agriculture

MDPI

Article

Light-YOLO: A Lightweight and Efficient YOLO-Based Deep Learning Model for Mango Detection

Zhengyang Zhong [1,2], Lijun Yun [1,2,*], Feiyan Cheng [1,2], Zaiqing Chen [1,2] and Chunjie Zhang [1,2]

[1] College of Information, Yunnan Normal University, Kunming 650500, China; 2124100041@ynnu.edu.cn (Z.Z.); chengfy@ynnu.edu.cn (F.C.); zaiqingchen@gmail.com (Z.C.); 4486@ynnu.edu.cn (C.Z.)
[2] Engineering Research Center of Computer Vision and Intelligent Control Technology, Department of Education of Yunnan Province, Kunming 650500, China
* Correspondence: yunlijun@ynnu.edu.cn; Tel.: +86-138-8836-6965

Abstract: This paper proposes a lightweight and efficient mango detection model named Light-YOLO based on the Darknet53 structure, aiming to rapidly and accurately detect mango fruits in natural environments, effectively mitigating instances of false or missed detection. We incorporate the bidirectional connection module and skip connection module into the Darknet53 structure and compressed the number of channels of the neck, which minimizes the number of parameters and FLOPs. Moreover, we integrate structural heavy parameter technology into C2f, redesign the Bottleneck based on the principles of the residual structure, and introduce an EMA attention mechanism to amplify the network's emphasis on pivotal features. Lastly, the Downsampling Block within the backbone network is modified, transitioning it from the CBS Block to a Multi-branch–Large-Kernel Downsampling Block. This modification aims to enhance the network's receptive field, thereby further improving its detection performance. Based on the experimental results, it achieves a noteworthy mAP of 64.0% and an impressive mAP0.5 of 96.1% on the ACFR Mango dataset with parameters and FLOPs at only 1.96 M and 3.65 G. In comparison to advanced target detection models like YOLOv5, YOLOv6, YOLOv7, and YOLOv8, it achieves improved detection outcomes while utilizing fewer parameters and FLOPs.

Keywords: mango; lightweight; Light-YOLO; computer vision; deep learning

Citation: Zhong, Z.; Yun, L.; Cheng, F.; Chen, Z.; Zhang, C. Light-YOLO: A Lightweight and Efficient YOLO-Based Deep Learning Model for Mango Detection. *Agriculture* **2024**, *14*, 140. https://doi.org/10.3390/agriculture14010140

Academic Editor: Jiangbo Li

Received: 21 December 2023
Revised: 13 January 2024
Accepted: 16 January 2024
Published: 18 January 2024

1. Introduction

Mango is a crucial economic crop in tropical regions and stands as one of the globally significant fruits, holding a prominent place in consumer markets. However, due to the intricate growth conditions and unique characteristics of mango fruits, the majority of mango harvesting still heavily relies on manual labor. This not only demands a substantial workforce but also results in relatively lower harvesting efficiency. Utilizing machine vision technology to analyze real-time images, the processing results are then supplied to a machine, thereby achieving automatic harvesting. This not only reduces manual labor but also significantly enhances harvesting efficiency. In this process, the accurate identification of mangoes is considered the primary task for automated harvesting, and its recognition precision will directly impact the effectiveness of the automated harvesting operation. Therefore, a high-precision and high-speed mango identification method is crucial for automated harvesting operations.

Early target detection algorithms predominantly relied on manual feature extraction, which, over time, exhibited inherent limitations, with the performance of manual feature extraction gradually reaching a point of saturation. The advent of convolutional neural networks marked a transformative phase in the realm of object detection algorithms. In 2014, the R-CNN [1] algorithm was introduced, heralding the inception of object detection algorithms rooted in deep learning, which subsequently experienced a notable surge

in development and application. A prevalent approach among early target detection algorithms entailed a two-stage detection methodology. The representative algorithms are Fast R-CNN [2] and Faster R-CNN [3]. In the initial phase, the foremost task is the generation of a regional candidate box, while in the subsequent stage, the features of each candidate box are extracted. Ultimately, a positional box is generated, and the corresponding category is predicted. The inception of the one-stage detection algorithm has ushered in a novel solution. Diverging from the two-stage detection algorithm, the one-stage detection algorithm concurrently generates candidate boxes, executing classification and boundary box regression simultaneously. This affords the one-stage approach a commendable detection speed, albeit with a modest trade-off in accuracy. Representative algorithms embodying this paradigm include the YOLO series [4–12] and SSD [13].

In light of the rapid advancements in new-generation information technologies, such as artificial intelligence, traditional agriculture is experiencing an accelerated transformation into smart agriculture. Machine-vision-based target detection technology fosters the high-quality advancement of agricultural production, including multiple aspects such as farming, harvesting, and pest control. The orchard environment is characterized by its inherent complexity. Owing to factors such as variations in light conditions and weather, fruits may exhibit hues resembling those of branches and leaves, thereby introducing challenges to the accurate recognition of the algorithm. To enhance the accuracy of algorithms and mitigate instances of missed and false detections, a range of solutions has been proposed in both the domestic and international literature. Wu et al. [14] present an enhanced YOLOv4 model and a data augmentation technique yielding an average accuracy of 98.15% in the context of Apple detection. In a similar vein, Yan et al. [15] introduce a lightweight method for detecting apples by leveraging an improved YOLOv5s architecture, achieving an average accuracy of 86.75%. Sun et al. [16] made significant enhancements to the backbone network within the YOLOv5 algorithm, substituting the initial activation function with Hard-Swish. This modification resulted in a notable increase in pear detection accuracy, achieving a precision level of 97.6%. Ren et al. [17] propose a recognition approach for Yuluxiang pears based on YOLOv8, integrating an EMA attention mechanism and enhancing the IOU loss function to improve target recognition and positioning accuracy. The corresponding F1 and average accuracy stand at 84.47% and 88.83%, respectively. Although deep-learning-based object detection algorithms have exhibited significant success in various fruit detection domains, there exists an opportunity for enhancement in the methodologies pertaining to mango detection. Stein et al. [18] employed the Faster R-CNN algorithm to identify mangoes, achieving an F1 score of 88.1. While the fundamental realization of mango detection has been accomplished, there remains a discernible opportunity for refining detection accuracy. In a similar vein, Li et al. [19] introduced an approach for detecting mature mangoes on trees using an enhanced version of YOLOv3, attaining an average accuracy of 94.91 with a model size of 238 MB. Despite the attainment of high accuracy, the impractical size of the model impedes its deployment in agricultural production processes. In another endeavor, Xu et al. [20] proposed a methodology for the swift identification of green mangoes in intricate scenes, leveraging YOLOv3 to achieve an F1 score of 97.7. Notwithstanding the commendable lightweight nature of the model, with FLOPs and model sizes of 10.12 G and 44 MB, respectively, there remains a need for further optimization, particularly in reducing the model size. The particulars of the aforenamed methods are presented in Table 1.

This paper presents a lightweight model designed for the rapid and accurate detection of mangoes in natural environments. The proposed approach involves the incorporation of the BiC module and SkC module to redesign the neck network. Drawing enlightenment from the methodology employed in YOLOv6, this paper opts to halve the number of channels in both the neck and the head while maintaining the channel count in the backbone. To enhance the network's feature extraction capabilities, we introduce structural reparameterization technology in the C2f Block. Through a comparative evaluation of various subsampling structures, we determine that replacing the convolutional Downsam-

pling Block in the trunk section with a Multi-branch–Large-Kernel Downsampling Block approach yields optimal detection performance. The residual EMA Bottleneck Block is conceived by amalgamating the residual structure with the EMA attention mechanism to compress the parameters and FLOPs of the Bottleneck. The proposed Light-YOLO algorithm demonstrates exceptional detection performance with minimal parameters and FLOPs. This design effectively addresses the demands of agricultural production and provides valuable technical support for real-time, accurate detection of multiple mango targets via mango-picking robots.

Table 1. The specific details of the methodologies cited in the references.

Authors	Fruit Type	Size	Main Model	F1 (%)	mAP0.5 (%)	Params (M)	FLOPs (G)	MS (MB)
Wu et al. [14]	Red Apple	416 × 416	YOLOv4	96.54	98.15	37.8	12.7	158
Yan et al. [15]	Red Apple	640 × 640	YOLOv5	87.49	86.75	6.52	-	12.7
Sun et al. [16]	Pear	640 × 640	YOLOv5	96.1	97.6	-	10.1	8.3
Ren et al. [17]	Yuluxiang Pear	640 × 640	YOLOv8	84.47	88.83	7.19	18.6	14.07
Stein et al. [18]	Mangoes (ACFR)	500 × 500	Faster R-CNN	88.1	-	-	-	-
Li et al. [19]	Mangoes	608 × 608	YOLOv3	-	94.91	-	-	238
Xu et al. [20]	Green Mangoes	416 × 416	YOLOv3	97.7	-	-	10.12	44

2. Materials and Methods

2.1. Experimental Data

In this study, the mango dataset from the ACFR orchard fruit dataset at the University of Sydney is chosen as the experimental dataset to assess the efficacy of the object detection algorithm proposed herein. The mango images in the dataset were collected from a mango orchard at Simpson Farm in Bundaberg, Queensland, Australia, totaling 1964 images. Specifically, the training set consists of 1464 images, the validation set includes 250 images, and the test set comprises 250 images. As illustrated in Figure 1, the captured images have a resolution of 500 × 500 pixels and are saved in PNG format. The annotation file includes coordinate information for the minimum external rectangular box enclosing each mango and is stored in CSV format. The data from the annotation file are processed, and the annotation information is formatted in accordance with YOLO's annotation format for subsequent training and testing.

Figure 1. Image example of ACFR Mango dataset.

2.2. Object Detection Model Based on Deep Learning

In order to meet the needs of agricultural production for detection speed and lightweight, we chose the YOLO algorithm with better real-time performance as the benchmark algorithm for this research. Over the past years, the YOLO series algorithms have experienced rapid development, contributing to continuous improvements in detection performance. However, this progress presents a challenge in the form of significantly increased parameters and FLOPs. For instance, the lightweight model within the YOLO

series can be considered: When utilizing a 512 × 512 pixels input feature graph size, the latest version of the YOLOv5-N model exhibits the parameters of 1.76 M and the FLOPs of 4.1 G. However, the latest models, including YOLOv6-N, YOLOv7-Tiny, and YOLOv8-N, within the same size range, demonstrate higher parameters and FLOPs, with values of 4.63 M and 7.26 G, 6.01 M and 13 G, and 3.01 M and 8.1 G, respectively. These observations reveal that the performance enhancements come at the expense of sacrificing parameters and FLOPs. Diverging from the laboratory setting, computational resources within the realm of agricultural production are constrained. In contrast to the standard target detection network, the lightweight and real-time target detection network aligns markedly better with the agricultural production requisites. Consequently, enhancing network detection performance without inflating the parameters and FLOPs has emerged as the focal research direction of this study.

Examining the iterative updates of the YOLO algorithm reveals that the enhancements in these algorithms primarily center around the following facets:

(1) To enhance the backbone: Joseph et al. [5,6] introduced the Darknet19 structure and the Darknet53 structure;
(2) In pursuit of superior multi-scale prediction structures: Joseph et al. [6] incorporated the Feature Pyramid Network (FPN) [21]. Alexey et al. [7] introduced the Path Aggregation Network (PANet) [22]. Furthermore, Chuyi et al. [10] introduced the RepBi-PAN;
(3) For a more efficient feature extraction structure: Alexey et al. [7] proposed a cross-stage–partial-connection (CSP) Block. Jocher et al. [8] introduced the C3 Block based on cross-stage partial connection. Chuyi et al. [10] put forward a CSPStackRep Block utilizing structural reparameterization technology. Additionally, Wang [11] and Jocher et al. [12], respectively, proposed the Extended-ELAN and C2f Block;
(4) Implementation of a more robust IOU loss function [23–28];
(5) Adoption of a superior activation function [29–31];
(6) Deployment of more effective label allocation strategies.

Throughout the experiment, my attention was not directed towards (1), (3), (5), or (6). This decision arose from the recognition that the existing structures and methods had reached a level of maturity and excellence deemed satisfactory. This paper primarily entails modifications to two parts, denoted as (2) and (3), in addition to a redesign of the Downsampling Block.

2.3. Attention Mechanism

The attention mechanism is a technique that emulates human visual and cognitive systems. Its incorporation into convolutional neural networks enables the automatic learning and selective concentration on crucial information within the input, subsequently enhancing the model's performance and generalization capabilities. Widely recognized attention mechanisms encompass the spatial attention mechanism [32,33], channel attention mechanism [34,35], and the mixed attention mechanism [36,37]. The spatial attention mechanism is designed to amplify the importance of specific target regions while concurrently attenuating the influence of irrelevant background regions. The channel attention mechanism autonomously discerns the significance of individual feature channels through network learning, thereby enhancing crucial features while suppressing less critical ones. The hybrid attention mechanism amalgamates spatial domain attention with channel attention. This study employs the Efficient Multi-Scale Attention Module with Cross-Spatial Learning (EMA) [38], whose architecture bears resemblance to the CA attention mechanism [39], as depicted in Figure 2. Structurally, when presented with a specific input feature graph, the EMA attention mechanism initially partitions it into G sub-feature graphs along the channel dimension to facilitate the learning of diverse semantic features. Diverging from the dual parallel subnetworks of the CA attention mechanism, the EMA attention mechanism integrates three parallel subnetworks. In contrast to the dual parallel subnetworks of the CA attention mechanism, the EMA attention mechanism employs three parallel subnet-

works. Specifically, the first two parallel subnetworks are situated within the 1 × 1 branch, while the third parallel subnetwork is positioned in the 3 × 3 branch. The methodology adopted in the first two parallel subnetworks closely resembles that of the CA attention mechanism, involving two one-dimensional average pooling operations, concatenation, and 1 × 1 convolution operations. Notably, channel dimension reduction is omitted as the feature grouping operation has been executed beforehand. The ensuing step involved the decomposition of the output results from the 1 × 1 convolution, followed by the application of the Sigmoid operation to yield two distinct channel attention graphs. Subsequently, the two channel attention graphs corresponding to each group were amalgamated through a multiplication process. An additional concurrent subnetwork employs a 3 × 3 convolution to capture local cross-channel interactions, effectively enlarging the feature space. Through the amalgamation of contextual information across two distinct scales, namely 1 × 1 and 3 × 3, convolutional neural networks can enhance pixel-level attention to high-level feature maps. Subsequently, two output feature maps are created through cross-space learning, and the feature maps within each group are computed as the summation of the generated spatial attention weights. Ultimately, the summation of spatial attention weights undergoes a Sigmoid operation, and the final output feature map is derived through multiplication.

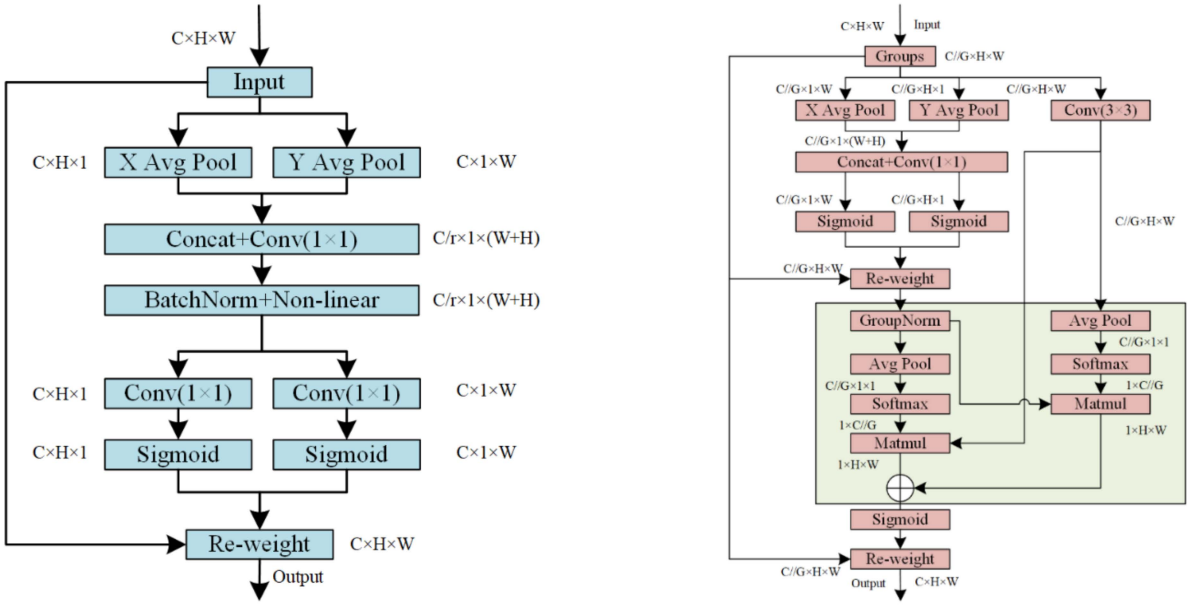


Figure 2. The structure of the CA attention mechanism and the EMA attention mechanism. (**a**) CA attention mechanism structure and (**b**) EMA attention mechanism structure.

2.4. Light-YOLO

Light-YOLO is devised based on the Darknet53 architecture, and the network structure is illustrated in Figure 3. The entire network is partitioned into three segments: the backbone, neck, and head. In the backbone, the overall structure adheres to the Darknet53 design, with the Focus Block replaced with a 3 × 3 CBS Block with a stride of 2. In the neck, we incorporate the BiC and SkC modules to facilitate the integration of features within the feature layer and enhance the localization accuracy of small targets. Simultaneously, we decrease the number of neck channels by half to mitigate computational demands. In the head network, we opt for an anchor-free frame, forsaking the anchor-based frame, while incorporating minor structural adjustments. Additionally, the CSP Block is replaced with the C2f Block, and a Multi-branch–Large-Kernel Downsampling Block is introduced.

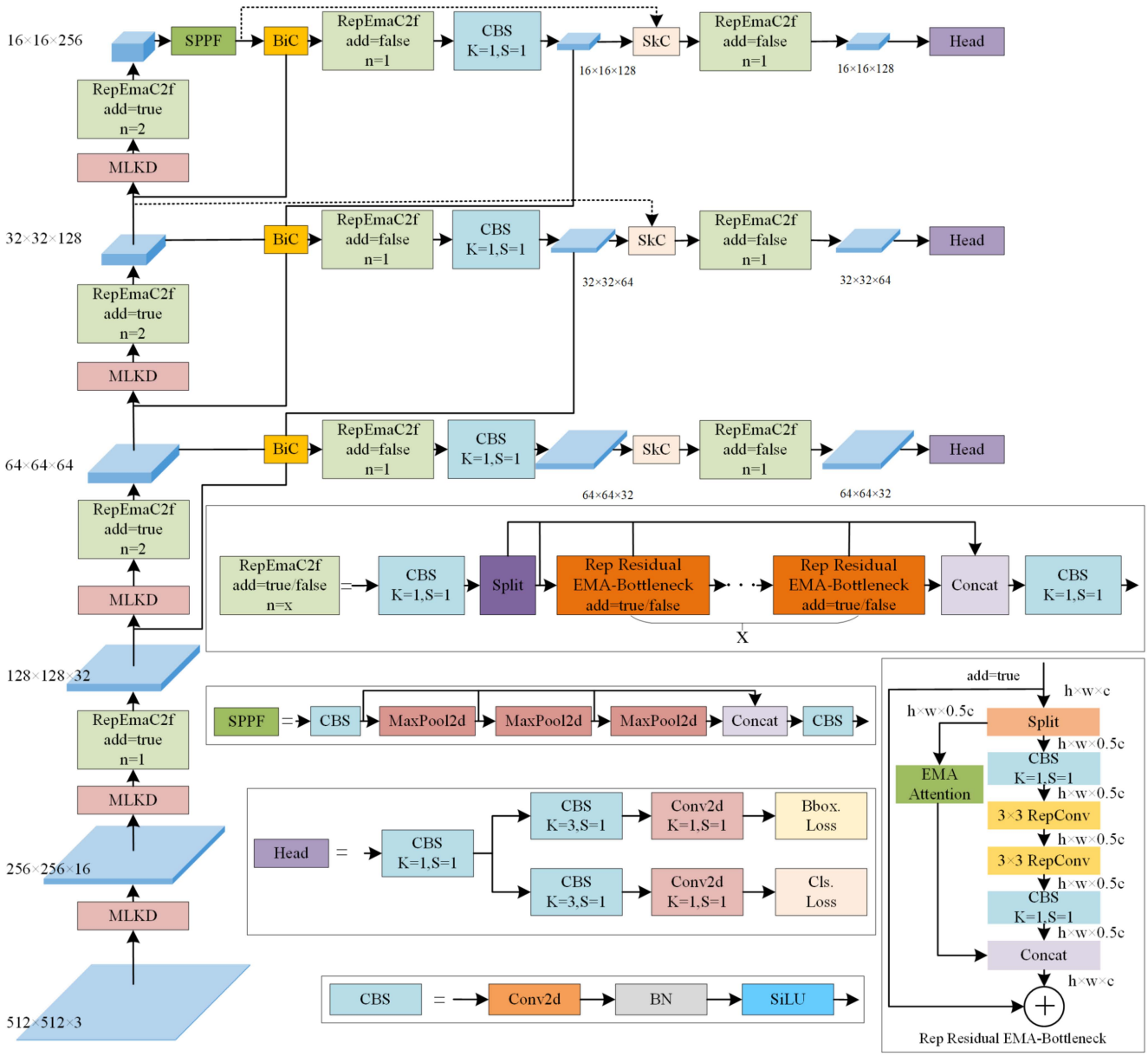


Figure 3. Light-YOLO model structure.

2.4.1. BiSC-PAN

As the intermediary segment of both the backbone and the head, the neck network bears the crucial task of amalgamating features with disparate resolutions from the backbone and augmenting the expressive capacity of output features. Its significance within the network architecture has been substantiated, potentially surpassing even the importance of the backbone [40–42]. The Feature Pyramid Network achieves multi-scale feature fusion through the incorporation of a top-down pathway. In the Path Aggregation Network, a bottom-up pathway is introduced in addition to the top-down pathway, effectively curtailing the information transmission distance between lower and upper-level features. This adjustment facilitates a smoother transmission of bottom-level information to upper-level features. Bidirectional Cross-Scale Connections and Weighted Feature Fusion (BIFPN) eliminate singular-output edge nodes that provide minimal contributions to fusion within the framework of the path aggregation network. Additionally, a jump link is introduced between the original input and output nodes at the same level. In a similar vein, RepBi-PAN introduces a bidirectional connection (BiC) module aiming to integrate feature

maps from adjacent layers. This approach enhances the accurate retention of positioning signals, thereby improving the precision of localizing small targets.

Building upon the aforenamed research, this paper introduces BiSC-PAN as a neck rooted in the path aggregation network and BiC module. Fruits, constituting small targets within the image, occupy a limited number of pixels. As downsampling operations increase, the feature information associated with these small targets in the high-level feature map undergoes a progressive diminution. In divergence from the C3 and C4 feature layers, the C5 feature layer has undergone a greater number of downsampling operations, thereby exacerbating the loss of features associated with small targets. To enhance the preservation of positioning signals within the high-level feature map, this study not only employs the bidirectional connection module on the C3 and C4 feature layers but extends its implementation to the C5 feature layer. However, since the C5 feature layer is already the highest feature layer, the bidirectional connection module in the C5 layer ignores the input from the layers positioned above it. Moreover, a skip joins module (SkC) is introduced to enhance the fusion of features. Positioned posterior to the Pi feature layer, the module is equipped with three inputs: Pi, Ni-1, and Ci. Finally, in order to further reduce the parameters and FLOPs of the lightweight network, adjustments have been made to the width of the neck. The structure of BiSC-PAN is illustrated in Figure 4.

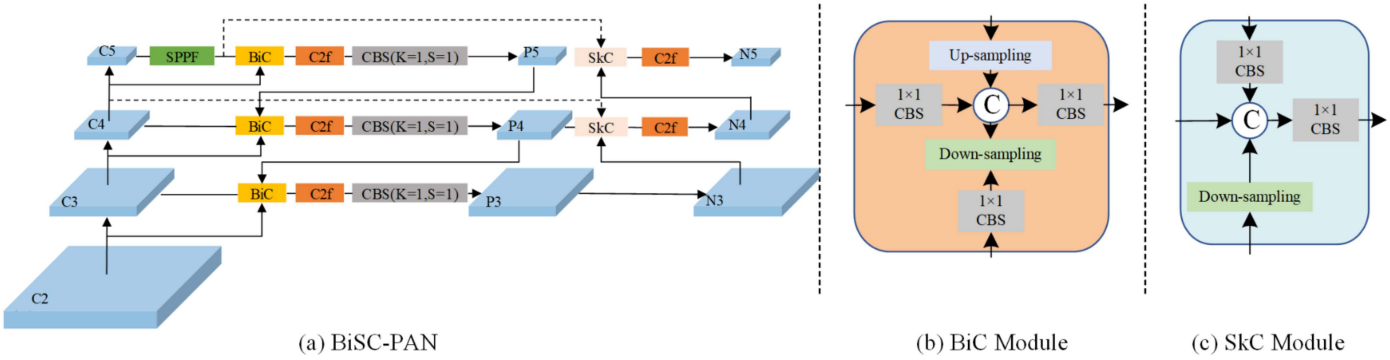
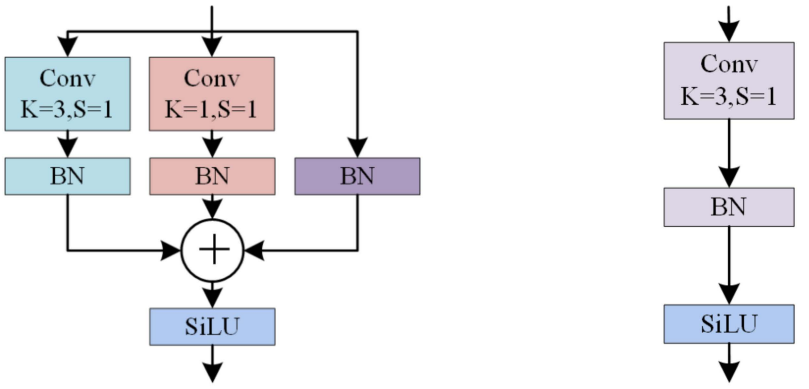


Figure 4. BiSC-PAN structure.

2.4.2. Structural Reparameterization

Structural reparameterization technology involves the creation of a series of structures designed for training purposes. Subsequently, it entails the transformation of the trained parameters into another set of parameters during the reasoning phase, thereby achieving an equivalent transformation of the overall structure. In the case of ACNet [43], the network replaces every 3 × 3 convolution with a 3 × 1 convolution, 1 × 3 convolution, and 3 × 3 convolution during the training stage. Subsequently, the outcomes of the three convolution layers are aggregated to derive the convolution layer's final output. During the inference phase, the three convolutional cores undergo fusion. Ding et al.'s RepVGG [44], leveraging structural reparameterization technology, has contributed to the revival of the VGG single-path minimalist architecture. In the course of training, parallel 1 × 1 convolution branches and identity mapping branches are added into each 3 × 3 convolution layer. Subsequently, during inference, the three branches are amalgamated to a unified 3 × 3 convolution layer. Motivated by the concepts discussed above, this study adopts the RepConv Block as a substitution for the 3 × 3 CBS Block. In the training phase, the RepConv Block is employed, featuring multiple branches to facilitate the training process. In the inference phase, each RepConv Block undergoes transformation into a 3 × 3 CBS Block. The RepConv Block structure is shown in Figure 5.



(**a**) Training (**b**) Inference

Figure 5. RepConv Block structure. (**a**) RepConv Block structure during training phase and (**b**) RepConv Block structure during inference phase.

2.4.3. Multi-Branch–Large-Kernel Downsampling Block

In contrast to other components within the network architecture, the downsampling structure is frequently overlooked. Common downsampling structures can be broadly categorized into two types. Firstly, there is the pooling operation with a stride of 2, typically employing Max pooling. The operational speed of the pooling operation is swift, owing to the fact that it solely diminishes feature dimensions without necessitating parameter learning. An alternative method involves the utilization of convolution, typically implemented as a convolution with a kernel size of 3 and a stride of 2. This approach operates at a marginally slower pace due to the concurrent necessity for feature extraction and parameter learning. YOLOv7 integrates both downsampling methods, featuring a dual-branch downsampling structure. One branch employs the Max pooling operation with a stride of 2, while the other employs convolution with a convolutional kernel size of 3 and a stride of 2.

Motivated by the preceding concepts, we initiated an exploration into the possibility of improving the downsampling method. In contrast to employing a 3×3 convolution, utilizing convolutions of 5×5 or 7×7 dimensions enhances the receptive field, thereby improving the efficacy of feature extraction. Initial consideration is to augment the size of the convolution kernel. However, it is imperative to note that this enhancement comes at the cost of a substantial escalation in FLOPs. Addressing this challenge, this study introduces a novel Multi-branch–Large-Kernel Downsampling (MLKD) Block. This Downsampling Block comprises four branches: 2×2 Max pooling, 3×3 CBS Block, 5×5 CBS Block, and 7×7 CBS Block. The channel ratio is set at 1:1:1:1. Drawing inspiration from the Inceptionv2 [45] network's implementation of a 5×5 convolution, the proposed block replaces the 5×5 and 7×7 convolutions with a two-layer series of 3×3 CBS Block and a three-layer series of 3×3 CBS Block. This strategic substitution not only effectively diminishes the parameters and FLOPs in comparison to a singular large-kernel convolution, but also adeptly mitigates the performance degradation associated with increased model depth. Ultimately, preceding the convolution process, a 1×1 CBS Block will be applied to adjust the number of channels, thereby further mitigating FLOPs. The configuration of the MLKD Block is illustrated in Figure 6.

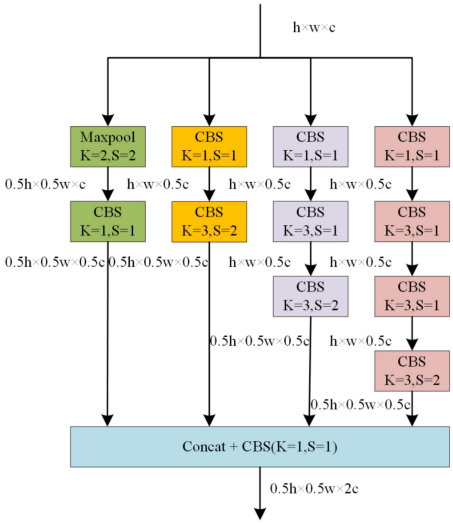


Figure 6. Multi-branch–Large-Kernel Downsampling Block structure.

2.4.4. Residual EMA-Bottleneck

The network's detection performance is influenced to a certain extent by the number of convolutional channels. Taking the YOLO series networks as an illustration, standard network structures denoted as s, m, l, and x typically exhibit a higher channel count compared to their lightweight counterparts. Nevertheless, the augmentation in the quantity of channels is frequently concomitant with an elevation in both the parameters and FLOPs. Consequently, this study initiates an exploration into whether a structure can be devised to sustain network performance while concurrently reducing the channel count. The Bottleneck, a pivotal component of the C2f Block, is recurrently employed throughout the network. Its parameters and FLOPs have a direct impact on the overall network's parameters and FLOPs. To enhance the network's lightweight characteristics, this paper integrates the concept of the residual structure, resulting in the redesign of the Bottleneck termed the Residual EMA-Bottleneck, as delineated in Figure 7. Primarily, the input flow of the Bottleneck undergoes a split operation, segmenting the original input flow into two equal parts. Subsequently, one of the parts traverses two 3 × 3 CBS Blocks for feature extraction, while the other concentrates on local features solely through an EMA attention mechanism module. Finally, the two outputs are stacked to form the ultimate output.

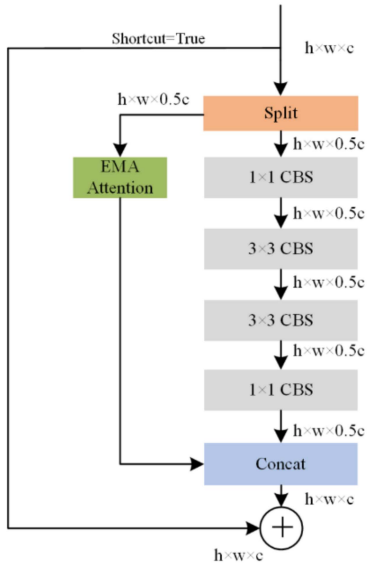


Figure 7. Residual EMA-Bottleneck structure.

2.5. Experimental Platform and Parameters

We performed all experiments on a platform with the following specifications: CPU, Intel(R) Core(R) i5-13600k; RAM, 64 GB; GPU, RTX 2080Super with 8 GB memory; the Windows11 operating system; CUDA version 11.6; Python version 3.8; Torch version 1.12.1; and CUDNN version 8.8 for deep learning computations in pycharm2022.

The network input image size was 512 × 512 × 3 pixels, the optimization strategy selected is the stochastic gradient descent (SGD) algorithm. Warmup and exponential moving average (EMA) techniques were also implemented in the experimental setup. Throughout the experiment, the training set from the ACFR Mango dataset was utilized for the purpose of model training, and the subsequent evaluation of target detection performance was carried out on the test set. We trained the model for 300 epochs using a batch size of 8 for the dataset. To enhance the information content of each image and bolster the model's capability to detect small targets, Mosaic data augmentation was implemented during the initial 285 rounds of training. Pretrained models were not employed in any of the experiments, each model underwent training from an initial state. Table 2 shows the specific Hyperparameter Settings.

Table 2. Specific Hyperparameter settings.

Training Parameter	Value
Initial learning rate	0.01
Final OneCycleLR learning rate	0.0001
Momentum	0.937
Optimizer weight decay	0.0005
Warmup epochs	3.0
Warmup initial momentum	0.8
Warmup initial bias lr	0.1

2.6. Evaluation Metrics

To holistically assess the efficacy of this model, key evaluation indices encompass accuracy (P), recall rate (R), mean average accuracy (mAP), number of network parameters (Params), floating-point operations per second (FLOPs), and average time. The definitions for P, R, mAP, and average time are articulated as follows:

$$P = \frac{T_p}{T_p + F_p} \times 100\% \quad (1)$$

$$R = \frac{T_p}{T_p + F_n} \times 100\% \quad (2)$$

$$AP = \int_0^1 P(R)d(R) \quad (3)$$

$$mAP = \frac{\sum AP}{N} \times 100\% \quad (4)$$

$$\text{average time} = \text{average Inference time} + \text{average NMS time} \quad (5)$$

where T_p represents the count of accurately predicted mangoes via the model, F_p denotes the count of mangoes erroneously predicted via the model, and F_n signifies the count of mangoes omitted via the model. P denotes the ratio of accurately predicted mangoes via the model, while R signifies the ratio of correctly predicted mangoes via the model in relation to the total count of mangoes. N is the number of detection categories. Since there is only one mango in this paper, N = 1. AP is the area under the P and R curves. mAP is the average AP value of all mango categories in the dataset, and in this paper, AP is equal to mAP. Model parameters refer to the count of parameters within the model architecture. FLOPs, representing Gigabits of floating-point operations per second, serve as an assessment of the computational complexity of a network. This paper delineates two

types of accuracy: mAP and mAP0.5. The average time is the sum of the average inference time and the average NMS time.

3. Results and Analysis

3.1. Ablation Experiments

3.1.1. BiSC-PAN

We performed a series of experiments aimed at substantiating the contributions of the BiC module and the SkC module to the network's detection performance. As highlighted in Table 3, the preliminary stage of experimentation involves the evaluation of network detection performance in the absence of both the bidirectional connection module and skip connection module. Subsequently, the BiC module and SkC module are incrementally incorporated into the network. It is observed that the inclusion of only the BiC module enhances the network's detection performance by 0.1%. Furthermore, incorporating the SkC module in conjunction with the bidirectional connection module yields an additional improvement of 0.3% in the network's detection performance. These findings underscore the efficacy of both the BiC module and the SkC Module in enhancing network detection performance.

Table 3. BiSC-PAN ablation experiment.

Method	mAP^{test}	$mAP0.5^{test}$
PANet	62.5%	95.3%
PANet + BiC	62.6%	95.5%
PANet + BiC + SkC	62.9%	95.7%

3.1.2. Structural Reparameterization

The 3 × 3 CBS Block within the C2f Block is focalized in the Bottleneck, featuring two 3 × 3 CBS Blocks in each Bottleneck. To assess the efficacy of structural reparameterization technology, this study systematically examined three distinct reparameterization structures. The experimental outcomes are presented in Table 4 and the three distinct reparameterization structures are shown as Figure 8. Initially, the 3 × 3 CBS Block above the Bottleneck was replaced, leading to a 0.2% improvement in the model performance compared to the original configuration. Subsequently, testing involved the replacement of only the 3 × 3 CBS Block below the Bottleneck, resulting in a 0.1% enhancement in the model's performance compared to its original counterpart. These findings underscore the capability of structural reparameterization technology to effectively enhance performance to a certain degree. Finally, by replacing all 3 × 3 CBS Blocks within the Bottleneck, a 0.3% improvement in the model's performance was observed relative to the original configuration. Owing to the utilization of structure recombination technology, there was no escalation in the parameters or FLOPs throughout the entire reasoning process.

Table 4. Structural reparameterization ablation experiment.

Method	mAP^{test}	Params	FLOPs
C2f	62.9%	2.74 M	3.56 G
C2f(a)	63.1% $_{(+0.2)}$	2.74 M	3.56 G
C2f(b)	63.0% $_{(+0.1)}$	2.74 M	3.56 G
C2f(c)	63.2% $_{(+0.3)}$	2.74 M	3.56 G

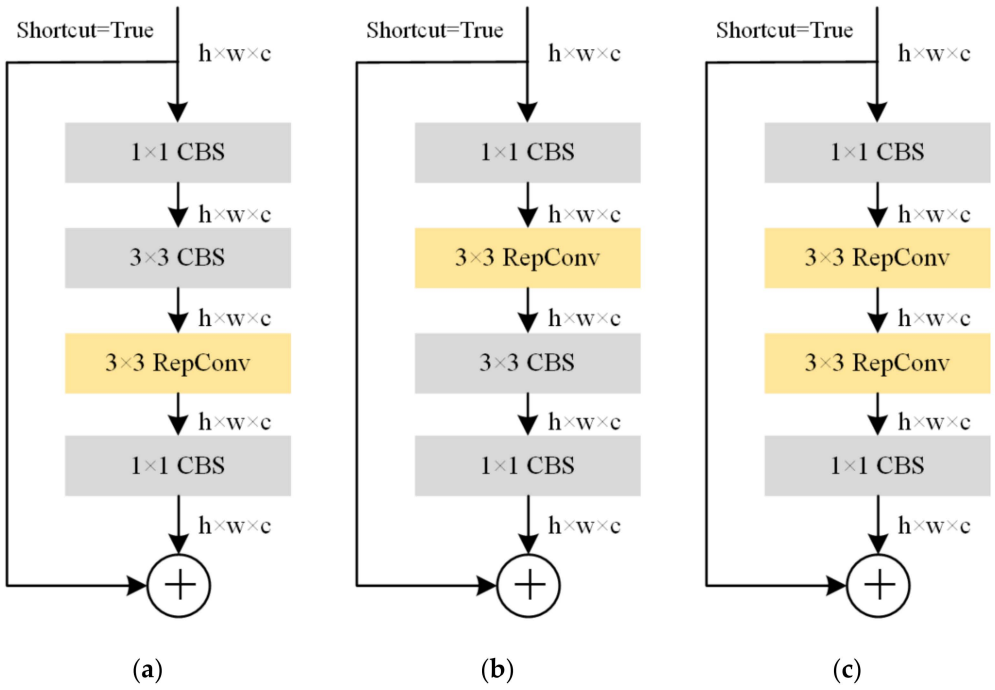


Figure 8. Three distinct reparameterization structures. (**a**) Replace the CBS Block below the C2f Block with the RepConv Block. (**b**) Replace the CBS Block above the C2f Block with the RepConv Block. (**c**) Replace all CBS Blocks in the C2f Block with the RepConv Block.

3.1.3. Multi-Branch–Large-Kernel Downsampling Block

This paper conducts a series of experiments to validate the efficacy of the proposed MLKD Block. The outcomes are presented in Table 5. Initially, the downsampling method described in YOLOv7 is individually applied to the backbone network and the neck network. It can be seen that the amalgamation of Max pooling and 3 × 3 convolution proves efficacious in diminishing both the model's parameter count and computational load. Relative to the G1, the parameters for the G2, G3, and G4 register reductions of 5.47%, 5.47%, and 11.31%, respectively. Likewise, the FLOPs for G2, G3, and G4 are correspondingly diminished by 3.37%, 2.53%, and 6.18%, respectively. As evidenced by G2, G3, and G4, the employed downsampling structure demonstrates applicability when selectively applied to either the backbone network or the neck network, resulting in enhanced average precision. Moreover, the augmentation effect is notably superior when exclusively applied to the backbone network. In light of the aforementioned findings, we ascertain the optimal deployment site for the MLKD Block and proceed to implement it within the backbone network to substantiate its efficacy. As depicted in Table 6, the initial approach involved substituting "5 × 5" CBS Block and "7 × 7" CBS Block with the 2 × 2 Max pooling and 3 × 3 CBS Block, resulting in a marginal 0.3% improvement in average accuracy. Nevertheless, there was a substantial increase in both the parameters and FLOPs, with the latter reaching 8.48 G. This marked a 138.20% surge compared to the widely adopted 3 × 3 CBS downsampling method. The incorporation of the MLKD Block not only enhances average accuracy, but also marginally diminishes the network's parameters. Moreover, it results in only a slight increase in FLOPs compared to the use of a 3 × 3 CBS downsampling method.

Table 5. The downsampling ablation experiment was performed by combining convolution and Max pooling. × means this part is not used, √ means this part is used.

Group	Max Pooling + Conv		mAP^{test}	Params	FLOPs
	Backbone	Neck			
G1	×	×	63.2%	2.74 M	3.56 G
G2	√	×	63.5% (+0.3)	2.59 M (−5.47%)	3.44 G (−3.37%)
G3	×	√	63.3% (+0.1)	2.59 M (−5.47%)	3.47 G (−2.53%)
G4	√	√	62.6% (−0.6)	2.43 M (−11.31%)	3.34 G (−6.18%)

Table 6. Multi-branch–Large-Kernel Downsampling Block ablation experiment.

Method	Branches	Proportion	mAP^{test}	Params	FLOPs
3 × 3 Conv	1	-	63.2%	2.74 M	3.56 G
Max pooling + 3 × 3 Conv	2	1:1	63.5% (+0.3)	2.59 M (−5.47%)	3.44 G (−3.37%)
5 × 5 Conv + 7 × 7 Conv	2	1:1	63.8% (+0.6)	3.37 M (+22.99%)	8.48 G (+138.20%)
MLKD Block	4	1:1:1:1	63.9% (+0.7)	2.69 M (−1.82%)	4.55 G (+27.81%)

3.1.4. Residual EMA-Bottleneck

To assess the efficacy of the Residual EMA-Bottleneck, the present study conducted a series of experiments, the results of which are detailed in Table 7. Initially, a 1 × 1 CBS Block is employed to halve the number of channels within a Bottleneck, aiming to gauge the impact of channel reduction on network performance. As anticipated, this reduction leads to a 0.8% decline in overall network performance, accompanied by a decrease in both parameters and FLOPs. Following this, the Residual Bottleneck of the C2f Block is substituted with the residual Bottleneck, sans the EMA attention mechanism supplementation. There has been a notable reduction in both the parameters and FLOPs, with decreases of 27.14% and 20.44%, respectively. However, the network's performance has decreased by 0.8%. It is observed that a halving of channel count and the incorporation of a residual structure can further diminish the number of network parameters and computational burden. However, this modification concurrently introduces a loss of precision. Ultimately, the application of the Residual EMA-Bottleneck to the C2f Block is executed. There is a reduction in both the parameters and FLOPs, concurrently leading to a 0.1% improvement in network performance.

Table 7. Residual EMA-Bottleneck ablation experiment.

Method	mAP^{test}	Params	FLOPs
Original network	63.9%	2.69 M	4.55 G
Reduce channels	63.1% (−0.8)	2.01 M (−25.28%)	3.69 G (−18.90%)
Residual Bottleneck	63.1% (−0.8)	1.96 M (−27.14%)	3.62 G (−20.44%)
Residual EMA-Bottleneck	64.0% (+0.1)	1.96 M (−27.14%)	3.65 G (−19.78%)

3.2. Comparison of Lightweight Networks

Compared to other object detection models, the YOLO model demonstrates superior real-time performance, and its lower computational cost makes it more suitable for deployment on mobile and edge devices. Currently, the YOLOv5 model remains the mainstream algorithm in the field of object detection, with continuous updates leading to significant improvements in detection speed and performance. YOLOv6, YOLOv7, and YOLOv8 represent the latest detection models. For this reason, we have chosen the lightweight versions of these models for a comparative analysis against Light-YOLO, aiming to further validate the detection performance of Light-YOLO. Furthermore, we assessed the detection performance of YOLOv6-N under distillation. The results obtained are summarized in Table 6. The data presented in Table 8 illustrate that, in comparison to YOLOv5-N and YOLOv6-N,

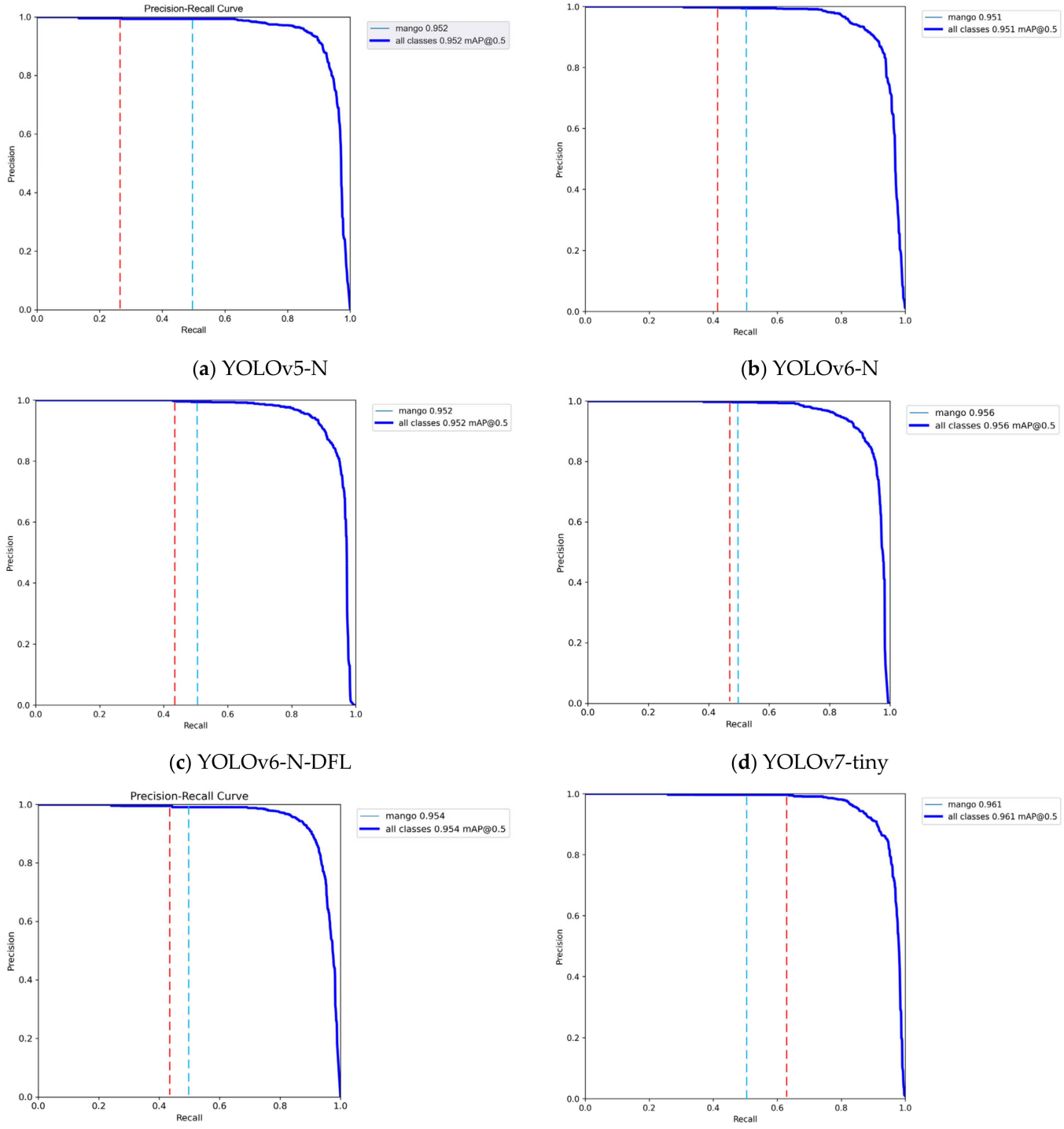
Light-YOLO exhibits an augmented accuracy of 1.1% and 2.1%, respectively. Furthermore, under the condition of a threshold set at 0.5, the accuracy of Light-YOLO demonstrates enhancements of 0.9% and 1%, respectively. In contrast to the distilled YOLOv6-N and YOLOv7-tiny, Light-YOLO exhibits an increase in accuracy of 1.3% and 2.1%, respectively. Additionally, under a threshold of 0.5, the accuracy improves by 0.9% and 0.5%, respectively. Relative to YOLOv8-N, Light-YOLO demonstrates comparable accuracy, with a noteworthy 0.7% improvement under the threshold condition of 0.5. Regarding the parameters and FLOPs, in comparison to YOLOv5-N, the parameters of Light-YOLO experience a slight 11.36% increase, yet the FLOPs diminish by 10.98%. Contrasting with YOLOv6-N, the parameters of Light-YOLO decrease by 57.67%, while the FLOPs are reduced by 49.72%. In contrast to YOLOv7-tiny and YOLOv8-N, Light-YOLO reduces parameters by 67.39% and 34.88%, and FLOPs by 71.92% and 54.94%, respectively. Furthermore, in terms of precision and recall, it is evident that Light-YOLO exhibits better recall at 91.0%. When compared to YOLOv5-N, YOLOv6-N, YOLOv6-N-DFL, YOLOv7-tiny, and YOLOv8-N, Light-YOLO surpasses them in recall by 2.0%, 0.7%, 2.7%, 3.1%, and 2.3%, respectively. This demonstrates that the Light-YOLO model is more accurate in correctly identifying positive samples. However, due to its higher recall rate, the model's precision is not particularly high at 90.9%, slightly lower than some models in Table 8. As shown in Figure 9, through the analysis of the PR curves of these lightweight models, it can be observed that in other models, when the recall is less than 0.5, the precision shows a significant decreasing trend with the increase of recall. However, in contrast, the precision of Light-YOLO only exhibits a noticeable decline after the recall reaches 0.6. Therefore, the area under the PR curve for Light-YOLO is larger, indicating superior overall performance at different recall. Part of the metric curve of the comparison models is shown in Figure 10.

3.3. Display of Visual Results

To further substantiate the practical detection efficacy of the Light-YOLO model, the test set images underwent detection using the optimal weights derived from training. During the detection process, the confidence threshold and intersection ratio were set at 0.5 and 0.3, respectively. Simultaneously, the detection performance of four models—YOLOv5-N, YOLOv6-N-DFL, YOLOv7-tiny, and YOLOv8-N—was assessed under identical parameters. Three images were specifically chosen for visualization. As shown in Figure 11, it becomes evident that other YOLO algorithms erroneously identify the area as mango due to the similar color of the bottom area, whereas Light-YOLO accurately discerns that the region is not mango. As shown in Figure 12, the inferior detection performance of other YOLO algorithms on mangoes with large occlusion areas is notable, whereas Light-YOLO adeptly identifies mangoes even in such conditions. Finally, as shown in Figure 13, the YOLOv5-N algorithm exhibits detection errors, while other YOLO algorithms manifest both error and omission detections. Remarkably, only Light-YOLO achieves flawless detection of all targets without any errors. In summary, although Light-YOLO exhibits a diminutive number of parameters and computational requirements, its overall detection performance is on par with contemporary mainstream networks, owing to the integration of BiSC-PAN, structural reparameterization, MLKD, and the Residual EMA-Bottleneck. Furthermore, in certain scenarios, Light-YOLO outperforms other models in detection efficacy.

Table 8. Comparison of lightweight networks.

Model	Params	FLOPs	Size	Average Time (bs = 1)	Precision	Recall	mAP^{test}	$mAP0.5^{test}$
YOLOv5-N	1.76 M	4.10 G	512	4.0 ms	93.2%	89.0%	62.9%	95.2%
YOLOv6-N	4.63 M	7.26 G	512	6.7 ms	90.3%	90.3%	61.9%	95.1%
YOLOv6-N-DFL	4.63 M	7.26 G	512	6.7 ms	93.5%	88.3%	62.7%	95.2%
YOLOv7-tiny	6.01 M	13.0 G	512	5.6 ms	92.6%	87.9%	61.9%	95.6%
YOLOv8-N	3.01 M	8.10 G	512	6.0 ms	92.8%	88.7%	64.0%	95.4%
Light-YOLO	1.96 M	3.65 G	512	10.8 ms	90.9%	91.0%	64.0%	96.1%



(**a**) YOLOv5-N (**b**) YOLOv6-N

(**c**) YOLOv6-N-DFL (**d**) YOLOv7-tiny

(**e**) YOLOv8-N (**f**) Light-YOLO

Figure 9. Comparison of PR curves of different lightweight models. (**a**) YOLOv5-N, (**b**) YOLOv6-N, (**c**) YOLOv6-N-DFL, (**d**) YOLOv7-tiny, (**e**) YOLOv8-N, and (**f**) Light-YOLO. The red line represents the value of recall when precision experiences its second decline.

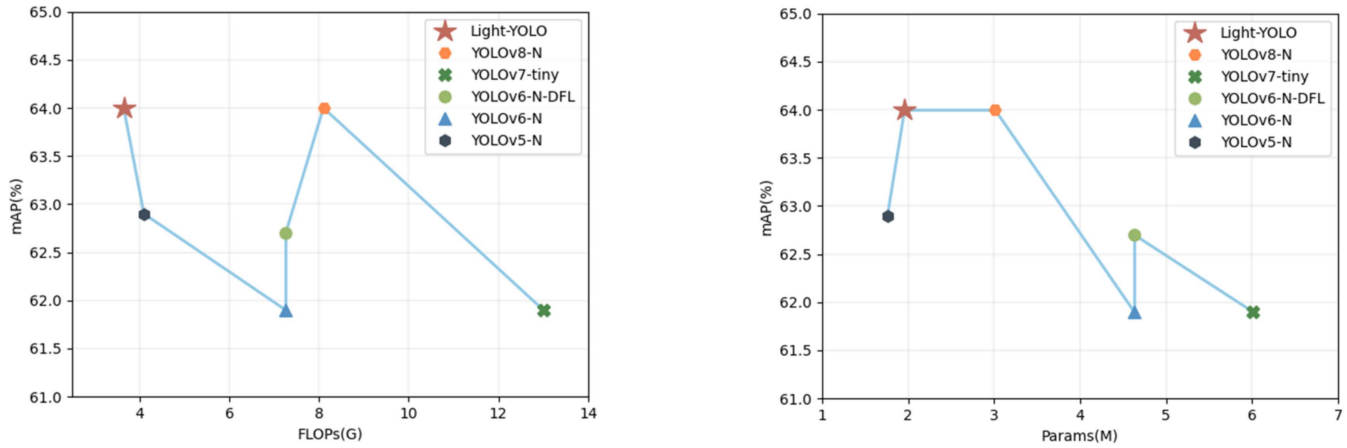


Figure 10. Comparison of lightweight networks. (**a**) Relationship curve between mAP and FLOPs and (**b**) Relationship curve between mAP and parameters.

Figure 11. Detection results of different lightweight detections in a multiple mango environment. (**a**) YOLOv5-N; (**b**) YOLOv6-N-DFL; (**c**) YOLOv7-Tiny; (**d**) YOLOv8-N; and (**e**) Light-YOLO. The green box represents the actual boxes of the mango in the Ground Truth. The red box represents predictive boxes, and the yellow circle represents missed or false detection.

Figure 12. Detection results of different lightweight detections in a speck mango environment. (**a**) YOLOv5-N; (**b**) YOLOv6-N-DFL; (**c**) YOLOv7-Tiny; (**d**) YOLOv8-N; and (**e**) Light-YOLO. The green box represents the actual boxes of the mango in the Ground Truth. The red box represents predictive boxes, and the yellow circle represents missed or false detection.

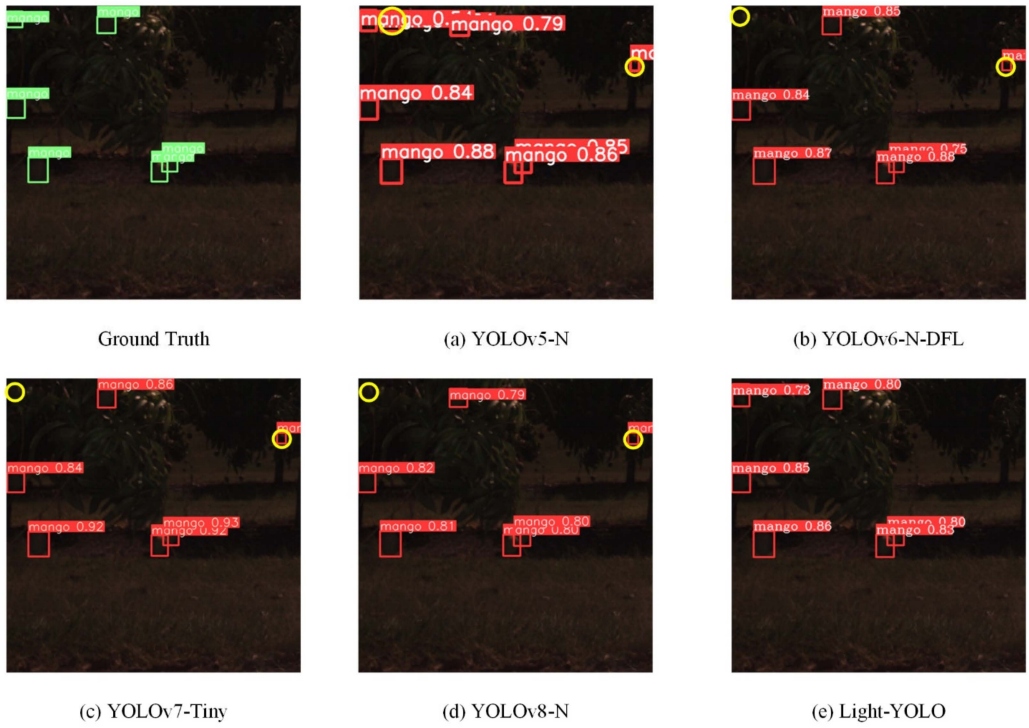


Figure 13. Detection results of different lightweight detections in a multiple mango and dark environment. (**a**) YOLOv5-N; (**b**) YOLOv6-N-DFL; (**c**) YOLOv7-Tiny; (**d**) YOLOv8-N; and (**e**) Light-YOLO. The green box represents the actual boxes of the mango in the Ground Truth. The red box represents predictive boxes, and the yellow circle represents missed or false detection.

4. Discussion

While the model advanced in this manuscript attains precision and swiftness in mango detection, it is not exempt from limitations, such as the following: (1) The incorporation of the MLKD Block, although enhancing detection accuracy, introduces a reduction in the model's inference speed due to the computational demands of the kernel. (2) The ACFR mango dataset is deficient in mango images captured within well-lit environments. Although the increased challenges posed by a dark environment on the detection algorithm can more effectively underscore the algorithm's superiority, the detection performance of Light-YOLO in well-lit environments remains unvalidated. (3) The model's accuracy in detecting heavily occluded mangoes requires enhancement. In upcoming research endeavors, we will strive to acquire additional mango images captured in well-illuminated natural settings. Subsequently, these images will be integrated into the ACFR Mango dataset to authenticate the detection efficacy of the Light-YOLO algorithm across various environmental contexts. Simultaneously, efforts will be directed towards identifying novel methods to optimize the MLKD Block, thereby enhancing the model's inference speed. Additionally, we plan to extend the application of Light-YOLO to other fruit datasets, optimizing it accordingly to accommodate a broader spectrum of fruit detection scenarios.

5. Conclusions

To address the requirements of agriculture and achieve rapid and precise detection of mangoes in natural environments, this paper introduces a lightweight and real-time object detection architecture, denoted as Light-YOLO. In the course of this investigation, we undertake the optimization of the neck network and introduce BiSC-PAN. Simultaneously, we integrate the reparameterization technique into the C2f Block of the network. Additionally, we introduce the MLKD Block and the Residual EMA-Bottleneck, successfully integrating them into the network. Experimental results demonstrate that, in comparison to alternative algorithms, the Light-YOLO algorithm exhibits outstanding detection performance while maintaining minimal parameters and FLOPs. Specifically, the parameters and FLOPs are only 1.96 M and 3.65 G, respectively, with mAP and mAP0.5 achieving values of 64.0% and 96.1%. It can satisfy the demands of agricultural deployment while furnishing a robust visual foundation for the expeditious and precise realization of mango detection.

In the future, our research will primarily focus on the following aspects. Firstly, we believe that the Large-Kernel Downsampling Block has demonstrated excellent performance in handling small targets. Therefore, we will strive to explore superior structures for the Large-Kernel Downsampling Block, employing methods such as dilated convolution and depth wise separable convolution to further reduce parameters and FLOPs. Secondly, considering the lack of mango images under optimal lighting conditions, we plan to employ drones to capture mango images in the orchards of the Jinsha River Basin in Yunnan Province, China. This aims to expand the mango dataset and conduct in-depth research into the impact of lighting conditions on the effectiveness of mango detection. Finally, we have noted the release of a mobile version of the YOLOv5 algorithm. Consequently, we will conduct comprehensive research on the relevant algorithms, with the objective of deploying Light-YOLO on mobile devices to assess its detection performance. Additionally, we will make efforts to design a fruit detection algorithm specifically tailored for optimal functionality on mobile or edge devices.

Author Contributions: Conceptualization, Z.Z. and L.Y.; methodology, Z.Z.; software, Z.Z.; validation, Z.Z. and L.Y.; formal analysis, Z.Z., L.Y. and F.C; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z., L.Y., F.C., Z.C. and C.Z.; visualization, Z.Z.; project administration, L.Y.; funding acquisition, L.Y. and F.C. 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 number 62265017.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The dataset used in the experiment described in this article comes from the ACFR Orchard Fruit Dataset provided by the agriculture team at the Australian Centre for Field Robotics, The University of Sydney, Australia. The dataset can be found in the ACFR Orchard Fruit Dataset of the University of Sydney: https://data.acfr.usyd.edu.au/ag/treecrops/2016-multifruit/ (accessed on 17 January 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef] [PubMed]
4. 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, 27–30 June 2016; pp. 779–788.
5. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
6. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
7. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
8. Jocher, G. YOLOv5 Release v6.1. 2022. Available online: https://github.com/ultralytics/YOLOv5/releases/tag/v6.1 (accessed on 17 January 2024).
9. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
10. Li, C.; Li, L.; Geng, Y.; Jiang, H.; Cheng, M.; Zhang, B.; Ke, Z.; Xu, X.; Chu, X. Yolov6 v3. 0: A full-scale reloading. *arXiv* **2023**, arXiv:2301.05586.
11. 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.
12. Jocher, G. Ultralytics YOLOv8. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 17 January 2024).
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14, pp. 21–37.
14. Wu, L.; Ma, J.; Zhao, Y.; Liu, H.J.A. Apple detection in complex scene using the improved YOLOv4 model. *Agronomy* **2021**, *11*, 476. [CrossRef]
15. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F.J.R.S. A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619.
16. Sun, H.; Wang, B.; Xue, J. YOLO-P: An efficient method for pear fast detection in complex orchard picking environment. *Front. Plant Sci.* **2023**, *13*, 1089454. [PubMed]
17. Ren, R.; Sun, H.; Zhang, S.; Wang, N.; Lu, X.; Jing, J.; Xin, M.; Cui, T.J.A. Intelligent Detection of Lightweight "Yuluxiang" Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy* **2023**, *13*, 2418. [CrossRef]
18. Stein, M.; Bargoti, S.; Underwood, J.J.S. Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors* **2016**, *16*, 1915. [CrossRef] [PubMed]
19. Li, G.-J.; Huang, X.-J.; Li, X.-H. Research on Mango Detection and Classification by Computer Vision. *J. Shenyang Agric. Univ.* **2021**, *52*, 70–78. (In Chinese)
20. Xu, Z.-F.; Jia, R.-S.; Sun, H.-M.; Liu, Q.-M.; Cui, Z.J.A.I. Light-YOLOv3: Fast method for detecting green mangoes in complex scenes using picking robots. *Appl. Intell.* **2020**, *50*, 4670–4687.
21. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
22. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
23. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
24. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
25. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7 February 2020; pp. 12993–13000.

26. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef] [PubMed]
27. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T.J.N. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
28. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
29. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
30. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. *Proc. Icml.* **2013**, *30*, 3.
31. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef] [PubMed]
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
33. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
35. 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.
36. 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.
37. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
38. 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, 4–10 June 2023; pp. 1–5.
39. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
40. Ghiasi, G.; Lin, T.-Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
41. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
42. Jiang, Y.; Tan, Z.; Wang, J.; Sun, X.; Lin, M.; Li, H.G. A Heavy-Neck Paradigm for Object Detection. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
43. Ding, X.; Guo, Y.; Ding, G.; Han, J. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1911–1920.
44. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
45. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.