Late-Stage Pear Inflorescence Detection Based on an Improved Ghost-YOLOv5s-BiFPN Algorithm

Authors: Xia Ye, Xiaohui Lei, Qi Yannan, Xu Tao, Yuan Quanchun, Pan Jian, Jiang Saike, Lü Xiaolan, Lü Xiaolan

Date: 2023-02-17T00:00:00+00:00

Abstract

Flower thinning is an important agronomic practice in pear production. Mechanized intelligent flower thinning is a rapidly developing approach today, and the classification and detection of flowers and buds are fundamental requirements to ensure proper operation of flower thinning machinery. To address the problem of pear inflorescence detection and classification in current intelligent pear orchard production, this study proposes a Ghost-YOLOv5s-BiFPN algorithm based on improved YOLOv5s for recognizing inflorescences in horizontal trellis pear orchards. After annotating and augmenting field-collected images of pear buds and flowers, they are fed into the algorithm for training to obtain a detection model. Ghost-YOLOv5s-BiFPN replaces the original Path Aggregation Network (PAN) structure with a weighted Bi-directional Feature Pyramid Network (BiFPN), enabling effective fusion of multi-scale target features extracted by the network. Additionally, the Ghost module replaces traditional convolution, reducing model parameters and improving operational efficiency without compromising accuracy. Field experimental results demonstrate that the improved Ghost-YOLOv5s-BiFPN algorithm achieves detection accuracies of 93.2% and 89.4% for buds and flowers in pear inflorescences, respectively, with an average precision of 91.3% for both targets, a detection time of 29 ms per image, and a model size of 7.62 M. Compared with the original YOLOv5s algorithm, detection precision and recall are improved by 4.2% and 2.7%, respectively, while detection time and model parameters are reduced by 9 ms and 46.6%, respectively. The algorithm proposed in this study can accurately recognize and classify pear buds and flowers, providing technical support for the subsequent implementation of intelligent flower thinning in pear orchards.

Full Text

Detection of Pear Inflorescence Based on Improved Ghost-YOLOv5s-BiFPN Algorithm

XIA Ye[1,2], LEI Xiaohui[1], QI Yannan[1], XU Tao[1], YUAN Quanchun[1], PAN Jian[1], JIANG Saike[1], LYU Xiaolan[1]*

[1]Institute of Agricultural Facilities and Equipment, Jiangsu Academy of Agricultural Sciences / Key Laboratory of Modern Horticultural Equipment, Ministry of Agriculture and Rural Affairs, Nanjing 210014, China
[2]Institute of Agricultural Engineering, Jiangsu University, Zhenjiang 210200, China

Abstract: Flower thinning is a crucial agronomic practice in pear production. Mechanized and intelligent flower thinning represents a rapidly developing approach, where the classification and detection of flowers and buds constitute fundamental requirements for ensuring proper operation of thinning machinery. This study addresses the challenges of pear inflorescence detection and classification in current intelligent pear orchard production by proposing an improved YOLOv5s-based recognition algorithm called Ghost-YOLOv5s-BiFPN for Y-shaped trellis pear orchards. The detection model was obtained by annotating and augmenting field-collected images of pear buds and flowers, then training the algorithm with this expanded dataset. The Ghost-YOLOv5s-BiFPN algorithm employs a weighted bidirectional feature pyramid network (BiFPN) to replace the original path aggregation network structure, enabling effective fusion of multi-scale target features. Simultaneously, Ghost modules replace traditional convolutions, reducing model parameters and improving operational efficiency on devices without compromising accuracy. Field experimental results demonstrate that the Ghost-YOLOv5s-BiFPN algorithm achieved detection accuracies of 93.21% for buds and 89.43% for flowers in pear inflorescences, with an average accuracy of 91.32%. The average detection time per image was 29 ms, and the model size was 7.62 MB. Compared with the original YOLOv5s algorithm, detection accuracy improved by 4.18%, while detection time and model parameters decreased by 9 ms and 46.63%, respectively. The proposed algorithm enables precise identification and classification of pear buds and flowers, providing technical support for subsequent intelligent flower thinning in pear orchards.

Keywords: pear inflorescence; intelligent recognition; YOLOv5s; BiFPN; lightweight model

---

1 Introduction

Pear trees produce far more flowers than fruit, making flower thinning an essential agronomic practice in pear production management. Currently,

manual methods dominate pear flower thinning, which is labor-intensive, time-consuming, and wastes tree nutrients. Although mechanical thinning tools exist, they operate through random striking, resulting in imprecise operation. Consequently, intelligent flower thinning technology is becoming increasingly important in orchards, with inflorescence detection and recognition being the primary task for intelligent thinning.

In recent years, various detection algorithms have been widely applied in agricultural picking and monitoring applications [1-4], with significant progress achieved in fruit recognition research. Du et al. [5] employed an improved Mask R-CNN algorithm with enhanced path aggregation to identify grape flower spikes and fruit stems, locating thinning clamping points through set logic algorithms with 83.3% accuracy. Chen et al. [6] utilized an improved Single Shot MultiBox Detector (SSD) algorithm with MobileNetV3 lightweight modules for tomato flower recognition, achieving 92.57% accuracy and 0.079 s/f detection speed, substantially improving model efficiency. Long et al. [7] incorporated Convolutional Block Attention Module (CBAM) attention mechanisms into YOLOv4' s cross-stage partial residual modules for strawberry fruit recognition across different growth stages, obtaining average precisions of 92.38%, 82.45%, 68.01%, and 92.31% for flowering, fruit expansion, green fruit, and ripening stages, respectively. Wu et al. [8] improved YOLOv4 for apple flower detection using channel pruning methods, reducing parameters by 96.74% while maintaining 97.31% average precision. Farjon et al. [9] employed Faster-RCNN with transfer learning and professional grower annotations to discriminate different apple flower blooming stages in canopies, achieving 68% average precision with results highly consistent with manual discrimination.

Among existing methods, R-CNN [10] class algorithms as two-stage detectors offer high accuracy but suffer from low efficiency and substantial computational requirements, making them unsuitable for resource-constrained embedded devices. SSD [11] provides speed advantages over R-CNN but has limited detection precision. YOLO [12] as a one-stage detector achieves real-time processing at 45 fps for standard versions and 155 fps for lightweight versions [13], with YOLOv5 employing CSPDarkNet53 backbone networks that reduce computation while maintaining accuracy through cross-stage hierarchical structure [14, 15]. However, original YOLO models struggle to achieve required processing efficiency on computationally limited embedded devices for intelligent thinning tasks and often overlook small-sized targets in pear flower recognition. Irregular branch growth, dense flowers, varying target sizes, and severe occlusion in typical orchard environments further impact detection accuracy.

Addressing these challenges, this study focuses on flowering-stage pear flowers as research objects, detecting flowers and buds under various conditions. The weighted bidirectional feature pyramid network enhances multi-scale feature fusion capabilities, while lightweight modules streamline network layers to reduce parameters for embedded device deployment.

2 Materials and Methods

2.1 Data Acquisition

Pear inflorescence data were collected using a Sony DSC-RX100 digital single-lens reflex camera in Nanjing and surrounding areas. The variety was "Sucui No. 1," with collection dates from March 10 to March 30, 2022. Images were captured in batches during both bright daylight and low evening sunlight conditions, yielding 2,163 original images saved as *.jpg files at 5472$×$3648 pixel resolution. As the orchard employed a horizontal trellis planting pattern [Figure 1: see original paper], data collection was performed along tree rows with individual branches as units.

The dataset comprised 1,658 training images and 505 validation images. To prevent overfitting from insufficient data, an OpenCV-based processing program performed data augmentation through brightness adjustment, rotation, Gaussian noise addition, and sharpness modification. Specific transformations included brightness reduction to 60% and 45% of original, 0-180° rotation, Gaussian noise variance of 0.01, and sharpness reduction of 0-20%. Each image underwent random combination transformations while ensuring no duplicate augmented images from the same source. To minimize repeated annotation effort, the program applied identical positional transformations to existing target annotations, directly generating annotated augmented data. Two examples of random combination augmentation strategies are shown in [Figure 2: see original paper]. A 10-fold augmentation strategy expanded the original dataset to 21,630 annotated images.

2.2 Algorithm Implementation

2.2.1 YOLOv5 Object Detection Algorithm YOLO transforms object detection into a probability regression problem by dividing images into anchor boxes and predicting bounding box parameters. It directly obtains target categories and estimated probabilities, significantly improving detection speed compared to two-stage RCNN networks. The standard YOLO processes 45 frames per second in real-time [12], while lightweight versions achieve 155 fps [13].

YOLOv5 employs CSPDarkNet53 as its backbone network, which divides feature maps into two parts and merges them through cross-stage hierarchical structures, reducing computation while maintaining accuracy. The neck network utilizes Feature Pyramid Networks (FPN) + Path Aggregation Network (PAN) structure, where FPN transmits strong semantic features top-down and PAN conveys strong localization features bottom-up, enabling bidirectional aggregation of features from different backbone layers. The prediction head performs probability regression using grid-based anchor boxes on multi-scale feature maps.

The YOLOv5 feature extraction network primarily comprises CBS modules, CSP_X blocks, and Spatial Pyramid Pooling (SPP) layers. CBS modules con-

sist of Convolution + Batch Normalization + SiLU activation, extracting features, normalizing them to accelerate learning, and mapping them while removing redundancy. CSP_X blocks cascade CBS with X residuals. SPP layers convert feature maps into fixed-size vectors, enabling fusion of local and global features from different input sizes. The complete network structure is shown in [Figure 3: see original paper].

For pre-thinning detection of buds and flowers, this study selected the smallest YOLOv5s version (minimum depth and feature map width) and fine-tuned its parameters. Input images were resized to 640$×$640 pixels, with learning rate, batch size, and iterations set to 1%, 32, and 200, respectively. Two classes were defined: flowers and buds, with petals-appearing buds classified as flowers.

Initial YOLOv5s model training showed effective recognition and classification but revealed large parameter counts consuming substantial computational resources, preventing efficient operation on lightweight embedded devices. Additionally, small target recognition performance was poor, necessitating network modifications for improved feature fusion and lightweight transformation for outdoor deployment.

2.2.2 Ghost-Integrated YOLOv5 Algorithm To further reduce parameters, Ghost networks [16] were introduced into YOLOv5. Ghost networks utilize Ghost convolution modules that address redundancy in traditional CNN structures. Conventional CNNs require extensive floating-point operations for ideal accuracy, while lightweight models like MobileNet [17-19] and ShuffleNet [20, 21] reduce computations but don' t effectively handle redundant feature maps.

Ghost convolution first generates basic intrinsic feature maps through 1$×$1 convolution, then applies linear transformations ϕ_1, ϕ_2, ⋯, ϕ to each feature map to produce redundant features, which are fused with original features to increase channel count. This linear operation approach generates redundant feature maps at lower cost than standard convolution. The Ghost module was implemented in PyTorch and integrated into YOLOv5, replacing traditional convolutions in the backbone network (CSP and CBS structures) while keeping the neck and prediction layers unchanged.

During early training, increasing epochs produced numerous convolutional layers with gradient vanishing issues. The ReLU activation function in Ghost modules produces zero output in the negative half-axis, preventing neurons from learning effective features. The Hard-Swish activation function from MobileNetV3 was adopted as an alternative, preserving Swish' s unbounded-above, bounded-below characteristics while replacing exponential operations with lower computational cost, making it more suitable for embedded deployment. Hard-Swish activates negative gradient information when $x < 0$, making it ideal for this task environment.

2.2.3 Weighted Bidirectional Feature Pyramid Network During network training, features from large targets are preserved through deep convolution while small target features may disappear. Therefore, feature layers from different depths for the same target must be fused. YOLOv5 uses Path Aggregation Network (PANet) [22] for multi-scale feature fusion, as shown in [FIGURE:7(a)], which bidirectionally propagates features to transfer strong semantic information from deep layers to shallow layers and strong localization information from shallow layers to deep layers.

However, PANet simply adds different features together, which can produce unequal contributions from different resolution features since target sizes vary across images. Large features dominate the fusion while small features contribute minimally. To address this, Weighted Bidirectional Feature Pyramid Network (BiFPN) [23] was adopted, as shown in [FIGURE:7(b)]. BiFPN incorporates attention mechanisms by adding learnable weights to dynamically adjust each scale' s contribution during fusion. It also adds residual connections to enhance feature expression and omits single-input/single-output nodes that don' t participate in feature fusion to reduce computation.

To improve small target detection, a 160$\times$160 $feature layer was added to YOLOv5's feature fusion section, with t$ and fused with the new 160$\times$160 $layer. An additional 160\times$160 detection layer was incorporated in the prediction head for small target detection. The modified neck network and prediction head are shown in [Figure 8: see original paper].

---

3 Results and Discussion

3.1 Model Performance Evaluation

Model performance was evaluated from both accuracy and computational efficiency perspectives. For pear flower recognition, computational efficiency directly impacts subsequent thinning operations and was considered the primary evaluation metric.

3.1.1 Model Accuracy Metrics Accuracy evaluation relied on four metrics: Precision (P), Recall (R), Mean Average Precision (mAP), and F1 score, where higher values indicate better performance. These metrics are calculated using formulas (2)-(5):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (2) \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (3) \tag{2}$$

$$\text{mAP} = \frac{\sum_{i=1}^{N(\text{Class})} \text{AP}_i}{N(\text{Class})} \quad (4) \tag{3}$$

$$\text{F1 score} = \frac{2 \times P \times R}{P + R} \quad (5) \tag{4}$$

where True Positive (TP) represents correctly identified positive samples, False Positive (FP) represents negative samples incorrectly identified as positive, and False Negative (FN) represents positive samples incorrectly identified as negative. Average Precision (AP) is the mean precision for each sample class.

3.1.2 Model Efficiency Metrics Efficiency evaluation comprised three metrics: Parameters, Floating Point Operations (GFLOPs), and average detection time. Parameters are determined by network structure, with each parameter typically stored as 32-bit in PyTorch, making model size an alternative parameter count indicator. GFLOPs represent required computation quantity, while average detection time was calculated as the mean of 10 test images.

3.2 Experimental Results

3.2.1 Effectiveness of BiFPN and Ghost Modules To validate BiFPN effectiveness, a network with BiFPN replacement was trained. Grad-CAM visualization compared network attention between original PANet and BiFPN structures. As shown in [Figure 9: see original paper], BiFPN produced higher heatmaps for target regions and lower attention to irrelevant environmental areas compared to PANet, demonstrating superior extraction of complete flower features while reducing environmental distraction.

Ghost module effectiveness was verified by visualizing feature maps after the first CBS module in YOLO layers. [Figure 10: see original paper] shows that similar redundant feature maps are generated during convolution, confirming the necessity of Ghost module' s linear processing for redundancy reduction.

3.2.2 Ablation Study Ablation experiments assessed individual and combined effects of BiFPN and Ghost modules. Training was conducted in PyTorch on a desktop server with Intel® CoreTM E5 V3 CPU, 32 GB RAM, and 12 GB GeForce GTX 3090 GPU, using Ubuntu 20.04 and CUDA 11.4 acceleration. Input size was 640$×$640, training epochs set to 1000, initial learning rate 0.001, with hyperparameter evolution for dynamic adjustment.

[Table 1] compares performance parameters. YOLOv5s-BiFPN improved mAP and recall by 5.1% and 4.2% respectively over original YOLOv5s, with only 1.2%

parameter increase and 3 ms detection time increase. Ghost-YOLOv5s reduced parameters, model size, and GFLOPs by 47.6%, 45.3%, and 48.7% respectively, with 11 ms faster detection, though mAP decreased by 0.9% and recall by 0.7%. Ghost-YOLOv5s-BiFPN combined both improvements, increasing mAP and recall by 4.2% and 2.7% while reducing parameters, model size, and GFLOPs by 46.6%, 44.4%, and 47.5%, respectively, with 9 ms faster detection. This demonstrates that Ghost convolution significantly reduces parameters without substantial accuracy loss, while BiFPN enhances small target detection, making the combined approach suitable for embedded deployment.

Table 1 Comparison of performance parameters between improved YOLOv5s and original YOLOv5s

Model	mAP/%	Recall/%	F1 Score/%	Parameters	GFLOPs	Avg. Detection Time/ms	Model Size/M
YOLOv5s	87.1	87.2	87.1	7,015,519	16.0	38	14.5
YOLOv5s-BiFPN	92.2	91.4	91.8	7,101,064	16.5	41	14.7
Ghost-YOLOv5s	86.2	86.5	86.3	3,678,423	8.2	27	7.9
Ghost-YOLOv5s-BiFPN	91.3	89.9	90.6	3,743,968	8.4	29	8.1

3.2.3 Pear Inflorescence Detection Results The proposed Ghost-BiFPN YOLOv5 method was tested on 118 pear bud and flower images containing 633 flowers and 304 buds. The model detected 572 flowers and 290 buds, with 538 true flowers and 271 true buds identified, achieving recall rates of 85.3% and 89.4%, and precision rates of 89.4% and 93.2% for flowers and buds, respectively.

As shown in [Figure 11: see original paper], the model performs well under various lighting conditions: strong sunlight [FIGURE:11(a)], uniform overcast light [FIGURE:11(d)], direct sunlight, and backlighting [FIGURE:11(c)]. It successfully detects occluded targets [FIGURE:11(5)] and flowers with heterochromatic stamens [FIGURE:11(7)], demonstrating generalization capability. However, some missed detections and duplicate detections occur, primarily due to overlapping targets causing IoU-based suppression [FIGURE:11(4,6)]. Bud recall typically exceeds flower recall because buds have more uniform features while flowers exhibit greater morphological complexity.

4 Conclusion

This study proposes an improved YOLOv5s model integrating BiFPN and Ghost modules for pear inflorescence recognition under horizontal trellis systems. Key

findings include:

1. The improved Ghost-YOLOv5s-BiFPN model achieved 91.3% mAP and 89.9% recall on the pear inflorescence test set, reducing parameters by 46.6% and detection time to 29 ms per image compared with original YOLOv5s. While Ghost modules slightly reduced accuracy, the trade-off was acceptable given the substantial model lightweighting. Future work will explore channel pruning for optimal lightweighting strategies.

2. Although the model effectively detects individual targets, performance on overlapping targets requires improvement. Future research will modify annotation strategies, augment overlapping target datasets for transfer learning, and optimize IoU parameters. To address lower flower recall, CBAM attention mechanisms will be considered to enhance model focus on such targets.

The algorithm provides accurate pear bud and flower identification and classification, offering technical support for intelligent flower thinning implementation in pear orchards.

References

[1] ZHANG F, CHEN Z, BAO R, et al. Recognition of dense cherry tomatoes based on improved YOLOv4-LITE lightweight neural network[J]. Transactions of the CSAE, 2021, 37(16): 270-278.

[2] LIU T, TENG G, YUAN Y, et al. Winter jujube fruit recognition method based on improved YOLO v3 under natural scene[J]. Transactions of the CSAM, 2021, 52(5): 17-25.

[3] KANG H, CHEN C. Fruit detection, segmentation and 3D visualisation of environments in apple orchards[J]. Computers and Electronics in Agriculture, 2020, 171: 105302.

[4] WANG Y, LYU J, XU L, et al. A segmentation method for waxberry image under orchard environment[J]. Scientia Horticulturae, 2020, 266: 109309.

[5] DU W, WANG C, ZHU Y, et al. Fruit stem clamping points location for table grape thinning using improved mask R-CNN[J]. Transactions of the CSAE, 2022, 38(1): 169-177.

[6] CHEN X, WU P, ZU S, et al. Study on identification method of thinning flower and fruit of tomato based on improved SSD lightweight neural network[J]. China Cucurbits and Vegetables, 2021, 34(9): 38-44.

[7] LONG J, GUO W, LIN S, et al. Strawberry growth period recognition method under greenhouse environment based on improved YOLOv4[J]. Smart Agriculture, 2021, 3(4): 99-110.

[8] WU D, LYU S, JIANG M, et al. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments[J]. Computers and Electronics in Agriculture, 2020, 178: 105742.

[9] FARJON G, KRIKEB O, HILLEL A, et al. Detection and counting of flowers on apple trees for better chemical thinning decisions[J]. Precision Agriculture, 2020, 21(3): 503-521.

[10] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2014: 580-587.

[11] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]// European Conference on Computer Vision. Berlin, German: Springer, 2016: 21-37.

[12] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]// The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2016: 779-788.

[13] SHAFIEE M J, CHYWL B, LI F, et al. Fast YOLO: A fast you only look once system for real-time embedded object detection in video[J/OL]. arXiv: 1709.05943, 2017.

[14] REDMON J, FARHADI A. YOLOv3: An incremental improvement[J/OL]. arXiv: 1804.02767, 2018.

[15] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: Optimal speed and accuracy of object detection[J/OL]. arXiv: 2004.10934, 2020.

[16] HAN K, WANG Y, TIAN Q, et al. Ghostnet: More features from cheap operations[C]// The IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2020: 1580-1589.

[17] HOWARD A G, ZHU M, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[J/OL]. arXiv: 1704.04861, 2017.

[18] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: Inverted residuals and linear bottlenecks[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, New York, USA: IEEE, 2018.

[19] HOWARD A, SANDLER M, CHU G, et al. Searching for MobileNetV3[C]// The IEEE/CVF International Conference on Computer Vision. Piscataway, New York, USA: IEEE, 2019: 1314-1324.

[20] ZHANG X, ZHOU X, LIN M, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices[C]// The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2018: 6848-6856.

[21] MA N, ZHANG X, ZHENG H T, et al. ShuffleNet V2: Practical guidelines for efficient CNN architecture design[C]// The European Conference on Computer Vision (ECCV). Piscataway, New York, USA: IEEE, 2018: 116-131.

[22] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation[C]// The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2018: 8759-8768.

[23] TAN M, PANG R, LE Q V. EfficientDet: Scalable and efficient object detection[C]// The IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2020: 10781-10790.

Note: Figure translations are in progress. See original paper for figures.