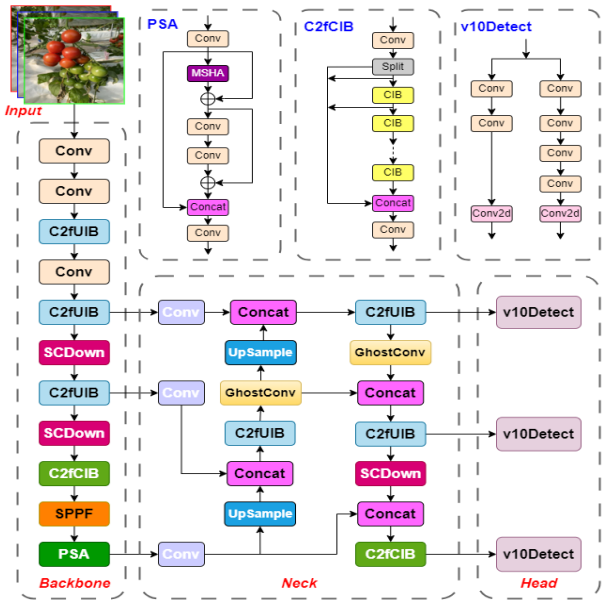

Fig. 1. Structure of the proposed LMCNet model.

3.2 The C2fUIB Block

The original backbone network employs the C2f module for feature extraction. However, detecting tomato fruits in specific agricultural environments often faces computational inefficiency due to resource constraints. To address this, we integrate the UIB module from the MobileNetV4 network to optimize YOLOv10's feature extraction capabilities. This approach not only enhances computational efficiency but also significantly reduces the algorithm's parameter size and computational overhead. The UIB search block is illustrated in Fig. 2. Its modular and adjustable design makes it well-suited for efficient network architectures and adaptable to a variety of optimization tasks.

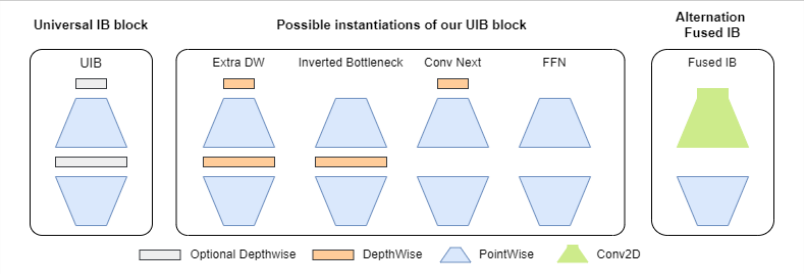

Fig. 2. Universal Inverted Bottleneck Module.

Assuming the input feature map is X and the output feature map is Y, the computation flow of the Universal Inverted Bottleneck (UIB) module is as follows:
Pre-DWConv:

$$X_1 = DWConv_{k_1}(X) \tag{1}$$

Channel Expansion (1×1 Convolution):

$$X_2 = PWConv_{expand}(X_1) \tag{2}$$

Intermediate DWConv:

$$X_3 = DWConv_{k_2}(X_2) \tag{3}$$

Channel Projection (1×1Convolution):

$$Y_{c,h,w} = BN(\sum_{k=1}^{C_{mid}} W_{c,k} * X_{3_{k,h,w}} + b_c) \tag{4}$$

Where k denotes the kernel size, W represents the convolution kernel weights, and b_c denotes the bias term.

As shown in Fig. 3, the C2f module processes input data by applying two convolutional layers, extracting abstract features at multiple levels. Additionally, the input data is branched to enhance the network's non-linear and representational capabilities, improving its effectiveness in capturing complex data patterns.

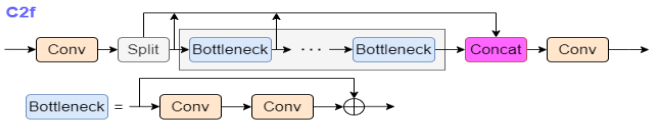

Fig. 3. C2f Module Structure.

We integrate the UIB block into the original C2f module by replacing its Bottleneck component, resulting in the creation of the C2fUIB module, as shown in Fig. 4. This new module not only preserves YOLOv10's strengths in multi-scale feature extraction

and fusion but also improves efficiency through a lightweight design, significantly reducing computational resource requirements.

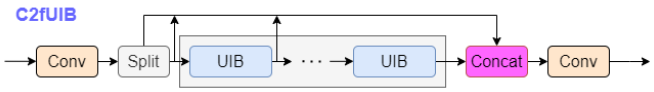

Fig. 4. C2fUIB Module Structure.

3.3 CCFM Structure

In this study, an improved Cross-Scale Feature Fusion Mod-ule (CCFM) structure replaces the Path Aggregation Network (PANet) in the neck of the YOLOv10 model. This modification enables the fusion of feature information at different scales, enhancing the model's adaptability to scale variations and improving its performance in detecting small-scale targets, as shown in Fig. 5. The enhanced CCFM structure incorporates a Fusion module consisting of convolutional layers within the fusion paths. These layers combine features from neighboring scales, effectively merging detailed features with contextual information to boost the model's overall performance.

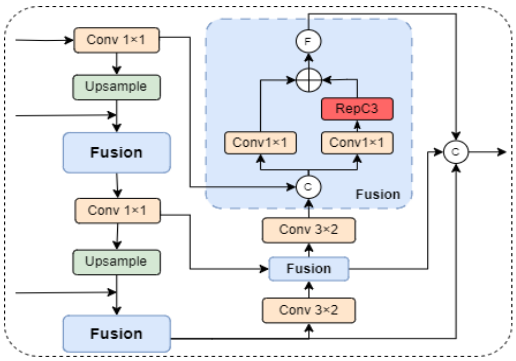

Fig. 5. CCFM Structure.

Let $F_l \in \mathbb{R}^{C\times H_l\times W_l}$ denote the low-level semantic features, and $F_m \in \mathbb{R}^{C\times H_m\times W_m}$ denote the high-level semantic features. The output feature F_{out} is computed as:

$$F_{out} = \sigma(BN(W_{3\times3} * \sigma(BN(W_{1\times1} * Concat(u(F_l), F_m + b))))) \tag{5}$$

Where $u(\,)$ denotes the upsampling function, σ is the activation function, W represents the convolutional kernel weights, and b is the bias term.

Additionally, the feature fusion module in the CCFM structure has been upgraded by replacing the RepC3 module with the C2fUIB module, further improving detection performance for small objects. The improved fusion modules complement one another through their respective design principles, optimizing the feature fusion process and significantly enhancing the overall performance of tomato fruit detection.

3.4 GhostConv Module

The original CCFM model structure in the neck network employs a conventional Conv module. In this study, we intro-duce the GhostConv module from the GhostNet [17] network to replace the standard Conv module. This substitution reduces the model's parameter count and memory footprint, facilitating lightweight deployment. The conventional convolution structure, shown in Fig. 6(a), typically involves convolving the input image, followed by batch normalization and non-linear activation.

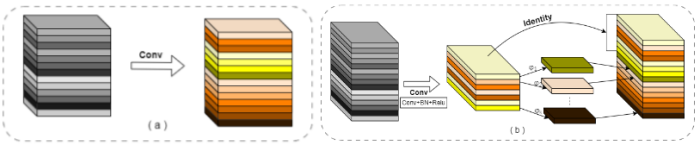

Fig. 6. Standard Convolution and GhostConv Model

The structure of the GhostConv module, illustrated in Fig. 6(b), begins with a standard convolution to compress the input channels and generate smaller feature maps. These feature maps then undergo a grouping operation: each channel is subjected to a linear transformation to produce a Ghost feature map. The feature map obtained from the initial convolution is mapped identically. Finally, the two groups of feature maps are merged along the channel dimension to produce the output feature map.

$$Y^{'} \in \mathbb{R}^{C^{'} \times H \times W}, \ C^{'} = C_{out}/s \tag{6}$$

$$Y = Concat(Y', \{CheapOp(Y_i')\}_i^{C'}) \tag{7}$$

Where Y' is the base feature map generated by standard convolution, and $CheapOP(\,)$ is a linear transformation applied to generate the feature map.

Unlike traditional convolution, the linear transformation in GhostConv does not involve batch normalization or non-linear activation. This approach enables the extraction of abundant intrinsic feature information with minimal computation, enhancing the efficiency of the convolution operation. As a result, the GhostConv module achieves a more lightweight algorithm design while maintaining recognition accuracy.

3.5 Evaluation Metrics

The improved YOLOv10 model algorithm adopts common evaluation metrics for object detection tasks. The public announcements are as follows:

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN} \tag{9}$$

$$AP = \int_0^1 P(R)dR \tag{10}$$

$$mAP_{50} = \frac{1}{N}\sum_{i=1}^{n} AP_i \tag{11}$$

Where TP (True Positive) denotes the count of regions accurately identified as targets by the model; FP (False Positive) refers to the number of regions without targets mistakenly classified as containing targets; FN (False Negative) indicates the number of regions with targets that the model fails to detect; P (Precision) refers to the fraction of correctly predicted positives among all predicted positives; R (Recall) represents the fraction of true positives identified among all actual positives; mAP50 signifies the mean average precision across all classes at an IoU of 0.5.

4 Experiments

4.1 Dataset

This study uses experimental image data from the Laboro Tomato dataset [18], consisting of 804 original images captured on a farm using two cameras with resolutions of 3024×4032 and 3120×4160. The dataset covers a range of scenarios, including single-object, multi-object, fruit overlap, and occlusion, allowing tomato fruits to be captured under various lighting conditions and real-world settings.

Before data augmentation, the raw images underwent pre-processing steps, such as scaling and normalization, to improve the effectiveness of model training on the dataset. This study applies data augmentation techniques, including horizontal flipping, vertical flipping, center cropping, brightness adjustment, contrast adjustment, and the addition of Gaussian noise. Augmentation methods were applied randomly to expand the dataset.

4.2 Experimental Environment

Table 1. Experimental Environment and Training Parameters.

Category	Details
Operating System	Linux Ubuntu 16.04
Development Environment	VSCode; PyTorch (v2.0.1); CUDA 11.7; Python 3.9
Input Image Size	640×640×3
Training Epochs	200
Optimizer	Stochastic Gradient Descent (SGD)
Batch Size	24
Initial Learning Rate	0.01
Momentum	0.937
Weight Decay	0.0005

4.3 Comparative Experiment

After incorporating the C2fUIB module, CCFM structure, and GhostConv module, we conducted a comparison of the improved model with several state-of-the-art object detection models, including REDETR-ResNet50, YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv10, as shown in Table 2.

Table 2. Comparison of Model Performance

Model	Precision (%)	Recall (%)	mAP50 (%)	GFLOPs	Parameter (M)	Model Size (MB)
REDETR-ResNet18	88.5	80.9	84.7	58.3	20.10	40.5
YOLOv5s	86.5	82.9	87.4	24.1	9.12	18.5
YOLOv8s	86.7	82.1	88.4	28.7	11.14	22.5
YOLOv9c	88.7	83.7	90.1	103.7	25.53	51.6
YOLOv10s	86.9	80.3	87.6	24.8	8.07	16.6
Ours	**88.2**	**86.2**	**90.2**	**14.8**	**4.62**	**9.7**

As shown in Table 2, the improved model surpassed REDETR-ResNet50, YOLOv5, YOLOv8, YOLOv9 and YOLOv10 in mAP50 by 5.5%, 2.8%, 1.8%, 0.1% and 2.6%, respectively. Furthermore, its parameter count was reduced by 76.05%, 47.57%, 56.89%, 81.2%, and 41.57%, respectively. Although the YOLOv9 model achieved slightly better accuracy, it comes at the cost of significantly higher parameter count and computational load compared to our model. These results demonstrate that the proposed model effectively balances detection accuracy and computational efficiency, embodying lightweight characteristics. In addition, the model's enhanced detection speed and efficient resource utilization make it highly suitable for deployment on devices with limited computational resources.

4.4 Ablation experiments

To assess the impact of replacing specific modules on tomato fruit maturity detection, modules were introduced progressively, and the detection performance of the resulting fused models was analyzed. The results of the ablation experiments are presented in Table 3, while Fig. 7 provides a comparison of parameter counts and GFLOPs.

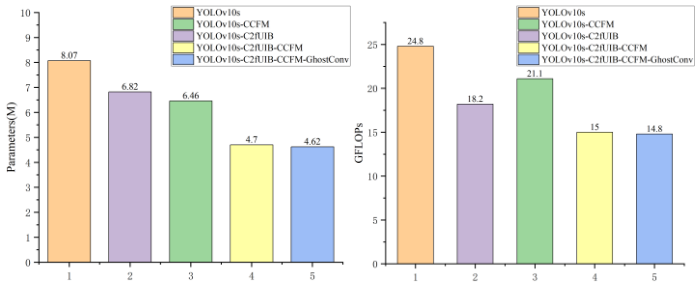

Fig. 7. Parameter Count and GFLOPs of Ablation Experiment Results

From the results of the ablation experiment, it is evident that replacing the C2f module with the C2fUIB module in the YOLOv10 model (Model 2 in Table I) leads to notable improvements. The accuracy increases from 86.9% to 87.2%, the recall rate improves from 80.3% to 83.3%, and the mAP50 rises from 87.6% to 89.2%. Additionally, the number of parameters decreases from 8.07M to 6.82M, and the GFLOPs reduce to 18.2.

This demonstrates that the UIB module from the MobileNetV4 network model enhances accuracy while simultaneously reducing parameters and computational complexity.

Table 3. Ablation Experiment Results with Different Modules

Model	C2fUIB	CCFM	GhostConv	Precision(%)	Recall(%)	mAP50(%)
1	-	-	-	86.9	80.3	87.6
2	√	-	-	87.2	83.3	89.2
3	-	√	-	86.6	83.7	89.2
4	√	√	-	87.6	83.5	89.6
5	√	√	√	88.2	86.2	90.2

Furthermore, with the incorporation of the improved CCFM structure (Model 4 in Table I), although the accuracy remains largely unchanged, there is a significant reduction in both the parameter count and computational load. The number of parameters drops to 4.7M, and the GFLOPs decrease to 15.0. These results indicate that integrating the CCFM structure with the C2fUIB module effectively minimizes model complexity and computational demands, positively impacting the overall performance of the improved network model.

When the improvements of the C2fUIB module, CCFM structure, and GhostConv module were combined (Model5), accuracy, recall, and mAP50 reached 88.2%, 86.2%, and 90.2%, respectively. The parameter count decreased to 4.62M, and GFLOPs dropped to 14.8. The analysis of the ablation experiment results indicates that these enhancements led to significant performance gains in tomato fruit detection, particularly when all the improvements were combined. The detection accuracy improved, while the model's parameter size and computational load were significantly reduced. As shown in Fig. 8, the ablation experiment results of the improved model illustrate the variation curves for accuracy, recall, and mAP50. It is evident that the accuracy of all models shows an upward trend, with the improved model achieving the best performance.

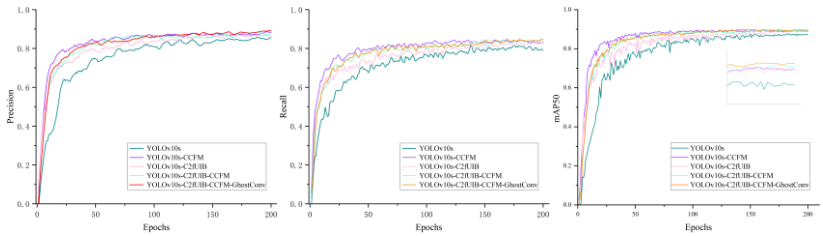

Fig. 8. Accuracy Change Curve of Ablation Experiment

Overall, the improved model demonstrates excellent performance across all evaluation metrics. These enhancements not only boost precision, recall, and mAP but also significantly reduce the model's parameters and computational complexity, leading to faster detection speed. These optimizations confirm the effectiveness of the introduced

modules in feature extraction and fusion, resulting in strong performance in the tomato fruit ripeness detection task.

5 Conclusion

In this study, we proposed an enhanced YOLOv10 model for detecting tomato fruit and assessing its maturity. The model introduces a lightweight design based on YOLOv10 and incorporates the UIB structure, which not only improves the detection of small objects but also reduces the computational resources required. Additionally, we integrated the improved CCFM structure for feature fusion, effectively combining multi-scale features to enhance the model's ability to detect targets while also reducing the number of network parameters. Compared to the original model, the improved network achieves superior detection performance, surpassing advanced detection models such as REDETR-ResNet50, YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv10, while significantly lowering both the model's parameter count and computational complexity.

Future work will focus on collecting fruit samples from various agricultural environments to enhance the model's generalization capabilities. We plan to explore multi-modal data fusion and improve the model's adaptability to diverse data types, aiming to boost operational efficiency on mobile devices and embedded systems while reducing computational and storage overhead. Ultimately, the goal is to provide more effective support for agricultural robots and automation equipment.

Acknowledgments. I would like to thank the support from the fund project of the Science and Technology Program of Inner Mongolia Autonomous Region (2020GG0264)

References

1. Chunjiang, Z.: State-of-the-art and recommended developmental strategic objectives of smart agriculture. Smart agriculture **1**(1), 1 (2019)
2. Hua, X., Li, H., Zeng, J., Han, C., Chen, T., Tang, L., Luo, Y.: A review of target recognition technology for fruit picking robots: from digital image processing to deep learning. Applied Sciences **13**(7), 4160 (2023)
3. Wang, Z., Xun, Y., Wang, Y., Yang, Q.: Review of smart robots for fruit and vegetable picking in agriculture. International Journal of Agricultural and Biological Engineering **15**(1), 33–54 (2022)
4. Chen, W., Liu, M., Zhao, C., Li, X., Wang, Y.: Mtd-yolo: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. Computers and Electronics in Agriculture **216**, 108533 (2024)
5. Bhargava, A., Bansal, A.: Fruits and vegetables quality evaluation using computer vision: A review. Journal of King Saud University-Computer and Information Sciences **33**(3), 243–257 (2021)
6. Surya Prabha, D., Satheesh Kumar, J.: Assessment of banana fruit maturity by image processing technique. Journal of food science and technology **52**, 1316–1327 (2015)

7. Lin, G., Tang, Y., Zou, X., Cheng, J., Xiong, J.: Fruit detection in natural environment using partial shape matching and probabilistic Hough transform. Precision Agriculture **21**, 160–177 (2020)
8. Dong, S., Wang, P., Abbas, K.: A survey on deep learning and its applications. Computer Science Review **40**, 100379 (2021)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
10. Girshick, R.: Fast r-cnn. arXiv preprint arXiv:1504.08083 (2015)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1137–1149 (2016)
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
13. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: Yolov10: Realtime end-to-end object detection. arXiv preprint arXiv:2405.14458 (2024)
14. Zheng, H., Wang, G., Li, X.: Yolox-dense-ct: a detection algorithm for cherry tomatoes based on yolox and densenet. Journal of Food Measurement and Characterization **16**(6), 4788–4799 (2022)
15. Cai, Y., Cui, B., Deng, H., Zeng, Z., Wang, Q., Lu, D., Cui, Y., Tian, Y.: Cherry tomato detection for harvesting using multimodal perception and an improved yolov7-tiny neural network. Agronomy **14**(10), 2320 (2024)
16. Qin, D., Leichner, C., Delakis, M., Fornoni, M., Luo, S., Yang, F., Wang, W., Banbury, C., Ye, C., Akin, B., et al.: Mobilenetv4: Universal models for the mobile ecosystem. In: European Conference on Computer Vision. pp. 78–96. Springer (2025)
17. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1580–1589 (2020)
18. Laboro Tomato: Instance Segmentation Dataset, https://github.com/laboroai/LaboroTomato, last accessed 2024/9/11