mathematics

MDPI

Article

ATC-YOLOv5: Fruit Appearance Quality Classification Algorithm Based on the Improved YOLOv5 Model for Passion Fruits

Changhong Liu [1,*], Weiren Lin [1], Yifeng Feng [1], Ziqing Guo [2] and Zewen Xie [3]

[1] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China
[2] School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China
[3] School of Physics and Material Science, Guangzhou University, Guangzhou 510006, China
* Correspondence: lch@gzhu.edu.cn

Abstract: Passion fruit, renowned for its significant nutritional, medicinal, and economic value, is extensively cultivated in subtropical regions such as China, India, and Vietnam. In the production and processing industry, the quality grading of passion fruit plays a crucial role in the supply chain. However, the current process relies heavily on manual labor, resulting in inefficiency and high costs, which reflects the importance of expanding the application of fruit appearance quality classification mechanisms based on computer vision. Moreover, the existing passion fruit detection algorithms mainly focus on real-time detection and overlook the quality-classification aspect. This paper proposes the ATC-YOLOv5 model based on deep learning for passion fruit detection and quality classification. First, an improved Asymptotic Feature Pyramid Network (APFN) is utilized as the feature-extraction network, which is the network modified in this study by adding weighted feature concat pathways. This optimization enhances the feature flow between different levels and nodes, allowing for the adaptive and asymptotic fusion of richer feature information related to passion fruit quality. Secondly, the Transformer Cross Stage Partial (TRCSP) layer is constructed based on the introduction of the Multi-Head Self-Attention (MHSA) layer in the Cross Stage Partial (CSP) layer, enabling the network to achieve a better performance in modeling long-range dependencies. In addition, the Coordinate Attention (CA) mechanism is introduced to enhance the network's learning capacity for both local and non-local information, as well as the fine-grained features of passion fruit. Moreover, to validate the performance of the proposed model, a self-made passion fruit dataset is constructed to classify passion fruit into four quality grades. The original YOLOv5 serves as the baseline model. According to the experimental results, the mean average precision (mAP) of ATC-YOLOv5 reaches 95.36%, and the mean detection time (mDT) is 3.2 ms, which improves the mAP by 4.83% and the detection speed by 11.1%, and the number of parameters is reduced by 10.54% compared to the baseline, maintaining the lightweight characteristics while improving the accuracy. These experimental results validate the high detection efficiency of the proposed model for fruit quality classification, contributing to the realization of intelligent agriculture and fruit industries.

Keywords: computer vision; deep learning; fruit quality classification; passion fruit; YOLOv5

MSC: 68U10

Citation: Liu, C.; Lin, W.; Feng, Y.; Guo, Z.; Xie, Z. ATC-YOLOv5: Fruit Appearance Quality Classification Algorithm Based on the Improved YOLOv5 Model for Passion Fruits. *Mathematics* **2023**, *11*, 3615. https://doi.org/10.3390/math11163615

Academic Editors: Hongang Qi, Yan Liu and Jun Miao

Received: 25 July 2023
Revised: 13 August 2023
Accepted: 18 August 2023
Published: 21 August 2023

1. Introduction

Quality grading of fruits is an important role in the fruit supply chain [1], as it provides a uniform language for describing the quality and condition of fruits, reduces waste, increases efficiency, and enhances quality for fruit producers and consumers. In addition, quality grading is a part of food supply chain management (FSCM), which covers various aspects such as production, processing, distribution, and consumption of food products and reflects the importance of factors such as food quality, safety, and freshness within a

limited time. For the inedible, defective fruits, their presence during processing and sales can potentially contaminate other healthy fruits, posing risks to consumer health. It can also compromise the aesthetic appeal of the fruits, thus affecting consumer purchasing enthusiasm [2] and resulting in potential economic losses. Regarding the edible, qualified fruits, they need to undergo further quality grading. The premium-grade fruits are directly sold as fresh products to consumers with high demands for fruit quality. On the other hand, fruits of a slightly lower grade may be used as raw materials in the fruit processing industry, such as for making jam, fruit preserves, or beverage ingredients.

In recent years, with the rapid development of artificial intelligence technology, an increasing number of researchers are focusing on how to apply artificial intelligence technology to fruit cultivation and production processes [3,4], especially using CNN for fruit detection and quality [5–7]. Intensification and intelligence in the fruit industry have become popular trends worldwide. Fruit quality grading is one of the most important applications in the agricultural field, and the application of deep learning techniques in this area is receiving more and more attention [8]. Goyal et al. [9] developed a fruit-identification and quality-detection model based on YOLOv5. Cheng et al. [10] proposed a YOLOv4-based appearance-grading model for categorizing tomatoes. Shankar et al. [11] combined hyperparameter optimization and deep transfer learning to propose an automatic classification method for fruits. Gururaj et al. [12] explored the use of deep learning techniques to identify color variations on mango surfaces caused by defects, replacing traditional, costly, and subjective manual mango quality grading. Koirala et al. [13] developed "MangoYOLO", based on the YOLOv2 and YOLOv3 models, for real-time fruit detection and yield estimation. More recent studies have continued exploring the use of deep learning for fruit quality analysis. Patil et al. [14] compared the performance of different machine learning algorithms, including convolutional neural network (CNN) [15], artificial neural network (ANN), and support vector machine (SVM) [16], in grading or classifying dragon fruits based on surface color features. They assessed the performance differences among these algorithms. Naranjo-Torres et al. [8] conducted a detailed investigation on the application of automatic detection methods based on CNN in fresh fruit production. It categorized fruit quality grading into two types: external feature-based and internal feature-based grading. Compared to the fruits mentioned above, passion fruit has a dark purple surface and the color difference between the different qualities of the fruit is relatively small, which is a difficulty for both human classification and deep learning network classification.

Some scholars have proposed the usage of techniques such as neural networks for different passion fruit detection tasks. Tu et al. [17] conducted research on machine vision algorithms that automatically detect passion fruit based on surface color features at different maturity stages. Tu et al. [18] studied the MS-FRCNN model for estimating passion fruit production. Lu et al. [19] proposed a 3D analysis of passion fruit surface based on deep learning. Duangsuphasin et al. [20] designed a passion fruit classification model using convolutional neural networks (CNNs). More scholarly research focuses on yield estimation of passionfruit and maturity detection in complex natural environments [17,21–23], while there is scarce literature on applying deep learning techniques to quality grading of passion fruit. Therefore, exploring how to use deep learning technology to replace manual methods for passion fruit quality grading is a worthwhile research problem for subsequent processing in the passion fruit industry. Based on this situation, we investigated the application of deep learning technology in the passion fruit production and processing industry. In order to improve the detection efficiency, an improved APFN is constructed in ATC-YOLOv5, which integrates adaptive spatial fusion operations and weighted feature-fusion pathways to prevent the loss of passion fruit features. Moreover, considering the importance of long-range dependencies in the network learning and detection process, the MHSA layer in the Transformer block is introduced into the Bottleneck structure, constructing an improved TRCSP layer, which reduces the parameters and computational requirements. In addition, coordinate attention blocks are added to the neck to further enhance the attention to the passion fruit feature information. The objective of this study is to achieve efficient

quality-classification detection of passion fruits in indoor environments using improved object-detection algorithms, which can reduce labor costs and the probability of errors due to subjective factors and provide consumers with higher-quality passion fruit products.

2. Related Work

Fruit quality classification is important for reducing losses and ensuring consumer satisfaction. Researchers have explored techniques like Red-Green-Blue Dense Scale Invariant Features Transform Locality-constrained Linear Coding (RGB-DSIFT-LLC) features [17], Histogram Oriented Gradients (HOG) and color features in outdoor scenes [24], along with utilizing an electronic nose sensor to classify fruits based on their aroma [25].

Recent advancements in AI technology and deep learning have rapidly advanced object detection, including the classification of passion fruits based on quality. Compared to traditional methods, deep learning techniques, particularly convolutional neural networks (CNNs), consistently demonstrate a superior performance. Fruit quality classification is transitioning from traditional computer vision (CV) approaches to deep learning methods. Scholars have already begun applying these techniques to fruit and quality classification. Gill et al. [26] introduced a high-quality dataset of images containing various classes of fruits and trained several neural networks (NN) or algorithms such as K-nearest neighbors (KNN), SVM, random forests (RF), and multilayer perceptrons (MLP) on that dataset for comparing the differences among different sorting methods. According to their study, CNN could reach an accuracy of 98.35%, which ranked first, while SVM performed the worst, with an accuracy of only 86.11%. It should be noted that SVM used to be one of the most prevailing sorting methods, but it is clear that recently CNN outperforms SVM and many other traditional classing methods. Joseph et al. [27] compare CNN to other machine learning methods including KNN, SVM, and decision trees.

These comparisons demonstrate the advantages of CNNs over traditional approaches for CV and deep learning. CNNs can autonomously learn features, enhancing versatility and accuracy. Though more computationally demanding than conventional algorithms, advances in GPUs have enabled the adoption of CNNs. More CV algorithms now utilize CNN structures. For fruit classification, CNNs achieve a significantly higher accuracy compared to manual feature engineering.

Object detectors are either one-stage, directly predicting boxes and labels (faster but less accurate, e.g., YOLO, SSD [28]), or two-stage, using region proposals first before classification (slower but more robust, e.g., R-CNN [29]).

Among these algorithms, the YOLO series is a family of object-detection models known for a good performance balance between speed and accuracy, making them suitable for edge devices. YOLO [30] was the first version, using a Darknet [31] framework and framing detection as regression. YOLOv2 [32] improved the accuracy and speed by adding batch normalization, anchor boxes, multi-scale training, and Darknet-19 architecture. YOLOv3 [33] further improved the performance with residual connections, upsampling, feature pyramids, and Darknet-53. YOLOv4 [34] incorporated techniques like Cross Stage Partial Networks (CSPNet) [35], Mish activation [36], SAM [37], Path Aggregation Network (PANet) [38], and DropBlock [39] with Darknet-53/74. YOLOv5 [40] used PyTorch and introduced auto-anchor generation, mosaic augmentation, label smoothing [41], and EfficientNet-based architecture.

2.1. YOLOv5

YOLOv5 [40] has five different scales of network structure, specifically named YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, that is, nano (n), small (s), medium (m), large (l), and extra-large (x) models.

YOLOv5 takes 640 × 640 pixel images as input and divides them into a grid of cells, where both the horizontal and vertical pixel counts are multiples of 32. Each cell predicts bounding boxes, confidence scores reflecting objects, and class probabilities. It supports augmentations at the test time and model assembly. The backbone uses CSPNet [35], which reduces duplicate gradients to improve optimization. The neck is based on PANet [38], generating multi-scale feature maps for detection. Anchor boxes are applied to features to output vectors with probabilities, scores, and boxes. The head uses anchor boxes to output these final detection vectors. The loss function combines cross-entropy for classification, binary cross-entropy for objectness, and a generalized IoU loss for localization. The network structure of the original YOLOv5 is shown in Figure 1.

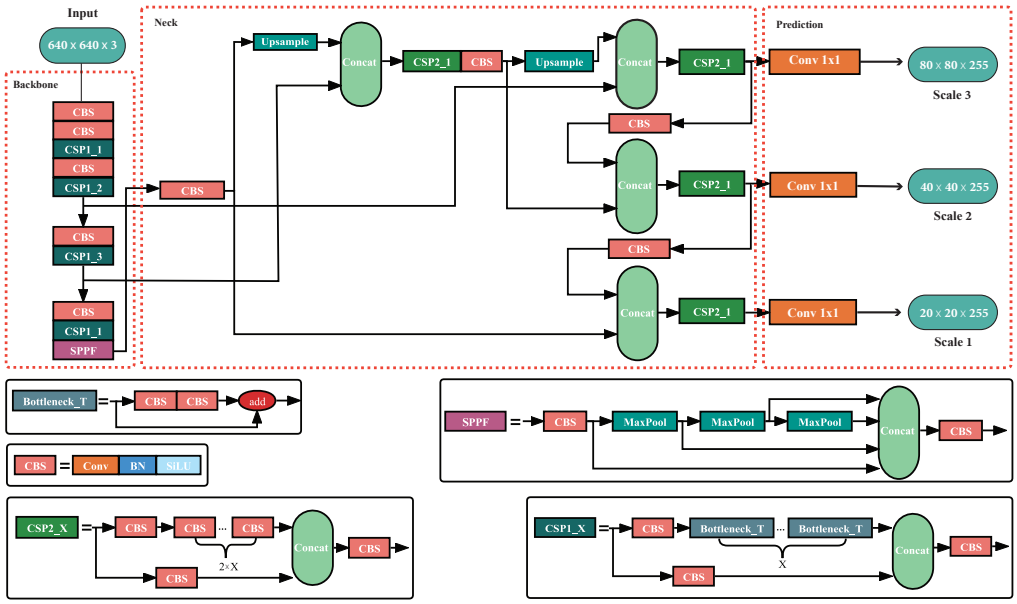


Figure 1. The network structure of the original YOLOv5 object-detection algorithm.

The following is an additional description of some of the structures in Figure 1. Conv, BN, and SiLU are the convolution layer, normalization, and activation function, respectively, which together form the CBS layer. Bottleneck_T refers to the Bottleneck structure with the residual connection. SPPF refers to the Spatial Pyramid Pooling - Fast (SPPF) layer. CSP1_*X* is a Cross Stage Partial (CSP) layer containing *X* Bottleneck_T. CSP2_*X* is a CSP layer containing *X* Bottleneck without residual connections.

2.2. Data Augmentation

Data augmentation [42] is a technique to increase diversity in the training set by creating new samples from the original data. This helps improve the model performance and robustness for computer vision tasks. Augmentation introduces variations while maintaining representativeness. It increases the training set size, improves generalization, and reduces the overfitting risk. Useful cases include lacking sufficient data, lacking diversity in the existing data, and needing robustness to noise or distortions.

Specific techniques include horizontal flipping to introduce new orientations, brightness adjustment [43] for new lighting conditions, and Gaussian blurring [44] to simulate noise. Horizontal flipping [45] is fast, retains labels, and helps recognize flipped variants; Perez and Wang [46] proved its effectiveness. Brightness augmentation creates new lighting scenarios. Gaussian blurring prevents overfitting and improves the robustness to noise.

2.3. Coordinate Attention Blocks

The coordinate attention blocks [47] represent a novel attention mechanism strategically devised to enhance deep learning models' capacity in capturing crucial spatial information and dependencies within the input data. This innovative approach achieves its objective by effectively incorporating position details into the channel information, thereby taking into account the interplay between spatial and channel aspects while effectively addressing the long-range dependency challenges.

Hou et al. [47] proposed a simple coordinate attention mechanism that can be flexibly plugged into classic mobile networks, such as MobileNetV2 [48], MobileNeXt [49], and EfficientNet [50], with nearly no computational overhead. The coordinate attention mechanism is shown in Figure 2.

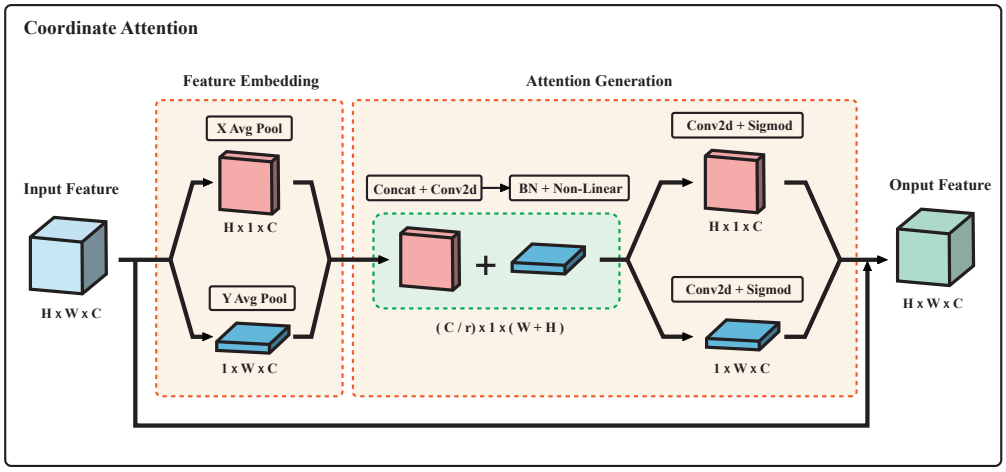


Figure 2. Coordinate attention mechanism diagram. 'H' refers to height, 'W' refers to width, and 'C' refers to channel. 'r' is the reduction ratio for controlling the block size. 'X Avg Pool' and 'Y Avg Pool' refer to a one-dimensional horizontal global pool and a one-dimensional vertical global pool, respectively, and 'BN' refers to the BatchNorm operation.

2.4. Bottleneck Transformer Blocks

Bottleneck Transformer Blocks [51], also known as BoTNet in that paper, is a backbone architecture that incorporates self-attention [52] for multiple computer vision tasks including image classification, object detection, and instance segmentation. BoTNet is based on the idea of replacing the spatial convolutions with global self-attention in the final three bottleneck blocks of a ResNet. The authors show that BoTNet achieves state-of-the-art results on the COCO [53] Instance Segmentation benchmark [54] and ImageNet [55] classification benchmark [56] while being faster and more parameter-efficient than previous models.

BoTNet can improve the performance of visual detection tasks by incorporating self-attention into the bottleneck blocks of a ResNet12. Self-attention can capture long-range dependencies and global information in an image, which can be useful for tasks such as object detection and instance segmentation. A BoTNet block can also reduce the computational cost and memory footprint compared to using self-attention on the entire feature map. A simplified diagram of the Multi-Head Self-Attention (MHSA) layer is shown in Figure 3.

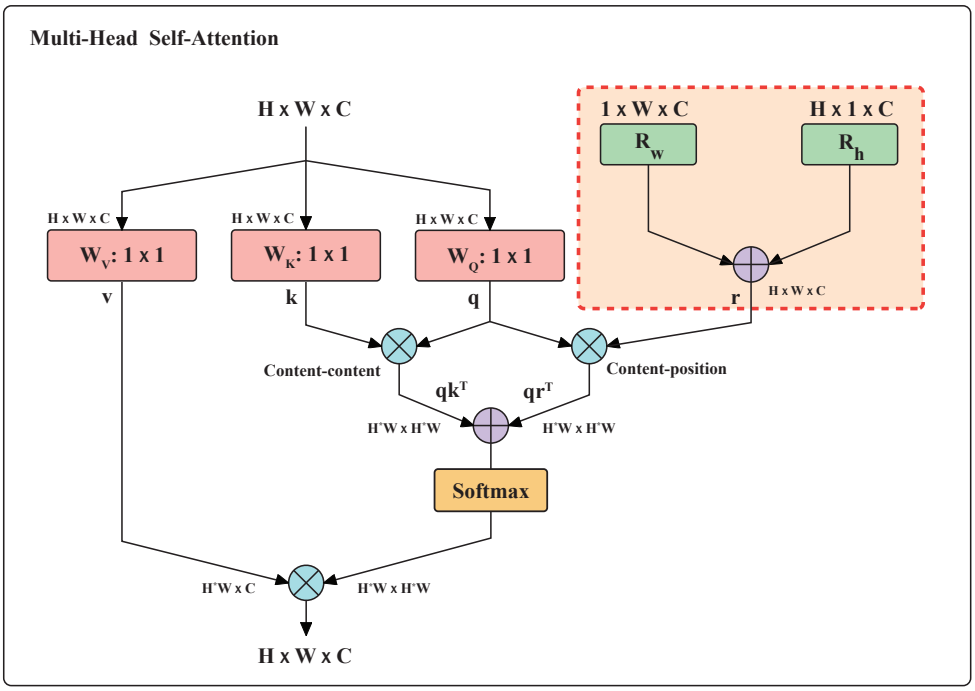


Figure 3. Simplified diagram of Multi-Head Self-Attention (MHSA) layer in the BoTNet. R_h and R_w are relative position encodings for the height and width of feature maps. q, k, r, v refer to the query, key, position, and value encodings, respectively, and $qk^T + qr^T$ refer to attention logits. $\oplus$ and $\otimes$ refer to element-wise sum and matrix multiplication, respectively. 1×1 represents a pointwise convolution.'H' refers to the height, 'W' refers to the width, and 'C' refers to the channel.

3. Methods

3.1. Overview of ATC-YOLOv5

ATC-YOLOv5 is a model derived from improvements made to YOLOv5. Its experimental detection results surpass those of the original YOLOv5 model. First, APFN [57] is employed as the main body of the feature-extraction network, enabling adaptive feature extraction across different scale levels. Combined with BiFPN [58], it is enhanced by incorporating weighted feature fusion across nodes on both the same and different scale levels, allowing the fusion of more comprehensive features without incurring excessive costs. Second, by introducing the MHSA layer in the Transformer block, the CSP layer is optimized to obtain the TRCSP layer, which reduces the parameter and computational complexity compared to the original CSP, facilitating the acquisition of more abundant correlated feature information by the network. Lastly, coordinate attention blocks are integrated into the feature-extraction network, strengthening the network's feature-extraction capabilities. The structure of ATC-YOLOv5 is shown in Figure 4.

3.2. Improved Feature Pyramid Network Based on AFPN

In the latest version of YOLOv5, the PANet [38] structure is utilized in the Neck section, which enables us to obtain outputs from different CSP layers of the backbone network and performs feature extraction at different scales. In this paper, we introduce APFN [57] as the main body in the feature-extraction network and combine it with BiFPN [58] to adaptively extract richer feature information across levels.

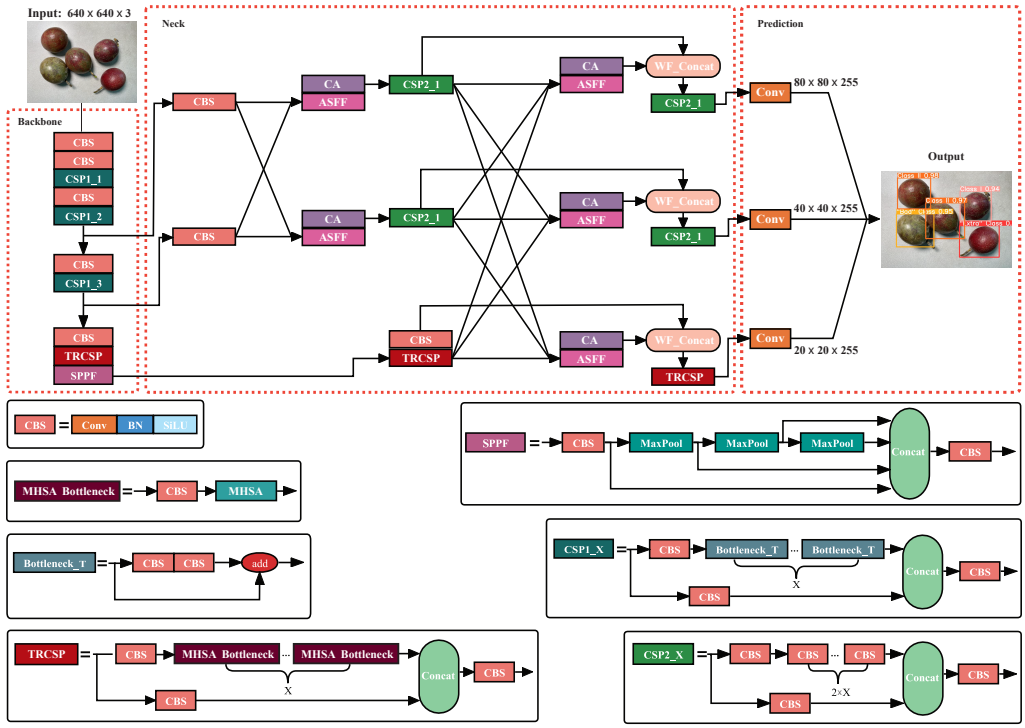


Figure 4. The network structure of ATC-YOLOv5. The backbone is the improved CSP-Darknet53 after adding the TRCSP layer, and the neck part uses the improved Asymptotic Feature Pyramid Network (APFN). In the backbone and neck, TRCSP is introduced only in the highest level of the feature layer. MHSA refers to the Multi-Head Self-Attention layer. MHSA Bottleneck refers to the structure after the introduction of the MHSA layer to the Bottleneck. A description of the rest of the structure can be found in a paragraph following Figure 1.

Unlike the sequential fusion of high-level and low-level features in PANet [38], APFN [57] employs an asymptotic feature-fusion approach. It first fuses the feature layers from lower-scale levels and gradually integrates higher-level feature layers. Simultaneously, an adaptive spatial fusion operation is utilized in the multi-level feature-fusion process to suppress feature-information conflicts between different scales, allowing better fusion of feature layers across non-adjacent scales. Building upon some aspects of BiFPN, we introduce additional feature-fusion pathways between different nodes at the same scale level in APFN. Considering the varying contributions of feature information between nodes and to avoid excessive increases in the network parameters or computational complexity, we also incorporate a weighted feature-fusion method used in BiFPN, which consumes fewer computational resources, into the additional feature-fusion pathways. Based on the fast normalized fusion [58], we improve the Concat layer to a Weighted Feature Concat (WFConcat) layer. The formula for fast normalized fusion is as follows:

$$O = \sum_i \frac{w_i}{\varepsilon + \Sigma_j w_j} \cdot I_i \tag{1}$$

where O represents the output of the weighted feature fusion, I_i represents the individual inputs of the weighted feature fusion, and w represents the learnable weight. For the form of w, a multi-dimensional tensor is used in this paper, and $\varepsilon = 0.0001$ represents a small value that serves to stabilize the values in Equation (1).

With the introduction of the improved APFN, progressive feature fusion is enabled between layers with both the same and different scales, and additional weighted feature-fusion pathways are added. This allows the new feature-extraction network to optimize the flow and fusion of feature information across scale levels and nodes while maintaining a lower parameter count and avoiding excessive computational overhead. As a result,

the feature-fusion efficiency and detection accuracy of the network for different qualities of passion fruit are further improved. The improved AFPN is illustrated in Figure 5.

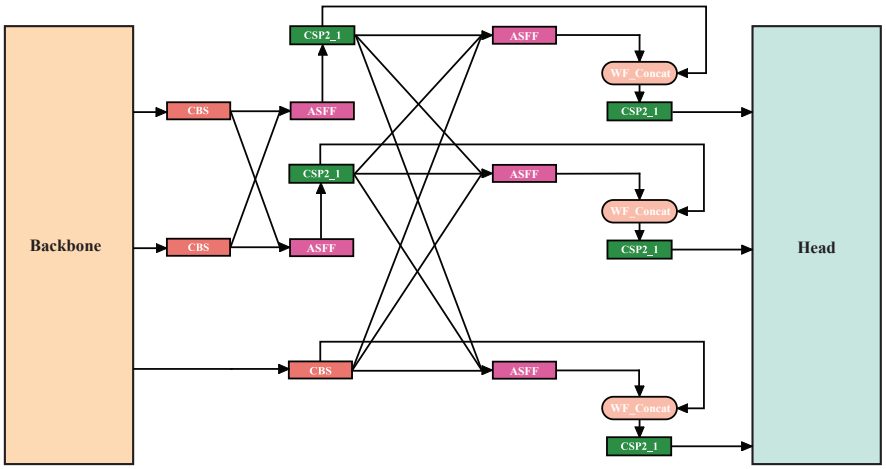


Figure 5. Improved APFN structure diagram. ASFF denotes the adaptively spatial feature-fusion operation. WF_Concat denotes weighted feature concat.

3.3. Improved CSP Model Based on MHSA Bottleneck

In the YOLOv5 network, the CSP layer is one of the main structures constituting the backbone network and the feature-extraction network. The CSP structure divides the input into two branches, with one branch undergoing convolutional operations through the CBS layer, and the other branch passing through the Bottleneck operation after the convolutional operation in the CBS layer. Finally, the features from both branches are merged. CSP can be divided into two types, CSP1_*X* and CSP2_*X*, which are applied to the backbone network and the feature-extraction network. The difference between them lies in the fact that the CSP1_*X* structure employs X Bottleneck modules with residual connections, while the CSP2_*X* structure uses X Bottleneck modules without residual connections.

The Bottleneck module used in the original CSP structure undergoes 1×1 convolution (CBS) and 3×3 convolution (CBS) operations, as shown in Figure 6, where Bottleneck_T with residual connections is applied to the CSP1_*X* layer, and Bottleneck_F is applied to the CSP2_*X* layer. The structure of the Bottleneck is similar to that of residual networks, but it has a faster computation speed. As an important component in the CSP layer, it not only performs feature extraction but also resolves the issue of gradient vanishing caused by network stacking.

Although convolution operations can effectively extract local features, in order to improve the performance of the network, structures based on convolutional operations sometimes require multilayer stacking [59], such as the 3×3 convolution (CBS) in the aforementioned Bottleneck. This not only increases the parameters and computational complexity but also performs poorly in modeling long-range dependencies required for object-detection tasks. On the other hand, self-attention mechanisms, as important components in Transformer blocks [51] for NLP tasks, can learn rich hierarchical correlated features in long sequences. Therefore, inspired by BoTNet, we replace the 3×3 convolution (CBS) in the Bottleneck of the original CSP layer with the Multi-Head Self-Attention (MHSA) layer proposed in Transformer [52], resulting in the improved MHSA Bottleneck module, as shown in Figure 6.

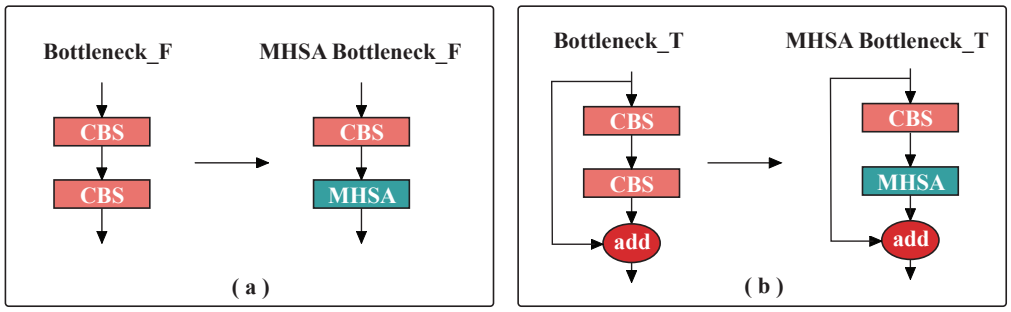


Figure 6. Bottleneck diagrams before and after improvement. (**a**) Bottleneck without residual connection, (**b**) Bottleneck with residual connection. CBS refers to a layer that consists of a convolution layer, normalization, and activation function together. MHSA refers to the Multi-Head Self-Attention layer.

After replacing the Bottleneck module in CSP with the improved MHSA Bottleneck module, we obtain the improved Transformer Cross Stage Partial (TRCSP) structure. Additionally, considering that performing self-attention operations multiple times across scale levels can significantly increase the memory and computational requirements [52], we only replace the CSP layer with the TRCSP layer in the layers with the highest number of channels of both the backbone network and the feature-extraction network. Compared to the original CSP structure, TRCSP reduces the parameters and computational complexity after applying the improved Transformer Bottleneck module. It enables the network to achieve a better performance in modeling long-range dependencies, allowing the network to learn more abundant passion fruit features, which is beneficial for passion fruit quality classification. The structure of the TRCSP is shown in Figure 7.

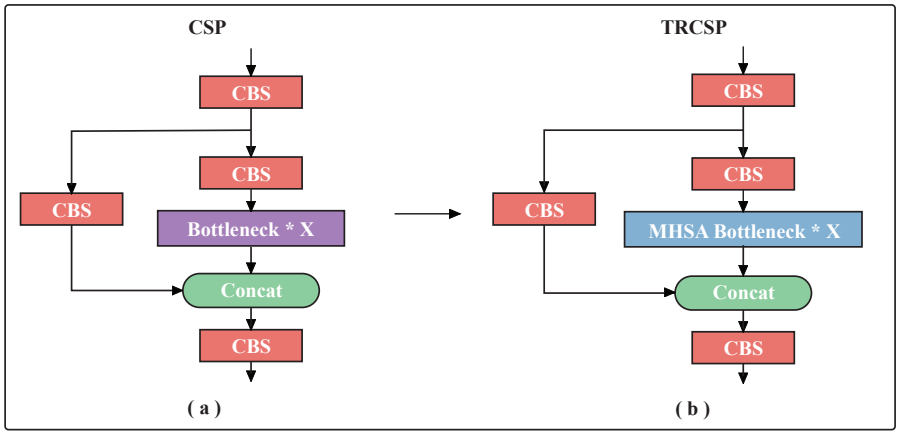


Figure 7. CSP layer diagrams before and after improvement. (**a**) Original CSP layer, (**b**) improved CSP layer. 'Bottleneck * *X*' refers to a combination of *X* Bottlenecks, and 'MHSA Bottleneck * *X*' refers to a combination of *X* MHSA Bottlenecks.

3.4. Coordinate Attention Module in Neck

In response to issues of the traditional attention mechanisms performing poorly in remote dependency modeling because they use convolutional computation and can only capture local relationships, a new method called coordinate attention mechanism [47] has been proposed that performs feature-perception operations along the spatial coordinate direction. The coordinate attention mechanism consists of two main processes: coordinate attention embedding and coordinate attention generation. In the process of coordinate attention embedding, the channel attention is decomposed into two one-dimensional feature-encoding processes, which perform feature aggregation along the two spatial directions (the x and y directions). Each channel is encoded along the horizontal coordinate and vertical coordinate using pooling kernels, resulting in a pair of direction-aware feature

maps. In the coordinate attention generation process, the obtained feature map tensors from the two different directions are adjusted in their channel dimensions through convolutional operations to match the number of channels in the input. Finally, an activation function is applied to obtain the output of the attention mechanism block. In summary, the process of coordinate attention can be represented by the following formula:

$$z_c^h = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{2}$$

where h represents the height parameter of the pooling kernels, z_c^h represents the output of the c-th channel at height h, and x_c represents the input of the c-th channel,

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w) \tag{3}$$

where w represents the width parameter of the pooling kernels and z_c^w represents the output of the c-th channel at width w,

$$u = \delta\Big(\mathrm{Conv}\Big(\Big[z^h, z^w\Big]\Big)\Big) \tag{4}$$

where $[\cdot, \cdot]$ represents the concatenation operation along the spatial dimension, δ represents a non-linear activation function, u represents the intermediate feature mapping obtained by combining the feature information horizontally and vertically, and Conv represents convolutional transformation in Equation (4),

$$g^h = \sigma\Big(\mathrm{Conv}\Big(u^h\Big)\Big) \tag{5}$$

$$g^\omega = \sigma(\mathrm{Conv}(u^\omega)) \tag{6}$$

where g^ω represents the tensor with the same channel number as the input obtained by the transformation of u^ω, g^h represents the tensor with the same channel number as the input obtained by the transformation of u^h, σ represents the sigmoid function, and u^h and u^w represent the two tensors obtained by the decomposition of u along the spatial dimension,

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{7}$$

where y_c represents the output of the c-th channel in Equation (7).

In order to improve the network's ability to locate, recognize, and extract better feature maps and information for passion fruit quality classification tasks, we have inserted coordinate attention modules after each adaptive spatial fusion operation in the feature-extraction network. With the assistance of coordinate attention, the network can adaptively learn the feature information and fine-grained details of the target objects. It can also accurately capture the precise position and long-range dependency information of the objects of interest. This helps the overall model to learn and detect more effectively.

4. Results and Discussion

4.1. Dataset Details

The datasets were created by capturing images of passion fruit using an iPhone 11. The images were taken indoors with various lighting conditions to maximize the color and appearance variation. Desk lamps with a high color rendering index (CRI) were primarily used, along with some fluorescent and natural lighting. In total, 1114 raw HEIC images (4032×3024 resolution) were captured. These were converted to JPEG format and resized to 640×480 for deep learning. Some low-quality images were excluded. The remaining images were augmented via processing. Yolo-mark was used to label the datasets. In order

to facilitate compatibility with the YOLO, SSD, and Faster R-CNN [60] formats, annotations were being stored as extensible markup language (XML) files in the PASCAL VOC format.

The dataset is partitioned into two distinct sub-datasets: a training set with 735 images and a testing set with 379 images. Then, three data-augmentation techniques were introduced to the training set, which enlarged the number of images in the training set by four times. With some minor adjustments, the final training set has 2540 images and the final test set, 379 images.

The images were divided into four classes: Extra Class, Class I, Class II, and Bad Class according to the Codex Standard for Passion Fruit (CODEX STAN 316-2014) [61]. This standard specifies the minimum quality, classification, sizing, tolerances, packaging, and labeling requirements for commercial passion fruit varieties. It sets quality criteria such as whole, firm, clean fruit free of defects and damage. Fruits are classified into three classes—Extra, Class I, and Class II—based on an increasing allowance for defects. The size can be determined by count, diameter, or weight, but it is not taken into account for the machine learning model provided, since the users of this passion fruit quality classifier can classify the incoming fruits by size directly by having the fruit pass through a slideway with a series of different pore sizes. Tolerances are provided for the percentage of fruit not meeting the class requirements. A few minor adjustments were made to the official standard to suit our mission. We classified the fruit based on the rules listed in Tables 1 and 2.

Table 1. Passion fruit classification standard.

Grade	Requirements for Each Grade
Extra Class (superior quality)	Must be free of defects; only very slight superficial defects are acceptable.
Class I (good quality)	The following slight defects are acceptable: 1. a slight defect in shape. 2. slight defects of the skin such as scratches, not exceeding more than 10% of the total surface area of the fruit. 3. slight defects in colouring. (The defects must not, in any case, affect the flesh of the fruit)
Class II (average quality)	Satisfy the minimum requirements (2) but do not qualify for inclusion in the higher classes. The following defects are allowed: 1. defects in shape, including an extension in the zone of the stalk. 2. defects of the skin such as scratches or rough skin, not exceeding more than 20% of the total surface area of the fruit. 3. defects in colouring. (The passion fruits must retain their essential characteristics as regards the quality, keeping quality, and presentation)
"Bad" class (NOT ready for sale)	A passion fruit will be classified as this grade if it does not meet any of the minimum requirements (2)

Table 2. Minimum requirements for the classification of passion fruit.

Classification Details
fresh in appearance (without rotting)
clean, free of any visible foreign matter
practically free of pests and damage caused by them affecting the general appearance
free of abnormal external moisture
the stem/stalk should be present
free of cracking

4.2. Experimental Settings

The server parameters used to conduct our experiments are listed in Table 3.

The batch size was set to 64. In order to keep the dataset consistent in each run (the data-augmentation techniques involve some random processes) and across all the NN we ran (different algorithms would apply different data-augmentation techniques), we turned

off all random data augmentations that were included in the YOLOv5 model itself. Other parameters were kept the same as the original YOLOv5s.

Table 3. Server properties and environmental information.

Parameter	Configuration
CPU	Intel (R) Xeon (R) Platinum 8358P
GPU	NVIDIA RTX3090
CUDA version	Cuda 11.3
Python version	Python 3.8
Deep learning framework	PyTorch 1.11.0
Operating system	ubuntu 20.04

4.3. Evaluation Indicator

In this research, we employ precision, recall, mean average precision (mAP), $F1$ *score*, and mean detection time (mDT) as performance-evaluation metrics for the proposed network model. In object detection, AP is typically calculated at various confidence thresholds to generate a precision–recall (P–R) curve. However, it can be useful to examine the performance of a model at a specific confidence threshold, such as 50%, so we use $mAP50$ specifically to evaluate the performance. On the other hand, mDT indicates the average detection time of the model, typically with a unit of ms, which determines its suitability for real-time detection. The following equations illustrate the calculation methods for precision, recall, AP, mAP, $F1$ *score*, and mDT:

$$P_c = \frac{TP_c}{FP_c + TP_c} \tag{8}$$

$$R_c = \frac{TR_c}{FN_c + TP_c} \tag{9}$$

$$AP_c = \sum \int_0^1 P(R_c) dR_c \tag{10}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{11}$$

$$F1\ score = \frac{2 \times P \times R}{P + R} \tag{12}$$

$$mDT = \frac{1}{M} \sum_{i=1}^{M} t_i \tag{13}$$

In the context of this research, we denote the precision, recall, and detection class as P, R, and c, respectively. Furthermore, TP, FN, and FP symbolize true positive, false negative, and false positive instances, respectively. N represents the total number of classification classes. M means the count of detected images in a certain time t. In our study, we obtain the mDT by detecting all the images in the test set, so M is a constant with a value of 379.

4.4. Reasons for Data-Augmentation Applications

We augmented the images by flipping, Gaussian noise, and brightness adjustment. The illumination conditions under which images are captured in real application scenarios often vary significantly, leading to variations in brightness levels across different instances of the same scene. These variations can pose challenges for deep learning models when attempting to generalize from a limited range of training examples. By incorporating the brightness data-augmentation technique [43], we intend to expose the model to a more diverse set of brightness conditions, enabling it to learn representations that are robust to variations in

illumination. Additionally, the application of brightness data augmentation may alleviate the need for collecting a larger dataset, potentially reducing the data-acquisition costs and computational requirements.

Considering that the classification process of passion fruit may take place during conveyor belt movement, it is possible that our neural network will have to detect the image in motion blur. The presence of moving passion fruits poses a challenge to accurately identify and classify them. This is particularly evident when users aim to capture images before all the fruits have settled to expedite the image-acquisition process. Additionally, irregularly shaped or deformed fruits tend to take longer to come to a complete stop, further complicating the detection process. Incorporating Gaussian blur [44] data augmentation during the training phase enables the network to focus on the relevant features and structures, leading to improved generalization. It can also contribute to smoother gradients during training, facilitating optimization and potentially leading to faster convergence.

The accurate detection of passion fruits becomes challenging since there are several possible scenarios for the classification of passion fruit. To mitigate this issue, we utilize horizontal flipping [45] data augmentation. The primary motivation behind employing horizontal flipping data augmentation is to expose the model to diverse orientations of passion fruits, thereby enabling it to handle variations in real-world scenarios effectively. By flipping a portion of the images horizontally, we create a new set of training data that includes both the original and flipped versions. This augmentation strategy effectively doubles the size of the training dataset, providing the model with more examples to learn from. Moreover, the presence of flipped passion fruit instances in the dataset encourages the model to learn features and patterns that are invariant to horizontal flipping. Consequently, the model becomes more robust to variations in passion fruit placement, as it learns to recognize the fruits regardless of their orientation. This increased resilience enhances the model's performance in detecting passion fruits in diverse settings, contributing to improved accuracy and reliability in practical applications.

4.5. Ablation Studies

To evaluate and validate the effectiveness of ATC-YOLOv5, we designed ablation experiments using a self-made passion fruit dataset. The ablation experiments including various improvement strategies are shown in Table 4, and the corresponding results are shown in Table 5. For a clear comparison of the model's performance with different enhancements, we primarily used the mean average precision (mAP), F1 score, precision (P), recall (R), number of parameters, mean detection time (mDT), and giga floating-point operations per second (GFLOPs) as evaluation metrics.

According to the results in Table 5, different schemes of the algorithm showed varying degrees of improvement compared to the baseline. In the ablation experiment, we first tested the replacement of FPN in the original network with AFPN. Due to its progressive feature fusion across different level feature layers, it reduces the impact of the semantic gap between non-adjacent layers and improves their fusion effectiveness. Specifically, AFPN fuses the most abstract top-level features extracted from YOLOv5's backbone. The replacement of AFPN resulted in a 1.35% increase in mAP, a reduction of 9.26% in the network parameters, and a decrease of 5.06% in GFLOPs.

In addition, based on AFPN's structure, the iAFPN scheme added modifications to feature fusion between non-adjacent feature layers of the same level. In AFPN, Adaptively Spatial Feature Fusion (ASFF) [62] is used for feature fusion. However, the algorithmic process of ASFF involves softmax, gradient, and backpropagation, which is obviously more complex than Fast normalized fusion [58] with only one formula in BiFPN. Therefore, to preserve the network's ability to learn the contribution differences between different feature layers without significantly increasing the parameter count, the newly introduced feature-fusion operation no longer uses adaptive spatial fusion but uses weighted feature fusion. iAFPN retains the advantages of AFPN in enhancing the fusion of feature layers between different levels and enables richer feature fusion between layers of the same level.

Consequently, iAFPN achieves a 2.59% improvement in mAP compared to the baseline, surpassing AFPN's mAP results. Moreover, iAFPN only increases the parameter count by 0.09 million and GFLOPs by 0.1 compared to AFPN.

Table 4. Different combinations for YOLOv5 algorithm improvements in ablation experiments.

Scheme	AFPN	Improved AFPN	Coordinate Attention Module	TRCSP
YOLOv5s (baseline)				
AFPN-YOLOv5s	√			
iAFPN-YOLOv5s		√		
iAFPN-CA-YOLOv5s		√	√	
iAFPN-TRCSP-YOLOv5s		√		√
ATC-YOLOv5 (ours)		√	√	√

Additionally, in the iAFPN-CA scheme, we introduce the coordinate attention mechanism, which allows the network to accurately capture the precise location and long-range dependency information of the objects of interest. With almost no parameter consumption, iAFPN-CA improves mAP by 0.55%. In the iAFPN-TRCSP scheme, we incorporate the Transformer Cross Stage Partial (TRCSP) layer, which achieves better results in long-range dependency modeling, into the highest abstract feature layer. Compared to iAFPN-CA, iAFPN-TRCSP increases mAP by 1.21% and reduces the parameter count and computational cost by 0.21 million and 0.2, respectively.

ATC-YOLOv5 integrates all the above improvements. Compared to the baseline, ATC-YOLOv5 achieves a 4.83% increase in mAP, a 2.75% improvement in F1, a 1.25% improvement in P, a 4.28% improvement in R, a reduction of 0.74 million parameters, an 11.1% faster mDT, and a decrease of 0.9 in GFLOPs. It is worth noting that although the iAFPN-TRCSP scheme had the optimal parameters and mDT in the ablation experiment results, the differences in these parameters between ATC-YOLOv5 and iAFPN-TRCSP are very small, making ATC-YOLOv5 still a lightweight network with better performance in terms of accuracy, detection speed, and model size compared to the baseline.

A comparison of the mAP changes between the ATC-YOLOv5 algorithm and the baseline algorithm during training is shown in Figure 8. The mAP is used as one of the most important metrics for model evaluation and combines both the precision and recall factors of the model. According to Figure 8, it can be seen that the training epoch required for ATC-YOLOv5 to reach the mAP near the highest value is less than that of the baseline algorithm. Moreover, the final value of the mAP of ATC-YOLOv5 is also higher than that of the baseline algorithm. This indicates that ATC-YOLOv5 can achieve better detection accuracy requirements.

Table 5. The results of ablation experiments. The bolded parameters are the most optimal.

Algorithms	mAP50/%	F1/%	P/%	R/%	Param */M	mDT/ms	GFLOPs
YOLOv5s (baseline)	90.53	86.55	86.63	86.48	7.02	3.6	15.8
AFPN-YOLOv5s	91.88	87.55	86.53	88.6	6.37	3.6	15
iAFPN-YOLOv5s	93.12	87.74	87.62	87.86	6.46	3.7	15.1
iAFPN-CA-YOLOv5s	93.67	89.00	**90.19**	87.85	6.47	3.3	15.1
iAFPN-TRCSP-YOLOv5s	94.88	87.32	85.22	89.52	**6.26**	**3.1**	14.9
ATC-YOLOv5 (ours)	**95.36**	**89.30**	87.88	**90.76**	6.28	3.2	**14.9**

* Param: the number of parameters.

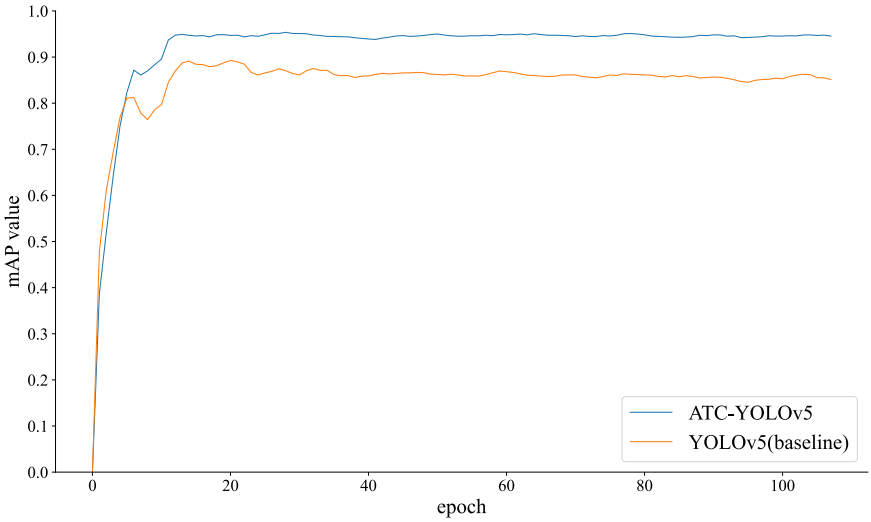


Figure 8. Comparison of mAP results between ATC-YOLOv5 and baseline algorithm.

4.6. Performance Comparison and Analysis

To objectively evaluate the effectiveness of the proposed method, we conducted comparative experiments with mainstream object-detection algorithms. The experiments were conducted in the same environment using identical datasets. The selection of image resolution holds paramount importance as it significantly influences the structure of the model network. Opting for high-resolution images amplifies the computational requirements and training duration, whereas adopting low-resolution images may result in a diminished training accuracy. Hence, in our comparison experiments, we set different image resolutions for each algorithm to account for these factors. The result of the comparison is listed in Table 6.

Table 6. Results for different well-known algorithms. The bolded parameters are the most optimal.

Algorithms	Input Size	mAP50/%	F1/%	P/%	R/%	Param */M	mDT/ms	GFLOPs
YOLOv3 [33]	480 × 640	80.96	81.48	79.34	83.37	97.34	8.4	116.0
Darknet-YOLOv4 [34]	480 × 640	81.56	83.67	79.61	86.85	140.09	10.08	105.8
YOLOv6-s [63]	640 × 640	82.91	79.10	87.42	75.22	18.5	7.38	45.2
YOLOv7-tiny [64]	640 × 640	70.38	69.45	69.63	69.32	**6.01**	5.8	**13.0**
YOLOv8-s [65]	640 × 640	87.75	79.98	78.60	81.40	11.13	8.00	28.4
SSD300 [28]	300 × 300	75.24	60.45	87.78	46.17	24.1	8.26	15.3
ATC-YOLOv5 (ours)	640 × 640	**95.36**	**89.30**	**87.88**	**90.76**	6.28	**3.2**	14.9

* Param: the number of parameters.

The proposed ATC-YOLOv5 method demonstrates a better performance compared to other state-of-the-art object-detection algorithms across multiple evaluation metrics.

In terms of accuracy, ATC-YOLOv5 achieves the highest mean average precision (mAP50) of 95.36%, outperforming the next-best method YOLOv8-s by 7.61%. This indicates the effectiveness of the proposed improvements to YOLOv5 in enhancing the detection accuracy. In addition to accuracy improvements, ATC-YOLOv5 also has efficiency gains compared to prior arts. It obtains the fastest mean detection time per image of just 3.2 ms, which is over two times faster than the next-fastest approach SSD300. Remarkably, ATC-YOLOv5 accomplishes these accuracy and speed improvements with a comparable model size of 6.28 million parameters. It even has fewer parameters than top performers like YOLOv8-s and Darknet-YOLOv4. The only method with fewer parameters is the highly compact YOLOv7-tiny. However, this comes at a huge reduction in accuracy.

On the other hand, the classification criteria include a variety of factors; thus, some passion fruits may be similar in color but belong to different quality classes due to differences in skin quality or shape. The common appearance features of fruit [1] used for computer vision include color features, morphological features, and texture features. Different features have different reflections on the quality of the fruit, and they may be more complex when combined, which makes it difficult for the model to recognize and learn. For example, due to the close resemblance in appearance between Class I and Extra Class passion fruits, the only difference lies in the slightly less vibrant color and surface skin defects of Class I compared to the Extra Class. As a result, the baseline algorithm is prone to confusion between the two categories, as shown in Figure 9a, where it mistakenly identifies the Extra Class passion fruit as Class I. Similarly, for the passion fruits that should be classified as Class II, the main distinguishing factor is the darker purple color. However, the baseline algorithm tends to misclassify the passion fruit in Figure 9c, which also has a dark color, as Class II. In reality, this type of passion fruit exhibits a greenish color and other defects, indicating that it is unripe or rotten and should be categorized as the disqualified Bad Class. In contrast, both Figure 9b,d of ATC-YOLOv5's detection results produce accurate identifications, showcasing the advantage of ATC-YOLOv5 in distinguishing the detailed features of similar-looking passion fruits.

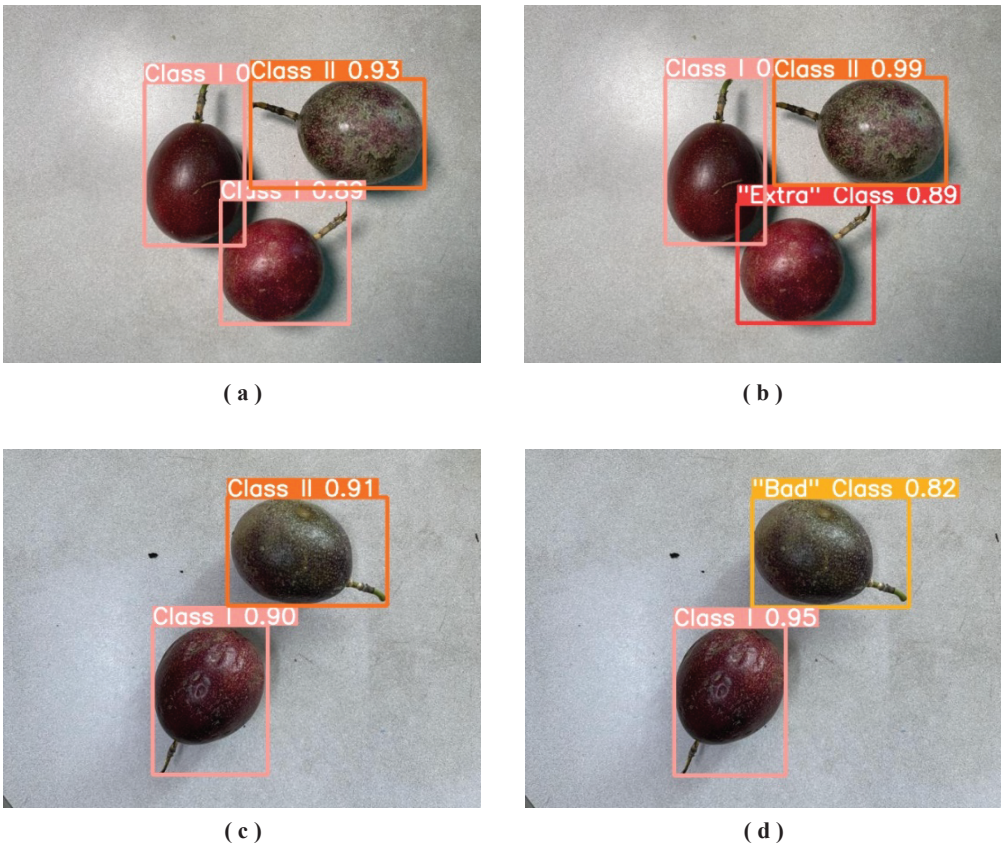
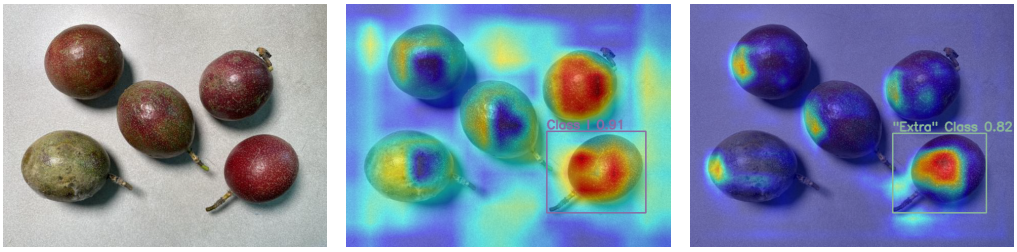


Figure 9. The detection results of passion fruit quality grading. (**a**,**c**) are the detection results of YOLOv5 (baseline). (**b**,**d**) are the detection results of ATC-YOLOv5 (ours).

The results validate the effectiveness of the proposed improvements in ATC-YOLOv5, enabling superior accuracy and speed with an efficient model size. The method advances state-of-the-art object detection across multiple competitive benchmarks.

To intuitively demonstrate the advantages of ATC-YOLOv5, we also present its feature attention effect in the form of heatmaps, as shown in Figure 10.



(a) (b) (c)

Figure 10. When the appearances are more similar, the network's classification of Extra class passion fruit and the heatmap results. (**a**) Input image. (**b**) Detection and heatmap results of YOLOv5 (baseline). (**c**) Detection and heatmap results of ATC-YOLOv5 (ours).

Based on Figure 10, it can be observed that the baseline algorithm mistakenly identifies the Extra Class as Class I during detection, and due to their similarity, the network's attention is also scattered between the two. On the other hand, ATC-YOLOv5 can correctly distinguish between Class I with minor skin defects and the ExtraClass with almost no skin defects, and the network's attention is mostly focused on the correct targets.

One of the additional improvement directions of ATC-YOLOv5 is network lightweightness. In the context of smart agriculture and automated fruit production, if ATC-YOLOv5 is used for passion fruit quality grading tasks, it is likely to be deployed on automated sorting machines or robots [66], which imposes certain requirements on the network's size, detection speed, and computational efficiency. Among the introduced improvements in this study, both iAFPN and TRCSP can increase the mAP while reducing the number of parameters and GFLOPs. Additionally, TRCSP reduces the use of convolution layers, resulting in a better mDT compared to the baseline, enabling faster detection with fewer computational resources. In contrast, Cheng et al. [10] constructed an appearance-grading model based on the YOLOv4 algorithm for tomato grading, but they did not consider the lightweightness of the model in their experiments. Tu et al. [17] proposed a network that achieved a detection accuracy of 92.71%, but the detection speed per image was still 72.14 ms, leaving room for optimization in terms of detection speed. Overall, the improvements introduced in ATC-YOLOv5 make it more suitable for deployment on hardware devices or robots.

Although the proposed ATC-YOLOv5 has achieved excellent results in terms of accuracy and lightweightness, it still has some limitations, which provide opportunities for future research directions. For instance, the network training is simulated under the conditions of passion fruit transportation, sorting, or production processes, which may limit its detection performance for unpicked passion fruit in the cultivation environment. Therefore, future work could focus on improving the model's environmental generalization ability to adapt to various background environments.

5. Conclusions

To achieve the quality grading and classification of passion fruit based on its appearance, this paper proposes ATC-YOLOv5, a lightweight passion fruit quality classification algorithm, built upon YOLOv5. The iAFPN is obtained by improving the AFPN, which introduces multiple weighted feature-fusion pathways. The original YOLOv5's FPN structure is replaced with the enhanced iAFPN, and the fusion between non-adjacent feature layers with significant semantic gaps is optimized. The TRCSP layer is created by incorporating the MHSA layer into the CSP layer, thereby replacing the CSP layer in the highest feature level of the network. The coordinate attention mechanism is integrated to assist in feature extraction. According to the experimental results, the mAP of ATC-YOLOv5 reaches 95.36%, the mDT is 3.2 ms, the number of parameters is 6.28 million, and the GFLOPs are 14.9. Compared to the baseline or other object-detection algorithms, ATC-YOLOv5 achieves a 4.83% increase in mAP and an 11.11% improvement in detection speed, while reducing the parameters and GFLOPs by 10.54% and 5.7%, respectively. ATC-YOLOv5's

better detection accuracy and lightweight network features provide the basis for its application in the detection and grading of other fruits. This study was trained and tested on a self-made passion fruit dataset that supports the development of intelligent agriculture and fruit production. In future work, we will focus on further improving the algorithm's detection performance in different environments and enhancing its compatibility with robots or drones.

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

Funding: The authors acknowledge the funding of the following science foundations: the Science and Technology Planning Project of Guangzhou, China (202102010392), the Science and Technology Planning Project of Guangdong Province, China (2020A1414050067), the Teaching Reform Project in Guangzhou Universities, China (2022CXCYZX001, 2023QTJG0604).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to lab privacy.

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

References

1. Bhargava, A.; Bansal, A. Fruits and vegetables quality evaluation using computer vision: A review. *J. King Saud Univ. Comput. Inf.* **2021**, *33*, 243–257. [CrossRef]
2. Wang, C.; Liu, S.; Wang, Y.; Xiong, J.; Zhang, Z.; Zhao, B.; Luo, L.; Lin, G.; He, P. Application of convolutional neural network-based detection methods in fresh fruit production: A comprehensive review. *Front. Plant Sci.* **2022**, *13*, 868745 [CrossRef] [PubMed]
3. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
4. Li, K.; Wang, J.; Jalil, H.; Wang, H. A fast and lightweight detection algorithm for passion fruit pests based on improved YOLOv5. *Comput. Electron. Agric.* **2023**, *204*, 107534. [CrossRef]
5. Sharma, A.K.; Nguyen, H.H.C.; Bui, T.X.; Bhardwa, S.; Van Thang, D. An Approach to Ripening of Pineapple Fruit with Model Yolo V5. In Proceedings of the 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Pune, India, 7–9 April 2022; pp. 1–5.
6. Bortolotti, G.; Mengoli, D.; Piani, M.; Grappadelli, L.C.; Manfrini, L. A computer vision system for in-field quality evaluation: Preliminary results on peach fruit. In Proceedings of the 2022 IEEE Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Perugia, Italy, 3–5 November 2022; pp. 180–185.
7. Mirhaji, H.; Soleymani, M.; Asakereh, A.; Mehdizadeh, S.A. Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions. *Comput. Electron. Agric.* **2021**, *191*, 106533. [CrossRef]
8. Naranjo-Torres, J.; Mora, M.; Hernández-García, R.; Barrientos, R.J.; Fredes, C.; Valenzuela, A. A review of convolutional neural network applied to fruit image processing. *Appl. Sci.* **2020**, *10*, 3443. [CrossRef]
9. Goyal, K.; Kumar, P.; Verma, K. AI-based fruit identification and quality detection system. *Multimed. Tools Appl.* **2022**, *82*, 24573–24604. [CrossRef]
10. Cheng, Y.H.; Tseng, C.Y.; Nguyen, D.M.; Lin, Y.D. YOLOv4-Driven Appearance Grading Filing Mechanism: Toward a High-Accuracy Tomato Grading Model through a Deep-Learning Framework. *Mathematics* **2022**, *10*, 3398. [CrossRef]
11. Shankar, K.; Kumar, S.; Dutta, A.K.; Alkhayyat, A.; Jawad, A.J.M.; Abbas, A.H.; Yousif, Y.K. An automated hyperparameter tuning recurrent neural network model for fruit classification. *Mathematics* **2022**, *10*, 2358. [CrossRef]
12. Gururaj, N.; Vinod, V.; Vijayakumar, K. Deep grading of mangoes using Convolutional Neural Network and Computer Vision. *Multimed. Tools Appl.* **2022**, 1–26. [CrossRef]
13. Koirala, A.; Walsh, K.; Wang, Z.; McCarthy, C. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. *Precis. Agric.* **2019**, *20*, 1107–1135. [CrossRef]
14. Patil, P.U.; Lande, S.B.; Nagalkar, V.J.; Nikam, S.B.; Wakchaure, G. Grading and sorting technique of dragon fruits using machine learning algorithms. *J. Agric. Food Res.* **2021**, *4*, 100118. [CrossRef]
15. Zhang, Z.; Wang, H.; Xu, F.; Jin, Y.Q. Complex-Valued Convolutional Neural Network and Its Application in Polarimetric SAR Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7177–7188. [CrossRef]
16. Hearst, M.; Dumais, S.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]

17. Tu, S.; Xue, Y.; Zheng, C.; Qi, Y.; Wan, H.; Mao, L. Detection of passion fruits and maturity classification using Red-Green-Blue Depth images. *Biosyst. Eng.* **2018**, *175*, 156–167. [CrossRef]
18. Tu, S.; Pang, J.; Liu, H.; Zhuang, N.; Chen, Y.; Zheng, C.; Wan, H.; Xue, Y. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. *Precis. Agric.* **2020**, *21*, 1072–1091. [CrossRef]
19. Lu, Y.; Wang, R.; Hu, T.; He, Q.; Chen, Z.S.; Wang, J.; Liu, L.; Fang, C.; Luo, J.; Fu, L.; et al. Nondestructive 3D phenotyping method of passion fruit based on X-ray micro-computed tomography and deep learning. *Front. Plant Sci.* **2023**, *13*, 1087904. [CrossRef]
20. Duangsuphasin, A.; Kengpol, A.; Rungsaksangmanee, P. The Design of a Deep Learning Model to Classify Passion Fruit for the Ageing Society. In Proceedings of the 2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C), Virtual, 4–5 August 2022; pp. 15–19.
21. Behera, S.K.; Rath, A.K.; Sethy, P.K. Fruits yield estimation using Faster R-CNN with MIoU. *Multimed. Tools Appl.* **2021**, *80*, 19043–19056. [CrossRef]
22. Maheswari, P.; Raja, P.; Apolo-Apolo, O.E.; Pérez-Ruiz, M. Intelligent fruit yield estimation for orchards using deep learning based semantic segmentation techniques—A review. *Front. Plant Sci.* **2021**, *12*, 684328. [CrossRef]
23. Renjith, P.N.; Muthulakshmi, A. Comprehensive Systematic Review on Fruit Maturity Detection Technique. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; pp. 1234–1240.
24. Tan, K.; Lee, W.S.; Gan, H.; Wang, S. Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes. *Biosyst. Eng.* **2018**, *176*, 59–72. [CrossRef]
25. Adak, M.F.; Yumusak, N. Classification of E-Nose Aroma Data of Four Fruit Types by ABC-Based Neural Network. *Sensors* **2016**, *16*, 304. [CrossRef] [PubMed]
26. Gill, H.S.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S.; Alassery, F. Fruit Image Classification Using Deep Learning. *CMC-Comput. Mater. Contin.* **2022**, *71*, 5135–5150. [CrossRef]
27. Joseph, J.L.; Kumar, V.A.; Mathew, S.P. Fruit Classification Using Deep Learning. In *Innovations in Electrical and Electronic Engineering*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 617–626.
28. 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; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
29. 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 (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
30. 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 (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
31. Koo, Y.; Kim, S.; Ha, Y.g. OpenCL-Darknet: Implementation and optimization of OpenCL-based deep learning object detection framework. *World Wide Web* **2021**, *24*, 1299–1319. [CrossRef]
32. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
33. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
34. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
35. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9–10.
36. Misra, D. Mish: A Self Regularized Non-Monotonic Neural Activation Function. *arXiv* **2019**, arXiv:abs/1908.08681.
37. Foret, P.; Kleiner, A.; Mobahi, H.; Neyshabur, B. Sharpness-Aware Minimization for Efficiently Improving Generalization. *arXiv* **2020**, arXiv:2010.01412.
38. 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 (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
39. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Dropblock: A regularization method for convolutional networks. *Adv. Neur.* **2018**, *31*.
40. Jocher, G.; Nishimura, K.; Mineeva, T.; Vilariño, R. YOLOv5 (2020). GitHub Repository. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 July 2023).
41. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
42. Mumuni, A.; Mumuni, F. Data augmentation: A comprehensive survey of modern approaches. *Array* **2022**, 100258. [CrossRef]
43. Kandel, I.; Castelli, M.; Manzoni, L. Brightness as an augmentation technique for image classification. *Emerg. Sci. J.* **2022**, *6*, 881–892. [CrossRef]
44. Gedraite, E.S.; Hadad, M. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In Proceedings of the ELMAR-2011, Zadar, Croatia, 14–16 September 2011; pp. 393–396.

45. Hussain, Z.; Gimenez, F.; Yi, D.; Rubin, D. Differential data augmentation techniques for medical imaging classification tasks. In Proceedings of the AMIA Annual Symposium Proceedings, American Medical Informatics Association, Washington, DC, USA, 4–8 November 2017; Volume 2017, p. 979.
46. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv* **2017**, arXiv:1712.04621.
47. 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 (CVPR), Virtual, 20–25 June 2021; pp. 13713–13722.
48. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
49. Huber, J.F. Mobile next-generation networks. *IEEE Multimed.* **2004**, *11*, 72–83. [CrossRef]
50. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
51. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 20–25 June 2021; pp. 16519–16529. [CrossRef]
52. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
53. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
54. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask scoring r-cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6409–6418.
55. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA 20–25 June 2009; pp. 248–255.
56. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **2015**, *115*, 211–252. [CrossRef]
57. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic Feature Pyramid Network for Object Detection. *arXiv* **2023**, arXiv:2306.15988.
58. 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 (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
59. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. Resnest: Split-attention networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022; pp. 2736–2746.
60. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neur.* **2015**, *28*, 1137–1149. [CrossRef] [PubMed]
61. Alimentarius, C. Standard for passion fruit Codex Stan 316-2014. In Proceedings of the Codex Committee on Fresh Fruits and Vegetables (18th Session), Phuket, Thailand, 24–28 February 2014.
62. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516.
63. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
64. 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. *arXiv* **2022**, arXiv:2207.02696.
65. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. GitHub Repository. Available online: https://github.com/ultralytics/ultralytics (accessed on 1 August 2023).
66. Yu, J.; Miao, W.; Zhang, G.; Li, K.; Shi, Y.; Liu, L. Target Positioning and Sorting Strategy of Fruit Sorting Robot Based on Image Processing. *Trait. Signal.* **2021**, *38*, 797–805. [CrossRef]