

Article

Instance Segmentation of Sugar Apple (*Annona squamosa*) in Natural Orchard Scenes Using an Improved YOLOv9-seg Model

Guanquan Zhu ^{1,†}, Zihang Luo ^{2,†}, Minyi Ye ¹, Zewen Xie ³, Xiaolin Luo ², Hanhong Hu ⁴, Yinglin Wang ⁵, Zhenyu Ke ⁴, Jiaguo Jiang ¹ and Wenlong Wang ^{4,*}

¹ School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China; guanquanzhu@gzhu.edu.cn (G.Z.)

² School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China; 1275174812@gzhu.edu.cn (Z.L.); luo640@gzhu.edu.cn (X.L.)

³ School of Physics and Materials Science, Guangzhou University, Guangzhou 510006, China

⁴ School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China

⁵ School of Life Sciences, South China Normal University, Guangzhou 510631, China

* Correspondence: wllwang@gzhu.edu.cn; Tel.: +86-20-39366923

[†] These authors contributed equally to this work.

Abstract: Sugar apple (*Annona squamosa*) is prized for its excellent taste, rich nutrition, and diverse uses, making it valuable for both fresh consumption and medicinal purposes. Predominantly found in tropical regions of the Americas and Asia, its harvesting remains labor-intensive in orchard settings, resulting in low efficiency and high costs. This study investigates the use of computer vision for sugar apple instance segmentation and introduces an improved deep learning model, GCE-YOLOv9-seg, specifically designed for orchard conditions. The model incorporates Gamma Correction (GC) to enhance image brightness and contrast, improving target region identification and feature extraction in orchard settings. An Efficient Multiscale Attention (EMA) mechanism was added to strengthen feature representation across scales, addressing sugar apple variability and maturity differences. Additionally, a Convolutional Block Attention Module (CBAM) refined the focus on key regions and deep semantic features. The model's performance was evaluated on a self-constructed dataset of sugar apple instance segmentation images captured under natural orchard conditions. The experimental results demonstrate that the proposed GCE-YOLOv9-seg model achieved an F1 score (*F1*) of 90.0%, a precision (*P*) of 89.6%, a recall (*R*) level of 93.4%, a mAP@0.5 of 73.2%, and a mAP@[0.5:0.95] of 73.2%. Compared to the original YOLOv9-seg model, the proposed GCE-YOLOv9-seg showed improvements of 1.5% in the F1 score and 3.0% in recall for object detection, while the segmentation task exhibited increases of 0.3% in mAP@0.5 and 1.0% in mAP@[0.5:0.95]. Furthermore, when compared to the latest model YOLOv12-seg, the proposed GCE-YOLOv9-seg still outperformed with an F1 score increase of 2.8%, a precision (*P*) improvement of 0.4%, and a substantial recall (*R*) boost of 5.0%. In the segmentation task, mAP@0.5 rose by 3.8%, while mAP@[0.5:0.95] demonstrated a significant enhancement of 7.9%. This method may be directly applied to sugar apple instance segmentation, providing a promising solution for automated sugar apple detection in natural orchard environments.



Academic Editor: Dimitre Dimitrov

Received: 28 April 2025

Revised: 4 June 2025

Accepted: 11 June 2025

Published: 13 June 2025

Citation: Zhu, G.; Luo, Z.; Ye, M.; Xie, Z.; Luo, X.; Hu, H.; Wang, Y.; Ke, Z.; Jiang, J.; Wang, W. Instance Segmentation of Sugar Apple (*Annona squamosa*) in Natural Orchard Scenes Using an Improved YOLOv9-seg Model. *Agriculture* **2025**, *15*, 1278. <https://doi.org/10.3390/agriculture15121278>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: deep learning; computer vision; YOLOv9-seg; sugar apple (*Annona squamosa*)

1. Introduction

Sugar apple (*Annona squamosa*), also known as custard apple, is a tropical fruit highly valued for its rich nutritional content and potential medicinal properties. It is widely

avored by consumers due to its sweet taste, soft pulp, and high antioxidant levels [1]. With increasing demand in both domestic and international markets, the sugar apple has become a fruit of considerable economic importance [2]. However, current harvesting practices still rely heavily on manual labor [3–5]. Given the fruit’s delicate and easily bruised surface, physical contact during manual picking often leads to damage, resulting in post-harvest losses and reduced market value [6]. Moreover, growing labor costs further inflate overall production expenses [7,8]. Therefore, developing an automatic detection and recognition method for sugar apples in natural orchard environments holds great potential to minimize harvest-related losses and to lower operational costs.

In recent years, to address the challenge of fruit recognition in orchard environments, researchers have proposed various methods. These approaches primarily rely on traditional image processing techniques and handcrafted feature extraction [9], such as color thresholding, edge detection, morphological operations, and geometry-based object recognition using features like roundness or the aspect ratio [10–13]. While these methods have shown effectiveness under controlled conditions with stable lighting and simple backgrounds, their robustness and generalizability are significantly limited in natural orchard scenes [14,15]. Complex illumination, frequent occlusions, and the high visual similarity between fruits and leaves pose serious challenges for traditional techniques. As a result, improving fruit detection accuracy under such complex conditions has become a key area of research focus.

With the rapid development of computer vision and artificial intelligence technologies, an increasing number of studies have adopted deep learning methods to address the challenge of fruit recognition in natural orchard environments. Convolutional neural networks (CNNs), known for their powerful feature extraction capabilities, have become a core technology in this field. Sa et al. [16] proposed a fruit detection method combining Faster R-CNN with multispectral imagery, successfully achieving accurate apple detection in orchards. Bargoti and Underwood [17] used Faster R-CNN to detect various fruits such as apples, mangoes, and oranges, demonstrating strong generalizability under natural lighting and occlusion conditions. In addition, the YOLO (You Only Look Once) family of object detectors, known for their high real-time performance, has been widely applied in orchard scenarios. Rahnemoonfar and Sheppard [18] trained a YOLO model on synthetically generated datasets, achieving high accuracy in tomato detection even under occlusion and overlap. In addition, instance segmentation techniques have been widely applied for the detection of various types of fruits. For example, studies have achieved the high-precision detection of tomatoes by combining deep instance segmentation, data synthesis, and color analysis methods [19]; Mask R-CNN models integrated with attention mechanisms have also been used for the instance segmentation of apples, enabling the precise segmentation of fruit regions and the effective differentiation of individual instances [20]. However, compared to these fruits, sugar apples present more unique and complex challenges in instance segmentation due to their irregular shapes and delicate surface characteristics [21].

Compared with earlier versions such as YOLOv5, YOLOv7, and YOLOv8, YOLOv9 introduces improvements in feature extraction, attention mechanisms, and multi-scale processing, significantly enhancing its performance in small object detection and robustness under occlusion [22]. At the same time, it maintains a high detection speed and model lightweight characteristics. Compared with Mask R-CNN, YOLOv9 may be slightly less precise in pixel-level segmentation, but it offers clear advantages in real-time performance, inference speed, and resource efficiency, making it more suitable for high-efficiency fruit detection tasks in agricultural settings [23]. In recent years, researchers have also proposed enhancements based on YOLOv9. For instance, Lu et al. introduced MAR-YOLOv9 [24], a lightweight, cross-dataset augmented agricultural object detection method that achieves

higher detection accuracy with lower computational complexity, further extending the practical applicability of YOLOv9 in agricultural computer vision. Compared with traditional image processing methods, deep learning models demonstrate stronger robustness and adaptability in complex environments, providing a more promising technical foundation for automated orchard detection systems.

Nowadays, many researchers have begun to focus on applying artificial intelligence algorithms for the automatic processing of sugar apples. Xie et al. [25] proposed the ECD-DeepLabv3+ model, an improved semantic segmentation model, for sugar apple maturity detection. Sanchez et al. [26] used image processing through CNN to determine the maturity of sugar apples, and created a system that can detect and classify sugar apples. Thite et al. [27] created a dataset for sugar apple lesions, laying a foundation for the subsequent automatic detection of sugar apple diseases using an artificial intelligence algorithm. Tonmoy et al. [28] proposed an effective architecture using multi-head attention and lightweight convolution to achieve state-of-the-art performance in sugar apple scab identification. Gaikwad, Sukanya S. et al. [29] analyzed a sugar apple plant health and leaf spot disease dataset with AlexNet and SqueezeNet, finding that SqueezeNet outperformed AlexNet.

However, research on sugar apple fruit recognition in natural orchard environments remains relatively limited. Referring to the application of deep learning in fruit recognition, Dong et al. [30] primarily used image classification tasks for mango recognition, but this approach had weak information expression capabilities during the model inference stage and struggled to provide fine-grained information such as fruit location, contours, and quantity. Therefore, its representational power is inferior to methods like object detection, semantic segmentation, and instance segmentation [16,31]. Moreover, some studies [25–29] have not fully considered the impact of complex backgrounds, lighting variations, and fruit occlusions in natural orchard environments, factors which significantly affect the model's recognition performance. In contrast to prior studies primarily focused on fruit maturity assessment or disease classification, this study targets real-time instance segmentation in complex natural orchard environments. It seeks to address the challenges introduced by variable lighting conditions, occlusions, and fruit clustering during in-field detection, thereby improving the precision and robustness of fruit recognition models.

Compared to image classification, object detection, and semantic segmentation, instance segmentation enables the precise extraction of object contours at the pixel level and effectively distinguishes different instances within the same class. When handling complex scenarios involving object overlap and occlusion, instance segmentation demonstrates superior robustness and detection accuracy. Therefore, instance segmentation offers a more precise and feasible technical solution for the automatic detection of sugar apples in natural orchard environments.

However, the accuracy of image classification, object detection, and instance segmentation varies significantly across different tasks and application scenarios. In natural orchard environments, due to complex backgrounds, significant lighting variations, and frequent fruit occlusion by foliage, instance segmentation faces greater technical challenges, resulting in relatively lower overall detection accuracy. In previous studies, Xie et al. [25] proposed the ECD-DeepLabv3+ semantic segmentation model, achieving an average accuracy of 94.58% in post-harvest sugar apple maturity segmentation tasks, while the CNN-based sugar apple maturity detection model developed by Sanchez et al. [26] achieved an accuracy of 86.84%. Instance segmentation can distinguish between different categories and instances within the same category. The GCE-YOLOv9-seg model achieved a segmentation recall of 89.4% and a detection recall of 89.6% in natural orchard environments (single-object tasks), highlighting the advantages of instance segmentation over object detection and

semantic segmentation in sugar apple detection tasks. Jroni et al. [31] emphasize the importance of deep learning-based fruit detection systems in improving the efficiency and precision of agricultural operations. Moreover, sugar apple instance segmentation not only enables precise fruit localization but also provides critical data support for applications such as fruit counting, yield estimation, automated harvesting, and intelligent agricultural management [19,32].

To address the challenges posed by varying illumination, occlusion from leaves and branches, and the clustering of sugar apple fruits in natural orchard environments, this study aims to achieve the precise instance segmentation of sugar apples in complex orchard scenarios. We propose an improved instance segmentation method, GCE-Yolov9-seg, specifically designed for the instance segmentation of sugar apples in natural conditions. This method incorporates Gamma Correction (GC) for image enhancement, improving image quality under low-light and backlight conditions. During model training, it integrates the Convolutional Block Attention Module (CBAM) and the Efficient Multi-scale Attention (EMA) mechanism to enhance the model's ability to recognize fruits obscured by foliage and branches, thereby significantly improving detection accuracy.

This approach demonstrates strong robustness and adaptability to complex and dynamic orchard environments, providing a feasible technical path for the automated harvesting of sugar apples. To evaluate the performance of GCE-YOLOv9-seg, an instance segmentation dataset consisting of 1078 natural orchard images of sugar apples was constructed. Experimental results demonstrate that excellent performance across various metrics, including recall, the F1 score, mAP@50, and mAP@[50:95], is achieved by the proposed model.

The main contributions of this paper are as follows:

- (1) This paper presents the first investigation into the application of instance segmentation for detecting sugar apples (*Annona squamosa*) in natural orchard settings.
- (2) This paper proposes GCE-Yolov9-seg, an improved instance segmentation model that outperforms the original model in sugar apple detection, with gains in the F1 score, recall, mAP@50, and mAP@[50:95].
- (3) This paper creates an instance segmentation dataset for sugar apple images under natural conditions to evaluate the performance of GCE-Yolov9-seg.

2. Materials and Methods

2.1. Dataset

This study aims to investigate the application of instance segmentation techniques for the detection and segmentation of sugar apples in natural orchard environments. Due to the absence of publicly available instance segmentation datasets specifically targeting sugar apple detection and segmentation in orchard settings, a dedicated dataset was constructed. Given the critical importance of training on diverse and representative data for object detection models [20], a total of 1078 images were collected, encompassing both top-view and bottom-view perspectives, as well as various challenging conditions such as direct light, backlight, and occlusion by branches and leaves, thereby accurately reflecting the diversity and complexity of natural orchard environments. As shown in Figure 1, the data were collected at WanAn Sugar Apple Orchard (22.87 °N, 113.35 °E) in Nansha District, Guangzhou City, Guangdong Province. This area is located at the core of the Guangdong–Hong Kong–Macao Greater Bay Area and is characterized by a typical South Asian tropical monsoon climate, which is highly suitable for the cultivation and growth of tropical and subtropical fruit trees. Climatologically, the region experiences an average annual temperature ranging from 21.0 to 23.0 degrees Celsius, with approximately 1600 mm of annual precipitation and about 130 rainy days per year [33].

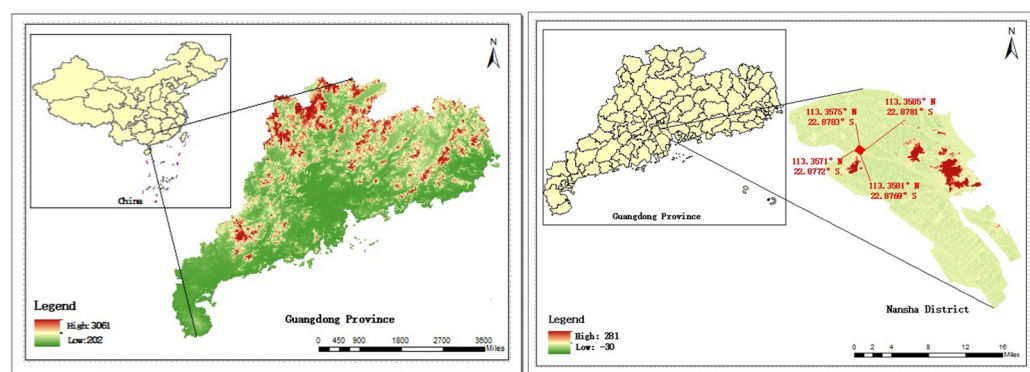


Figure 1. A map of the study area.

Following data collection, the dataset was randomly and evenly divided into a training set (647 images), a validation set (215 images), and a test set (216 images) according to a 6:2:2 ratio. The dataset comprises a total of 1,078 annotated images of sugar apples, categorized into three typical orchard imaging conditions: frontal lighting, backside lighting, and occlusion by foliage. As shown in Table 1, 249 images were collected under frontal lighting (128 for training, 62 for validation, and 59 for testing), 158 under backside lighting (109 for training, 22 for validation, and 27 for testing), and 671 under foliage occlusion (410 for training, 131 for validation, and 130 for testing). The distribution of each category across the training, validation, and test sets was well-balanced, ensuring that the model received sufficient exposure to each condition during training and evaluation. This balanced and representative composition contributed significantly to enhancing the model's generalization ability and robustness in real-world complex orchard environments.

Table 1. Image distribution across categories in the dataset.

Category	Train	Val	Test	Total
Frontal lighting	128	62	59	249
Backside lighting	109	22	27	158
Obscured by foliage	410	131	130	671
Total	647	215	216	1078

The images were captured using different smartphones (iPhone 13 Pro Max, OPPO Reno12, and IQOO Neo7 SE), resulting in varying resolutions (3024×3024 , 3072×3072 , and 3456×3456 pixels). To standardize the dataset, image preprocessing was performed using the Python Imaging Library (version 10.2.0). High-quality scaling was achieved using the LANCZOS resampling method, ensuring the preservation of overall image quality while handling images in the RGBA format. All images were uniformly resized to a resolution of 512×512 pixels. Representative sample images from the constructed dataset are shown in Figure 2.



Figure 2. Some images from the self-made dataset of sugar apples in a natural orchard environment.

2.2. YOLOv9-seg

With the advancement of artificial intelligence (AI), computer vision technologies have found widespread applications across various domains such as agriculture [34],

environmental monitoring [35], and medicine [36]. Within the framework of computer vision, three major tasks are commonly addressed: image classification, object detection, and image segmentation. Compared with the former two, image segmentation has garnered significant attention due to its ability to precisely delineate different objects at the pixel level. Image segmentation can be further categorized into semantic segmentation and instance segmentation. Semantic segmentation focuses on classifying each pixel according to its semantic category, whereas instance segmentation goes a step further by not only identifying the semantic class of each pixel but also distinguishing between different instances of the same object class within an image.

In the field of instance segmentation, the YOLO (You Only Look Once) series has emerged as a prominent solution. The YOLO series of object detection models are known for their high speed and accuracy, making them particularly suitable for real-time applications [37]. While traditional YOLO models have achieved remarkable success in object detection, recent advancements have extended the YOLO framework to instance segmentation tasks, leading to the development of segmentation-capable versions.

We investigated the performance of the YOLO series instance segmentation models (as shown in Section 3.2), and the experimental results demonstrate that YOLOv9-seg excels across various evaluation metrics, outperforming other models overall. Therefore, we selected this model as the preferred choice for this task. The YOLOv9-seg instance segmentation model, based on the YOLO architecture (as shown in Figure 3), integrates object detection and instance segmentation into a unified framework, enabling the precise segmentation of each detected object. The model utilizes an improved deep convolutional neural network (CNN) to extract multi-scale features and incorporates contextual information to generate pixel-level masks for each target. In instance segmentation tasks, YOLOv9-seg outputs not only the object categories and bounding boxes but also detailed object boundaries through a mask generation network, thereby enhancing both segmentation accuracy and processing speed. By adopting an end-to-end training strategy that jointly optimizes object detection and instance segmentation tasks, the model ensures a high degree of consistency between bounding boxes and segmentation masks, achieving efficient and precise instance segmentation even in complex environments.

2.3. Architecture of the proposed GCE-YOLOv9-seg

This paper proposes an instance segmentation model based on YOLOv9-seg, specifically designed for sugar apple (*Annona squamosa*) recognition in orchard environments, significantly enhancing detection and segmentation performance under natural conditions. As illustrated in Figure 4a and Algorithm 1, three key improvements were introduced to the model architecture:

- i. During the image input stage, a Gamma Correction (GC) image enhancement technique was employed to preprocess the orchard-acquired sugar apple images. By improving image brightness and contrast, this method enhances the model's ability to distinguish target regions and extract critical features effectively.
- ii. After the fusion of the backbone feature map P3 and the upsampled features, an Efficient Multiscale Attention (EMA) module was introduced. This module strengthens the model's capability to capture and express multi-scale spatial features from sugar apple images collected in complex orchard environments. Consequently, it enables a more accurate extraction of salient features from targets of varying scales and poses, improving recognition and segmentation performance in scenarios with dense distributions, occlusions, and varying lighting conditions.
- iii. After fusing backbone features P3 and P4, followed by downsampling and concatenation, a Convolutional Block Attention Module (CBAM) was incorporated to enhance

the model's sensitivity to medium-scale targets. Additionally, another CBAM was integrated after the RepNCSPELAN4 module, which follows the fusion of P4 and P5, to further strengthen the model's attention to critical regions within deep semantic features.

Algorithm 1: GCE-YOLOv9-seg: A Gamma Correction and Attention-Enhanced YOLOv9-seg for Sugar Apple Segmentation

Input: Raw sugar apple RGB image I
Output: Pixel-wise segmentation map O

- 1 $F1 \leftarrow \text{Upsample}(F2, P3);$
 $\{3, P4, P5\} \leftarrow \text{Backbone}(I);$ // Extract multi-scale features
// Feature map resolution details:
- 2 // P3: stage 3, stride 8
// P4: stage 5, stride 16
// P5: stage 7, stride 32
- 3 $F1 \leftarrow \text{Upsample}(F2, P3);$
- 4 $F1 \leftarrow \text{Concat}(F1, P4);$
- 5 $F1 \leftarrow \text{RepNCSPELAN4}(F1);$
- 6 $F2 \leftarrow \text{Upsample}(F1);$
- 7 $F2 \leftarrow \text{Concat}(F2, P3);$
- 8 $F2 \leftarrow \text{EMA}(F2);$ // EMA fusion after $P4 \uparrow + P3$
- 9 $P3\text{seg} \leftarrow \text{RepNCSPELAN4}(F2);$
- 10 $P3\text{ds} \leftarrow \text{Adown}(P3\text{seg});$
- 11 $P4\text{seg} \leftarrow \text{Concat}(P3\text{ds}, F1);$
- 12 $P4\text{seg} \leftarrow \text{CBAM}(P4\text{seg});$ // CBAM after $P3 \downarrow + F1$
- 13 $P4\text{seg} \leftarrow \text{RepNCSPELAN4}(P4\text{seg});$
- 14 $P4\text{ds} \leftarrow \text{Adown}(P4\text{seg});$
- 15 $P5\text{seg} \leftarrow \text{Concat}(P4\text{ds}, P5);$
- 16 $P5\text{seg} \leftarrow \text{RepNCSPELAN4}(P5\text{seg});$
- 17 $P5\text{seg} \leftarrow \text{CBAM}(P5\text{seg});$ // CBAM after $P4 \downarrow + P5$
- 18 $O \leftarrow \text{Segment}([P3\text{seg}, P4\text{seg}, P5\text{seg}]);$ // Final segmentation output
- 19 **return** O

2.3.1. Gamma Correction (GC)

In the fields of image processing and computer vision, image enhancement techniques are widely utilized to improve image quality [38]. This is particularly important under natural environmental conditions, where sugar apple orchard images often suffer from complex backgrounds and low contrast. To address the challenges related to image quality in the instance segmentation of sugar apples, this study investigated several commonly used image enhancement methods, including Contrast Limited Adaptive Histogram Equalization (CLAHE) [39], Gamma Correction (GC) [40], Histogram Equalization (HE) [41], and Intensity Contrast Mapping (ICM) [42]. Figure 5 visualizes the effects of different image enhancement algorithms. Each of these methods has distinct characteristics, improving image details and visual effects through local or global contrast enhancement, brightness adjustment, and nonlinear transformations. These improvements provide clearer and more accurate image inputs for the subsequent instance segmentation tasks. The experimental results demonstrated that GC achieved the best performance for sugar apple instance segmentation under orchard conditions (as shown in Section 3.4).

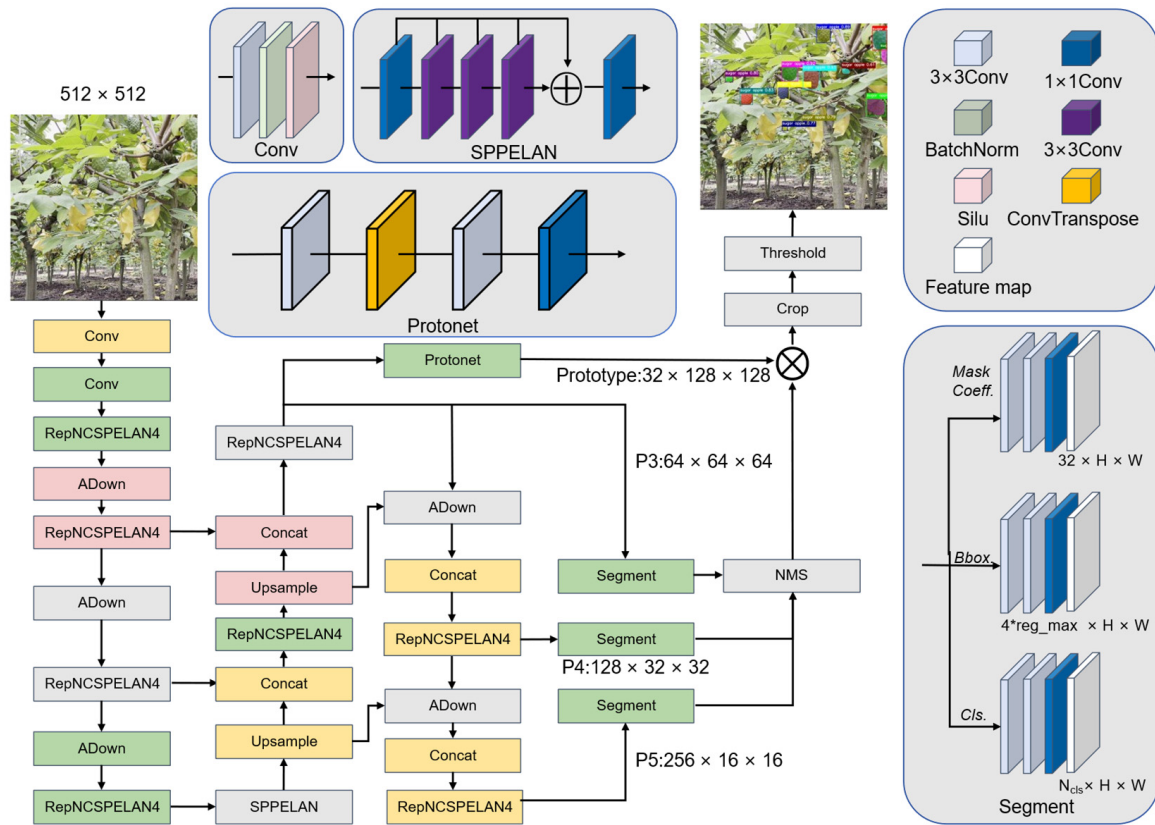


Figure 3. Architecture of YOLOv9-seg.

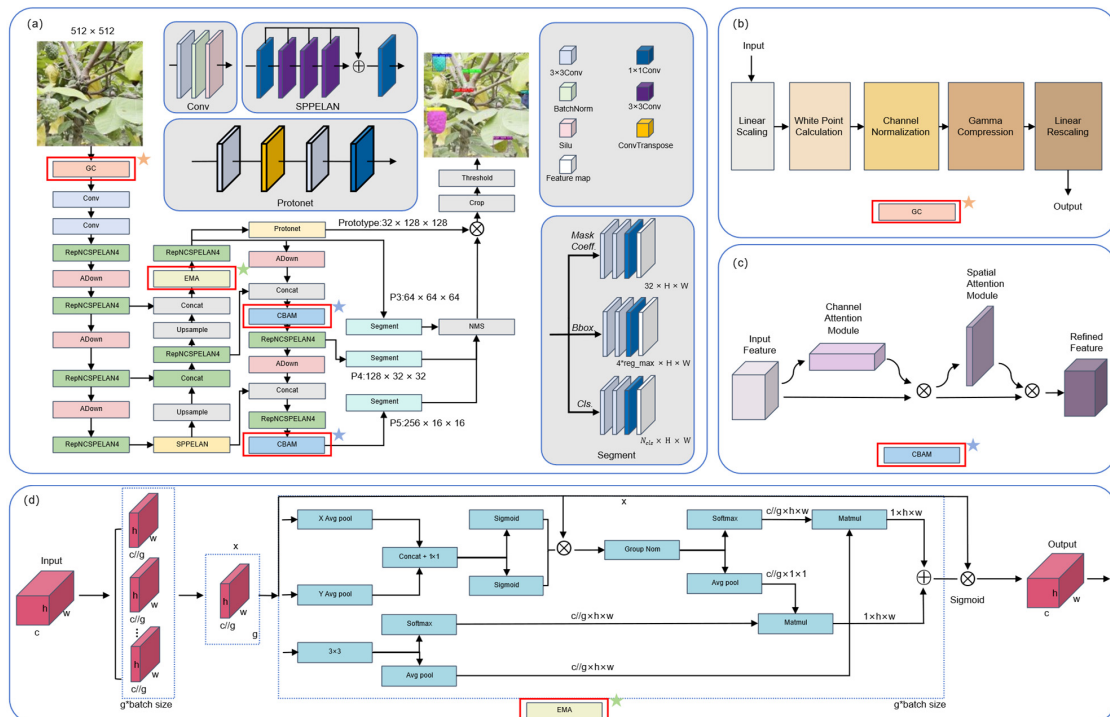


Figure 4. The architecture of the proposed GCE-YOLOv9-seg: (a) the overall structure of GCE-YOLOv9-seg; (b) the Gamma Correction module; (c) the Convolutional Block Attention Module (CBAM); and (d) the Efficient Multiscale Attention (EMA) module.



Figure 5. Visualization of different image enhancement algorithms.

Gamma Correction (GC) is a classic and effective image enhancement technique, with its processing flow illustrated in Figure 4b. GC aims to adjust image brightness through nonlinear gray-level transformation, aligning it more closely with the human visual system's perception of luminance. By introducing a Gamma value (γ), this method compresses or expands the dynamic range of an image, effectively enhancing detail visibility in dark regions and suppressing overexposure in bright areas. Consequently, it improves the overall contrast and visual quality of an image.

$$I_{out}^{(i)}(x, y) = 255 \cdot \left(\frac{I_{in}^{(i)}(x, y) / 255}{\max(I_{in}^{(i)} / 255)} \right)^{\gamma} \quad (1)$$

where $I_{in}^{(i)}(x, y)$ denotes the pixel value at position (x, y) in the i -th channel of the original image, $\gamma = 0.7$ is the Gamma value, $\max(I_{in}^{(i)} / 255)$ represents the maximum value of the normalized pixels in the i -th channel used to prevent brightness overflow, and $I_{out}^{(i)}(x, y)$ denotes the pixel value after enhancement.

In this study, Gamma Correction was applied during the preprocessing stage of input images collected from orchard environments. Considering challenges such as low illumination and shadow occlusion in real-world shooting conditions, the experimental results (as detailed in Section 3.6.1) demonstrated that the best enhancement effect was achieved with a Gamma value (γ) of 0.7. Therefore, a Gamma value of 0.7 was adopted to perform nonlinear enhancement, aiming to improve image brightness uniformity and detail clarity under complex lighting conditions. This preprocessing step not only enhanced the visual quality of the images but also improved the subsequent model's ability to extract key features related to sugar apples, thereby effectively enhancing the overall robustness and accuracy of the detection process.

2.3.2. Convolutional Block Attention Module (CBAM)

The Convolutional Block Attention Module (CBAM) is a lightweight attention mechanism designed for image processing and computer vision tasks, with its schematic diagram shown in Figure 4c [43]. CBAM aims to enhance the feature representation capability of convolutional neural networks by explicitly modeling channel attention and spatial attention. Specifically, the channel attention submodule focuses on extracting discriminative semantic features and strengthening the network's response to critical channel information, while the spatial attention submodule guides the network to focus on salient regions in the spatial dimension by capturing the response intensities at different positions within the feature maps. Figure 4c illustrates the detailed structure of the CBAM.

The CBAM extracts global contextual information from the input feature map by performing global average pooling (GAP) and global max pooling (GMP) along the spatial dimensions. Specifically, for an input feature map $F \in R^{C \times H \times W}$, GAP and GMP operations are first applied independently, generating two channel descriptor vectors. These vectors are then passed through a shared multi-layer perceptron (MLP) for nonlinear transformation. After element-wise summation and a Sigmoid activation function, the channel

attention weight map M_c is obtained. This process can be expressed by the following formula:

$$M_c(F) = \sigma(MLP(GAP(F)) + MLP(GMP(F))) \quad (2)$$

where $\sigma(\cdot)$ denotes the Sigmoid function, and $MLP(\cdot)$ represents the shared multilayer perceptron.

The resulting channel attention weight $M_c \in R^{C \times 1 \times 1}$ is multiplied element-wise with the input feature map F to obtain the weighted feature map F' , which is computed as follows:

$$F' = M_c(F) \odot F \quad (3)$$

On the channel-attended feature map $F' \in R^{C \times H \times W}$, the spatial attention submodule further enhances feature representation by modeling the positional importance along the spatial dimension. First, global average pooling and global maximum pooling are performed along the channel dimension of F' to obtain two 2D feature maps of size $R^{1 \times H \times W}$. These two feature maps are then concatenated along the channel dimension to form a tensor of size $R^{2 \times H \times W}$, which is then passed through a convolutional layer with a 7×7 convolution kernel. After convolution and activation by the Sigmoid function, the spatial attention map M_s is generated. This process can be expressed as follows:

$$M_s(F') = \sigma(f^{7 \times 7}([AvgPool(F'); MaxPool(F')])) \quad (4)$$

where $f^{7 \times 7}(\cdot)$ represents the convolution operation with a 7×7 kernel, and $\sigma(\cdot)$ denotes the Sigmoid activation function. Finally, the spatial attention map $M_s \in R^{1 \times H \times W}$ is element-wise multiplied by the weighted feature map F' to produce the output feature map F'' of the spatial attention submodule, computed as follows:

$$F'' = M_s(F') \odot F' \quad (5)$$

2.3.3. Efficient Multiscale Attention (EMA)

Efficient Multiscale Attention (EMA) is a mechanism based on cross-spatial learning [44], with its schematic diagram shown in Figure 4d. EMA aims to enhance the capability of convolutional neural networks in feature extraction and attention weight computation through multiscale feature fusion and the optimization of feature maps. The core idea of EMA is to introduce multiple parallel sub-networks, where cross-channel attention and cross-spatial attention learning are integrated within these sub-networks. This approach further strengthens the network's ability to capture fine-grained features and global contextual information in images.

Initially, EMA performs spatial attention $A_{spatial}^{(i)}$ and channel attention $A_{channel}^{(i)}$ on the feature map X_i of each scale. Spatial attention adjusts the weight of regions in the feature map by focusing on spatial positions, while channel attention emphasizes the importance of different channels. Specifically, the spatial attention is computed as follows:

$$A_{spatial}^{(i)} = Conv(Sigmoid(W_{spatial} \cdot X_i)) \quad (6)$$

where $W_{spatial}$ is the weight matrix of the spatial attention module, and $Sigmoid(\cdot)$ is the activation function. The channel attention is computed as follows:

$$A_{channel}^{(i)} = Sigmoid(MLP(Global Pooling(X_i))) \quad (7)$$

where MLP is the multi-layer perceptron responsible for generating the attention weights for each channel, and Global Pooling represents a Global Pooling operation, such as global average pooling or global max pooling, used to extract global features.

After obtaining the spatial and channel attention for each scale, EMA fuses the feature maps of different scales via weighted summation. To allow the feature maps of different scales to contribute differently to the final representation, the information from each scale is combined using weighted summation, resulting in a comprehensive multi-scale feature map representation. The mathematical expression is as follows:

$$X_{EMA} = \sum_{i=1}^n \alpha_i \left(X_i \odot A_{spatial}^{(i)} \odot A_{channel}^{(i)} \right) \quad (8)$$

where X_i represents the feature map of the i^{th} scale, $A_{spatial}^{(i)}$ and $A_{channel}^{(i)}$ denote the spatial and channel attention for the i^{th} scale, α_i is the weighting coefficient for the i^{th} scale, and X_{EMA} is the fused multi-scale feature map.

Compared to traditional attention mechanisms, EMA effectively reduces computational resource demands while maintaining strong model performance, making it an ideal choice for image processing in resource-constrained environments. The design of EMA aligns with the objectives of this study, which aims to develop a high-performance instance segmentation network for the detection and segmentation of sugar apples. Therefore, the ECA module was incorporated into the model after the fusion of the backbone feature P3 and the upsampled features to enhance accuracy when capturing critical information from the images.

2.4. Evaluation Indicators

In this study, we evaluated the performance of the proposed model in the instance segmentation task for sugar apples in a natural orchard environment from two aspects: model performance and model complexity. To assess model performance, we employed five evaluation metrics: the F1 score ($F1$), precision (P), recall (R), $mAP@0.5$, and $mAP@[0.5:0.95]$. The F1 score provides a comprehensive evaluation of the balance between precision and recall, reflecting the model's ability to identify true positives. Precision (P) measures the accuracy of the model's positive predictions, while recall (R) evaluates the model's ability to identify true positive samples. $mAP@0.5$ evaluates the model's segmentation accuracy when the Intersection over Union (IoU) threshold is set to 0.5, while $mAP@[0.5:0.95]$ evaluates the model's segmentation performance across multiple IoU thresholds ranging from 0.5 to 0.95, providing a comprehensive assessment of the model's segmentation capability. Using these five metrics, we can evaluate the model's performance from various aspects and ensure its effectiveness in the instance segmentation task.

By comparing the predicted results with the dataset labels, four basic metrics are derived: true positive (TP), false positive (FP), true negative (TN), and false negative (FN), along with the average precision for each class (AP_i), the number of classes (N), and $mAP@IoU$ at different IoU thresholds. These basic metrics are subsequently used to calculate key evaluation indicators, including $F1$, P , R , $mAP@0.5$, and $mAP@[0.5:0.95]$. Additionally, they provide strong support for model performance evaluation. The following are the formulas for calculating the related metrics:

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$mAP@0.5 = \frac{1}{N} \sum_{i=1}^N AP_i \quad (12)$$

$$mAP@[0.5 : 0.95] = \frac{1}{10} \sum_{i=1}^{10} mAPIoUi \quad (13)$$

2.5. Loss Function

The loss function of the YOLOv9-seg model mainly consists of two parts: detection loss and instance segmentation loss, aiming to simultaneously optimize bounding box localization, category classification, and mask prediction. The detection loss includes bounding box regression loss, classification loss, and object confidence loss, which improve the model's detection accuracy and localization precision. The instance segmentation loss focuses on pixel-level mask prediction, enhancing the accuracy and detail of segmentation. The specific loss functions are as follows:

The detection loss L_{detect} consists of three components:

$$L_{detect} = L_{box} + L_{cls} + L_{obj} \quad (14)$$

where L_{box} is the bounding box regression loss, L_{cls} is the classification loss, and L_{obj} is the object confidence loss.

These components are specifically formulated as follows:

$$L_{detect} = 1 - CIoU(b, b^*) - \sum_{c=1}^C y_c \log(\hat{p}_c) - [y_{obj} \log(\hat{p}_{obj}) + (1 - y_{obj}) \log(1 - \hat{p}_{obj})] \quad (15)$$

where b and b^* are the predicted and ground truth bounding boxes, C is the number of classes, y_c is the ground truth class label, $\hat{p}_c$ is the predicted class probability, y_{obj} is the object presence label, and $\hat{p}_{obj}$ is the objectness confidence.

The segmentation loss consists of two main components:

$$L_{seg} = L_{BCE} + L_{Dice} \quad (16)$$

where L_{BCE} is the binary cross-entropy loss, and L_{Dice} is the Dice loss.

These components are specifically formulated as follows:

$$L_{seg} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{m}_i) + (1 - y_i) \log(1 - \hat{m}_i)] + \left(1 - \frac{2 \sum_{i=1}^N y_i \hat{m}_i + \epsilon}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{m}_i + \epsilon} \right) \quad (17)$$

where N is the total number of pixels, $y_i \in \{0, 1\}$ is the ground truth mask label, and $\hat{m}_i \in [0, 1]$ is the predicted mask probability. The first term is the binary cross-entropy loss, and the second term is the Dice loss, which improves the overlap between the predicted and true masks. ϵ is a small constant to avoid division by zero.

3. Results

3.1. Experiment Details

The hardware and software specifications of the experimental platform used in this study are provided in Table 2.

Table 2. Experimental platform’s environment settings.

Parameter	Configuration
CPU	Intel Xeon E5-2678 V3 processor
GPU	NVIDIA GeForce RTX 3090
CUDA version	CUDA 12.2
Operating system	Windows 10
Programming language	Python 3.10
Deep learning framework	Pytorch 2.6.0

To ensure the fairness of the experiment, we maintained consistency in the base parameters throughout the training process. By standardizing the experimental conditions and hyperparameter settings, we effectively eliminated the interference of external variables, ensuring that the model’s performance was solely influenced by the algorithm’s optimization and the structural design. More details can be found in Table 3.

Table 3. Experimental hyperparameter settings.

Parameter	Setting
Epoch	100
Batch Size	16
Optimizer	SGD
Initial Learning Rate	1×10^{-2}
Learning Rate Factor	1×10^{-2}

3.2. Baseline Model Results

3.2.1. Quantitative Analysis of the Baseline Model

In this section, a systematic evaluation of existing instance segmentation methods is conducted, covering models such as Mask R-CNN, YOLOv5-seg, YOLOv6-seg, YOLOv7-seg, YOLOv8-seg, YOLOv9-seg, YOLOv10-seg, YOLOv11-seg, and YOLOv12-seg [45–53]. All experiments were trained and tested on a unified dataset (the self-made orchard sugar apple dataset) to ensure the consistency of independent variables and thereby guarantee the comparability of the results. The quantitative evaluation results for each model are summarized in Table 4.

Table 4. The performance of the eight models.

Model	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]	Params (M)	GFLOPs
Mask R-CNN	82.9%	84.2%	81.6%	91.6%	75.8%	43.9	133.08
YOLOv5-seg	85.1%	88.4%	82.0%	88.4%	63.2%	1.88	6.9
YOLOv6-seg	88.3%	90.4%	86.4%	92.0%	70.9%	4.00	7
YOLOv7-seg	88.5%	89.5%	87.6%	92.3%	67.6%	37.87	142.6
YOLOv8-seg	87.5%	89.3%	85.8%	91.0%	67.1%	3.26	12.1
YOLOv9-seg	88.5%	90.4%	86.6%	93.1%	72.2%	27.36	144.2
YOLOv10-seg	87.5%	90.4%	84.8%	90.9%	66.0%	2.85	11.8
YOLOv11-seg	87.9%	89.4%	86.4%	90.8%	65.9%	2.84	10.4
YOLOv12-seg	87.2%	90.0%	84.6%	89.6%	65.3%	2.82	10.4

According to the data presented in Table 4, YOLOv9-seg demonstrates outstanding performance across various evaluation metrics, particularly in instance segmentation accuracy. It achieves the highest Segment mAP@0.5 score of 93.1%, outperforming Mask R-CNN (91.6%), YOLOv7 (92.3%), YOLOv6 (92.0%), and YOLOv8 (91.0%). Furthermore, its Segment mAP@[0.5:0.95] reaches 72.2%, also surpassing YOLOv6 (70.9%) and YOLOv7 (67.6%). In addition, YOLOv9-seg attains an F1 score of 88.5%, achieving a good balance between

high precision (90.4%) and high recall (86.6%). Although its parameter count (27.36M) and computational complexity (144.2 GFLOPs) are higher than those of lightweight models, YOLOv9-seg exhibits a significant advantage in accuracy, making it well-suited for applications that demand high-quality segmentation. As a result, it was selected as the preferred model for this task.

3.2.2. Qualitative Analysis of the Baseline Model

To better validate the performance of the baseline models, we conducted a qualitative analysis of night models (Mask R-CNN, YOLOv5-seg, YOLOv6-seg, YOLOv7-seg, YOLOv8-seg, YOLOv9-seg, YOLOv10-seg, YOLOv11-seg, and YOLOv12-seg). Using our custom dataset, which includes complex backgrounds (such as leaf and branch occlusions), the experimental results show that YOLOv9-seg exhibits better performance in terms of instance segmentation accuracy, segmentation details, and robustness under complex backgrounds. It is worth mentioning that to ensure the rigor of the experiment, all of the images used to evaluate model performance were not included in the model training.

Figure 6a–e presents the instance segmentation results of sugar apple in complex backgrounds, such as leaf and branch occlusion, across various models. Arrows in Figure 6 highlight regions with missed or incorrect detections. As shown in Figure 6, compared to the other seven baseline models, the YOLOv9-seg instance segmentation model exhibits the best performance. Figure 6a shows that YOLOv5-seg and YOLOv6-seg have errors in instance segmentation, misclassifying leaves as sugar apples, while YOLOv7-seg incorrectly detects branches as sugar apples and Mask R-CNN does not detect sugar apples. In contrast, YOLOv9-seg performs better in the instance segmentation of distant sugar apples, with higher confidence and overall superior performance compared to other models. Figure 6b,c, and d indicate that Mask R-CNN, YOLOv5-seg, YOLOv6-seg, YOLOv8-seg, YOLOv10-seg, YOLOv11-seg, and YOLOv12-seg all exhibit segmentation errors, such as misclassifying leaves as sugar apples or insufficient instance segmentation. In comparison, YOLOv9-seg effectively segments the sugar apples in cases of leaf and branch occlusion, demonstrating more stable and superior performance. Figure 6e shows that when handling the instance segmentation of edge sugar apples, YOLOv5-seg, YOLOv6-seg, YOLOv8-seg, YOLOv10-seg, and YOLOv12-seg fail to detect edge sugar apples, while YOLOv9-seg performs the best, successfully addressing the edge instance segmentation issue compared to other models.

3.3. Backbone Replacement Results

In this section, we replaced the backbone of the YOLOv9-seg instance segmentation model to explore the performance of different backbone networks on the orchard sugar apple instance segmentation task. The backbones evaluated include several popular architectures such as MobileNet, EfficientNet, Starnet, LCNNet, RepVit, FasterNet, and GhostNet [54–60]. To ensure the comparability of the experimental results, all experiments were conducted under consistent conditions using the same custom orchard sugar apple dataset for training and testing. The quantitative evaluation results for each backbone network are summarized in Table 5.

According to the data presented in Table 5, the selection of the original backbone is justified by its optimal performance in both segmentation accuracy and overall balance. Specifically, the original backbone achieved a Segment mAP@0.5 of 93.1% and a Segment mAP@[0.5:0.95] of 72.2%, both the highest among all compared models, outperforming other lightweight backbones such as GhostNetv1 (92.4%, 71.3%) and EfficientNet (92.3%, 71.9%). Additionally, the original backbone maintained a well-balanced performance across the F1 score (88.5%), precision (90.4%), and recall (86.6%), ensuring stable and

accurate results in practical detection tasks. Although its parameter count (27.36 M) and computational complexity (144.2 GFLOPs) are higher than some lightweight backbones, the original backbone, with its stronger feature extraction capability, provides a more reliable performance advantage, making it the preferred choice for tasks requiring higher segmentation quality.

Table 5. Backbone replacing results of YOLOv9-seg.

Backbone	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]	Params (M)	GFLOPs
Original	88.5%	90.4%	86.6%	93.1%	72.2%	27.36	144.2
MobileNetV2	88.6%	90.3%	86.9%	92.5%	71.2%	21.34	106.6
MobileNetV3	87.3%	90.2%	84.5%	90.7%	66.8%	19.83	102.4
MobileNetV4	86.9%	90.1%	84.0%	89.7%	65.3%	21.61	105.5
EfficientNet	88.6%	90.7%	86.6%	92.3%	71.9%	23.12	108.6
StarNet	86.7%	91.8%	82.1%	88.2%	63.8%	19.09	101.8
LCNetV2	85.9%	91.2%	81.1%	88.9%	64.9%	23.42	113
RepViT	89.1%	89.8%	88.4%	92.7%	70.4%	23.66	115.4
FasterNet	88.5%	89.8%	87.3%	92.5%	70.9%	31.88	133.4
GhostNetv1	89.4%	88.8%	90.0%	92.4%	71.3%	21.71	104.9



Figure 6. The results of the instance segmentation on the eight baseline models.

3.4. Image Enhancement Results

3.4.1. Quantitative Analysis of the Image Enhancement Experiment

In this section, we investigate several image enhancement algorithms, specifically exploring the performance of CLAHE, GC, HE, and ICM in the task of sugar apple instance segmentation under natural orchard environments. To maintain the original model's parameter count and computational complexity, all image enhancement operations were applied to the input images prior to model training. By employing different enhancement techniques, we analyzed their impact on the performance of the YOLOv9-seg model for sugar apple instance segmentation in natural orchard conditions. The quantitative evaluation results for the four enhancement algorithms are summarized in Table 6.

Table 6. Image augmentation replacing results of YOLOv9-seg.

Algorithm	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]
Original	88.5%	90.4%	86.6%	93.1%	72.2%
CLAHE	88.7%	90.2%	87.3%	93.3%	73.0%
GC	89.6%	90.3%	88.9%	93.1%	72.7%
HE	88.6%	88.7%	88.6%	93.1%	72.6%
ICM	89.4%	90.3%	88.6%	93.4%	72.9%

As shown in Table 6, the choice of Gamma Correction (GC) as the image enhancement algorithm is justified by its outstanding performance across multiple key metrics, particularly its ability to significantly improve overall detection performance while maintaining segmentation accuracy. Specifically, GC achieved the highest F1 score of 89.6% among all methods, indicating the best balance between precision and recall; its recall rate reached 88.9%, clearly surpassing that of the original images (86.6%) and CLAHE (87.3%). In addition, GC achieved a Segment mAP@0.5 of 93.1%, consistent with that of the original images, and a Segment mAP@[0.5:0.95] of 72.7%, which is higher than the original images (72.2%) and HE (72.6%), though slightly lower than CLAHE (73.0%) and ICM (72.9%). Nevertheless, GC demonstrated a significant advantage in recall and the F1 score. Therefore, GC offers notable improvements in model stability and target recognition capabilities, making it the optimal choice in terms of overall performance.

3.4.2. Qualitative Analysis of the Image Enhancement Experiment

To evaluate the performance of four image enhancement algorithms, we conducted a qualitative analysis of the CLAHE, GC, HE, and ICM image enhancement methods. The experiments were conducted using a self-constructed dataset, which includes complex background scenes such as leaf and branch occlusions. The analysis results show that, in terms of instance segmentation accuracy and detail preservation, the GC image enhancement algorithm demonstrated the best overall performance.

Figure 7a–g shows the instance segmentation results of sugar apples under complex backgrounds, such as leaf and branch occlusions. It can be observed that the GC image enhancement algorithm outperforms the other three enhancement methods in terms of overall performance. Specifically, Figure 7a–d focuses on evaluating the target detection capability in instance segmentation. Figure 7a,b shows that the CLAHE and HE enhancement algorithms exhibit over-detection and missed detection issues, where branches and leaves are incorrectly identified as sugar apples, or occluded targets fail to be detected. In contrast, the results in Figure 7c,d indicate that the GC enhancement algorithm performs better in detecting distant sugar apples, with more accurate target localization. Figure 7e–g focuses on the comparison of segmentation accuracy, covering typical perspective scenarios such as near-distance branch occlusion, a front view, and an upward view. In these cases, the performance of the four image enhancement algorithms is relatively similar, as all methods can effectively segment the contours of the sugar apples, indicating similar segmentation abilities in boundary detail extraction under specific perspectives and occlusion conditions.

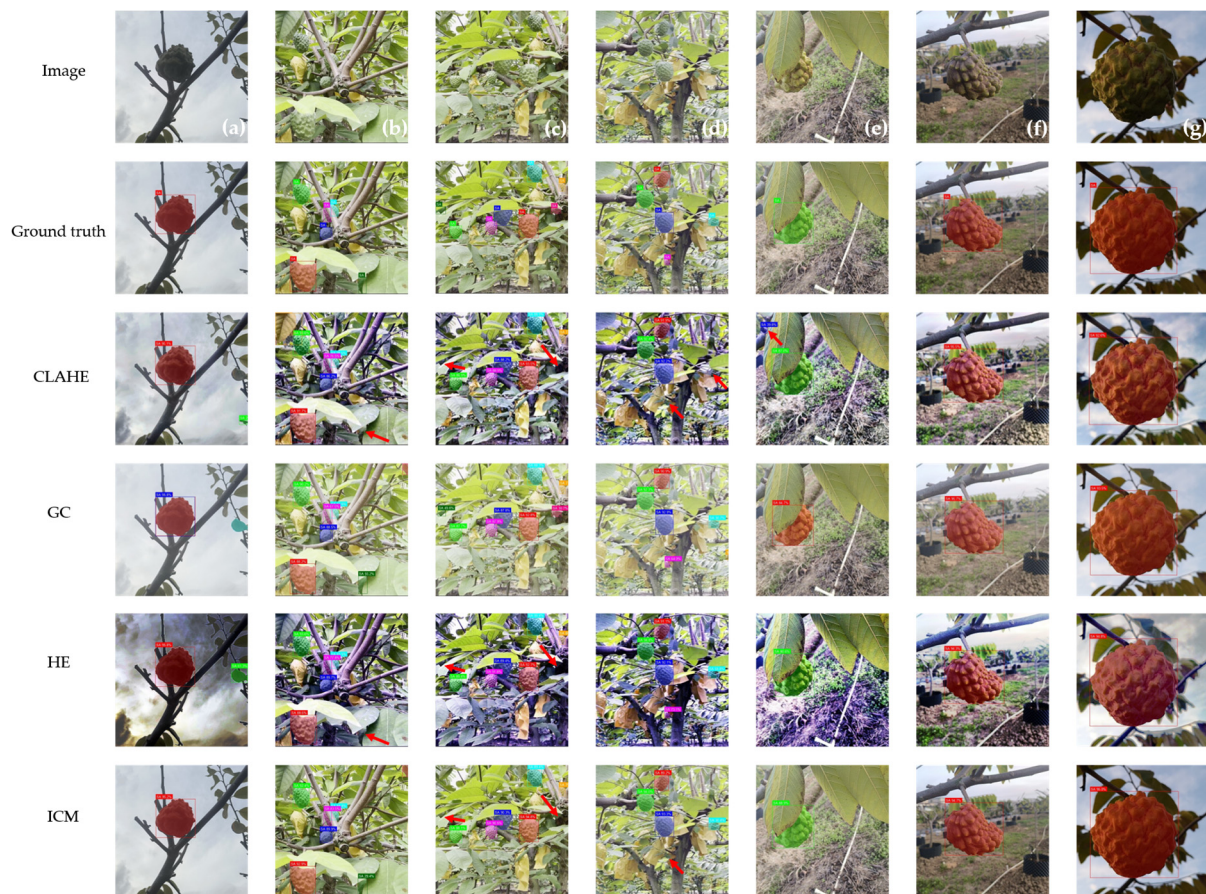


Figure 7. The results of instance segmentation on the four image enhancement algorithms.

3.5. YOLOv9-seg Before and After Improvement Results

3.5.1. Quantitative Analysis of the Improved YOLOv9-seg

In this section, we investigate the performance of the GCE-YOLOv9-seg and YOLOv9-seg models on the task of sugar apple instance segmentation in natural orchard environments. The quantitative evaluation results are summarized in Table 7.

Table 7. Results of YOLOv9-seg and GCE-YOLOv9-seg.

Model	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]	Params (M)	GFLOPs
YOLOv9-seg	88.5%	90.4%	86.6%	93.1%	72.2%	27.36	144.2
GCE-YOLOv9-seg	90.0%	90.4%	89.6%	93.4%	73.2%	27.95	162

According to the experimental results shown in Table 7, the proposed GCE-YOLOv9-seg model demonstrates significant performance improvements over the baseline YOLOv9-seg model in the sugar apple instance segmentation task within orchard environments. Specifically, for the object detection task, the F1 score and recall (R) reached 90.0% and 89.6%, respectively, representing improvements of 1.5% and 3.0% compared to the original model. In the segmentation task, mAP@0.5 and mAP@[0.5:0.95] achieved 93.4% and 73.2%, respectively, showing increases of 0.3% and 1.0%. Although the GCE-YOLOv9-seg model exhibited slightly higher parameter count (27.95 M) and computational complexity (162 GFLOPs) compared to YOLOv9-seg, its enhanced feature extraction capability provided superior performance, particularly in scenarios with higher demands for segmentation

quality. These results demonstrate that GCE-YOLOv9-seg outperforms YOLOv9-seg across multiple key metrics, validating its effectiveness in instance segmentation tasks.

3.5.2. Qualitative Analysis of the GCE-YOLOv9-seg

In order to better validate the performance of the proposed model, we conducted a qualitative analysis of the optimal baseline model, YOLOv9-seg, and the improved model, GCE-YOLOv9-seg. The experimental results indicate that the proposed GCE-YOLOv9-seg model outperforms the baseline model in terms of instance segmentation accuracy, segmentation detail, and robustness in real orchard environment backgrounds. It is worth noting that, to ensure the rigor of the experiment, all images used for evaluating the model's performance were not involved in the model's training process.

Figure 8a–g shows the instance segmentation results of sugar apples in a natural orchard environment using the YOLOv9-seg and GCE-YOLOv9-seg models. Compared to the baseline model, the proposed model (GCE-YOLOv9-seg) demonstrates the best performance in segmentation results. Figure 8a–d shows that in complex orchard environments (such as tree branch and leaf occlusion, and low-light conditions), YOLOv9-seg exhibits false detection, while the proposed model (GCE-YOLOv9-seg) maintains high accuracy and significantly outperforms the baseline model overall. Figure 8d–g reveals that the YOLOv9-seg model has obvious segmentation errors or under-segmentation issues when segmenting sugar apples (e.g., incorrectly segmenting grass as sugar apples). In contrast, the proposed model (GCE-YOLOv9-seg) can accurately segment sugar apples and shows significantly higher prediction accuracy.



Figure 8. The results of instance segmentation on YOLOv9-seg and GCE-YOLOv9-seg.

However, GCE-YOLOv9-seg and YOLOv9-seg exhibit certain limitations under low-light and grassy background conditions. As illustrated in Figure 9a–e, under low illumination and backlit environments, the insufficient lighting causes the features of sugar apples and leaves to appear similar, resulting in false positive detections (leaves misclassified as sugar apples) and false negatives for both models. Furthermore, in grassy background scenarios, both GCE-YOLOv9-seg and YOLOv9-seg demonstrate false positive detections (grass and branches misclassified as sugar apples) as well as false negatives.

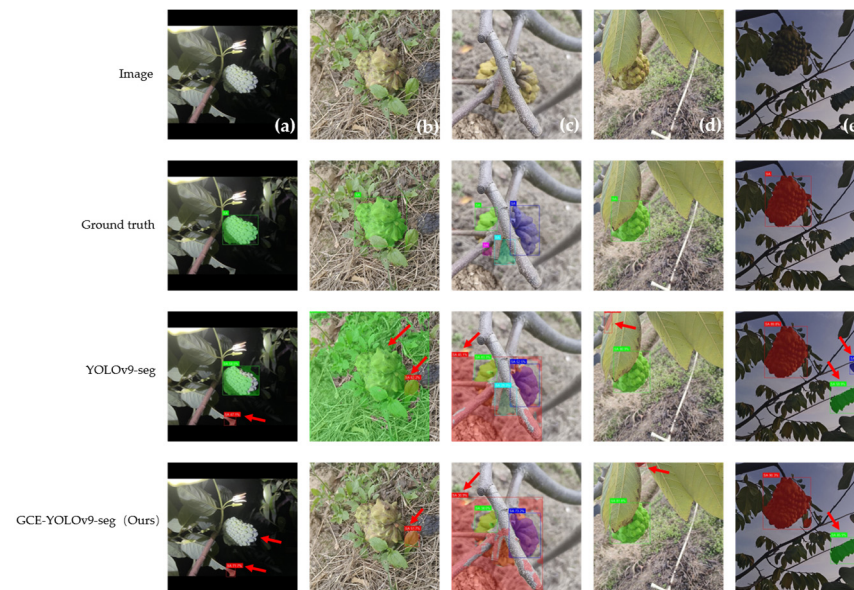


Figure 9. Limitations of instance segmentation using YOLOv9-seg and GCE-YOLOv9-seg.

3.6. Ablation Analysis

3.6.1. Ablation Analysis of Gamma Correction

In this subsection, Gamma Correction is described as a nonlinear operation used for encoding and decoding image brightness, where the Gamma (γ) value can be adjusted according to the characteristics of the dataset to optimize feature enhancement. The choice of γ significantly impacts the results [61]. To determine the optimal Gamma parameter for Gamma Correction and improve image enhancement effects [62], we designed an ablation study that tested different Gamma values ranging from 0.1 to 1.0. By comparing and analyzing the F1 score, precision (P), recall (R), as well as segmentation mAP@0.5 and mAP@[0.5:0.95] under each Gamma value, as shown in Table 8, we evaluated the impact of these parameters on model performance, thereby providing a basis for selecting the final parameter.

Table 8. Different Gamma values for the results of GCE-YOLOv9-seg.

Value of Gamma	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]
0.1	86.1%	88.0%	84.3%	90.5%	66.4%
0.2	89.0%	91.2%	86.9%	92.1%	70.3%
0.3	89.5%	91.4%	87.6%	92.1%	70.8%
0.4	89.0%	91.5%	86.7%	92.4%	71.1%
0.5	89.7%	91.0%	88.4%	93.0%	73.1%
0.6	89.4%	91.6%	87.3%	93.0%	72.8%
0.7	90.0%	90.4%	89.6%	93.4%	73.2%
0.8	89.4%	91.3%	87.5%	93.0%	73.0%
0.9	89.5%	89.4%	89.6%	93.1%	73.7%
1.0	89.8%	90.1%	89.5%	92.8%	72.7%

In the low Gamma range of 0.1–0.3, image brightness is insufficiently enhanced, leaving dark regions poorly illuminated. As a result, fruit boundaries and fine details remain obscured, making it difficult for the model to accurately extract features. According to the principles of Gamma Correction, this range offers limited dynamic range stretching, especially under conditions common in natural orchards, such as shade, backlighting, or darker fruit colors. These factors lead to weak contrast and frequent missed detections. Experimentally, with Gamma = 0.1, the F1 score is 86.1%, the precision (P) is 88.0%, the recall (R) rate is 84.3%, mAP@0.5 is 90.5%, and mAP@[0.5:0.95] is only 66.4%. While Gamma = 0.2

improves performance slightly ($F1 = 89.0\%$, $mAP@[0.5:0.95] = 70.3\%$), and $\Gamma = 0.3$ raises $F1$ to 89.5% , $mAP@[0.5:0.95]$ still stays under 71% . Overall, the performance in this range is relatively low, confirming that inadequate brightness hampers the model's feature extraction capabilities.

In the mid Γ range of $0.4\text{--}0.7$, image brightness is more reasonably enhanced, exposing shadowed details without causing overexposure in bright areas. Γ Correction in this range stretches mid- and low-brightness pixels effectively, improving contrast between fruits and background elements, which enhances model accuracy. In terms of results, $\Gamma = 0.4$ yields an $F1$ of 89.0% , a P of 91.5% , an R of 86.7% , and a $mAP@[0.5:0.95]$ of 71.1% . With $\Gamma = 0.5$, $F1$ increases to 89.7% , R to 88.4% , and a $mAP@[0.5:0.95]$ to 73.1% . $\Gamma = 0.6$ gives a slightly lower $F1$ of 89.4% and a $mAP@[0.5:0.95]$ of 72.8% . Notably, $\Gamma = 0.7$ achieves the highest $F1$ score of 90.0% in this group, with R at 89.6% , $mAP@0.5$ at 93.4% , and $mAP@[0.5:0.95]$ at 73.2% . This confirms that this range offers a balance between visibility and information preservation, leading to stable and robust performance across all metrics.

In the high Γ range of $0.8\text{--}1.0$, the brightness enhancement tends to be excessive. According to Γ Correction theory, this range compresses pixel values nonlinearly, causing bright areas to saturate and lose detail. Overexposed regions make it harder for the model to distinguish fruit boundaries from the background, especially under sunlight or high reflectance. Experimentally, $\Gamma = 0.8$ has an $F1$ of 89.4% , a P of 91.3% , an R of 87.5% , and a $mAP@[0.5:0.95]$ of 73.0% . $\Gamma = 0.9$ achieves the highest $mAP@[0.5:0.95]$ in the table (73.7%) but drops in $F1$ (89.5%) and P (89.4%). With $\Gamma = 1.0$, while $F1$ remains at 89.8% and R at 89.5% , $mAP@[0.5:0.95]$ decreases to 72.7% . These results indicate that this range leads to fluctuating and less reliable model performance, likely due to over-brightened, low-contrast features.

In conclusion, $\Gamma = 0.7$ stands out as the optimal choice across all intervals. From an enhancement perspective, $\Gamma = 0.7$ lies near the inflection point of the Γ curve, providing balanced brightening of dark areas without sacrificing highlight detail. This enhances the contrast between fruits, leaves, and the background, which significantly benefits feature extraction. Experimentally, $\Gamma = 0.7$ achieves top-tier results across all key metrics— $F1$ score (90.0%), recall (89.6%), $mAP@0.5$ (93.4%), and $mAP@[0.5:0.95]$ (73.2%)—either leading or very close to the best among all tested values. Compared with $\Gamma = 0.6$ and 0.8 , it improves $mAP@[0.5:0.95]$ by $0.4\text{--}0.5$ percentage points and delivers a higher $F1$. Against $\Gamma = 0.9$ and 1.0 , it provides better balance and consistency. These findings confirm that $\Gamma = 0.7$ is the most effective enhancement setting for preprocessing sugar apple images in natural orchard environments, offering a strong foundation for high-precision instance segmentation.

3.6.2. Ablation Analysis of the GCE-YOLOv9-seg

Table 9 presents a comprehensive ablation study to assess the contribution of GC image enhancement, CBAM attention, and EMA modules to the performance of GCE-YOLOv9-seg. Method (1), the baseline model, achieves an $F1$ score of 88.5% , a precision of 90.4% , a recall rate of 86.6% , a segmentation $mAP@0.5$ of 93.1% , and a $mAP@[0.5:0.95]$ of 72.2% . When GC is added in Method (2), the $F1$ score increases to 89.6% ($+1.1\%$), recall improves to 88.9% ($+2.3\%$), and segmentation $mAP@[0.5:0.95]$ slightly increases to 72.7% . This demonstrates that GC effectively improves feature contrast and enhances the model's ability to detect more objects, especially improving recall.

Table 9. The results of the ablation experiments.

Method	GC	CBAM	EMA	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]	Params (M)	Flops (G)
(1)				88.5%	90.4%	86.6%	93.1%	72.2%	27.36	144.2
(2)	✓			89.6%	90.3%	88.9%	93.1%	72.7%	27.36	144.2
(3)		✓		89.5%	91.8%	87.3%	93.6%	72.9%	27.79	146
(4)			✓	89.6%	90.1%	89.2%	93.8%	73.1%	27.74	161.7
(5)	✓	✓		89.8%	90.6%	89.1%	93.2%	71.7%	27.79	146
(6)	✓		✓	89.6%	89.9%	89.4%	93.1%	72.1%	27.74	161.7
(7)		✓	✓	89.4%	90.1%	88.8%	93.4%	72.9%	27.95	162
(8)	✓	✓	✓	90.0%	90.4%	89.6%	93.4%	73.2%	27.95	162

When only CBAM is introduced in Method (3), the model achieves the highest precision among all methods at 91.8% (+1.4% over baseline) and improves F1 to 89.5%, indicating CBAM's effectiveness in focusing attention on informative spatial and channel features. Meanwhile, the segmentation performance also improves to 72.9%. Method (4), which adds only EMA, delivers the highest recall (89.2%) and a segmentation mAP@[0.5:0.95] of 73.1% among single-module variants, confirming EMA's strength in multi-scale attention and stabilizing training. Comparing the three single-module methods, GC contributes the most to recall, the CBAM to precision, and EMA to segmentation quality and general detection robustness.

Among the two-module combinations, Method (5) (GC+CBAM) increases the F1 to 89.8% and precision to 90.6%, though segmentation mAP@[0.5:0.95] drops to 71.7%. Method (6) (CBAM+EMA) balances detection and segmentation, reaching an 89.4% F1, 90.1% precision, and 72.9% segmentation mAP. Method (7) (GC+EMA) achieves the highest recall (89.4%) among dual-module models and 72.1% segmentation mAP. Finally, Method (8), which combines all three modules, yields the best results across all metrics: an F1 of 90.0% (+1.5% over baseline), a precision of 90.4%, a recall rate of 89.6% (+3.0%), a segmentation mAP@0.5 of 93.4%, and a mAP@[0.5:0.95] of 73.2% (+1.0%). These results show that the three modules provide complementary improvements, and their joint use leads to the most balanced and comprehensive enhancement in detection and segmentation performance.

3.7. Sensitive Analysis

3.7.1. Sensitive Analysis of the Batch Size

This paper focuses on analyzing batch size, a key hyperparameter that significantly affects the training stability and performance of the GCE-YOLOv9-seg model. Because batch size impacts model convergence, feature learning, and both detection and segmentation accuracy, this study conducts a sensitivity analysis to identify the optimal batch size for achieving the best overall results.

The sensitivity analysis results, as shown in Table 10, reveal clear trends across batch sizes of 4, 8, 16, and 32. Starting with a batch size of 4, the model achieves an F1 score of 88.4%, a precision of 89.1%, a recall rate of 87.8%, and a segmentation mAP@[0.5:0.95] of 71.7%. Increasing the batch size to 8 and 16 steadily improves all metrics, with batch size 16 delivering the best balance: an F1 of 90.0%, a precision of 90.4%, a recall rate of 89.6%, and a segmentation mAP@[0.5:0.95] of 73.2%. When the batch size grows to 32, recall and segmentation mAP@[0.5:0.95] reach their highest points at 91.0% and 73.3%, respectively, yet the precision decreases to 88.8%, causing a slight drop in F1 to 89.9%. This indicates that while larger batch sizes enhance recall and segmentation consistency, they may reduce the model's ability to discriminate false positives. Consequently, batch size 16 is recommended as the optimal setting, offering the best trade-off between detection accuracy, segmentation quality, and training stability.

Table 10. The results of the sensitive analysis of the batch size.

Value of Batch Size	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]
4	88.4%	89.1%	87.8%	92.7%	71.7%
8	89.5%	90.2%	88.9%	93.2%	72.5%
16	90.0%	90.4%	89.6%	93.4%	73.2%
32	89.9%	88.8%	91.0%	93.2%	73.3%

3.7.2. Sensitive Analysis of the Initial Learning Rate

The initial learning rate is a critical hyperparameter in training the GCE-YOLOv9-seg model, directly affecting the convergence speed and performance. A learning rate that is too low can lead to slow training and the model getting stuck in local optima, while a rate that is too high may cause unstable training and even performance degradation. Sensitivity analysis of the initial learning rate helps determine the optimal parameter setting, improving the model's generalization ability and accuracy.

Table 11 shows significant differences in the model's F1 score, precision (P), recall (R), and instance segmentation mAP metrics under different initial learning rates. The overall trend indicates that as the learning rate increases from 0.002 to 0.01, all metrics improve significantly. The F1 score rises from 86.9% to 90.0%, the precision and recall reach 90.4% and 89.6% respectively, and the instance segmentation mAP@0.5 and mAP@[0.5:0.95] improve to 93.4% and 73.2%, demonstrating the best detection and segmentation performance. However, when the learning rate further increases to 0.02, although the Segment mAP@0.5 slightly increases to 93.5%, the F1 score and recall decline, indicating performance fluctuations. At a learning rate of 0.05, precision reaches its highest value of 91.3%, but the F1 score and recall drop to 89.0% and 86.9%, and the mAP metrics decline significantly, suggesting that an excessively high learning rate causes unstable training and reduced generalization. In summary, an initial learning rate of 0.01 enables the GCE-YOLOv9 model to achieve a balanced and optimal performance in terms of accuracy, recall, and instance segmentation, making it the most suitable choice in this study.

Table 11. The results of the sensitive analysis of the initial learning rate.

Value of Initial Learning Rate	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]
0.002	86.9%	88.3%	85.5%	90.2%	63.8%
0.005	87.8%	88.2%	87.5%	91.9%	70.0%
0.01	90.0%	90.4%	89.6%	93.4%	73.2%
0.02	89.6%	90.6%	88.6%	93.5%	73.6%
0.05	89.0%	91.3%	86.9%	92.0%	70.2%

3.8. Generalizability Analysis

To validate the generalization capability of the proposed model in this study, an open-source dataset was selected for testing. The dataset, sourced from the Baidu Paddle platform, contains 758 images across five fruit categories: the Braeburn apple, the Golden Delicious apple, the Topaz apple, Peach, and Pear. The dataset is available at <https://aistudio.baidu.com/datasetdetail/114414> (accessed on 29 May 2025).

Table 12 shows that the GCE-YOLOv9-seg model achieves significant improvements over the YOLOv9-seg model across all metrics: an increase of 2.5% in the F1 score (F1), 2.7% in precision (P), 2.2% in recall (R), 0.8% in segmentation mAP@0.5, and 1.0% in mAP@[0.5:0.95]. The results demonstrate that the proposed GCE-YOLOv9-seg outperforms the baseline YOLOv9-seg model, reflecting its strong generalization capability and stability.

Table 12. The results of GCE-YOLOv9-seg and YOLOv9-seg on an open-source dataset.

Model	F1	P	R	Segment mAP@0.5	Segment mAP@[0.5:0.95]	Params (M)	GFLOPs
YOLOv9-seg	95.1%	93.1%	97.2%	98.7%	98.2%	27.36	144.2
GCE-YOLOv9-seg	97.6%	95.8%	99.4%	99.5%	99.2%	27.95	162

4. Discussion

The accurate and rapid perception and identification of sugar apples in orchard environments is a challenging task due to their external characteristics resembling those of leaves and being susceptible to occlusion by leaves and branches. Additionally, their surfaces are easily damaged by impacts, necessitating a non-destructive detection approach. Existing studies have shown that computer vision-based methods perform well in various tasks related to sugar apple detection (e.g., using an improved DeepLabv3+ model for detecting sugar apple maturity, achieving a mean Intersection over Union (MIoU) of 89.95% and a mean Pixel Accuracy (MPA) of 94.58% [25], employing YOLO algorithms for sugar apple detection with an accuracy of 86.84% [26], using an improved CNN approach to classify sugar apple fruit diseases with an accuracy of 99.15% [28], and designing a CNN-based model for sugar apple leaf disease classification, achieving an accuracy of 78.3% [29]). However, current computer vision methods for sugar apple tasks either lack a systematic response to environmental interferences in complex orchard settings, or provide inference results with relatively low levels of information representation (e.g., methods based on image classification), making it difficult to achieve accurate recognition and segmentation of sugar apple fruits under complex conditions.

Furthermore, most current computer vision-based techniques for sugar apple tasks have not incorporated instance segmentation. However, compared to image classification, semantic segmentation, and object detection, instance segmentation offers better individual distinction and more detailed contour extraction. While semantic segmentation classifies every pixel in the image, it cannot distinguish between individual sugar apples of the same class. Object detection can identify the general location of the object, but does not provide precise edge information. Instance segmentation, on the other hand, retains pixel-level classification accuracy and can distinguish the specific contours of each individual sugar apple, making it especially suitable for scenarios involving dense fruit, significant overlap, or substantial individual shape variation. Therefore, introducing instance segmentation not only helps to improve the accuracy of sugar apple detection and counting but also provides higher-quality visual information for subsequent processing tasks like surface quality detection and size estimation.

In the field of instance segmentation, YOLO-based algorithms feature end-to-end high-speed inference and robustness, enabling the efficient and accurate detection and segmentation of sugar apples in complex orchard environments. They are suitable for real-time fruit monitoring and automated harvesting. Moreover, this study, addressing the challenges posed by the aforementioned interference in natural orchard environments, introduces a custom sugar apple instance segmentation dataset from an orchard environment. In natural orchard settings, lighting conditions are often uneven, especially in areas with leaf occlusion and backlighting, where the fruit's edges and texture details are obscured, reducing detection and segmentation accuracy.

In traditional sugar apple detection in natural environments, YOLOv9-seg has shown significant advantages compared to other YOLO series models [63]. By introducing the DualConv convolution layer, SENetV2, and the improvement of the EIou loss function, YOLOv9-seg achieved better results in banana ripeness detection in natural environments [64], especially in terms of error segmentation and precision, with outstanding

performance in detecting objects under occlusion. Based on our experimental results, YOLOv9-seg demonstrated the best performance in the sugar apple instance segmentation task in natural orchard environments compared to other commonly used networks. However, the experimental results indicate that, under the influence of objects with similar features, there are still some deficiencies, and further optimization is required.

This study aimed to further improve the instance segmentation performance based on the YOLOv9 model. Existing studies have shown that integrating pixel-level Gamma Correction (GC) with deep learning methods can effectively enhance details in dark regions, improve image contrast, and significantly boost the performance of downstream tasks [65–67]. The fundamental mechanism lies in dynamically adjusting the image's luminance distribution, reducing information loss, and thereby optimizing the input quality for feature extraction. Previous research also indicates that traditional Histogram Equalization (HE) methods fail to account for the local features of the image [68], and gray-level transformation methods may cause blurring and the loss of grayscale information under low-light conditions [69]. Based on these findings, this study introduces the GC module at the input layer, enabling the model to better adapt to natural low-light environments, highlighting weak feature regions, enhancing edge feature representation, and ultimately improving the overall performance of sugar apple instance segmentation.

To further enhance the model's ability to perceive key information in complex natural environments, previous research has shown that the Convolutional Block Attention Module (CBAM) can maintain high detection accuracy and robustness under conditions such as dense target distributions [70–72]. Traditional channel attention mechanisms often show insufficient perception and localization bias under conditions like partial occlusion [73]. In contrast, CBAM explicitly models spatial and channel attention, guiding the model to focus on key regions and features, thereby significantly improving feature extraction effectiveness. Therefore, this study introduces the CBAM module at key points after the backbone feature fusion to further enhance the model's ability to perceive weak feature regions and object edges, improving detection performance and stability in the sugar apple instance segmentation task.

Furthermore, to strengthen the model's multi-scale feature perception and representation, prior research has shown that integrating the Efficient Multiscale Attention (EMA) module into lightweight network structures can significantly improve feature extraction and detection performance. Combining EMA with spatial attention further achieves comprehensive feature integration, effectively improving object removal detection accuracy [74,75]. Inspired by these findings, this study introduces the EMA module after the fusion of backbone and upsampled features, dynamically modeling spatial relationships across different scales. This enhances the model's ability to distinguish small targets from complex backgrounds and results in significant performance improvements in the sugar apple instance segmentation task. Ablation experiments further validated the effectiveness of the proposed module designs in enhancing model performance.

The proposed method holds promise as an effective solution for the intelligent perception of sugar apples in natural orchard environments, potentially reducing labor costs, increasing yield, and promoting the development of precision agriculture. However, its robustness and applicability still require validation across a broader range of real-world orchard settings. The dataset constructed in this study was collected using only three smartphone models with relatively uniform resolution, which may introduce device-dependent biases and limit the model's generalization capability. In future work, the research will be extended to multi-class tasks by incorporating various sugar apple diseases and pests, such as anthracnose, blank canker, diplodia rot, leaf spots on fruit, leaf spots on leaves, and mealybugs.

Extensive research has been conducted on the multi-class classification and detection of citrus diseases such as canker and black spot [76], as well as apple leaf diseases, including apple scab and rot [77]. Meanwhile, other studies have focused on the single-class detection of apples [78] and pineapples [79] in orchard environments. Since sugar apples typically exist as a single class in natural orchard environments and samples of different disease categories are scarce, collecting multi-class data poses significant challenges. Therefore, this study mainly focuses on single-class sugar apple detection in natural orchard settings, covering complex conditions including direct light, backlight, and occlusion by branches and leaves.

Furthermore, although ablation experiments demonstrated that a fixed Gamma Correction parameter of 0.7 yields optimal results in the Gamma Correction (GC) image enhancement module, this parameter lacks a theoretical basis. Existing studies emphasize the need to balance accuracy and computational efficiency when integrating attention modules into object detection frameworks, Anandakrishnan et al. [80] use depth-wise convolution modules to reduce the number of parameters. Future research will explore adaptive Gamma Correction or dynamic parameter tuning to improve performance under varying illumination conditions. Meanwhile, the introduction of CBAM and EMA modules significantly enhances the model's ability to perceive critical features but also increases computational complexity and model burden. Future efforts should focus on designing lightweight attention mechanisms that effectively integrate CBAM and EMA into the YOLOv9-seg architecture, balancing detection accuracy with computational efficiency to meet the demands of embedded devices and real-time applications. Additionally, developing more specialized loss functions targeting occlusion and lighting variations unique to agricultural object detection is essential to improve boundary accuracy under complex conditions.

The current loss function, while effective for general object detection tasks, has not been optimized for the impact of occlusions and lighting variations in agricultural object detection. In agricultural scenarios, crop occlusions and lighting changes can significantly affect detection accuracy, so the current loss function may not achieve optimal results under these specific conditions. Although we used the standard loss function, no modifications were made to the original YOLOv9-seg loss function. Future research could consider introducing specialized loss functions that account for factors such as occlusion and lighting variations to enhance the model's adaptability in complex agricultural environments.

Currently, precision agriculture primarily focuses on key areas such as fruit counting [81], pest and disease detection [82], and automated harvesting [83]; thus, the research outcomes hold significant practical implications for agricultural applications. The improved YOLOv9-seg model demonstrates excellent performance in sugar apple detection and instance segmentation tasks, potentially facilitating the development of automated harvesting technologies, enhancing yield estimation accuracy, improving picking efficiency, and reducing labor costs, thereby advancing precision agriculture. At the same time, it is important to acknowledge the limitations of the study, including dataset biases and limited model generalization. Future work should expand the scope of data collection and strengthen validation and optimization under diverse environmental conditions to ensure stable deployment in actual agricultural production.

5. Conclusions

This paper delves into the feasibility of using deep learning-based computer vision methods for the intelligent perception of sugar apples in natural orchard environments. The paper first explored nine mainstream instance segmentation models (including Mask R-CNN and YOLOv5-seg through YOLOv12-seg) and found that YOLOv9-seg performed the

best for this task. To further enhance the instance segmentation performance of YOLOv9-seg, an improved model, GCE-YOLOv9-seg, was proposed. By introducing Gamma Correction image enhancement, the model's performance was significantly improved. Additionally, the performance was further enhanced by incorporating Efficient Multiscale Attention (EMA) and Convolutional Block Attention Module (CBAM).

The experimental results on a self-made dataset showed that GCE-YOLOv9-seg achieved an F1 score (F1) of 90.0%, a precision (P) of 90.4%, and a recall (R) rate of 89.6% for object detection tasks. For instance segmentation tasks, mAP@0.5 and mAP@[0.5:0.95] reached 93.4% and 73.2%, respectively. Compared to the original YOLOv9-seg model, GCE-YOLOv9-seg showed a 1.5% improvement in the F1 score, a 3.0% increase in recall for object detection tasks, and a 0.3% increase in mAP@0.5 and a 1.0% increase in mAP@[0.5:0.95] for segmentation tasks. Additionally, ablation experiments demonstrated that the optimal Gamma Correction parameter for image enhancement was 0.7, and validated that the modules introduced had a positive impact on the overall performance improvement. The method proposed in this paper provides a potential and efficient solution for the intelligent perception of sugar apples in precision agriculture.

Author Contributions: Conceptualization, G.Z., Z.L. and Z.X.; Methodology, G.Z., Z.L. and Z.X.; Validation, G.Z. and Z.L.; Investigation, G.Z., Z.L., M.Y., J.J. and X.L.; Resources, W.W.; Data curation, G.Z., Z.L., H.H., Z.K. and X.L.; Writing—original draft preparation, G.Z., Z.L., M.Y., Z.X. and Y.W.; Writing—review and editing, G.Z., Z.X. and W.W.; Visualization, G.Z., Z.L. and X.L.; Supervision, W.W.; Project administration, W.W., G.Z., Z.L. and Z.X.; Funding acquisition, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 82001983, 52275097); the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2019A1515111202); the Science and Technology Program of Guangzhou, China (No. 202002030269); the Tertiary Education Scientific research project of Guangzhou Municipal Education Bureau (No.2024312250); and the Student Innovation Training Program of Guangzhou University (No. 202411078050).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The dataset used in this research is available upon reasonable request from the corresponding author. This dataset is not available to the public because of the laboratory privacy concerns.

Acknowledgments: The author sincerely thanks Chen Yongneng and other staff members of the Wan'an Sugar Apple Orchard in Nansha District, Guangzhou, Guangdong Province, for their support of this work.

Conflicts of Interest: There are no conflicts of interest for this article.

References

1. Shehata, M.G.; Abu-Serie, M.M.; Abd El-Aziz, N.M.; El-Sohaimy, S.A. Nutritional, phytochemical, and in vitro anticancer potential of sugar apple (*Annona squamosa*) fruits. *Sci. Rep.* **2021**, *11*, 6224. [[CrossRef](#)] [[PubMed](#)]
2. Van Damme, P.; Scheldeman, X. Commercial development of cherimoya (*Annona cherimola* Mill.) in Latin America. In Proceedings of the First International Symposium on Cherimoya 497, Loja, Ecuador, 16–19 March 1999; pp. 17–42.
3. Pargi Sanjay, J.; Gupta, P.; Balas, P.; Bambhaniya, V. Comparison between manual harvesting and mechanical harvesting. *J. Sci. Res. Rep.* **2024**, *30*, 917–934.
4. Anjom, F.K.; Vougioukas, S.G.; Slaughter, D.C. Development of a linear mixed model to predict the picking time in strawberry harvesting processes. *Biosyst. Eng.* **2018**, *166*, 76–89. [[CrossRef](#)]
5. Gallardo, R.K.; Galinato, S.P. *2019 Cost Estimates of Establishing, Producing, and Packing Honeycrisp Apples in Washington*; Washington State University Extension: Washington, DC, USA, 2020.

6. Martinez-Romero, D.; Serrano, M.; Carbonell, A.; Castillo, S.; Riquelme, F.; Valero, D. Mechanical damage during fruit post-harvest handling: Technical and physiological implications. In *Production Practices and Quality Assessment of Food Crops: Quality Handling and Evaluation*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 233–252.
7. Hall, R.E. The process of inflation in the labor market. *Brook. Pap. Econ. Act.* **1974**, *1974*, 343–393. [[CrossRef](#)]
8. King, R.G.; Watson, M.W. Inflation and unit labor cost. *J. Money Credit. Bank.* **2012**, *44*, 111–149. [[CrossRef](#)]
9. Xiao, F.; Wang, H.; Li, Y.; Cao, Y.; Lv, X.; Xu, G. Object detection and recognition techniques based on digital image processing and traditional machine learning for fruit and vegetable harvesting robots: An overview and review. *Agronomy* **2023**, *13*, 639. [[CrossRef](#)]
10. Arivazhagan, S.; Shebiah, R.N.; Nidhyandhan, S.S.; Ganesan, L. Fruit recognition using color and texture features. *J. Emerg. Trends Comput. Inf. Sci.* **2010**, *1*, 90–94.
11. Tan, S.H.; Lam, C.K.; Kamarudin, K.; Ismail, A.H.; Rahim, N.A.; Azmi, M.S.M.; Yahya, W.M.N.W.; Sneah, G.K.; Seng, M.L.; Hai, T.P. Vision-based edge detection system for fruit recognition. *J. Phys. Conf. Ser.* **2021**, *2107*, 012066. [[CrossRef](#)]
12. Jana, S.; Parekh, R. Shape-based fruit recognition and classification. In Proceedings of the International Conference on Computational Intelligence, Communications, and Business Analytics, Kolkata, India, 24–25 March 2017; pp. 184–196.
13. Moreda, G.; Muñoz, M.; Ruiz-Altisent, M.; Perdigones, A. Shape determination of horticultural produce using two-dimensional computer vision—A review. *J. Food Eng.* **2012**, *108*, 245–261. [[CrossRef](#)]
14. Mehl, P.; Chao, K.; Kim, M.; Chen, Y. Detection of defects on selected apple cultivars using hyperspectral and multispectral image analysis. *Appl. Eng. Agric.* **2002**, *18*, 219.
15. Mehl, P.M.; Chen, Y.-R.; Kim, M.S.; Chan, D.E. Development of hyperspectral imaging technique for the detection of apple surface defects and contaminations. *J. Food Eng.* **2004**, *61*, 67–81. [[CrossRef](#)]
16. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. Deepfruits: A fruit detection system using deep neural networks. *Sensors* **2016**, *16*, 1222. [[CrossRef](#)] [[PubMed](#)]
17. Bargoti, S.; Underwood, J. Deep fruit detection in orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3626–3633.
18. Rahnemoonfar, M.; Sheppard, C. Real-time yield estimation based on deep learning. In Proceedings of the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping II, Anaheim, CA, USA, 10–11 April 2017; pp. 59–65.
19. Fawzia Rahim, U.; Mineno, H. Highly accurate tomato maturity recognition: Combining deep instance segmentation, data synthesis and color analysis. In Proceedings of the 2021 4th Artificial Intelligence and Cloud Computing Conference, Kyoto, Japan, 17–19 December 2021; pp. 16–23.
20. Wang, D.; He, D. Fusion of Mask RCNN and attention mechanism for instance segmentation of apples under complex background. *Comput. Electron. Agric.* **2022**, *196*, 106864. [[CrossRef](#)]
21. Yang, L.; Wei, Y.Z.; He, Y.; Sun, W.; Huang, Z.; Huang, H.; Fan, H. ishape: A first step towards irregular shape instance segmentation. *arXiv* **2021**, arXiv:2109.15068.
22. Yaseen, M. What is yolov9: An in-depth exploration of the internal features of the next-generation object detector. *arXiv* **2024**, arXiv:2409.07813.
23. Sapkota, R.; Ahmed, D.; Karkee, M. Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. *Artif. Intell. Agric.* **2024**, *13*, 84–99. [[CrossRef](#)]
24. Lu, D.; Wang, Y. MAR-YOLOv9: A multi-dataset object detection method for agricultural fields based on YOLOv9. *PLoS ONE* **2024**, *19*, e0307643. [[CrossRef](#)]
25. Xie, Z.; Ke, Z.; Chen, K.; Wang, Y.; Tang, Y.; Wang, W. A Lightweight Deep Learning Semantic Segmentation Model for Optical-Image-Based Post-Harvest Fruit Ripeness Analysis of Sugar Apples (*Annona squamosa*). *Agriculture* **2024**, *14*, 591. [[CrossRef](#)]
26. Sanchez, R.B.; Esteves, J.A.C.; Linsangan, N.B. Determination of sugar apple ripeness via image processing using convolutional neural network. In Proceedings of the 2023 15th International Conference on Computer and Automation Engineering (ICCAE), Sydney, Australia, 3–5 March 2023; pp. 333–337.
27. Thite, S.; Patil, K.; Jadhav, R.; Suryawanshi, Y.; Chumchu, P. Empowering agricultural research: A comprehensive custard apple (*Annona squamosa*) disease dataset for precise detection. *Data Brief* **2024**, *53*, 110078. [[CrossRef](#)]
28. Tonmoy, M.R.; Adnan, M.A.; Al Masud, S.M.R.; Safran, M.; Alfarhood, S.; Shin, J.; Mridha, M. Attention mechanism-based ultralightweight deep learning method for automated multi-fruit disease recognition system. *Agron. J.* **2025**, *117*, e70035. [[CrossRef](#)]
29. Gaikwad, S.S.; Rumma, S.S.; Hangarge, M. Classification of Fungi Infected Annona Squamosa Plant Using CNN Architectures. In Proceedings of the International Conference on Soft Computing and Pattern Recognition, Online, 15–17 December 2021; pp. 170–177.
30. Dong, Z.; Wang, J.; Sun, P.; Ran, W.; Li, Y. Mango variety classification based on convolutional neural network with attention mechanism and near-infrared spectroscopy. *J. Food Meas. Charact.* **2024**, *18*, 2237–2247. [[CrossRef](#)]

31. Jroni, Z.; Moussaid, A.; Hadi, M.Y. Exploring End-to-End object detection with transformers versus YOLOv8 for enhanced citrus fruit detection within trees. *Syst. Soft Comput.* **2024**, *6*, 200103. [\[CrossRef\]](#)
32. Xiao, F.; Wang, H.; Xu, Y.; Zhang, R. Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review. *Agronomy* **2023**, *13*, 1625. [\[CrossRef\]](#)
33. Li, L.; Chan, P.; Deng, T.; Yang, H.-L.; Luo, H.-Y.; Xia, D.; He, Y.-Q. Review of advances in urban climate study in the Guangdong-Hong Kong-Macau greater bay area, China. *Atmos. Res.* **2021**, *261*, 105759. [\[CrossRef\]](#)
34. Patrício, D.I.; Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Comput. Electron. Agric.* **2018**, *153*, 69–81. [\[CrossRef\]](#)
35. Zhuang, Y.; Chen, W.; Jin, T.; Chen, B.; Zhang, H.; Zhang, W. A review of computer vision-based structural deformation monitoring in field environments. *Sensors* **2022**, *22*, 3789. [\[CrossRef\]](#)
36. Metaxas, D.N. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 389.
37. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
38. Qi, Y.; Yang, Z.; Sun, W.; Lou, M.; Lian, J.; Zhao, W.; Deng, X.; Ma, Y. A comprehensive overview of image enhancement techniques. *Arch. Comput. Methods Eng.* **2021**, *29*, 583–607. [\[CrossRef\]](#)
39. Reza, A.M. Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement. *J. VLSI Signal Process. Syst. Signal Image Video Technol.* **2004**, *38*, 35–44. [\[CrossRef\]](#)
40. Rahman, S.; Rahman, M.M.; Abdullah-Al-Wadud, M.; Al-Quaderi, G.D.; Shoyaib, M. An adaptive gamma correction for image enhancement. *EURASIP J. Image Video Process.* **2016**, *2016*, 35. [\[CrossRef\]](#)
41. Pizer, S.M.; Amburn, E.P.; Austin, J.D.; Cromartie, R.; Geselowitz, A.; Greer, T.; ter Haar Romeny, B.; Zimmerman, J.B.; Zuiderveld, K. Adaptive histogram equalization and its variations. *Comput. Vis. Graph. Image Process.* **1987**, *39*, 355–368. [\[CrossRef\]](#)
42. Iqbal, K.; Salam, R.A.; Osman, A.; Talib, A.Z. Underwater image enhancement using an integrated colour model. *IAENG Int. J. Comput. Sci.* **2007**, *34*, 239–244.
43. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
44. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–9 June 2023; pp. 1–5.
45. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
46. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Poznanski, J.; Yu, L.; Rai, P.; Ferriday, R. *ultralytics/yolov5, v3.0*; Zenodo: Geneva, Switzerland, 2020.
47. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
48. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. 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, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
49. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-time flying object detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
50. Wang, C.-Y.; Yeh, I.-H.; Mark Liao, H.-Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; pp. 1–21.
51. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J. Yolov10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
52. Khanam, R.; Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725.
53. Tian, Y.; Ye, Q.; Doermann, D. Yolov12: Attention-centric real-time object detectors. *arXiv* **2025**, arXiv:2502.12524.
54. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
55. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
56. Gan, Y.Q.; Zhang, H.; Liu, W.H.; Ma, J.M.; Luo, Y.M.; Pan, Y.S. Local-global feature fusion network for hyperspectral image classification. *Int. J. Remote Sens.* **2024**, *45*, 8548–8575. [\[CrossRef\]](#)
57. Cui, C.; Gao, T.; Wei, S.; Du, Y.; Guo, R.; Dong, S.; Lu, B.; Zhou, Y.; Lv, X.; Liu, Q. PP-LCNet: A lightweight CPU convolutional neural network. *arXiv* **2021**, arXiv:2109.15099.

58. Wang, A.; Chen, H.; Lin, Z.; Han, J.; Ding, G. Repvit: Revisiting mobile cnn from vit perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 15909–15920.
59. Chen, J.; Kao, S.-h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
60. 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, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
61. Liao, J.; Wang, Y.; Zhu, D.; Zou, Y.; Zhang, S.; Zhou, H. Automatic segmentation of crop/background based on luminance partition correction and adaptive threshold. *IEEE Access* **2020**, *8*, 202611–202622. [\[CrossRef\]](#)
62. Zilvan, V.; Ramdan, A.; Supianto, A.A.; Heryana, A.; Arisal, A.; Yuliani, A.R.; Krisnandi, D.; Suryawati, E.; Kusumo, R.B.S.; Yuawana, R.S. Automatic detection of crop diseases using gamma transformation for feature learning with a deep convolutional autoencoder. *J. Teknol. Dan Sist. Komput.* **2024**, *10*.
63. Sharma, A.; Kumar, V.; Longchamps, L. Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. *Smart Agric. Technol.* **2024**, *9*, 100648. [\[CrossRef\]](#)
64. Wang, G.; Gao, Y.; Xu, F.; Sang, W.; Han, Y.; Liu, Q. A Banana Ripeness Detection Model Based on Improved YOLOv9c Multifactor Complex Scenarios. *Symmetry* **2025**, *17*, 231. [\[CrossRef\]](#)
65. Li, X.; Liu, M.; Ling, Q. Pixel-wise gamma correction mapping for low-light image enhancement. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *34*, 681–694. [\[CrossRef\]](#)
66. Weng, S.-E.; Miaou, S.-G.; Christanto, R.; Hsu, C.-P. Exposure Correction in Driving Scenes Using the Atmospheric Scattering Model. In Proceedings of the 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan), Taichung, Taiwan, 9–11 July 2024; pp. 493–494.
67. Tommandru, S.; Sandanam, D. Low-illumination image contrast enhancement using adaptive gamma correction and deep learning model for person identification and verification. *J. Electron. Imaging* **2023**, *32*, 053018. [\[CrossRef\]](#)
68. Li, C.; Zhu, J.; Bi, L.; Zhang, W.; Liu, Y. A low-light image enhancement method with brightness balance and detail preservation. *PLoS ONE* **2022**, *17*, e0262478. [\[CrossRef\]](#)
69. Guo, J.; Ma, J.; García-Fernández, Á.F.; Zhang, Y.; Liang, H. A survey on image enhancement for Low-light images. *Heliyon* **2023**, *9*, e14558. [\[CrossRef\]](#)
70. Liu, T.; Yuan, Y.; Teng, G.; Meng, X. Improved Deep Convolutional Neural Network-Based Method for Detecting Winter Jujube Fruit in Orchards. *Eng. Lett.* **2024**, *32*, 569–578.
71. De Moraes, J.L.; de Oliveira Neto, J.; Badue, C.; Oliveira-Santos, T.; de Souza, A.F. Yolo-Papaya: A papaya fruit disease detector and classifier using CNNs and convolutional block attention modules. *Electronics* **2023**, *12*, 2202. [\[CrossRef\]](#)
72. Tang, R.; Lei, Y.; Luo, B.; Zhang, J.; Mu, J. YOLOv7-Plum: Advancing plum fruit detection in natural environments with deep learning. *Plants* **2023**, *12*, 2883. [\[CrossRef\]](#)
73. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
74. Chen, J.; Ma, A.; Huang, L.; Li, H.; Zhang, H.; Huang, Y.; Zhu, T. Efficient and lightweight grape and picking point synchronous detection model based on key point detection. *Comput. Electron. Agric.* **2024**, *217*, 108612. [\[CrossRef\]](#)
75. Liu, W.; Wang, S.; Gao, X.; Yang, H. A Tomato Recognition and Rapid Sorting System Based on Improved YOLOv10. *Machines* **2024**, *12*, 689. [\[CrossRef\]](#)
76. Negi, A.; Kumar, K. Classification and detection of citrus diseases using deep learning. In *Data Science and Its Applications*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2021; pp. 63–85.
77. Sangeetha, K.; Rima, P.; Pranesh Kumar, M.; Preethees, S. Apple leaf disease detection using deep learning. In Proceedings of the 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 29–31 March 2022; pp. 1063–1067.
78. Yue, Y.; Cui, S.; Shan, W. Apple detection in complex environment based on improved YOLOv8n. *Eng. Res. Express* **2024**, *6*, 045259. [\[CrossRef\]](#)
79. Zhang, R.; Huang, Z.; Zhang, Y.; Xue, Z.; Li, X. Msgv-yolov7: A lightweight pineapple detection method. *Agriculture* **2023**, *14*, 29. [\[CrossRef\]](#)
80. Anandkrishnan, J.; Sangaiah, A.K.; Darmawan, H.; Son, N.K.; Lin, Y.-B.; Alenazi, M.J. Precise Spatial Prediction of Rice Seedlings From Large Scale Airborne Remote Sensing Data Using Optimized Li-YOLOv9. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *18*, 2226–2238. [\[CrossRef\]](#)
81. Gao, F.; Fang, W.; Sun, X.; Wu, Z.; Zhao, G.; Li, G.; Li, R.; Fu, L.; Zhang, Q. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. *Comput. Electron. Agric.* **2022**, *197*, 107000. [\[CrossRef\]](#)

82. Chen, Y.; Pan, J.; Wu, Q. Apple leaf disease identification via improved CycleGAN and convolutional neural network. *Soft Comput.* **2023**, *27*, 9773–9786. [[CrossRef](#)]
83. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and localization methods for vision-based fruit picking robots: A review. *Front. Plant Sci.* **2020**, *11*, 510. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.