IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2024.Doi Number*

# Fruit detection methods based on Deep Learning: A Systematic Literature Review

**XINYU GONG, QIUFENG WU**

School of Arts and Sciences, Northeast Agricultural University, Harbin 150006, China

Corresponding author: QIUFENG WU (qfwu@neau.edu.cn).

This work was supported in part by the Program for Young Talents of Basic Research in Universities of Heilongjiang Province under Grant YQJH2023200.

**ABSTRACT** With the in-depth advancement of agricultural intelligence, automation, and mechanization, artificial intelligence-driven fruit detection technology has developed rapidly. As an important task in the field of agricultural computer vision, fruit target detection faces numerous challenges. This paper systematically reviews recent breakthroughs and representative research in this field. Based on an in-depth analysis of existing studies, the paper categorizes deep learning models for fruit target detection into four distinct scenarios: small sample detection (addressing challenges in data acquisition and high annotation costs), complex scene detection (dealing with detection issues in environments with overlapping, occlusion, and illumination changes), small target scenario (enhancing the model's ability to recognize low-pixel, densely packed targets), and real-time scenario (developing lightweight detection algorithms to improve inference speed). The paper summarizes the innovative technical solutions and detection performance for each scenario, analyzes the latest advancements in fruit detection, and provides valuable support and reference for technological innovations in this field.

**INDEX TERMS** Artificial intelligence, object detection, Smart agriculture, Computer vision, Deep learning, systematic literature review.

## I. INTRODUCTION

The global agricultural sector is undergoing its fourth productivity revolution driven by intelligent sensing technologies, with fruit detection emerging as a pivotal component in harvesting robots' environmental perception systems. This technology serves three critical functions: yield prediction, quality monitoring, and robotic operation guidance—strategically aligned with achieving United Nations Sustainable Development Goals through post-harvest loss reduction (FAO reports indicate 30% average losses in developing nations)[1],[2]. International policymakers have accelerated commitments through three landmark initiatives: China's 14th Five-Year Plan (mandating >50% crop harvesting automation), Horizon Europe's €2.4 billion investment in agricultural AI, and the U.S. AI Policy Roadmap (prioritizing edge computing deployment)—demonstrate global commitments to technological advancement. However, a persistent implementation gap remains between policy objectives and practical adoption due to four entrenched technical challenges under open-field conditions: small-sample scenarios (SS) requiring ~4,000 human-hours per crop species for data annotation, complex-field scenarios (CS) suffering >35% Non-Maximum Suppression failures from occlusion, small-object scenarios (SO) achieving <60% mean average precision for 32×32-pixel objects, and real-time scenarios (RS) facing 12% model accuracy degradation under computational constraints.

This article first deconstructs the fruit detection problem into four categories of scenes: SS, CS, SO, and RS. For SS , meta-learning frameworks and data augmentation mitigate data scarcity in rare fruit cultivars. In CS, focal loss reweighting and adaptive NMS architectures counteract occlusion-induced feature ambiguity. ST employ hierarchical feature pyramids and coordinate attention mechanisms to recover semantic information from low-resolution targets. For RS, neural architecture search (NAS) combined with model distillation-pruning pipelines achieves Pareto-optimal speed-accuracy tradeoffs on edge devices. Our methodology integrates a three-stage analysis of 140 studies from Web of Science and CNKI databases (2018–2024), filtering to 81 scenario-specific technical improvements through keyword screening, empirical validity assessment, and critical synthesis.

This paper conducts a systematic review of four key challenging scenarios in fruit detection and evaluates effective deep learning-based optimization strategies tailored to each scenario, offering actionable guidance for advancing research in this field. Relevant literature was retrieved from the Web of Science and CNKI databases, followed by rigorous screening, critical analysis, and synthesis of improvement methodologies addressing these challenges. This article focuses on deep learning-based optimization methods for fruit detection. It begins with a comprehensive analysis of the current development status, underlying principles, and persistent challenges in the field. Subsequently, it explores targeted improvement strategies for four key scenarios and concludes by outlining potential future trends in fruit detection research.

## II. Overview of the development of fruit testing methods

This article retrieves 140 relevant literature through keywords such as "deep learning," "agriculture," "object detection," "fruit," "computer vision," and "recognition." However, some of the literature focuses on fruit counting, or does not use deep learning methods, or is not optimized for the specific scenarios involved in this article, and is therefore not suitable as a reference for a comprehensive review of fruit detection. In the end, 81 relevant literature were selected, and the specific countries and the top five universities in terms of publication volume are shown in Tables 1 and 2, respectively.

TABLE 1
QUANTITY OF ARTICLES PER COUNTRY

| **Country** | Num |
|---|---|
| China | 117 |
| England | 3 |
| Italy | 2 |
| Japan | 2 |
| USA | 2 |
| Belgium | 1 |
| Brazil | 1 |
| France | 1 |
| Germany | 1 |
| India | 1 |
| Korea | 1 |
| Malaysia | 1 |
| Netherlands | 1 |

TABLE 2
QUANTITY OF ARTICLES FROM TOP 5 UNIVERSITIES

| Country | Institution | Num |
|---|---|---|
| China | South China Agr University | 10 |
| China | Northwest Agriculture and Forestry University | 9 |
| China | South China Agricultural University | 7 |
| China | Jiangsu University | 6 |
| China | Shandong Normal University | 5 |

The development of fruit detection technology has gone through an evolution from traditional computer vision to deep learning paradigms. Early research mainly used image processing and feature extraction techniques, such as edge detection, color thresholding, and shape recognition, to identify and locate fruits in images. However, these methods performed poorly in complex background, lighting changes, or occlusion environments, making it difficult to meet the actual application needs of agriculture[[3]-[6]. With the rise of convolutional neural networks (CNNs), deep learning-based detection frameworks can automatically extract features and recognize complex patterns, significantly reducing the reliance on manual feature engineering. With the development of deep learning technology, frameworks like YOLO have been widely used due to their ability to achieve real-time and accurate fruit detection, and their excellent performance in complex environments.

During this process, the research focus gradually shifted to addressing challenges unique to agricultural scenarios, such as the detection of small target fruits, dense fruit clusters, and detection under varying lighting conditions. To tackle these issues, researchers introduced techniques like data augmentation, multi-scale feature pyramids, and attention mechanisms. These technological breakthroughs have benefited from the collaborative efforts of global research forces, with statistics showing that countries like China, the UK, and the US have contributed over 90% of high-quality fruit detection papers. Research institutions such as South China Agricultural University and Northwest A&F University have played a significant role in advancing the development of fruit detection systems. As research continues to deepen in various regions, the number of research papers and technological advancements in fruit detection has gradually increased worldwide.

Despite significant progress in data-driven deep learning methods, four core challenges remain in practical deployment: Firstly, high-performance models rely on a large number of labeled samples, which are difficult to obtain in real-world scenarios. To address this issue, researchers have proposed optimization methods aimed at dealing with SS through data set augmentation or enhancing the model's self-learning ability. Secondly, issues such as lighting changes, target overlap and occlusion, and background interference in CS seriously affect the accuracy and robustness of target detection, making it a hot topic in current research. Thirdly, for SO, visual features are difficult to obtain efficiently and are noisy, so how to improve the accuracy of small target detection has become another important issue in the field of computer vision. Finally, the complexity of model parameters and the huge amount of computation in the inference process slow down the inference speed, making the model too large and posing challenges for real-time deployment. To address this issue, researchers are seeking a balance between model detection performance and lightweighting to optimize the application effect in RS.

## III. Overview of fruit target detection

### *A. Introduction to fruit object detection methods*

As the core technology of computer vision, object detection aims to realize the dual perception of target localization and classification in images. In the fruit detection scenario, its core task is to extract the feature representation of the target

from the complex farmland image, and then realize the application of automatic harvesting, yield evaluation and quality grading. With the development of deep learning, the mainstream detection methods have gradually shifted from traditional manual feature extraction to automatic feature learning frameworks represented by CNN and Transformer. Current object detection algorithms based on deep learning can be divided into the following three categories.

#### 1) ANCHOR-BASED ALGORITHMS

Two-stage detectors, represented by Faster R-CNN, first generate candidate regions through the RPN network, then align features and perform classification and regression via RoI Pooling[7]. Their accuracy advantage stems from the meticulous screening of candidate regions, but computational redundancy is significant (e.g., RoI cross-layer feature fusion requires high-resolution feature maps). Single-stage detectors, represented by the YOLO (You Only Look Once) series and SSD, achieve end-to-end detection through grid-based predictions, significantly improving detection speed at the cost of some accuracy, making them more suitable for agricultural robot scenarios with high real-time requirements[8].

#### 2) ANCHOR-FREE ALGORITHMS

To avoid the sensitivity of anchor box parameters, algorithms such as FCOS predict the target center points and bounding boxes on a per-pixel basis, reducing the cost of hyperparameter tuning[9]. Such methods have potential advantages for SO detection but are sensitive to occlusion scenarios with dense fruit overlaps (e.g., ambiguity in center point prediction in crowded apple scenes).

#### 3) TRANSFORMER-BASED END-TO-END DETECTION ALGORITHM

Models represented by DETR model global contextual relationships through self-attention mechanisms, eliminating the need for NMS post-processing and demonstrating strong robustness in occlusion scenarios. However, due to the $O(N^2)$computational complexity of Transformers, they are difficult to deploy on agricultural edge devices (such as orchard mobile robots)[10].

Table 3 summarizes the key scenario requirements for agricultural fruit detection and the corresponding technological innovation directions.

TABLE 3
SUMMARY OF SCENARIOS

| Scene Type | Typical Characteristics | Technical Challenges | Optimization Direction Examples |
|---|---|---|---|
| CS | lighting changes, foliage occlusion, fruit overlap | feature confusion, edge blurring | adaptive data augmentation, context-aware loss function |
| SO | Dense Fruits, Large Scale Differences | Low-Resolution Feature Loss | Multi-Scale Feature Fusion, Attention-Guided RPN |
| RS | Dynamic response requirements for picking robots | model computation latency | lightweight network design, knowledge distillation compression meta-learning framework, synthetic data generation |
| SS | Rare variety annotation data scarcity | overfitting risk | |

### B. Evaluation index

The performance of object detection algorithms is often quantified and evaluated through multi-dimensional metrics. Below are the classic evaluation methods.

#### 1) CONFUSION MATRIX AND BASIC INDICATORS

TP：True Positive；FP：False Positive；FN：False Negative；TN：True Negative.

This leads to equations 1-4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{4}$$

#### 2) BOUNDARY BOX LOCALIZATION EVALUATION

IoU(Intersection over Union): a measure of the overlap between the predicted frame and the real frame.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

Average Precision (AP): Based on the area under the Precision-Recall Curve, it comprehensively reflects the performance balance at different recall rates.

$$AP = \int_0^1 Precision(r)dr \tag{6}$$

Mean Average Precision (mAP): The arithmetic mean of AP across multiple classes.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{7}$$

#### 3) ROBUSTNESS EVALUATION METRICS

Matthews Correlation Coefficient (MCC): Suitable for imbalanced data distributions.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{8}$$

## IV. ANALYSIS OF IMPROVED METHODS FOR FRUIT DETECTION TECHNOLOGY

This section provides an in-depth analysis of fruit detection based on deep learning from the perspective of technological evolution. It focuses on the challenges and hotspots in recent years, including SS, CS, SO, and RS. The unique perspectives of each scenario's methods are introduced in detail. Among the 81 references in this article, 5 are for SS, 38 are for CS, 38 are for SO, and 24 are for RS.

The paper is structured into four main sections, each focusing on a different aspect of fruit detection technology. The first section, "Development Overview," provides a historical perspective on the evolution of research focus and the development of object detection in deep learning. The second section, "Core concepts," delves into the fundamental principles of fruit target detection and the evaluation metrics used for target detection algorithms. The third section, "Research trends: methods for different scenarios," explores various methods tailored to specific detection challenges, such as SS, CS, SO, and RS. Each scenario is addressed with unique techniques, such as data augmentation, meta-learning, non-maximum suppression, loss function optimization, feature pyramid, attention mechanism, pruning, and knowledge distillation. Finally, the "Prospects for future research" section outlines potential directions for advancing the field, ensuring a forward-looking perspective on fruit detection technology. As shown in Fig. 1.

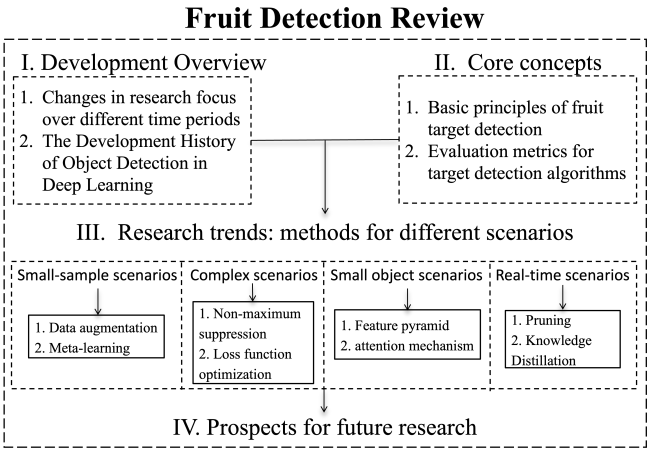


**FIGURE 1. Paper Structure.**

### *A. DETECTION OF SS*

In the field of deep learning, small-sample scenarios refer to the situation where the amount of training data is relatively small, and are common in areas where data acquisition and labeling costs are high, such as medical image analysis, remote sensing, and agriculture. The research problems of SS are mainly focused on insufficient data, overfitting, insufficient model generalization ability, limited data enhancement effect, improper model architecture design, poor application of transfer learning, imperfect self-supervised learning methods, and challenges of evaluation methods[11].

In the context of SS, the improved methods for fruit detection mainly revolve around data augmentation and generalized meta-learning. Table 4 provides a brief summary and analysis of these two approaches, outlining the backbone networks in the literature and the improvement techniques aimed at addressing the challenges of SS. Sections 4.1.1 and 4.1.2 focus on analyzing the network structures and improvements of the corresponding typical models.

TABLE 4
COMPARATIVE ANALYSIS OF SMALL SAMPLE FRUIT DETECTION METHODS

| Classification | Backbone | research object | Improvement Points | Enhancement effect | Reference number |
|---|---|---|---|---|---|
| Data augmentation | ResNet | Citrus | ResNet101<br>Data Augmentation | For the farm's only 200 labeled datasets, the current AP is 97.1%. | [11] |
| | | Pomegranate | ResNet18<br>Pseudo-label learning | Under simulated natural growth conditions, the F1 score is approximately 86.42%, and the processing time is about 0.15s per frame. | [12] |
| meta-learning | | Pomegranate | ResNet18<br>Pseudo-label learning | Under simulated natural growth conditions, the F1 score is approximately 86.42%, and the processing time is about 0.15s per frame. | [13] |
| | ResNet | Apple | ResNet50<br>Soft Teacher | For the MSU Apple Dataset V2, the current AP is 51.8%, which is a 17.4% improvement over the original model AP. | [13] |
| | | | ResNet50<br>TreeAttention | For the MSU Apple Dataset V2, the current AP is 51.8%, representing a 17.4% improvement over the original model's AP. | [14] |
| | CSPDarknet | Grape | CSPDarknet53<br>Geometric Consistency for Generating Pseudo Labels | In the vineyard environment, mAP0.5:0.95 is 47.81%, an improvement of 18.5%, with a parameter count of 7.2M and an inference time of 0.0064s/frame. | [15] |
| | Convolutional Auto-Encoder | Camellia fruit research object | Asymmetric Decomposition Convolution Kernel | In the actual image detection of the planting base environment, the classification accuracy is 87%, and the processing time is 12s/frame | [16] |
| | | | Direct connection training | In the actual image detection of the planting base environment, the classification accuracy is 87%, and the processing time is 12s per frame | [16] |

### 1) APPLICATION OF DATA AUGMENTATION IN FEW-SHOT SCENARIO DETECTION

Data augmentation refers to generating new training samples by transforming the original data, aiming to expand the size and diversity of the dataset. It focuses on the data level and does not involve changes to the model structure or learning strategies.

In the context of small-sample object detection, data augmentation techniques are widely used to enhance model performance and reduce reliance on large amounts of annotated data. Devanna R et al.[13] employed the ResNet18 backbone network in pomegranate detection, combined with pseudo-label learning methods, significantly reducing the workload of manual annotation while improving segmentation accuracy. Kim J W et al.[12] used the ResNet101 backbone network in citrus detection, increasing the amount of training data through data augmentation techniques, which reduced overfitting, although it also increased the model's sensitivity to lighting conditions. These studies indicate that data augmentation techniques play a crucial role in small-sample scenarios, alleviating data scarcity to some extent while improving detection accuracy. Specifically, analysis of the typical ResNet-18 network shows that it generates diverse training samples through data augmentation techniques (such as rotation, flipping, contrast adjustment, etc.). The augmented data can simulate variations in lighting, perspective, and fruit morphology, enabling the model to learn more diverse feature representations[17].

### 2) APPLICATION OF META-LEARNING IN FEW-SHOT SCENE DETECTION

Broadly speaking, meta-learning refers to the idea of "learning to learn". It does not aim to change the amount of data available, but rather optimizes model structure, refines training strategies, and utilizes additional data to generate pseudo-labels. This enables the model to automatically design hyperparameters and select optimizers, thereby quickly adapting to and outputting results in small-sample scenarios. These techniques ultimately enhance model adaptability and learning efficiency.

In the context of few-shot object detection, the core idea of meta-learning is to reduce reliance on manually annotated data by cleverly utilizing unlabeled data or optimizing network structures. For example, Devanna R et al.[13] and Johanson R et al.[14] have effectively leveraged unlabeled data through pseudo-label learning and semi-supervised learning frameworks (such as Soft Teacher) in tasks involving pomegranate and apple detection, respectively. Additionally, researchers have attempted to adapt to few-shot scenarios by improving network architectures and feature extraction methods. For instance, Johanson R et al. also employed a selective tiling strategy combined with the TreeAttention module to focus on regions of interest in images. Similarly, Ciarfuglia T A et al.[15] used CSPDarknet53 in grape detection to generate pseudo-labels by leveraging geometric consistency between video frames, replacing traditional bounding box label generation methods and further optimizing the detection process. Semi-supervised learning frameworks require additional computational resources to generate pseudo-labels, and the selective tiling strategy necessitates extra preprocessing steps. This raises several questions: How can we balance model performance and computational costs in few-shot scenarios? How can we more efficiently utilize unlabeled data, and how can we design network architectures that are better suited for few-shot detection?

Specifically, ResNet 18 can be used via the pseudo-labeling learning framework to generate high-quality pseudo-labels with a small amount of labeled data, guiding the model for self-supervised learning on unlabeled data, while ResNet-50 has the ability to extract common and highly discriminative features with its deep residual structure and bottleneck module design.

As shown in Fig. 2, the input layer of ResNet-50 is the same as that of ResNet-18, but its 4 residual stages are composed of bottleneck residual modules (Bottleneck Block) stacked together. Each module consists of 1x1 convolution (dimensionality reduction) → 3x3 convolution (feature extraction) → 1x1 convolution (dimensionality increase), and the final feature map is compressed into a 2048-dimensional vector through a global average pooling layer, and the classification result is output by the fully connected layer. Compared with ResNet-18, ResNet-50 has a deeper network structure (50 layers) and higher feature extraction capability.

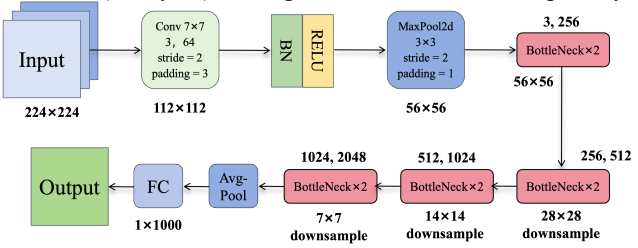


**FIGURE 2. Resnet50 network structure.**

Analysis of the typical Convolutional Auto-Encoder (CAE) network reveals that, compared to traditional symmetric convolutional kernels, the addition of asymmetric decomposed convolutional kernels by Zhang Xizhi et al.[16] in the detection of oil tea fruits better captures diverse feature information. By using the direct connection path method, more original data information is preserved, which to some extent reduces information loss during the encoding and decoding processes[18].

The Convolutional Auto-Encoder mainly consists of an encoder and a decoder, as shown in Fig. 3. The input image is randomly divided into two parts, one is visible and the other is masked. The ViT model outputs the feature representation $Z_v$ of the visible patches as the encoder. The features of the masked patches $Z_m$ are predicted through a cross-attention mechanism, and by adding constraints, the output of the Latent contextual regressor is aligned with the direct output of the encoder $\overline{Z_m}$ in the same encoding space. The decoder only

uses $Z_m$ and positional encoding as input to obtain the predicted output.

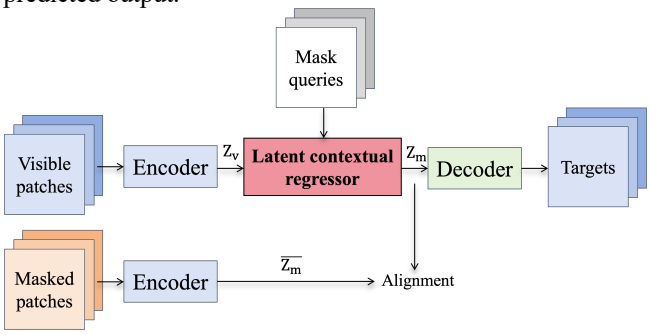


**FIGURE 3.** **Network structure of Convolutional Auto-Encoder.**

### *B. DETECTION OF CS*

In the field of deep learning, CS detection refers to detection tasks performed under conditions marked by complex data distributions, variable environmental factors, and various interfering elements. This is particularly challenging when there is significant occlusion between the target and the background, such as dense occlusion, or when there are drastic changes in lighting conditions, including strong light, weak light, shadows, and reflections. In these scenarios, detection algorithms must exhibit enhanced capabilities in target feature extraction, environmental adaptability, and robustness. These qualities are essential to overcoming challenges such as incomplete target features, complex and diverse backgrounds, and dynamic lighting conditions[19].

The methods aimed at improving fruit detection in CS primarily focus on optimizing loss functions and enhancing non-maximum suppression (NMS) techniques. Table 5 presents a summary and analysis of the optimization approaches in these two areas, highlighting the improved techniques and the corresponding backbone networks.

TABLE 5
COMPARATIVE ANALYSIS OF COMPLEX FRUIT DETECTION METHODS

| Classification | Backbone | research object | Improvement Points | Enhancement effect | Reference number |
|---|---|---|---|---|---|
| Loss function | ResNet | Hami melon | ResNet43<br>Ciou | In a natural daylight greenhouse environment, the AP is 89.6%, the parameter count is significantly reduced to 98.1MB compared to a standard model, the detection speed is 0.0104s per frame, which is 56.1% faster than YOLOv4. | [20] |
| | | Apple | ResNet50<br>①Focal loss<br>②Eiou loss | For multi-classification tasks, the mAP is 91.26%, which is a 5.02% improvement in accuracy compared to the traditional RetinaNet. The model size is 128MB, which is 75.4% and 47.5% smaller than Faster-RCNN and YOLOv4, respectively, and the detection time is 0.04272s per frame. | [21] |
| | | | ResNet50<br>KL divergence loss function | On the GreenApple dataset, AP is 62.3%, small target APS is 47.0%; on the MinneApple dataset, AP is 43.5%, small target APS is 42.2%; on the Pascal VOC dataset, AP is 51.4%, small target APS is 35.6%. The parameter count is 42.55M, an increase of 1.43M compared to the baseline model. | [22] |
| | | Tomato | CSPDarkNet53<br>Ciou | In a nighttime greenhouse environment, the mAP is 96.7%, an improvement of 3.3% compared to the regular model, with recognition accuracies for green and red fruits at 96.2% and 97.6% respectively, and a detection time of 0.01s per frame. | [23] |
| | | Tomato | CSPDarkNet53<br>①Variable Focal Loss<br>②Wise-IoU Regression Loss Function | In real-time detection under complex growth conditions, the mAP0.5 is 91.4%, which is a 4.3% improvement over the baseline model, with a model complexity (GFLOPs) of 42.4 and a detection time of 0.0166s per frame. | [24] |
| | Darknet | Kiwifruit | CSPDarknet53<br>Ciou | In a greenhouse environment, the mAP0.5 reached 96.3%, an improvement of 3.8% compared to the original model. The computational load of the model was reduced, and the detection time was 0.03597s per frame, a decrease of 0.00907s per frame compared to the original model. | [25] |
| | | | CSPDarknet53<br>①Focal loss<br>②Ciou loss | In a greenhouse environment, the mAP0.5 is 96.3%, an improvement of 3.8% compared to the original model, with reduced computational load. The detection time is 0.03597s/frame, a decrease of 0.00907s/frame compared to the original model. | [25] |
| | | | DarkNet19<br>Giou combined with focal loss function | Under different lighting, density, and occlusion conditions, the mAP is 96.3%, the model size is 106M, which is 47.8%, 48.8%, and 56.6% smaller compared to Faster-RCNN, SSD, and YOLOv4 respectively, with a detection time of 0.0505s/frame. | [26] |
| | | | CSPDarknet<br>Joint bounding box regression loss function with intersecting scales | On datasets with varying weather, lighting, maturity, and occlusion, the mAP0.5 is 97%, the number of parameters is approximately 28.993M, a reduction of 22.93% compared to the original model, and the detection time is approximately 0.0132s per frame, which is 0.003s faster than the original model. | [27] |
| | | | CSPDarkNet53<br>Adaptive Fully Connected Regression | Under different natural lighting conditions, AP0.5 is 96.4%, with a detection time of approximately 0.0172s per frame. | [28] |
| | | Apple | CSPDarkNet53<br>Giou | Under different lighting, occlusion levels, density, and smoothness conditions, the mAP is 98.71%, the model size is 14.08MB, which is 124.84MB and 124M smaller compared to YOLO v4tiny and RetinaNet models respectively, and the detection time is 0.0127s per frame. | [29] |
| | | | CSPDarknet<br>Focal Eiou loss | Under different varieties, lighting conditions, and occlusion scenarios in the forest farm data, the mAP is 96.65%, and the detection time is 0.03657s/frame. | [30] |
| | | Lychee | CSPDarkNet<br>①CIOU Loss<br>②PolyLoss | Under datasets with varying lighting, angles, and occlusion conditions, the mAP is 82.62%, the parameter count is 5.48M, which is a 44.8% reduction compared to the standard model, and the detection time is 0.0156s per frame. | [31] |
| | | Pear | CSPDarkNet53<br>Ciou | For datasets covering various CS captured on-site and downloaded from the internet, the mAP is 87.1%, the model size is 44.8MB, and the detection time is 0.025s per frame. | [32] |

| | | | | | |
|---|---|---|---|---|---|
| | | | CSPDarkNet53<br>①Regression Loss Function<br>②Normalized Gaussian Wasserstein Distance Metric | For the expo and orchard environment, the AP is 72.6%, which is a 22% improvement over the original model. The parameter count is 7.24M, an increase of 0.23M, and the detection time is approximately 0.0139s per frame. | [33] |
| | | Yellow Peach | CSPDarkNet53<br>Focal loss | In a natural orchard environment, the mAP is 88.5%, which is an improvement of 5.4% compared to the ordinary model, with a parameter count of 50.9M and a detection time of 0.019s per frame. | [34] |
| | | Strawberry | CSPDarkNet53<br>Weighted confidence loss function | In complex orchard environments, the F1 score is 96.1%, an improvement of 3.3% compared to the original model, with a parameter count of 8.3MB, reduced by 39.4%. | [35] |
| | | Camellia fruit | CSPDarkNet<br>Eiou | For complex natural scenes, the mAP is 80.4%, which is a 3.5% improvement over the original model, with a model size of 51.9MB and a detection time of 0.0476s per frame. | [36] |
| | | 20 common fruits and vegetables | CSPDarkNet53<br>Ciou | In a complex planting environment with dense targets and background interference, the mAP is 90%, which is an improvement of 4.4% compared to the original model. The number of parameters is 12M, an increase of 4.9M over the original model, and the detection time is 0.0172s per frame, which is 0.007s slower per frame. | [37] |
| | | Bananas, oranges, and apples | SAM CSPDarkNet53<br>Focal loss | For the Wine Grape Instance Segmentation Dataset (WGISD), the AP reached 94.25%, with a detection time of 0.0263s per frame. | [38] |
| | | Cherry | MobileNetv2<br>Giou | For datasets with varying lighting, angles, and occlusion conditions, the AP is 91.13%, with a reduction in the number of parameters compared to traditional models. The detection speed on GPU is 0.0169s/frame, and on CPU is 0.0809s/frame. | [39] |
| | MobileNet | Hami melon | MobileNetv3<br>Alpha iou | For datasets with no occlusion, occlusion by foliage, and mutual occlusion of targets, the mAP is 99.4%, the model size is 4.5MB, reduced by 41.5%, and the detection time is approximately 0.0227s/frame, reduced by about 13.69%. | [40] |
| | | Citrus | MobileNetv3<br>①GIoU<br>②Dice loss | In multi-classification tasks, the accuracy is 96%, with MobileNetV3 Small and MobileNetV3 Large having parameter counts of 2.5M and 5.5M, respectively. | [41] |
| | | Strawberry | CSPDarkNet53<br>Tiny Soft nms | Under different lighting, regions, varieties, and maturity datasets, the mAP is 91.08%, the parameter count is 8M, which is 5M less than the original model, the detection time is 0.0123s/frame, and it is 31.15 times, 3.38 times, and 6.45 times faster than Faster-RCNN (Resnet50), SSD300, and YOLOv4, respectively. | [42] |
| | | Grape | DarkNet53<br>Soft nms | In a greenhouse environment, the mAP0.5 is 96.3%, an improvement of 3.8% compared to the original model, with a detection time of 0.03597s per frame, a reduction of 0.00907s per frame compared to the original model. | [25] |
| | Darknet | Grape | CSPDarkNet53<br>Soft nms | For drone-captured images, the mAP0.5-0.95 is 98.5%, with a detection time of approximately 0.04s per frame, slightly slower than the original model. | [43] |
| NMS | | Citrus | CSPDarkNet53<br>Greedy confluence's nms | For multi-classification tasks, the mAP is 96.65%, which is a 1.7% improvement over the regular model, with a detection speed of approximately 0.0254s per frame. | [44] |
| | | Grape | CSPDarkNet53<br>Soft nms | In dense scenes for detecting small and dense objects, the mAP0.5 is 98.72%, with a detection speed of 0.20567s/frame, which is 0.20567s/frame faster than Faster-RCNN and 0.03913s/frame faster than YOLOv4. | [45] |
| | ResNet | Apple | ResNet43<br>Diou nms | In a natural daylight greenhouse environment, the AP is 89.6%, the parameter count is significantly reduced to 98.1MB compared to a standard model, the detection speed is 0.0104s per frame, which is 56.1% faster than YOLOv4. | [20] |
| | | Apple | ResNet50<br>Soft nms | In complex orchard environments with small and dense targets, the mAP0.5 for mature targets is 94.36%, while for immature targets it is 84%. | [46] |

### 1) APPLICATION OF LOSS FUNCTION IN CS DETECTION

In fruit detection research, the enhancement of loss functions primarily targets the optimization of bounding box regression and classification tasks. By introducing or combining different loss functions, several challenges, such as detecting overlapping occlusions and mitigating interference from complex backgrounds, have been effectively addressed. For instance, in detecting fruits such as apples, tomatoes, lychees, and yellow peaches, loss functions like CIoU, GIoU, and EIoU have been widely adopted in conjunction with backbone networks such as the ResNet and DarkNet series. This combination has significantly improved detection accuracy and reduced false positives.

To tackle challenges in specific scenarios, researchers have proposed innovative methods, such as adaptive fully connected regression loss functions and variable Focal Loss, further enhancing the model's detection capabilities[24][28]. Notably, analysis of the typical network MobileNetV2 reveals that, through careful selection and optimization of loss functions, it achieves a balance between classification and localization performance, all while maintaining the advantages of its lightweight design. In low-light scenarios, the Dice loss function helps the model better identify targets, particularly in situations with low contrast[41]; In cases of occlusion and overlap, the Alpha IoU loss function helps the model focus more effectively on the classification and localization of occluded regions by balancing the weights of classification and localization losses[40]; Additionally, compared to traditional L1 or L2 loss functions, the GIoU loss function has demonstrated superior performance in bounding box regression tasks, particularly in the presence of occlusion and overlap[29][39][41].

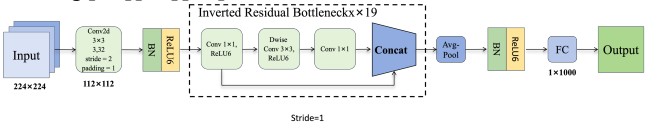


**FIGURE 4.** MobileNetv2 structure.

MobileNetV2 builds upon the depthwise separable convolution (DSC) technique introduced in MobileNetV1, effectively reducing computational cost and the number of parameters by decomposing the standard convolution into two stages: depthwise convolution and pointwise convolution. This separation allows each to handle channel and spatial information independently, thereby improving efficiency. Unlike its predecessor, MobileNetV2 incorporates inverted residual blocks, which expand the network's depth by processing shallow features while maintaining computational efficiency.

Each inverted residual block consists of three convolutional layers: a 1x1 expansion convolution, a 3x3 depthwise separable convolution, and a 1x1 projection convolution. Furthermore, MobileNetV2 utilizes linear bottleneck layers within the inverted residual blocks. These layers optimize feature representation by employing linear activation functions (e.g., ReLU6), which help prevent the loss of intermediate layer features, thereby improving the overall classification performance[47]. This architecture is illustrated in Fig. 4.

### 2) APPLICATION OF NMS IN CS DETECTION

In object detection tasks, NMS is a widely used post-processing technique designed to filter out redundant detection boxes and retain the results with the highest confidence scores. However, in CS, where objects may overlap, be occluded, or are densely distributed, traditional NMS can lead to missed or false detections. Therefore, improvements to NMS focus on enhancing detection accuracy and minimizing redundant prediction boxes, particularly in scenarios with densely packed objects or complex backgrounds.

For example, within the DarkNet series of backbone networks, various enhanced NMS methods, such as Soft NMS and Greedy Confluence NMS, have been proposed. These methods significantly improve detection performance by optimizing the strategy for selecting candidate boxes, particularly excelling in fruit detection tasks for fruits like apples, grapes, bananas, and cherries. Additionally, in the ResNet series, researchers have combined Diou NMS with Soft NMS, further optimizing detection accuracy and robustness, achieving excellent results in complex scenarios such as the detection of strawberries and cantaloupes.

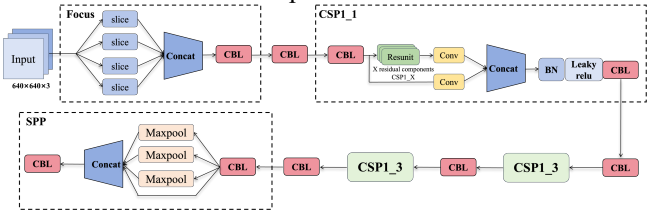


**FIGURE 5.** CSPDarknet53 structure.

CSPDarknet53 is an enhanced version of Darknet53, incorporating the Cross Stage Partial (CSP) architecture[48]. As illustrated in Fig. 5, the backbone consists of multiple CSP modules and residual blocks, with each CSP module comprising several convolutional layers and skip connections. In the CSP module, some of the convolutional layers are replaced with DSC to improve computational efficiency.

To address the issue of target overlapping and occlusion in CS, several improvements have been integrated into CSPDarknet53: The Tiny Soft NMS method dynamically adjusts the confidence level of overlapping frames using a Gaussian attenuation function (e.g., $S_i = S_i \cdot e^{-IoU^2/\sigma}$). This approach prevents the omission of detections due to the hard thresholding in traditional NMS, complementing CSP's high recall characteristics. Additionally, the Greedy Confluence strategy introduces multi-dimensional

weights, such as location sensitivity and category consistency (e.g., $Score = \alpha \cdot cls + \beta \cdot IoU + \gamma \cdot centerness$ ) to comprehensively assess the validity of detection boxes and mitigate the issue of single-confidence bias.

Experimental results on the VisDrone dense scene dataset demonstrate that these improvements enhance the mAP of CSPDarknet53 by 4.5% and reduce the false positive detection rate (FPPI) by 18%, highlighting the effectiveness of co-optimizing the backbone network with intelligent suppression strategies.

### C. DETECTION FOR SO

SO detection refers to object detection tasks performed under low-resolution imaging conditions, where valuable semantic information is often lost. In SO detection, traditional detection algorithms encounter several challenges, including difficulty in effectively extracting fine-grained features, susceptibility to background noise and other interference factors, and the use of inappropriate anchor box sizes, among others[49].

Advancements in fruit detection for SO primarily focus on feature pyramids and attention mechanisms. Table 6 summarizes and analyzes the research methods in these two areas, providing an organized overview of the improvement strategies, their corresponding enhancement effects, and the backbone networks employed for SO detection.

TABLE 6
COMPARATIVE ANALYSIS OF SO FRUIT DETECTION METHODS

| Classification | Backbone | research object | Improvement Points | Enhancement effect | Reference number |
|---|---|---|---|---|---|
| Feature Pyramid | ResNet | Apple | ResNet50<br>Bidirectional Feature Pyramid Network (BiFPN) | For multi-classification tasks, the mAP is 91.26%, which is a 5.02% improvement in accuracy compared to the traditional RetinaNet. The model size is 128MB, which is 75.4% and 47.5% smaller than Faster-RCNN and YOLOv4, respectively, and the detection time is 0.04272s per frame. | [21] |
| | | Pomegranate | ResNet18<br>Fast Spatial Pyramid Pooling (SPPF) | Under simulated natural growth conditions, the F1 score is approximately 86.42%, and the processing time is about 0.15s per frame. | [13] |
| | Darknet | Apple | DarkNet53<br>Spatial Pyramid Pooling (SPP) | In a greenhouse environment, the mAP0.5 is 96.3%, an improvement of 3.8% compared to the original model, with a detection time of 0.03597s per frame, a reduction of 0.00907s per frame compared to the original model. | [25] |
| | | | CSPDarknet<br>Path-Path Bifurcation Feature Pyramid Network (P2BiFPN) | In complex weather conditions such as rain and fog, the mAP0.5 for the MinneApple dataset is 80.4%, the model size is 5.06MB, and the detection time is approximately 0.0099s per frame. | [50] |
| | | Bayberry | CSPDarknet53<br>SPP | Under different lighting, occlusion levels, density, and smoothness conditions, the mAP is 98.71%, the model size is 14.08MB, which is 124.84MB and 124M smaller compared to YOLO v4tiny and RetinaNet models respectively, and the detection time is 0.0127s per frame. | [29] |
| | | | CSPDarknet53<br>Lightweight Bidirectional Feature Pyramid Network(light-BiFPN) | Under different colors, distribution densities, and light intensities, the mAP0.5 is 91.4%, the model size is 46.9MB, and the detection speed is approximately 0.0266s/frame. | [51] |
| | | | CSPResNest50<br>Recursive Feature Pyramid (RFP) | In complex orchard environments with varying lighting, occlusion, and density levels, the mAP is 94.6%, with a detection speed of approximately 0.0196s per frame. | [52] |
| | | Camellia fruit | CSPDarknet53<br>BiFPN | In a natural orchard environment, the mAP is 88.5%, which is an improvement of 5.4% compared to the ordinary model, with a parameter count of 50.9M and a detection time of 0.019s per frame. | [34] |
| | | | CSPDarknet53<br>BiFPN | In a nighttime environment using active light sources, the AP is 97.1%, with a detection speed of 0.025s/frame. | [53] |
| | | Blueberry | CSPDarknet53<br>BiFPN | For the expo and orchard environment, the AP is 72.6%, which is a 22% improvement over the original model. The parameter count is 7.24M, an increase of 0.23M, and the detection time is approximately 0.0139s per frame. | [33] |
| | | Citrus | CSPDarknet<br>SPPFCSPC: SPPF + Cross-Stage Partial Layer | For the high-density complex background dataset, the mAP is 91.9%, an improvement of 5.4% compared to the original model, with a parameter count of 5.34M, reduced by approximately 24%, and a detection time of 0.0251s per frame. | [54] |
| | | Pineapple | CSPDarknet53<br>SPPF | For the dataset of different lighting and maturity levels in open field environments, the mAP is 80.3%, with APs for immature, near-mature, and mature targets being 82.1%, 73.5%, and 86.6% respectively. The parameter count is 9.4M, and the detection time is 0.0181s per frame. | [55] |
| | GhostNet | Lychee | GhostNetV1<br>Coordinate attention | In the complex background of an orchard environment, the mAP0.5 is 95.72%, with the AP for immature targets at 95.91% and the AP for mature targets at 95.54%. The parameter count is 10.2M, which is approximately 84.5% less than the original model. The detection time is 0.02212s per frame, which is approximately 12.64% less than the standard model. | [56] |
| | | Strawberry | GhostNetV1<br>①SPP<br>②FPN | Under natural lighting conditions in a plantation environment, the mAP0.5 is 92.62%, an improvement of 5.77% compared to the original model. The AP for mature and immature targets are 95.28% and 89.97%, respectively. The model size is 4.68MB, which is 18.89MB smaller than the original model. The detection time is 0.00563s per frame, an increase of 0.00114s per frame. | [57] |
| Attention mechanism | CSPDarknet | Apple | CSPDarknet53<br>Convolutional layer combined with | Under the dataset of different lighting, angles, and times in natural scenes, the mAP0.5 is 98.23%, the model size is 27MB, and the detection time is 0.017s per frame. | [58] |

| | | | | | |
|---|---|---|---|---|---|
| Classification | | | Convolutional Block Attention Module (Conv_CBAM) | | |
| | | | CSPDarknet53<br>Visual Attention Mechanism Coordinated Attention Module (CA) | In a natural orchard environment, the mAP is 88.5%, which is an improvement of 5.4% compared to the ordinary model, with a parameter count of 50.9M and a detection time of 0.019s per frame. | [34] |
| | | Strawberry | CSPDarknet<br>Convolutional Attention Module | For complex environments with varying lighting, occlusion, and long distances, the mAP0.5 is 97.9%, the model size is 81.3MB, reduced by 430.9MB compared to the original model, and the detection time is 0.009s per frame, reduced by 0.006s per frame. | [59] |
| | | | CSPDarknet53<br>Normalized Attention Module (NAM) | In real-time detection under complex growth conditions, the mAP0.5 is 91.4%, which is a 4.3% improvement over the baseline model, with a model complexity (GFLOPs) of 42.4 and a detection time of 0.0166s per frame. | [24] |
| | | Citrus | CSPDarknet53<br>Convolutional Block Attention Module (CBAM) | For datasets covering various CS captured on-site and downloaded from the internet, the mAP is 87.1%, the model size is 44.8MB, and the detection time is 0.025s per frame. | [32] |
| | | Tomato | CSPDarknet<br>Global Attention Mechanism | For datasets with complex backgrounds and small targets, the mAP0.5 for both large and small targets is 92.1%, an improvement of 1.6% over the original model. The mAP0.5 for only small targets is 90.4%, an improvement of 2.5%. The model size is 2.4MB, and the detection time on GPU is 0.0172s per frame. | [60] |
| | | | CSPDarknet53<br>swin transformer detection head | In the greenhouse environment, targeting different growth stages, the mAP0.5 is 92.1%, the parameter count is 37.75M, which is an increase of 0.55M compared to the ordinary model, and the detection time is approximately 0.022s per frame. | [61] |
| | | Lychee | CSPDarknet53<br>①Fusion Squeeze Excitation Block (SE)<br>②Non-local block visual attention mechanism (NL) | For the young fruit dataset in a natural environment, the AP is 96.9%, the parameter count is 255M, an increase of 11M, and the detection time is 0.0316s per frame, an increase of 0.0026s per frame. | [62] |
| | | | CSPDarknet53<br>CA | Under different natural lighting conditions, AP0.5 is 96.4%, with a detection time of approximately 0.0172s per frame. | [28] |
| | | Strawberry | CSPDarknet<br>New Convolutional Block Attention Module (NCAM) | For complex environments with dense adhesion and severe occlusion, as well as targets of varying maturity, the mAP is 83.2%, an improvement of 2.4% over the original model, with a parameter count of approximately 7.02M, reduced by about 0.458M. | [63] |
| | | Apple | CSPDarknet53<br>①The transformer module with multi-head attention mechanism CoT<br>②CBAM | For complex natural environments with different occlusions, lighting, and shooting distances, the mAP0.5 is 98.6%, the parameter count is 28M, which is 8.5M less than the original model, and the detection time is 0.0086s/frame, reduced by 0.0024s/frame. | [64] |
| | | Lychee | CSPDarknet53<br>CBAM | In natural orchard environments with backlighting, different angles, and varying maturity levels, mAP0.5 is 92.31%, AP for immature targets is 90.92%, AP for semi-mature targets is 91.98%, AP for mature targets is 94.04%, the parameter count is 28M, reduced by 8.5M, detection time is 0.025s/frame, decreased by 0.004s/frame. | [65] |
| | | Cherry | CSPDarknet53<br>Coordinate Attention Mechanism | Under different colors, distribution densities, and lighting intensities, the mAP0.5 is 91.4%, the model size is 46.9MB, and the detection speed is approximately 0.0266s per frame. | [51] |
| | | Camellia fruit | STCNN<br>The windowed multi-head self-attention (SW-MSA) mechanism of Swin Transformer combines with RCNN | For complex field environment datasets, the mAP0.5 is 92.54%, showing improvement over YOLOv5, v6, v7, v8, and Faster RCNN, with a parameter count of 28M and a detection time of 0.163s per frame. | [66] |
| | Swin Transformer | | Swin Transformer | | |
| | | Strawberry | The Swin-B Transformer module integrates with TOOD (Task-Aligned One-Stage Object Detection) | For complex field environments with varying levels of maturity, the mAP0.5 is 74.1%, an improvement of 2.2% compared to the original model, with a parameter count of 59.4M, an increase of 13.3M, and a detection time of 0.017s/frame, an increase of approximately 0.001s/frame. | [67] |

### 1) APPLICATION OF FEATURE PYRAMID IN SO DETECTION

The design and optimization of the feature pyramid primarily focus on enhancing the ability to capture multi-scale features, especially the fine-grained features of small targets, thereby improving the detection accuracy and robustness of the model. For instance, the ResNet series of backbone networks, combined with the SPPF method, has been applied to the detection of fruits such as apples and pomegranates, significantly improving the recognition capability for small targets[13].

The core objective of the DarkNet series in feature pyramid design is to effectively extract and fuse multi-scale features, achieving better detection performance in challenging scenarios[50-55]. Additionally, the GhostNet backbone network has been optimized by incorporating coordinate attention and introducing SPP, which further enhances its detection performance for small targets like apples and strawberries0. Furthermore, the Swin-Base Transformer can replace traditional feature pyramids with CARAFE-FPN, a method that more effectively captures features across different scales[66]. A detailed analysis of the typical network GhostNetV1 reveals that it is an efficient, lightweight network designed to reduce computational load through techniques such as Ghost Convolution and DSC[67]. By fusing multi-scale features, the model not only improves its ability to capture the feature information of small targets but also maintains high efficiency and real-time performance.

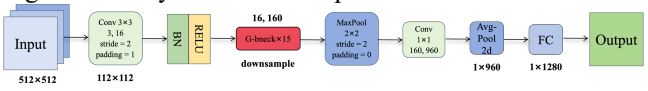


**FIGURE 6. GhostNetV1 network structure.**

As illustrated in Fig. 6, the input image is initially passed through a standard convolutional layer (typically using a 3x3 convolutional kernel with a stride of 2) to extract the initial features. This is followed by Batch Normalization and ReLU activation functions to stabilize the learning process and introduce non-linearity.The core module of GhostNetV1 is the Ghost Bottleneck, which consists of a series of operations: a 1x1 convolution, a depthwise convolution, Ghost Convolution, and an identity mapping. Additionally, GhostNetV1 integrates DSC within its architecture, optimizing computational efficiency while retaining critical feature information.

### 2) APPLICATION OF ATTENTION MECHANISM IN SO DETECTION

The design and optimization of attention mechanisms primarily focus on enhancing the model's ability to capture key features of the target, particularly the salient features of small objects, thereby improving detection accuracy and robustness. Researchers working with the CSPDarknet series have explored various attention modules, including CBAM, CA, SE, NL, NCAM, the Transformer-based CoT module with Multi-Head Attention, NAM, Global Attention Mechanism, and Coordinate Attention Mechanism, among others. These modules have been effectively applied to the detection of small targets such as citrus, tomatoes, strawberries, apples, blueberries, lychees, cherries, and oil tea fruits, enhancing the focus on the critical features of the targets.

For the Swin Transformer network, researchers have integrated its attention mechanism into other object detection models, leveraging its efficient feature extraction and attention computation capabilities to boost performance in small object detection. Specifically, the SW-MSA mechanism or the Swin-B module of the Swin Transformer has been directly incorporated into object detection models to replace or complement the original feature extraction modules[68].

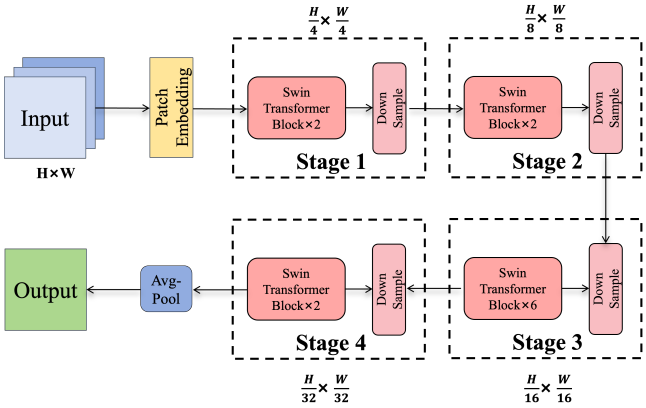


**FIGURE 7. Structure of Swin Transformer.**

As shown in Fig. 7, the Swin Transformer begins by dividing the input image into multiple small patches, which are then converted into embedding vectors. Each patch is mapped to a corresponding feature vector, forming a two-dimensional feature map. In each Transformer layer, the model computes self-attention within local windows, with the window positions sliding across the image to capture broader contextual information at multiple levels.

## *D. DETECTION FOR RS*

RS detection refers to the object detection task conducted in scenarios requiring rapid response and efficient processing, with the core goal of achieving model lightweighting and acceleration during the inference phase. The widespread application of deep learning in image detection tasks has led to high demands on computational and storage resources, making its deployment in real-time tasks challenging[69].

In RS, the improvement methods for fruit detection primarily focus on pruning and knowledge distillation. Table 7 provides a comprehensive summary and in-depth analysis of deep learning approaches in these two areas.

TABLE 7
COMPARATIVE ANALYSIS OF SMALL OBJECT FRUIT DETECTION METHODS

| Classification | Backbone | research object | Improvement Points | Enhancement effect | Reference number |
|---|---|---|---|---|---|
| Pruning | Darknet | Citrus | CSPDarknet<br>L2 regularization constraint | For the mobile camera and low-altitude drone dataset in mountainous orchard environments, the mAP is 93.32%, the model size is 21MB, which is 12MB smaller than the original model, and the detection time is 0.18s per frame. | [71] |
| | | Kiwifruit | CSPDarknet<br>Remove large target feature maps | Under datasets with varying lighting, angles, and occlusion conditions, the mAP is 82.62%, the parameter count is 5.48M, which is a 44.8% reduction compared to the standard model, and the detection time is 0.0156s per frame. | [31] |
| | | Blueberry | CSPDarknet<br>C3Ghost | For complex environments with dense adhesion and severe occlusion, as well as targets of varying maturity, the mAP is 83.2%, an improvement of 2.4% over the original model, with a parameter count of approximately 7.02M, reduced by about 0.458M. | [63] |
| | | | CSPDarknet<br>①Focus<br>②SPPF | For complex orchard environments with backlighting and different varieties, the mAP is 94.88%, an increase of 0.51% compared to the original model. The model size is 16.6MB, a reduction of 18.23%, and the detection time is 0.01006s per frame, a decrease of 20.03%. | [72] |
| | | Apple | CSPDarknet<br>PConv | For the natural orchard environment dataset with complex backgrounds, the AP is 93.86%, an improvement of 1.64% compared to the ordinary model. The parameter count is 8.83M, reduced by 2.3M, and the detection time is 0.0007s per frame, reduced by 0.0003s per frame. | [73] |
| | | | DarkNet53<br>Residual module combined with CSPNet | In a greenhouse environment, the mAP0.5 is 96.3%, an improvement of 3.8% compared to the original model, with a detection time of 0.03597s per frame, a reduction of 0.00907s per frame compared to the original model. | [25] |
| | | Winter Jujube | CSPDarknet<br>Residual module combined with CSPNet | In complex natural environments with varying lighting and occlusion conditions, the mAP is 92.2%, an improvement of 3% over the original model. The parameter count is approximately 2.078M, an increase of about 0.318M, and the detection time is 0.0134s per frame, an increase of 0.00257s per frame. | [74] |
| | | Guava | CSPDarknet<br>Ghost | Under different lighting conditions, shooting distances, and backgrounds, the AP is 92.3%, the parameter count is 6.2M, which is 11.4% less than the original model, and the average detection time is 0.025s per frame. | [75] |
| | | Cherry tomato | CSPDarknet53<br>①Ghost<br>②DSC | In multi-classification tasks under different lighting conditions, the fruit's mAP is 99.2%, the model parameter count is approximately 2.445M, which is 65.25 less compared to a standard model, and the detection time on GPU is 0.0045s/frame. | [76] |
| | | Orange fruit | CSPDarknet53<br>GhostConv | Under different colors, distribution densities, and lighting intensities, the mAP0.5 is 91.4%, the model size is 46.9MB, and the detection speed is approximately 0.0266s per frame. | [51] |
| | | Grape | CSPDarknet<br>PConv | For the high-density complex background dataset, the mAP is 91.9%, an improvement of 5.4% compared to the original model, with a parameter count of 5.34M, reduced by approximately 24%, and a detection time of 0.0251s per frame. | [54] |

| | | | | | |
|---|---|---|---|---|---|
| | | Camellia fruit | MobileNetv3<br>DSC | For various actual greenhouse datasets in intensive agricultural environments, AP0.5 is 99.74%, an improvement of 8.29% compared to the original model. The parameter count is 12.027M, a reduction of 81.33%, and the detection time is 0.00301s per frame, a decrease of 34.85%. | [77] |
| | MobileNet | Bayberry | MobileNetv2<br>DSC | For various natural and real orchard environments, the AP is 97.24%, the model size is 46.5MB, which is 197.5MB smaller than the original model, and the detection time is 0.01872s/frame, reduced by 0.01139s/frame. | [78] |
| | | Dragon fruit | MobileNetv2<br>①Lightweight Inverted Residual Convolution<br>②Remove the small target detection layer | In natural orchard environments with varying lighting, occlusion levels, and distances, the mAP0.5 is 99.2%, the model size is 6.01MB, a 57% reduction compared to the standard model, and the detection time is 0.0111s per frame, a 21.7% improvement. | [79] |
| | Ghostnet | Winter jujube | SE-CSPGhostnet<br>Ghost | In environments with varying lighting, occlusion levels, and fruit sparsity for dense and occlusion detection, the mAP is 96.87%, higher than YOLOv4, v5s, and v5x, with a parameter count of 11.003M, smaller than YOLOv4 and v5x, but larger than YOLOv5s by 2.939M, and a detection time of 0.0179s per frame. | [80] |
| | | Apple | GhostNetV1<br>①Ghost<br>②Separable Convolution<br>③Residual Edge | For the complex spatial relationship between the target and the background, the mAP0.5-0.95 is 98.94%, the number of parameters is about 15.504M, which is reduced by about 75.8% compared to the ordinary model, and the detection time is 0.027s/frame. | [81] |
| Knowledge Distillation | ShuffleNet | Grape | ShuffleNetV2<br>①Slim-Neck<br>②Soft Label Loss (Lsoft) weighted combination Hard Label Loss (Lhard) | In various lighting conditions, single targets, multiple targets, occluded targets, damaged fruits, and other orchard environments, the mAP0.5-0.95 is 90.8%, an improvement of 1.4% over the original model. The parameter count is 0.79M, reduced by 88.77%, and the detection time is 0.0091s per frame, reduced by 0.0003s per frame. | [82] |
| | Resnet | Apple | Resnet50<br>KL divergence loss | On the GreenApple dataset, the AP is 62.3%, and the small target APS is 47.0%; on the MinneApple dataset, the AP is 43.5%, and the small target APS is 42.2%; on the Pascal VOC dataset, the AP is 51.4%, and the small target APS is 35.6%. The parameter count is 42.55M, which is an increase of 1.43M compared to the baseline model. | [22] |

### 1) APPLICATION OF PRUNING IN RS DETECTION

In real-time fruit detection scenarios, researchers have applied pruning concepts to backbone networks such as Darknet, MobileNet, and GhostNet, employing various strategies such as replacing traditional convolutional modules, removing redundant channels, and introducing lightweight modules. These efforts have successfully reduced the computational load and parameter count of the models while maintaining or minimally affecting detection accuracy.

For example, Huang T et al.[70] used the CSPDarknet network in citrus detection, applied L2 regularization to constrain the batch normalization layers, and removed redundant channels with smaller weights, successfully accelerating the inference speed but with a slight loss in accuracy. Yang W et al.[62] replaced the C3 module in YOLOv5 with the C3Ghost module in blueberry detection, reducing the number of model parameters but increasing the model's complexity. Specifically, analysis of the typical network MobileNetV3 reveals that it has significantly reduced computational complexity and storage requirements through innovative designs such as DSC, SE modules, and the Hard Swish activation function. However, in practical applications, the computational and storage requirements of the model may still exceed the hardware resource limitations. Therefore, pruning methods, as an important model compression technique, have become key to further optimizing MobileNetV3's performance. For example, the introduction of deeply separable convolution of MobileNetV3 in PANet can improve the inference speed[82].

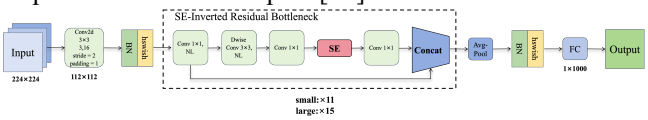


**FIGURE 8. MobileNetv3 structure.**

MobileNetV3 adopts a Depthwise Separable Convolutions structure similar to MobileNetV2 but optimizes the depth and width of the network. As shown in the Fig. 8, the main network consists of multiple Inverse Bottleneck Layers, each containing depthwise separable convolutions and a linear bottleneck. The highlights of MobileNetV3 are the introduction of the SE module and the Hard Swish activation function. The SE module enhances the model's ability to focus on important features by weighting each channel in the feature map; the Hard Swish activation function combines the advantages of ReLU and Swish, maintaining computational efficiency while providing better nonlinear expression capabilities.

### 2) APPLICATION OF KNOWLEDGE DISTILLATION IN RS DETECTION

In RS, knowledge distillation technology transfers knowledge from large teacher networks to lightweight student networks, improving detection performance while maintaining the model's lightweight and real-time characteristics. For example, Sun M et al.[22] achieved knowledge transfer from the teacher model to the student model in apple detection by introducing a KL scatter loss in the BFP Net of ResNet50, optimizing the feature representation without increasing the amount of significant computation.

The analysis of the typical network ShuffleNetV2 reveals that it significantly reduces computational complexity and model size through techniques such as grouped convolution, channel shuffling, and inverted residual blocks. However, the performance of ShuffleNetV2 may still be insufficient for high-precision detection in real-time lightweight scene detection tasks. For this reason, knowledge distillation methods are introduced to further optimize the performance of ShuffleNetV2[83]. For example, in jujube detection Feng J et al.[81] used YOLOv5m as the teacher network and ShuffleNetV2 as the backbone of the student network to improve the accuracy and generalization ability while maintaining the size and parameters of the student network.

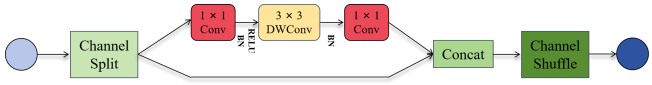


**FIGURE 9. ShuffleNetV2 structure.**

ShuffleNetV2 starts with feature extraction, using a convolutional layer to decompose the input RGB image into multiple feature maps. The main network consists of multiple ShuffleNet units, where Fig. 9 represents a unit containing Channel Split, 3x3 depth separable convolution, batch normalization of the convolved features, application of ReLU activation function and finally Channel Shuffle.

## V. CONCLUSION AND FUTURE WORK

This paper provides a comprehensive review of the research advancements in deep learning-based fruit detection, with a particular focus on the challenges and solutions across four core scenarios: SS, CS, SO, and RS detection. For the first time, we systematically deconstruct the fruit detection problem into these four categories, offering a detailed analysis of the key challenges and the corresponding technological breakthroughs for each scenario. Through an extensive review and synthesis of existing research, this paper presents enhanced methodologies for each scenario, offering valuable theoretical guidance for the continued optimization of fruit detection technologies.

Looking ahead, two major trends are anticipated to shape the future of fruit detection systems. First, fruit detection will evolve towards multifunctional integration, developing unified detectors capable of detecting multiple fruit types while also integrating functionalities such as

maturity assessment, fruit counting, disease monitoring, and robotic harvesting. Second, future systems are expected to increasingly rely on the fusion of multimodal information, including RGB images, depth data, and thermal imaging, as well as spatiotemporal features. This fusion will significantly enhance the robustness of detection models, particularly in challenging environments with complex backgrounds and small target detection.

## REFERENCES


[1] X. Lv, X. Zhang, H. Gao, T. He, Z. Lv, Z. Zhang, and L. Lili, "When crops meet machine vision: A review and development framework for a low-cost nondestructive online monitoring technology in agricultural production," *Agric. Commun.*, vol. 2, no. 1, 2024, Art. no. 100029.

[2] M. Gemtou, B. C. Guillén, and E. Anastasiou, "Smart Farming Technologies and Sustainability," in *Palgrave Studies in Digital Business and Enabling Technologies*, 2024, pp. 120-139. DOI: 10.1007/978-3-031-61749-2_6.

[3] D. Surya Prabha and J. Satheesh Kumar, "Sequential hybridization of genetic algorithm and fuzzy logic for enhanced edge detection of banana," *Int. J. Control Theory Appl.*, vol. 9, no. 10, pp. 4733–4745, 2016.

[4] S. Jana and R. Parekh, "Shape-based fruit recognition and classification," in *Computational Intelligence, Communications, and Business Analytics*, J. Mandal, P. Dutta, and S. Mukhopadhyay, Eds. Singapore: Springer, 2017, pp. 157–166. [Online]. Available: [https://doi.org/10.1007/978-981-10-6430-2_15](https://doi.org/10.1007/978-981-10-6430-2_15)

[5] H. Li, M. Zhang, Y. Gao, M. Li, and Y. Ji, "Green ripe tomato detection method based on machine vision in greenhouse," *Trans. Chinese Soc. Agric. Eng.*, vol. 33, no. z1, pp. 33328–334, 2017. DOI: 10.11975/j.issn.1002-6819.2017.z1.049.

[6] Y. He, Y. Yang, W. Sun, and W. Wang, "Research on automation of agricultural machinery based on computer vision identification technology," *J. Adv. Oxid. Technol.*, vol. 21, no. 2, 2018. DOI: 10.26802/jaots.2018.12742.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137-1149, Jun. 2017.

[8] S. Bhumla and D. K. Gupta, "A Review: Object Detection Algorithms," in Proc. 2023 Third Int. Conf. Secure Cyber Comput. Commun. (ICSCCC), May 2023, pp. 827-832.

[9] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, pp. 9627–9636.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in Proc. 16th European Conference on Computer Vision (ECCV), 2020, pp. 213–229. [Online]. Available: https://doi.org/10.1007/978-3-030-58452-8_13

[11] Y. Shi, D. Shi, Z. Qiao, Y. Zhang, Y. Liu, and S. Yang, "A Survey on Recent Advances in Few-Shot Object Detection," *Chinese Journal of Computers*, vol. 46, no. 8, pp. 1753–1780, Aug. 2023. DOI: 10.11897/SP.J.1016.2023.01753 (in Chinese with English abstract).

[12] J. W. Kim and M. Lee, "A Real-time Citrus Segmentation and Detection System using Mask R-CNN," *dcs*, vol. 19, no. 12, pp. 2385–2391, Dec. 2018, doi: 10.9728/dcs.2018.19.12.2385.

[13] R. P. Devanna, A. Milella, R. Marani, S. P. Garofalo, G. A. Vivaldi, S. Pascuzzi, R. Galati, and G. Reina, "In-Field Automatic Identification of Pomegranates Using a Farmer Robot," *Sensors*, vol. 22, no. 15, p. 5821, 2022. DOI: 10.3390/s22155821.

[14] R. Johanson, C. Wilms, O. Johannsen, and S. Frintrop, "S 3 AD: Semi-supervised Small Apple Detection in Orchard Environments," *in 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA: IEEE, Jan. 2024, pp. 7061–7070. doi: 10.1109/WACV57701.2024.00692.

[15] T. A. Ciarfuglia, I. M. Motoi, L. Saraceni, M. Fawakherji, A. Sanfeliu, and D. Nardi, "Weakly and semi-supervised detection, segmentation and tracking of table grapes with limited and noisy data," *Comput. Electron. Agric.*, vol. 205, 2023, Art. no. 107624.

[16] X. Zhang and L. Li, "Research of image recognition of camellia oleifera fruit based on improved convolutional auto-encoder," journal of Forestry Engineering, vol. 4, no. 3, pp. 118–124, 2019, doi: 10.13360/j.issn.2096-1359.2019.03.018 (in Chinese with English abstract).

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[18] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in Proc. 21st Int. Conf. Artif. Neural Networks (ICANN), Espoo, Finland, June 2011, pp. 52–59.

[19] J. Ruan, H. Cui, Y. Huang, T. Li, C. Wu, and K. Zhang, "A review of occluded objects detection in real CS for autonomous driving," *Green Energy and Intelligent Transportation*, vol. 2, no. 3, pp. 65–77, 2023. DOI: 10.1016/j.geits.2023.100092

[20] O. M. Lawal, "YOLOMuskmelon: Quest for Fruit Detection Speed and Accuracy Using Deep Learning," IEEE Access, vol. 9, pp. 15221–15227, 2021, doi: 10.1109/ACCESS.2021.3053167.

[21] J. Sun, L. Qian, W. Zhu , X. Zhou, C. Dai, and X. Wu, "Apple detection in complex orchard environment based on improved RetinaNet," *Trans. Chinese Soc. Agric. Eng.*, vol. 38, no. 15, pp. 314–322, 2022. DOI: 10.11975/j.issn.1002-6819.2022.15.034 (in Chinese with English abstract).

[22] M. Sun, L. Xu, X. Chen, Z. Ji, Y. Zheng, and W. Jia, "BFP Net: Balanced Feature Pyramid Network for Small Apple Detection in Complex Orchard Environment," *Plant Phenomics*, vol. 2022, p. 2022/9892464, Jan. 2022, doi: 10.34133/2022/9892464.

[23] B. He, Y. Zhang, J. Gong, G. Fu, Y. Zhao, and R. Wu, "Fast Recognition of Tomato Fruit in Greenhouses at Night Based on Improved YOLO v5," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 53, no. 5, pp. 201–208, 2022. DOI: 10.6041/j.issn.1000-1298.2022.05.020 (in Chinese with English abstract).

[24] A. Wang, W. Qian, A. Li, Y. Xu, J. Hu, Y. Xie, and L. Zhang, "NVW-YOLOv8s: An improved YOLOv8s network for real-time detection and segmentation of tomato fruits at different ripeness stages," *Computers and Electronics in Agriculture*, vol. 219, p. 108833, Apr. 2024, doi: 10.1016/j.compag.2024.108833.

[25] H. Zhao, Y. Qiao, H. Wang, and Y. Yue, "Apple fruit recognition in complex orchard environment based on improved YOLOv3," *Trans. Chinese Soc. Agric. Eng.*, vol. 37, no. 16, pp. 127–135, 2021. DOI: 10.11975/j.issn.1002-6819.2021.16.016 (in Chinese with English abstract).

[26] Y. Long, N. Li, Y. Gao, M. He, and H. Song, "Apple fruit detection under natural condition using improved FCOS network," *Trans. Chinese Soc. Agric. Eng.*, vol. 37, no. 12, pp. 307–313, 2021. DOI: 10.11975/j.issn.1002-6819.2021.12.035 (in Chinese with English abstract).

[27] Z. Zhang, J. Zhou, Z. Jiang, and H. Han, "Lightweight Apple Recognition Method in Natural Orchard Environment Based on Improved YOLO v7 Model," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 55, no. 3, pp. 231–242, 262, Mar. 2024. DOI: 10.6041/j.issn.1000-1298.2024.03.023 (in Chinese with English abstract).

[28] S. Kaukab, K. Komal, B. M. Ghodki, H. Ray, Y. B. Kalnar, K. Narsaiah, and J. S. Brar, "Improving real-time apple fruit detection: Multi-modal data and depth fusion with non-targeted background removal," *Ecological Informatics*, vol. 82, p. 102691, Sep. 2024. DOI: 10.1016/j.ecoinf.2024.102691.

[29] H. Song, Y. Wang, Y. Wang, S. Lv, and M. Jiang, "Camellia oleifera Fruit Detection in Natural Scene Based on YOLO v5s," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 53, no. 7, pp. 234–242, Jul. 2022. DOI: 10.6041/j.issn.1000-1298.2022.07.024 (in Chinese with English abstract).

[30] X. Zhu, F. Chen, Y. Zheng, X. Peng, and C. Chen, "An efficient method for detecting Camellia oleifera fruit under complex orchard environment," *Scientia Horticulturae*, vol. 330, p. 113091, Apr. 2024, doi: 10.1016/j.scienta.2024.113091.

[31] J. Zhou, W. Hu, A. Zou, S. Zhai, T. Liu, W. Yang, and P. Jiang, "Lightweight Detection Algorithm of Kiwifruit Based on Improved YOLOX-S," *Agriculture*, vol. 12, no. 7, p. 993, Jul. 2022. DOI: 10.3390/agriculture12070993.

[32] J. Xie, J. Peng, J. Wang, B. Chen, T. Jing, D. Sun, P. Gao, W. Wang, J. Lu, R. Yetan, et al., "Litchi Detection in a Complex Natural Environment Using the YOLOv5-Litchi Model," *Agronomy*, vol. 12, no. 12, p. 3054, Dec. 2022, doi: 10.3390/agronomy12123054.

[33] Z. Xiong, L. Wang, Y. Zhao, and Y. Lan, "Precision Detection of Dense Litchi Fruit in UAV Images Based on Improved YOLOv5

Model," *Remote Sensing*, vol. 15, no. 16, p. 4017, Aug. 2023, doi: 10.3390/rs15164017.

[34] X. Liu, G. Li, W. Chen, B. Liu, M. Chen, and S. Lu, "Detection of Dense Citrus Fruits by Combining Coordinated Attention and Cross-Scale Connection with Weighted Feature Fusion," *Applied Sciences*, vol. 12, no. 13, p. 6600, Jun. 2022, doi: 10.3390/app12136600.

[35] H. Sun, B. Wang, and J. Xue, "YOLO-P: An efficient method for pear fast detection in complex orchard picking environment," *Front. Plant Sci.*, vol. 13, p. 1089454, Jan. 2023, doi: 10.3389/fpls.2022.1089454.

[36] P. Liu and H. Yin, "YOLOv7-Peach: An Algorithm for Immature Small Yellow Peaches Detection in Complex Natural Environments," *Sensors*, vol. 23, no. 11, p. 5096, May 2023, doi: 10.3390/s23115096.

[37] W. Du, Y. Zhu, S. Li, and P. Liu, "Spikelets detection of table grape before thinning based on improved YOLOV5s and Kmeans under the complex environment," *Computers and Electronics in Agriculture*, vol. 203, p. 107432, Dec. 2022, doi: 10.1016/j.compag.2022.107432.

[38] C. Guo, S. Zheng, G. Cheng, Y. Zhang, and J. Ding, "An improved YOLO v4 used for grape detection in unstructured environment," *Front. Plant Sci.*, vol. 14, p. 1209910, Jul. 2023, doi: 10.3389/fpls.2023.1209910.

[39] S. Lü, S. Lu, Z. Li, T. Hong, Y. Xue, and B. Wu, "Orange recognition method using improved YOLOv3-LITE lightweight neural network," *Trans. Chinese Soc. Agric. Eng.*, vol. 35, no. 17, pp. 205–214, 2019. DOI: 10.11975/j.issn.1002-6819.2019.17.025 (in Chinese with English abstract).

[40] J. Huang, X. Zhao, F. Gao, X. Wen, S. Jin, and Y. Zhang, "Recognizing and detecting the strawberry at multi-stages using improved lightweight YOLOv5s," *Trans. Chinese Soc. Agric. Eng.*, vol. 39, no. 21, pp. 181–187, 2023. DOI: 10.11975/j.issn.1002-6819.202307186 (in Chinese with English abstract).

[41] C. Guo, C. Zhu, Y. Liu, R. Huang, B. Cao, Q. Zhu, R. Zhang, and B. Zhang, "End-to-End lightweight Transformer-Based neural network for grasp detection towards fruit robotic handling," *Comput. Electron. Agric.*, vol. 221, p. 109014, Jun. 2024. DOI: 10.1016/j.compag.2024.109014.

[42] H. Li, C. Li, G. Li, and L. Chen, "A real-time table grape detection method based on improved YOLOv4-tiny network in complex background," *Biosystems Engineering*, vol. 212, pp. 347–359, Dec. 2021, doi: 10.1016/j.biosystemseng.2021.11.011.

[43] H. Wang, J. Feng, and H. Yin, "Improved Method for Apple Fruit Target Detection Based on YOLOv5s," *Agriculture*, vol. 13, no. 11, p. 2167, Nov. 2023, doi: 10.3390/agriculture13112167.

[44] Y. Xu, M. Jiang, Y. Li, Y. Wu, and G. Lu, "Fruit target detection based on improved YOLO and NMS," *J. Electron. Meas. Instrumentation*, vol. 36, no. 4, pp. 114–123, 2022. DOI: 10.13382/j.jemi.B2104724 (in Chinese with English abstract).

[45] R. Gai, M. Li, Z. Wang, L. Hu, and X. Li, "YOLOv5s-Cherry: Cherry Target Detection in Dense Scenes Based on Improved YOLOv5s Algorithm," *J CIRCUIT SYST COMP*, vol. 32, no. 12, p. 2350206, Aug. 2023, doi: 10.1142/S0218126623502067.

[46] Y. Zhang, L. Zhang, H. Yu, Z. Guo, R. Zhang, and X. Zhou, "Research on the Strawberry Recognition Algorithm Based on Deep Learning," *Applied Sciences*, vol. 13, no. 20, p. 11298, Oct. 2023, doi: 10.3390/app132011298.

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520.

[48] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[49] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, and J. Han, "Towards Large-Scale Small Object Detection: Survey and Benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13467–13488, Nov. 2023. DOI: 10.1109/TPAMI.2023.3290594.

[50] L. Ma, L. Zhao, Z. Wang, J. Zhang, and G. Chen, "Detection and Counting of Small Target Apples under Complicated Environments by Using Improved YOLOv7-tiny," *Agronomy*, vol. 13, no. 5, p. 1419, May 2023, doi: 10.3390/agronomy13051419.

[51] A. Zhu, R. Zhang, L. Zhang, T. Yi, L. Wang, D. Zhang, L. Chen, "YOLOv5s-CEDB: A robust and efficiency Camellia oleifera fruit detection algorithm in complex natural scenes," *Computers and Electronics in Agriculture*, vol. 221, p. 108984, Jun. 2024, doi: 10.1016/j.compag.2024.108984.

[52] Y. Shi, J. Li, P. Zhang, and D. Wang, "Detecting and counting of spring-see citrus using YOLOv4 network model and recursive fusion of features," *Trans. Chinese Soc. Agric. Eng.*, vol. 37, no. 18, pp. 161–169, 2021. DOI: 10.11975/j.issn.1002-6819.2021.18.019 (in Chinese with English abstract).

[53] J. Xiong, Z. Huo, Q. Huang, H. Chen, Z. Yang, Y. Huang, and Y. Su, "Detection method of citrus in nighttime environment combined with active light source and improved YOLOv5s model," *Journal of South China Agricultural University*, vol. 45, no. 1, pp. 97–107, 2024. DOI: 10.7671/j.issn.1001-411X.202209010 (in Chinese with English abstract).

[54] C. Yang, J. Liu, and J. He, "A lightweight waxberry fruit detection model based on YOLOv5," *IET Image Processing*, vol. 18, no. 7, pp. 1796–1808, May 2024, doi: 10.1049/ipr2.13064.

[55] Z. He, S. R. Khanal, X. Zhang, M. Karkee, and Q. Zhang, "Real-time Strawberry Detection Based on Improved YOLOv5s Architecture for Robotic Harvesting in open-field environment," Oct. 12, 2023, *arXiv*: arXiv:2308.03998. Accessed: Sep. 18, 2024. [Online]. Available: http://arxiv.org/abs/2308.03998

[56] C. Zhang, F. Kang, and Y. Wang, "An Improved Apple Object Detection Method Based on Lightweight YOLOv4 in Complex Backgrounds," *Remote Sensing*, vol. 14, no. 17, p. 4150, Aug. 2022, doi: 10.3390/rs14174150.

J. Sun, Y. Chen, X. Zhou, J. Shen, and X. Wu, "Fast and accurate recognition of the strawberries in greenhouse based on improved YOLOv4-Tiny model," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 38, no. 18, pp. 195–203, 2022. DOI: 10.11975/j.issn.1002-6819.2022.18.021 (in Chinese with English abstract).

[57] S. Lyu, R. Li, Y. Zhao, Z. Li, R. Fan, and S. Liu, "Green Citrus Detection and Counting in Orchards Based on YOLOv5-CS and AI Edge System".

[58] J. Yang, Z. Qian, Y. Zhang, Y. Qin, and H. Miao, "Real-time recognition of tomatoes in complex environments based on improved YOLOv4-tiny," *Trans. Chinese Soc. Agric. Eng.*, vol. 38, no. 9, pp. 215–221, 2022. DOI: 10.11975/j.issn.1002-6819.2022.09.023 (in Chinese with English abstract).

[59] Q. Luo, C. Wu, G. Wu, and W. Li, "A Small Target Strawberry Recognition Method Based on Improved YOLOv8n Model," *IEEE Access*, vol. 12, pp. 14987–14995, 2024, doi: 10.1109/ACCESS.2024.3356869.

[60] Y. Bai, J. Yu, S. Yang, and J. Ning, "An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings," *Biosystems Engineering*, vol. 237, pp. 1–12, Jan. 2024, doi: 10.1016/j.biosystemseng.2023.11.008.

[61] H. Song, M. Jiang, Y. Wang, and L. Song, "Efficient detection method for young apples based on the fusion of convolutional neural network and visual attention mechanism," Transactions of the Chinese Society of Agricultural Engineering, vol. 37, no. 9, pp. 297–303, 2021. DOI: 10.11975/j.issn.1002-6819.2021.09.034 (in Chinese with English abstract).

[62] W. Yang, X. Ma, W. Hu, and Pengjie Tang, "Lightweight Blueberry Fruit Recognition Based on Multi-Scale and Attention Fusion NCBAM," *Agronomy*, vol. 12, no. 10, p. 2354, Sep. 2022, doi: 10.3390/agronomy12102354.

[63] C. Li, J. Lin, Z. Li, C. Mai, R. Jiang, and J. Li, "An efficient detection method for litchi fruits in a natural environment based on improved YOLOv7-Litchi," *Computers and Electronics in Agriculture*, vol. 217, p. 108605, Feb. 2024, doi: 10.1016/j.compag.2023.108605.

[64] P. Zhou, Y. Pei, R. Wei, Y. Zhang, and Y. Gu, "Real-time detection of orchard cherry based on YOLOV4 model," *Acta Agriculturae Zhejiangensis*, vol. 34, no. 11, pp. 2522–2532, 2022. DOI: 10.3969/j.issn.1004-1524.2022.11.021.

[65] F. Meng, J. Li, Y. Zhang, S. Qi, and Y. Tang, "Transforming unmanned pineapple picking with spatio-temporal convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 214, p. 108298, Nov. 2023, doi: 10.1016/j.compag.2023.108298.

[66] H. Liu, X. Wang, F. Zhao, F. Yu, P. Lin, Y. Gan, X. Ren, Y. Chen, and J. Tu, "Upgrading Swin-B Transformer-based model for accurately identifying ripe strawberries by coupling task-aligned

one-stage object detection mechanism," *Comput. Electron. Agric.*, vol. 218, p. 108674, Mar. 2024. DOI: 10.1016/j.compag.2024.108674.
[67] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 1580–1589.
[68] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 10012-10022).
[69] Z. Cao, L. Kooistra, W. Wang, L. Guo, and J. Valente, "Real-Time Object Detection Based on UAV Remote Sensing: A Systematic Literature Review," Drones, vol. 7, no. 10, p. 620, Oct. 2023. DOI: 10.3390/drones7100620.
[70] H. Huang, T. Huang, Z. Li, S. Lyu, and T. Hong, "Design of Citrus Fruit Detection System Based on Mobile Platform and Edge Computer Device," *Sensors*, vol. 22, no. 1, p. 59, Dec. 2021, doi: 10.3390/s22010059.
[71] G. Hu, J. Zhou, C. Chen, C. Li, L. Sun, Y. Chen, S. Zhang, and J. Chen, "Fusion of the lightweight network and visual attention mechanism to detect apples in orchard environment," *Trans. Chinese Soc. Agric. Eng.*, vol. 38, no. 19, pp. 131–142, 2022. DOI: 10.11975/j.issn.1002-6819.2022.19.015 (in Chinese with English abstract).
[72] B. Zhao, A. Guo, R. Ma, Y. Zhang, and J. Gong, "YOLOv8s-CFB: A lightweight method for real-time detection of apple fruits in complex environments," *J. Real-Time Image Process.*, vol. 21, no. 5, p. 164, 2024.
[73] C. Yu, J. Feng, Z. Zheng, J. Guo, and Y. Hu, "A lightweight SOD-YOLOv5n model-based winter jujube detection and counting method deployed on Android," *Computers and Electronics in Agriculture*, vol. 218, p. 108701, Mar. 2024, doi: 10.1016/j.compag.2024.108701.
[74] L. Wang, H. Zheng, C. Yin, Y. Wang, Z. Bai, and W. Fu, "Dense Papaya Target Detection in Natural Environment Based on Improved YOLOv5s," *Agronomy*, vol. 13, no. 8, p. 2019, Jul. 2023, doi: 10.3390/agronomy13082019.
[75] J. Zhao, X. Yao, Y. Wang, Z. Yi, Y. Xie, and X. Zhou, "Lightweight-Improved YOLOv5s Model for Grape Fruit and Stem Recognition," *Agriculture*, vol. 14, no. 5, p. 774, May 2024, doi: 10.3390/agriculture14050774.
[76] F. Zhang, Z. Chen, R. Bao, Z. Zhang, and Z. Wang, "Recognition of dense cherry tomatoes based on improved YOLOv4-LITE lightweight neural network," *Trans. Chinese Soc. Agric. Eng.*, vol. 37, no. 16, pp. 270–278, 2021. DOI: 10.11975/j.issn.1002-6819.2021.16.033 (in Chinese with English abstract).
[77] J. Liu, Y. Li, L. Xiao, W. Li, and H. Li, "Recognition and location method of orange based on improved YOLOv4 model," *Trans. Chinese Soc. Agric. Eng.*, vol. 38, no. 12, pp. 173–182, 2022. DOI: 10.11975/j.issn.1002-6819.2022.12.020 (in Chinese with English abstract).
[78] J. Wang, Y. Su, J. Yao, M. Liu, Y. Du, X. Wu, L. Huang, and M. Zhao, "Apple rapid recognition and processing method based on an improved version of YOLOv5," *Ecological Informatics*, vol. 77, p. 102196, Nov. 2023, doi: 10.1016/j.ecoinf.2023.102196.
[79] J. Chen, A. Ma, L. Huang, Y. Su, W. Li, H. Zhang, and Z. Wang, "GA-YOLO: A Lightweight YOLO Model for Dense and Occluded Grape Target Detection," *Horticulturae*, vol. 9, no. 4, p. 443, Mar. 2023, doi: 10.3390/horticulturae9040443.
[80] F. Zhang, W. Cao, S. Wang, X. Cui, N. Yang, X. Wang, X. Zhang, and S. Fu, "Improved YOLOv4 recognition algorithm for pitaya based on coordinate attention and combinational convolution," *Front. Plant Sci.*, vol. 13, p. 1030021, Oct. 2022, doi: 10.3389/fpls.2022.1030021.
[81] J. Feng, C. Yu, X. Shi, Z. Zheng, L. Yang, and Y. Hu, "Research on Winter Jujube Object Detection Based on Optimized Yolov5s," *Agronomy*, vol. 13, no. 3, p. 810, Mar. 2023, doi: 10.3390/agronomy13030810.
[82] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, et al., "Searching for MobileNetV3," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, pp. 1314–1324.
[83] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 116–131.

**XINYU GONG** was born in Guangzhou, Guangdong Province, China in 2004. She is currently pursuing the bachelor's degree in Data Science and Big Data at the College of Arts and Sciences, Northeast Agricultural University. Her research interests include computer vision and deep learning.

**Qiufeng Wu** received B. S degree in Mathematics and Applied Mathematics from Harbin Normal University in 2002. He received M.S. degree in Management Science and Engineering from Northeast Agricultural University in 2007. He received Ph.D in computer application technology from Harbin Institute of Technology, China, in 2014. He is working as associated professor in College of Arts and Sciences in Northeast Agricultural University. His research interests include machine learning, computer vision and smart agriculture.