agronomy

MDPI

Article

Improved Feature Fusion in YOLOv5 for Accurate Detection and Counting of Chinese Flowering Cabbage (*Brassica campestris* L. ssp. *chinensis* var. *utilis* Tsen et Lee) Buds

Kai Yuan [1], Qian Wang [1], Yalong Mi [1], Yangfan Luo [1] and Zuoxi Zhao [1,2,*]

[1] College of Engineering, South China Agricultural University, Guangzhou 510642, China; 20211168010@stu.scau.edu.cn (K.Y.); wulaiqiufeng@stu.scau.edu.cn (Q.W.); miyalong@stu.scau.edu.cn (Y.M.); fit@stu.scau.edu.cn (Y.L.)

[2] Key Laboratory of Key Technology on Agricultural Machine and Equipment, South China Agricultural University, Ministry of Education, Guangzhou 510642, China

* Correspondence: zhao_zuoxi@scau.edu.cn; Tel.: +86-136-0004-9101

Abstract: Chinese flowering cabbage (*Brassica campestris* L. ssp. *chinensis* var. *utilis* Tsen et Lee) is an important leaf vegetable originating from southern China. Its planting area is expanding year by year. Accurately judging its maturity and determining the appropriate harvest time are crucial for production. The open state of Chinese flowering cabbage buds serves as a crucial maturity indicator. To address the challenge of accurately identifying Chinese flowering cabbage buds, we introduced improvements to the feature fusion approach of the YOLOv5 (You Only Look Once version 5) algorithm, resulting in an innovative algorithm with a dynamically adjustable detection head, named FPNDyH-YOLOv5 (Feature Pyramid Network with Dynamic Head-You Only Look Once version 5). Firstly, a P2 detection layer was added to enhance the model's detection ability of small objects. Secondly, the spatial-aware attention mechanism from DyHead (Dynamic Head) for feature fusion was added, enabling the adaptive fusion of semantic information across different scales. Furthermore, a center-region counting method based on the Bytetrack object tracking algorithm was devised for real-time quantification of various categories. The experimental results demonstrate that the improved model achieved a mean average precision (mAP@0.5) of 93.9%, representing a 2.5% improvement compared to the baseline model. The average precision (AP) for buds at different maturity levels was 96.1%, 86.9%, and 98.7%, respectively. When applying the trained model in conjunction with Bytetrack for video detection, the average counting accuracy, relative to manual counting, was 88.5%, with class-specific accuracies of 90.4%, 80.0%, and 95.1%. In conclusion, this method facilitates relatively accurate classification and counting of Chinese flowering cabbage buds in natural environments.

Keywords: Chinese flowering cabbage; spatial-aware; feature fusion; YOLOv5; object tracking

Citation: Yuan, K.; Wang, Q.; Mi, Y.; Luo, Y.; Zhao, Z. Improved Feature Fusion in YOLOv5 for Accurate Detection and Counting of Chinese Flowering Cabbage (*Brassica campestris* L. ssp. *chinensis* var. *utilis* Tsen et Lee) Buds. *Agronomy* **2024**, *14*, 42. https://doi.org/10.3390/agronomy14010042

Academic Editors: Luis Manuel Navas Garcia and Yanbo Huang

Received: 11 November 2023
Revised: 1 December 2023
Accepted: 16 December 2023
Published: 22 December 2023

1. Introduction

Chinese flowering cabbage (*Brassica campestris* L. ssp. *chinensis* var. *utilis* Tsen et Lee) is one of the specialty leafy vegetables in southern China, beloved for its rich nutritional content and tender texture, with a cultivation area reaching approximately 6.67 million hectares [1]. The timing of harvesting is crucial; immature Chinese flowering cabbage presents thin stems and small dimensions, while overripe Chinese flowering cabbage is unsuitable for transportation and falls short in taste and quality [2]. Therefore, it is necessary to detect the maturity of Chinese flowering cabbage before harvesting [3].

Traditional methods for assessing the maturity of leafy vegetables primarily involve farmers making subjective judgments based on the appearance, quality, and market demand [4]. Additionally, some destructive maturity tests, while relatively accurate, can only sample test vegetables after harvesting under laboratory conditions and may not truly

reflect the maturity of the entire plant population before harvest. In response, some non-destructive maturity detection methods have been proposed [5,6]. For instance, Michele et al. [7] designed a mechanically structured contact-type Radicchio maturity sensor. Radicchio at different maturity levels exhibits distinct heights, leading to varied displacements as the sensor passes through. Similarly, Birrell et al. [8] developed a vision system for selectively harvesting mature and disease-free iceberg lettuce. They initially employed YOLOv3 to detect the bounding boxes and positions of lettuce, subsequently feeding the bounding boxes into the Darknet network. The network classified them into three categories: mature, immature, and diseased, allowing for the exclusive harvesting of mature individuals.

The limited number of studies on the maturity detection of leafy vegetables by scholars can be attributed to the minimal differences between individuals at various maturity levels in vegetables, especially when compared to fruits [9]. There are fewer detectable characteristics in leafy vegetables, which poses challenges for non-destructive maturity detection research. Currently, there is no research on non-destructive detection of Chinese flowering cabbage maturity before harvest.

In accordance with the industry standards outlined in the Chinese flowering cabbage grading specifications, noticeable disparities in the morphology and color of cabbage buds at various maturity levels offer essential information for assessing their degree of maturity [10]. Maturity discrimination can be achieved by identifying and classifying the buds of Chinese flowering cabbage, which is essentially an object detection task in computer vision.

Research on image-based object detection technology in crop detection is already very extensive, with the main methods being traditional image processing and machine learning [11,12]. Traditional image processing predominantly relies on factors such as color [13], texture [14], and shape features [15] or incorporates methods that combine multiple features for crop recognition and detection [16,17]. Traditional machine learning methods typically involve a two-step process: manual extraction of target features followed by the use of trained classifiers for pixel-level classification to identify target regions [11,18]. However, it is important to note that traditional methods are susceptible to environmental variables, including fluctuations in lighting intensity, shooting angles and distances, and background variations [6].

In recent years, deep learning object detection methods based on convolutional neural networks have been widely applied in the agricultural domain [19,20]. Existing object detection methods can be broadly classified into one-stage and two-stage approaches. One-stage object detection models have faster detection speeds. In contrast, two-stage models are characterized by slower processing speeds and are often unsuitable for real-time detection scenarios [21]. Particularly, the introduction of visual attention mechanisms has enhanced the detection capability of small objects while ensuring inference speed [22,23]. For instance, Li et al. [24] incorporated the CBAM (Convolutional Block Attention Module) attention mechanism into the YOLOv5 model for wheat spike detection and counting. From the feature heatmap, it is evident that the model with the attention mechanism focuses more on targets, resulting in improved accuracy. Chen et al. [25] embedded the ECA (Efficient Channel Attention) attention mechanism into RetinaNet to detect the ripeness of pineapples, and the improved model achieved recognition accuracies exceeding 90% for pineapples at different ripeness levels.

For counting in continuous image sequences, it is crucial to avoid multiple counting of the same target. Object detection models typically lack access to temporal and target displacement information between frames, often resulting in repetitive counting of objects [26]. To mitigate this challenge, numerous studies have sought to combine object detection models with Multi-Object Tracking (MOT) algorithms, thereby enhancing counting accuracy. For instance, Wang et al. [27] introduced the MangoYOLO detection algorithm, which integrates Kalman filtering and the Hungarian algorithm to detect, track, and count fruit trees in video sequences. Li et al. [28] trained a YOLOv5 detection model for counting tea buds and implemented automatic counting using the improved DeepSORT algorithm. The

algorithm achieved a high correlation of 98% with manual counting results. Given the complexity of the Chinese flowering cabbage growth environment, with varying bud sizes, shapes, and growth positions at different levels of maturity. Consequently, the methods mentioned above may not be entirely suitable for this specific research context.

In summary, this study achieved real-time detection of Chinese flowering cabbage maturity in the field through bud detection. The main work and contributions of this study include: (1) Added a dynamically adjustable detection head with adaptive capabilities to the original YOLOv5 algorithm's neck, enhancing the feature fusion capability of the original algorithm and improving the detection accuracy of Chinese flowering cabbage buds. (2) Designed a center-region counting method based on the Bytetrack multi-object tracking algorithm, to some extent, avoiding the repeated counting of the same Chinese flowering cabbage buds. (3) Filled the gap in the field of pre-harvest maturity detection of Chinese flowering cabbage, providing references for the detection of other crops.

2. Materials and Methods

2.1. Image Acquisition and Data Sets

This study utilized RGB images and video data from two distinct varieties of Chinese flowering cabbage, namely "49-days" and "80-days". The data collection was carried out at the "QiLin" Experimental Farm of South China Agricultural University (113.38 E, 23.17 N). The data acquisition equipment consisted of two smartphones (Xiaomi 10 and Redmi K40 are both from Xiaomi Technology Co., Ltd., Beijing, China) and a SONI ILCE-5100 (Sony Group Corporation, Tokyo, Japan) digital camera, with pixel resolutions of 3000 × 4000, 5792 × 4344, and 2000 × 3008, respectively. The video data had a resolution of 1080 × 1920, and random single frames were extracted to construct the dataset. The images were captured from distances ranging from 30 to 80 cm from the top of the cabbage. The dataset included images under various lighting conditions and backgrounds with soil, totaling 4683 images. After removing blurry images, the dataset contained 4445 images. Among these, "49-days" images were taken 30~40 days after planting, totaling 1925 images, while "80-days" images were taken 50~60 days after planting, totaling 2520 images.

The images were annotated using Labelimg (Data annotation tools, v1.8.1, HumanSignal, SF, USA). In accordance with the industry standard for Chinese flowering cabbage, the maturity was categorized into three distinct stages: "growing", "ripe", and "over-ripe". This categorization was based on the extent of bud opening, as shown in Figure 1.

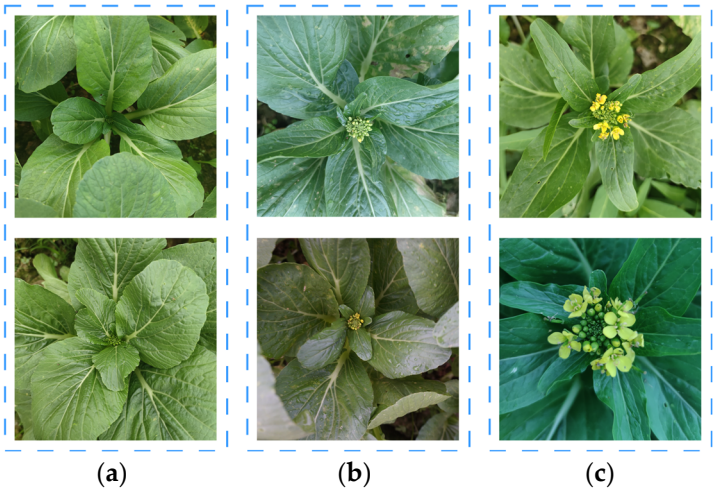

Figure 1. Three growth states of Chinese flowering cabbage buds. (**a**) Growing; (**b**) ripe; (**c**) over-ripe.

To ensure the quality and effectiveness of the annotations, objects that extended beyond 1/3 of the image borders were not included in the annotation process. Following annotation, XML files were generated to store category and target coordinate information.

The dataset was then divided into training, validation, and test sets in a ratio of 7:2:1, as depicted in Table 1.

Table 1. Dataset overview and description.

Set	Number of Images	Target Boxes			
		Ripe	Growing	Over-Ripe	Total
Train	3110	13,953	9317	4351	27,621
Validation	890	4245	2323	1125	7693
Test	445	2644	1506	558	4708
Total	4445	20,842	13,146	6034	40,022

2.2. Detection and Counting Method

This study can detect Chinese flowering cabbage buds in dynamic scenes, and the workflow is shown in Figure 2. The process of this method consists of three steps: (1) Utilizing the improved YOLOv5 algorithm to detect video frames captured by the camera, obtaining the location information and maturity categories in the current frame. (2) Jointly using the Bytetrack algorithm to track the targets and assign independent IDs. (3) When the motion of a target stabilizes and enters the central region, its category is determined, completing the counting.

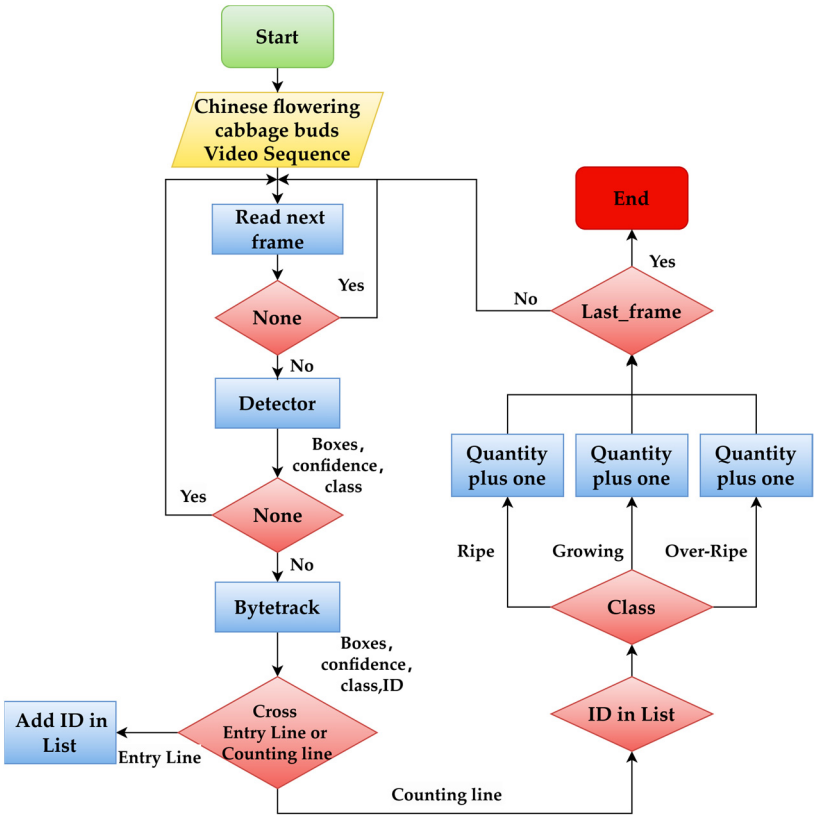

Figure 2. Overall workflow diagram.

2.2.1. Original YOLOv5

YOLOv5 is a one-stage object detection algorithm based on regression, comprising four main components: an input end, backbone, neck, and detection head [29]. The input end performs a series of preprocessing on the image, including online data augmentation, such as Mosaic, MixUp, and geometric transformations. The backbone is CSP-Darknet53, primarily composed of C3 modules utilizing the CSP (Cross Stage Partial) structure. A

C3 module, in turn, consists of three standard convolutional layers and Bottleneck units featuring residual connections [30]. The neck network adopts the Path Aggregation Network (PANet) to enhance the feature fusion capabilities of the network [31]. The detection head comprises three standard convolutional detectors, each responsible for detecting the feature maps at three different scales from the PANet output. It generates predictions for four position coordinates and class confidence scores.

YOLOv5 employs a matching strategy based on aspect ratios, which enables cross-anchor, cross-grid, and cross-branch predictions. This strategy substantially increases the number of positive samples, accelerates model convergence, and improves detection accuracy. During the inference process, non-maximum suppression is applied for post-processing to remove low-confidence boxes, resulting in the final detection results [32].

2.2.2. Bud Detection Method Based on Improved YOLOv5

This study addresses the detection task of Chinese flowering cabbage buds in continuous images in natural environments, which presents several challenges. On the one hand, the targets are relatively small, exhibiting diverse shapes, and the "growing" and "ripe" categories are highly similar, making it challenging to distinguish them from the background. Secondly, it is necessary to count various categories of buds in motion, which leads to image blurring and increases the difficulty of detection and continuous tracking [26].

The original algorithm's head directly processes the 3-scale feature maps from PANet through 1×1 convolutions to output prediction information of $na \times (nc + 5)$, where na represents the number of anchors, and nc is the number of classes. This type of head only considers the issue of different object scales in the images and does not address the problem of misidentification caused by different perspectives, leading to objects appearing in different shapes, rotations, and positions, reflecting a lack of spatial awareness. Additionally, YOLOv5's head unifies the tasks of classification and localization, even though these two tasks have entirely different objectives and constraints, resulting in a lack of task awareness. The original algorithm struggles to meet the task requirements, and tracking performance relies on the accuracy of the detector. Therefore, in this study, we made the following improvements to YOLOv5.

- Add a $4\times$ down-sampling layer.

On top of the original detection layers P3, P4, and P5 in the network, a P2 detection layer is added. P3, P4, and P5 correspond to 8, 16, and 32 times downsampling, while the P2 layer is derived from a $4\times$ downsampling feature map, which has a smaller receptive field, making it advantageous for detecting small-sized objects [33]. The specific operation involves upsampling the feature map of the 17th layer once and then concatenating it with the output of the 3rd layers in the backbone to form the network's P2 detection layer. Due to significant differences between the self-built dataset used in this study and the COCO (Common Objects in Context) dataset, and the addition of the P2 detection layer, the original Anchors are no longer applicable. Therefore, a re-clustering of Anchors was conducted to assign 3 new Anchors to the P2 detection layer. Ultimately, the sizes of the 12 groups of Anchors are as follows (((21,21), (31,31), (41,42)), ((63,64), (81,85), (109,106)), ((136,136), (171,169), (218,216)), ((310,275), (392,417), (576,522))).

- Feature fusion with a spatial-aware attention mechanism.

Expanding on the previously discussed enhancements, we introduced a unified object detection head framework named "DyHead" into the head network. DyHead integrates three distinct self-attention mechanisms: scale-awareness, spatial-awareness, and task-awareness [34]. In this study, the PANet structure in the YOLOv5 neck was removed because the spatial-awareness module relies on feature fusion, and only the FPN (Feature Pyramid Net) structure was retained, simplifying the model structure and preventing information loss caused by downsampling.

The P2, P3, P4, and P5 feature layers obtained from FPN first go through the spatial-aware module. As illustrated in Figure 3, taking the middle P3 layer as an example, it

initially undergoes a convolution operation to derive self-learned position biases (offset) and importance factors (mask). Additionally, the low-level feature layer P2 and high-level feature layer P4 undergo Deformable Convolution (DefV2-0, DefV2-1, and DefV2-2 represent three deformable convolution operations with different step sizes) to promote sparse attention learning. Subsequently, feature aggregation is performed at the same spatial positions with the middle feature layer (P3), resulting in temp_fea_High, temp_fea_Mid, and temp_fea_Low. The spatial-awareness module plays a vital role in introducing position biases to acquire deformation representation capabilities. Simultaneously, it utilizes an importance factor to adaptively weight the deformation-sampled positions, enabling the network to possess dynamic adjustment capability and better adapt to different forms of flowering bud targets.

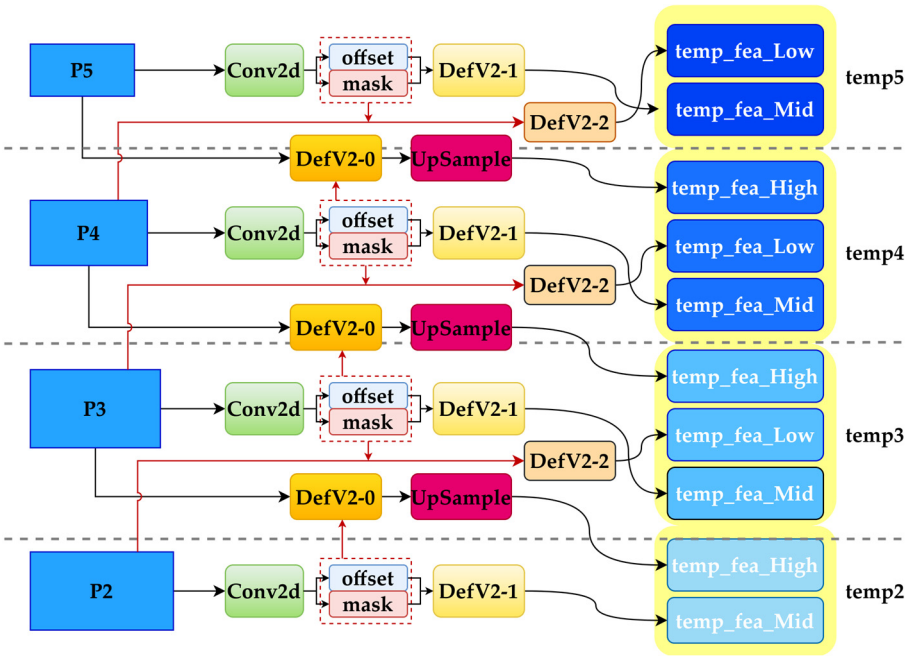

Figure 3. Spatial-aware module.

Following this, each temp undergoes a scale-awareness attention mechanism, as shown in Figure 4. The temps are first subjected to average pooling to compress the features and reduce the parameter count. Subsequently, they are connected to a fully connected layer (replaced by a 1×1 convolution), followed by a ReLu activation layer, and finally, a hard sigmoid activation layer to expedite training. Essentially, it assigns weights to feature layers at different levels, allowing the model to adaptively blend features based on the importance of features at that level.

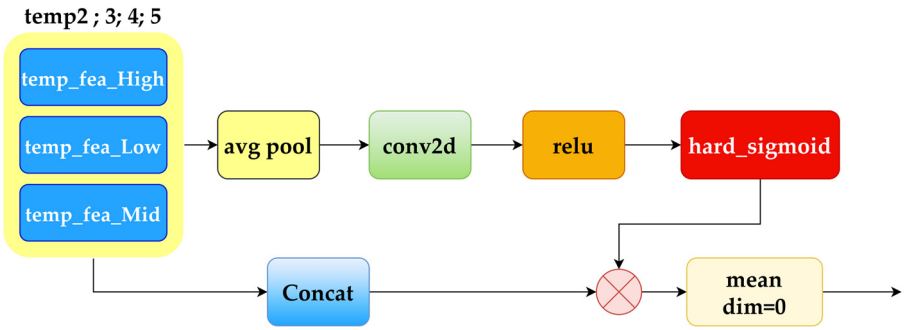

Figure 4. Scale-aware module (symbol $\otimes$ represents dot product).

Finally, there is the task-aware module, as shown in Figure 5, which adapts to the detection task by activating channels in the feature mapping to enhance performance. The

specific process involves reducing the feature dimension of input feature x through average pooling, followed by two fully connected layers and a normalization layer to map the feature space to the range of [–1, 1]. The functions implemented by these operations are similar to a hyper function $\theta(x)$ to generate four learnable parameters α_1, β_1, α_2, and β_2 for subsequent calculations [35]. Lastly, the activation function $f_\theta(x)$ is used to dynamically activate different channels of the input feature x, resulting in the final output of the task-aware block. More detailed information about the hyperfunction $\theta(x)$ and activation function $f_\theta(x)$ can be found in the literature [36].

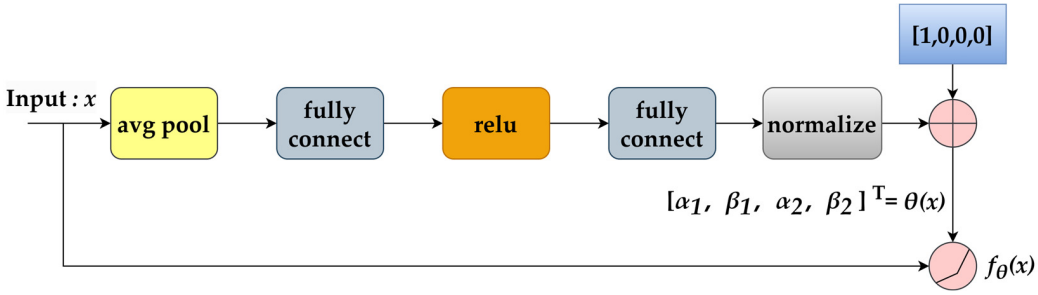

Figure 5. Task-aware module (symbol ⊕ represents addition).

The complete improved model is shown in Figure 6, and the improved sections are highlighted in pink. After passing through the dynamic detection head, four different-sized detection result maps are generated, with sizes of 320, 160, 80, and 40, corresponding to the detection of smaller, small, medium, and large targets. Classification and localization information is then obtained by subsequent standard convolution layers.

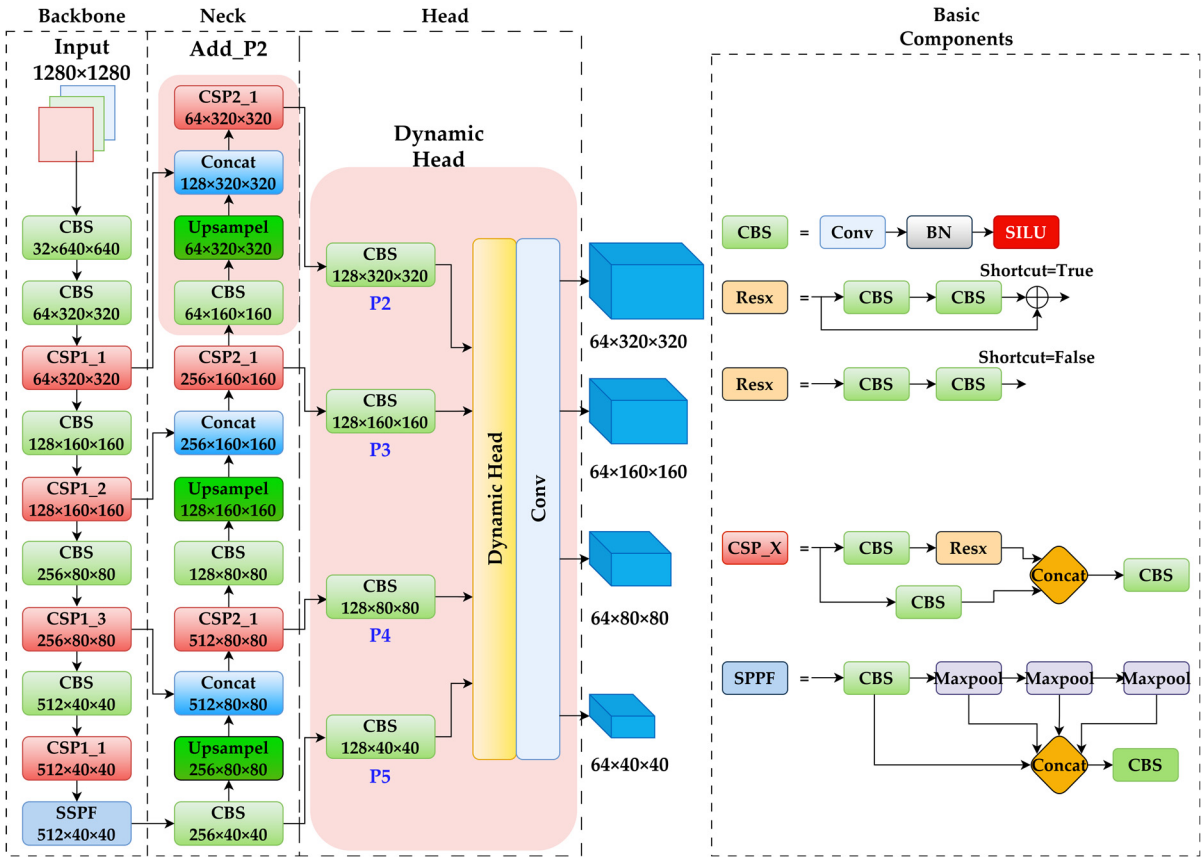

Figure 6. Complete model structure (symbol ⊕ represents addition).

2.2.3. Tracking and Counting Based on Bytetrack

Because of the image sensor's rapid acquisition speed, the same target may be captured in consecutive frames, resulting in the repeated counting of identical targets. To guarantee that each bud is counted only once, this study employs the Bytetrack object tracking algorithm. Bytetrack associates the detection results from YOLOv5 at different time points, assigning independent IDs to each target and enabling the accurate counting of Chinese flowering cabbage buds. The specific method is elucidated below.

Bytetrack is a detection-based tracking algorithm that employs a data association method called Byte [37]. Instead of simply discarding low-confidence detection results, it segregates the target detection boxes into low-score boxes and high-score boxes based on confidence scores and processes them separately. When using Bytetrack to track Chinese flowering cabbage buds, the initial step is to divide the detection results into high-score detection boxes and low-score detection boxes based on a confidence threshold and create corresponding trajectories. The first matching is performed using high-score boxes and track, with IoU (Intersection over Union) as the only similarity calculation, significantly improving matching speed. Unmatched high-score boxes and unmatched high-score trajectories (U_track) are retained. The second matching is performed using low-score boxes and U_track, with the retention of U_track continuing [37]. At this point, background false detections can be filtered out as they lack corresponding trajectories. Meanwhile, obscured targets can be recovered. Unmatched trajectories are retained for a certain lifespan, and if no boxes are matched during this period, they are deleted. For high-score boxes that are not matched to trajectories, they continue to be observed in frames, and if they are continuously detected, trajectories are assigned.

During the utilization of Bytetrack for bud tracking, the movement of the camera may cause detected buds to shift from the image's edges to its center. This can result in significant alterations in the aspect ratio of the detection boxes [26]. Consequently, the high-score detection boxes in the current frame may not be correctly matched with the previously assigned trackers. This situation leads to a modification in the assigned ID for the same flower bud, ultimately impacting counting accuracy. Therefore, this study does not directly use the IDs generated by the Bytetrack algorithm for counting. Instead, a counting method was devised based on a center region, as shown in Algorithm 1.

Firstly, two fixed-width and positioned lines are set at the center of the video window, as shown in Figure 7, one as the entry line and the other as the counting line, capable of adapting to the counting of targets moving in two different directions, up and down. Taking the example of targets moving from bottom to top, as the camera moves, the targets continuously tracked by Bytetrack gradually move toward the center of the image. The system checks whether the current target's center point (x^i,y^i) has crossed the entry line. If the target ID is not stored, it adds the target ID to the $Array_{down}$ storage list. As the targets continue to move upwards, their center points cross the counting line, and it checks whether the target ID appears in the $Array_{down}$ storage list. If it exists, it accumulates the total count of targets and determines the target class, achieving the counting of flower buds at different stages of maturity. This counting method only counts the target when it is fully visible, and the aspect ratio does not undergo significant changes, ensuring a certain level of stability and accuracy in target ID and counting.

Algorithm 1. A Special Tracking and Counting Method

Input: id^i; $class^i$; (x^i,y^i);
Output: $Array_{top}$; $Array_{down}$; *Total*; *nclass0*; *nclass1*; *nclass2*
Constant: $(0,y_{top1})$; $(width,y_{top2})$; $(0,y_{down1})$; $(width,y_{down2})$
if $0 \le x^i \le width$ **and** $y_{top1} \le y^i \le y_{top2}$ **then**
  if id^i **not in** $Array_{top}$ **then**
    id^i **add into** $Array_{top}$
    if id^i **in** $Array_{down}$ **then**
      $Total \leftarrow Tatal + 1$
      if $class^i == 0$ **then**
        $nclass0 \leftarrow nclass0 + 1$
      if $class^i == 1$ **then**
        $nclass1 \leftarrow nclass1 + 1$
      if $class^i == 2$ **then**
        $nclass2 \leftarrow nclass2 + 1$
      $Array_{down}$ *remove* id^i
else if $0 \le x^i \le width$ **and** $y_{down1} \le y^i \le y_{down2}$ **then**
  if id^i **not in** $Array_{down}$ **then**
    id^i **add into** $Array_{down}$
    if idi **in** $Array_{top}$ **then**
      $Tatal \leftarrow Tatal + 1$
      if $class^i == 0$ **then**
        $nclass0 \leftarrow nclass0 + 1$
      if $class^i == 1$ **then**
        $nclass1 \leftarrow nclass1 + 1$
      if $class^i == 2$ **then**
        $nclass2 \leftarrow nclass2 + 1$
      $Array_{top}$ *remove* id^i
end if
return *total*, *nclass0*, *nclass1*, *nclass2*

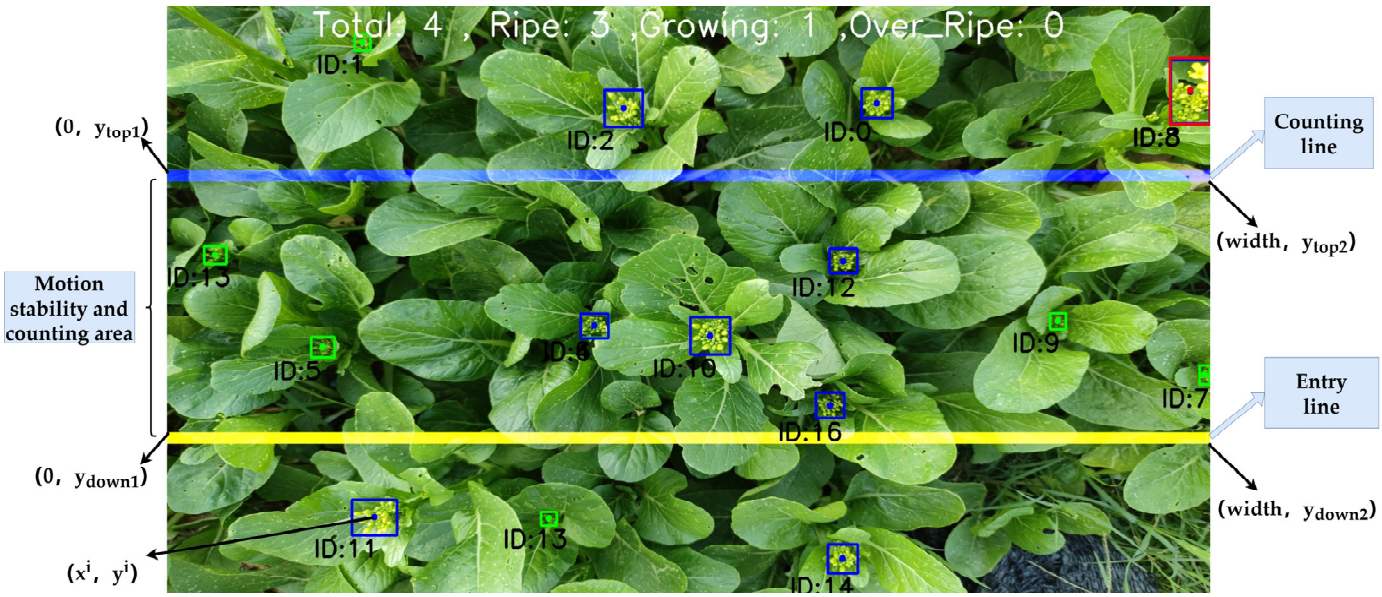

Figure 7. An example of the process of flower buds counting.

2.3. Evaluation Metrics

To validate the effectiveness of our improvement method, *Precision*, *Recall*, *mAP*, and *F*1 metrics are used to evaluate model performance. The calculation of these metrics is as shown in the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$AP = \int_0^1 Precision \cdot Recall \, dr \tag{3}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{N} \tag{4}$$

Precision represents the ratio of correctly detected results to the total detected results, while recall is defined as the proportion of correctly detected results among all true results, as shown in Equations (1) and (2). Where *TP* (True Positives) represents the number of correctly predicted positive class bounding boxes, *FN* (False Negatives) represents the number of positive class bounding boxes that were missed by the model, and *FP* (False Positives) represents the number of incorrectly predicted positive class bounding boxes. Compared to precision and recall, Average Precision (*AP*) can more comprehensively reflect the overall detection performance of a model [38]. *mAP* is the average of *AP* for each class, where *AP* is defined as the area under the precision–recall curve. *mAP@0.5* represents the *mAP* calculated at an IoU of 0.5, and *mAP@0.5:0.95* denotes the average *mAP* calculated with IoU thresholds moving from 0.5 to 0.95 at intervals of 0.05 [39]. *N* stands for the number of classes, which is equal to 3 in this study.

3. Results

3.1. Model Training Results and Ablation Experiment

Table 2 presents the software and hardware configuration used for model training. During training, pretrained weights were employed to enhance the model's initial performance. The batch size was fixed at eight, and the input image size was set to 1280 × 1280. Stochastic Gradient Descent (SGD) was used to update the network parameters with a momentum of 0.937 and weight decay of 0.0005. The learning rate was updated using linear decay, where the initial learning rate (lr) was set to 0.01, and the decay factor was set to 0.01.

Table 2. Experimental configuration.

Configuration	Parameter
Development environment	Anaconda3-2021.11 + Pycharm (v 2022.1.3)
CPU (Central Processing Unit)	Intel Core i9-11900 K
GPU (Graphic Processing Unit)	Nvidia GeForce RTX 3060Ti
Operating system	Windows 10
Accelerated environment	CUDA11.3 CUDNN8.3.0
Development of language	Python3.8

Figure 8 shows the changes in model metrics and losses after 100 training epochs. It is evident that during the initial 20 epochs of training, the model exhibited rapid convergence. This phase was characterized by substantial reductions in location loss, object loss, and class loss for both the training and validation datasets, along with a notable increase in model precision and recall. However, after 50 epochs, the model's progress began to stabilize. To

ensure optimal model performance, an early stopping strategy was used [40]. The training process was concluded at the 100th epoch. Specifically, we saved the model that achieved the highest sum of precision, recall, mAP@0.5, and mAP@0.5:0.95 scores. These scores were weighted using coefficients (0, 0, 0.1, and 0.9) to prioritize critical metrics.

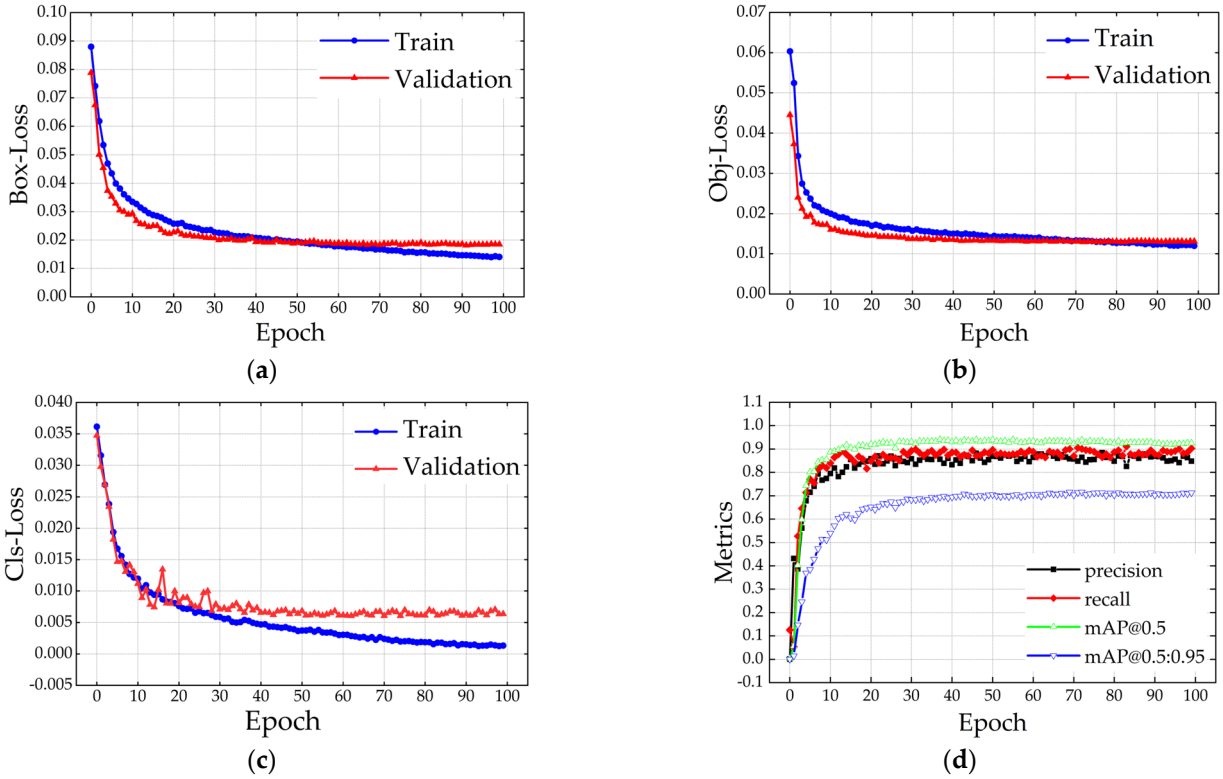

Figure 8. FPNDyH−Yolov5′s metrics variation during training. (**a**) Boxes loss; (**b**) object loss; (**c**) classification loss; (**d**) model metrics.

Figure 9 illustrates the results of evaluating our improved model using the test dataset. Figure 8a represents the PR curve of the model at an IoU threshold of 0.5, and the area enclosed by the curve is the Average Precision (AP). The AP values for different categories of flower buds are as follows: growing: 86.9%, ripe: 96.1%, over-ripe: 98.7%. Notably, the model achieves the highest detection precision for over-ripe buds, which is attributed to their fully open state, distinct yellow color, and high contrast with the background. In contrast, growing buds, surrounded by leaves and with a color similar to the background, exhibit relatively lower detection performance. Ripe buds, characterized by a pale yellow color and plump morphology, perform well in terms of detection. Figure 8b–d represents precision, recall, and F1 curves at different confidence levels, respectively. These curves collectively reveal the model's excellent performance on the test set, indicating strong fitting and generalization capabilities for all three maturity levels of cabbage. Examples of six detection results are shown in Figure 10, only a few targets of growing were missed (blue circle in Figure 10a,c), and the rest were detected correctly, indicating that the model performs well.

Ablation experiments involve removing or adding certain structures of the detection algorithm to observe their impact on performance [41]. In order to validate the effectiveness of the improved model, the ablative experiments were conducted on FPNDyH-YOLOv5. YOLOv5s without PANet were used as the baseline, retaining only the FPN. Ablation experiments were performed by adding the P2 detection layer and utilizing the spatial-aware attention mechanism, scale-aware attention mechanism, and task-aware attention mechanism. We considered precision and recall at the point of the maximum F1 value. The

experimental results are presented in Table 3, where "√" and "-" represent the selected and unselected methods, respectively.

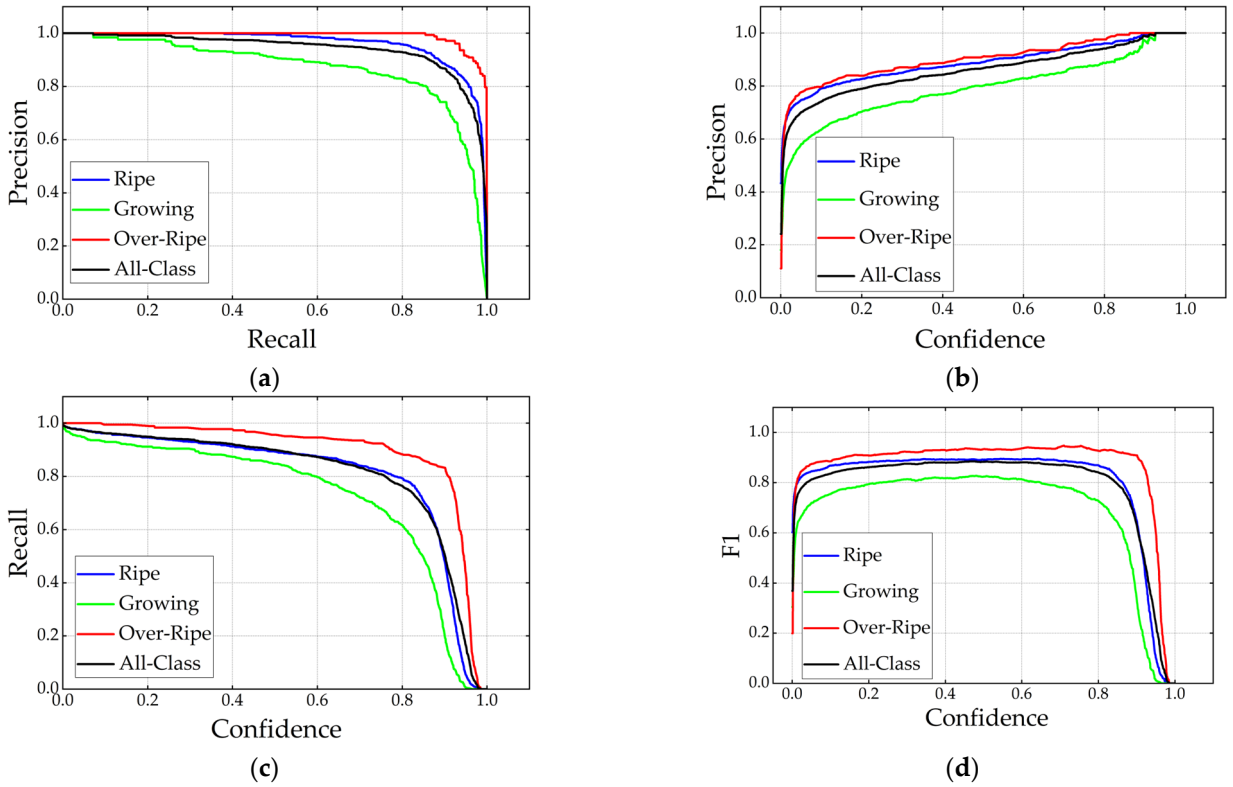

Figure 9. The performance of the improved model on the test set. (**a**) Precision–recall; (**b**) precision–confidence; (**c**) recall–confidence; (**d**) F1–confidence.

It can be observed that after adding the P2 detection layer to the FPN structure, the recall rate increased by 1.0%. The P2 detection layer only undergoes a 4× down-sampling, making it more sensitive to small object detection, resulting in more true positives being correctly detected. However, it also leads to many false-positive targets, causing a slight decrease in precision. After incorporating the spatial aware attention mechanism, all model metrics showed a significant improvement, with a direct increase of 3.9% in mAP@0.5. With the addition of scale awareness and task awareness, the model metrics further improved, although to a lesser extent. This suggests that the spatial awareness module played a dominant role [34]. Ultimately, the model's performance improved by 3.0%, 4.6%, and 5.2% compared to the baseline model when all three attention mechanisms were combined. To provide a more intuitive demonstration of the effectiveness of each attention module, we conducted visualizations of the feature maps of the same channel after the action of each module, as shown in Figure 11. It is clearly visible that the P2 detection layer contains more small targets. After passing through each attention module, the target features are enhanced, demonstrating the effectiveness of the improvement.

Table 3. Ablation experiments result.

Add_P2	Spatial-Aware Attention	Scale-Aware Attention	Task-Aware Attention	Precision (%)	Recall (%)	mAP@0.5 (%)
-	-	-	-	83.5	85.5	88.7
√	-	-	-	83.3	86.5	89.2
√	√	-	-	85.5	88.3	92.6
√	√	√	-	85.9	89.7	93.4
√	√	√	√	86.5	90.1	93.9

Where "√" and "-" represent the selected and unselected methods, respectively.

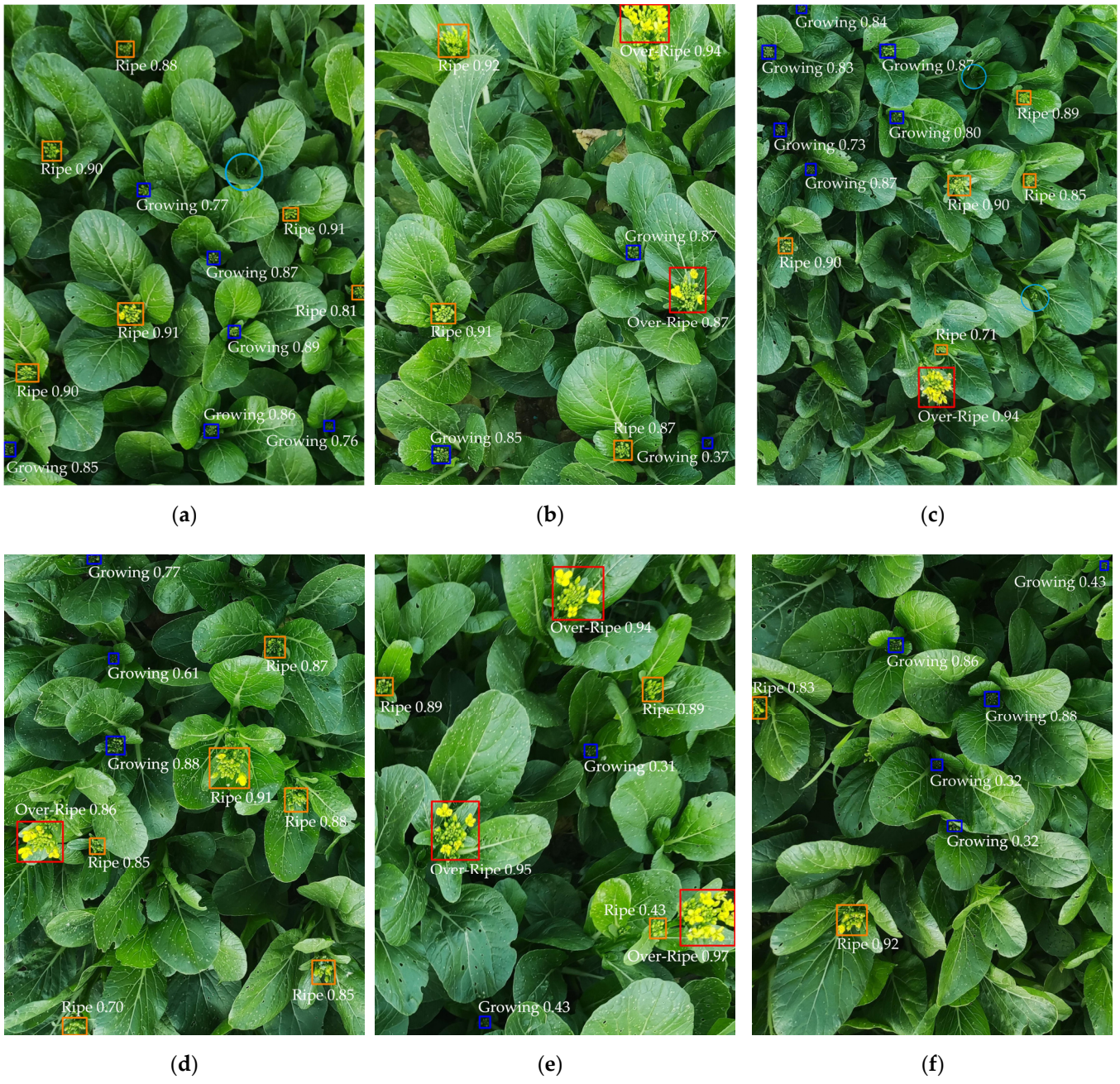

Figure 10. Examples of detection results. Confidence interval range: (**a**) growing: 0.77~0.89; ripe: 0.81~0.91; (**b**) growing: 0.37~0.87; ripe: 0.87~0.92; over-ripe: 0.87, 0.94; (**c**) growing: 0.73~0.87; ripe: 0.71~0.90; over-ripe: 0.94; (**d**) growing: 0.61~0.88; ripe: 0.70~0.91; over-ripe: 0.86; (**e**) growing: 0.31~0.43; ripe: 0.43~0.89; over-ripe: 0.94~0.97; (**f**) growing: 0.32~0.88; ripe: 0.83, 0.92.

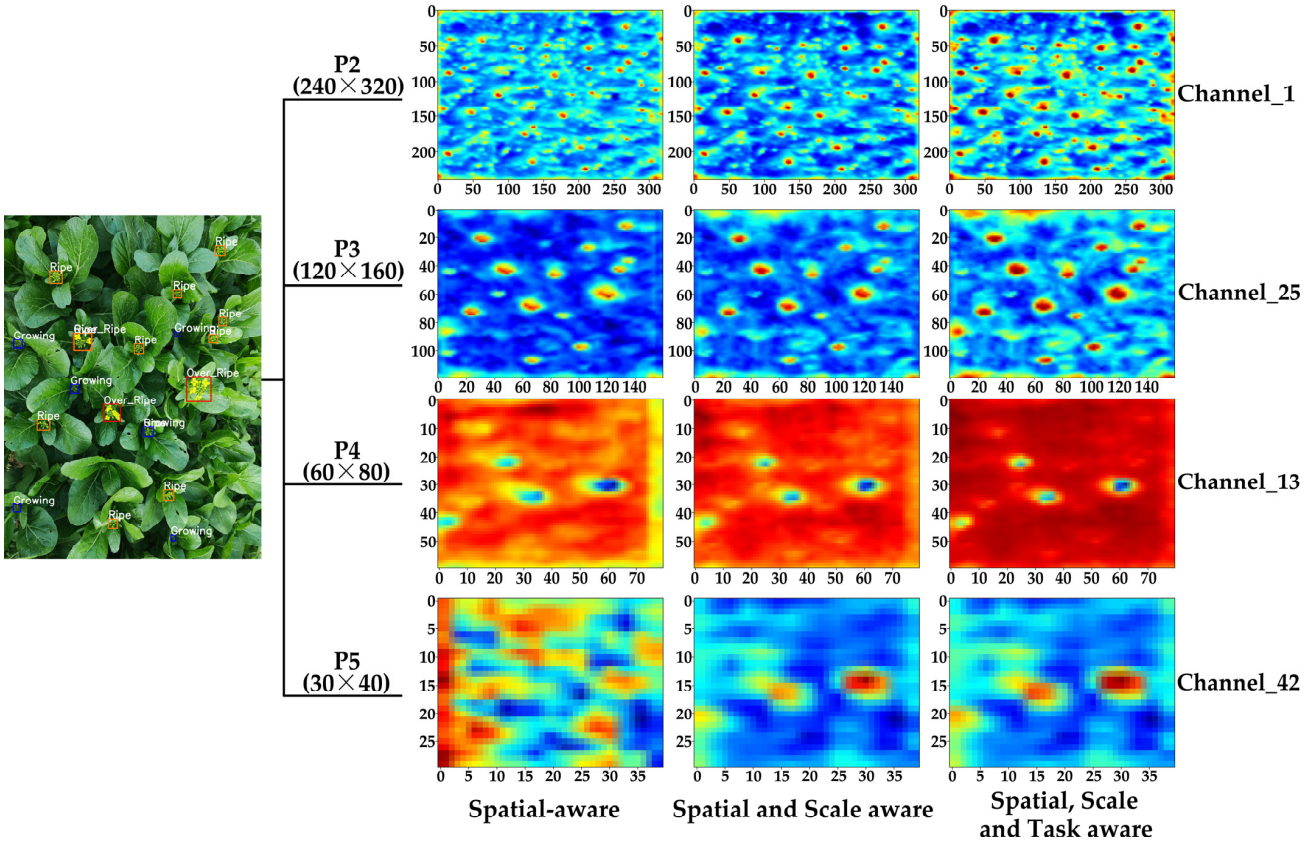

Figure 11. Feature heatmap.

3.2. Experiment on Different Feature Fusion Methods

This study conducted a comprehensive comparison of various feature fusion methods, building upon the addition of the P2 detection layer. Each method is denoted as follows: A represents the FPN structure, B represents the original PANet of YOLOv5, C represents the modified BiFPN for PANet, D represents the direct fusion of the output layers of the backbone using DyHead, E represents the FPN combined with DyHead, and F represents the combination of PANet with DyHead. Precision and recall were evaluated at the point of the maximum F1 value, and the experimental results are shown in Table 4. The results reveal that methods D, E, and F, which incorporate DyHead for dynamic feature fusion, outperform models that do not utilize them. Notably, our proposed method E (FPNDyH) achieves the highest mAP at an IoU threshold of 0.5, surpassing the original algorithm using method B (PANet) by 2.5 percentage points. While the precision of method E is slightly lower than that of method B, it boasts a remarkable 3.3% increase in recall. Models A and D have fewer parameters, but their metrics are inferior to E. In comparison to B, C, and F, our method exhibits the best performance with the least number of parameters and computational burden. In summary, our proposed FPNDyH method is more suitable for detecting Chinese flowering cabbage buds.

Table 4. Experimental results for different feature fusion methods.

Method	Layers	Parameters	Flops	Precision (%)	Recall (%)	map@0.5 (%)
A	194	4,837,152	58.0 G	83.3	86.5	89.2
B	260	7,172,000	75.2 G	87.1	86.8	91.4
C	278	7,231,214	77.6 G	85.4	87.2	90.3
D	165	4,486,860	48.8 G	86.3	88.2	91.6
E	231	5,003,980	63.8 G	86.5	90.1	93.9
F	297	7,349,068	81.0 G	86.2	89.8	92.7

3.3. Compared with Other Detection Models

To further verify the performance of the improved algorithm in Chinese flowering cabbage bud detection, we trained five typical object detection algorithms: SSD, Faster R-CNN, YOLOv4, YOLOX-s, and YOLOv7, and conducted comparative experiments using the same dataset. Several performance metrics were used, including precision, recall, parameters, flops, and mAP@0.5. The results are summarized in Table 5. Faster-RCNN is a typical two-stage object detection algorithm with the largest number of parameters and computational requirements and exhibits the poorest performance. The other algorithms are one-stage detection algorithms. SSD shows relatively weak performance in detecting small objects [42]. YOLOv4, YOLOX, and YOLOv7 are algorithms from the same YOLOv5 family and perform reasonably well, but they are all slightly inferior to our proposed FPNDyH-YOLOv5. A more intuitive comparison can be seen in Figure 12, which clearly illustrates that FPNDyH-YOLOv5 outperforms other algorithms across various metrics while also having the smallest number of parameters and computational requirements.

Table 5. Comparison with different models.

Model	Parameters	Flops	Precision (%)	Recall (%)	map@0.5 (%)
SSD-VGG	23,879,570	1.096 T	83.6	87.5	89.6
Faster-RCNN	136,729,994	1.176 T	81.6	85.3	87.2
YOLOv4	63,948,456	567.74 G	86.5	89.1	90.7
YOLOX-S	8,938,456	107.045 G	83.5	87.3	91.0
YOLOv7	9,324,824	106.7 G	85.9	89.6	92.6
FPNDyH-YOLOv5	5,003,980	63.8 G	86.5	90.1	93.9

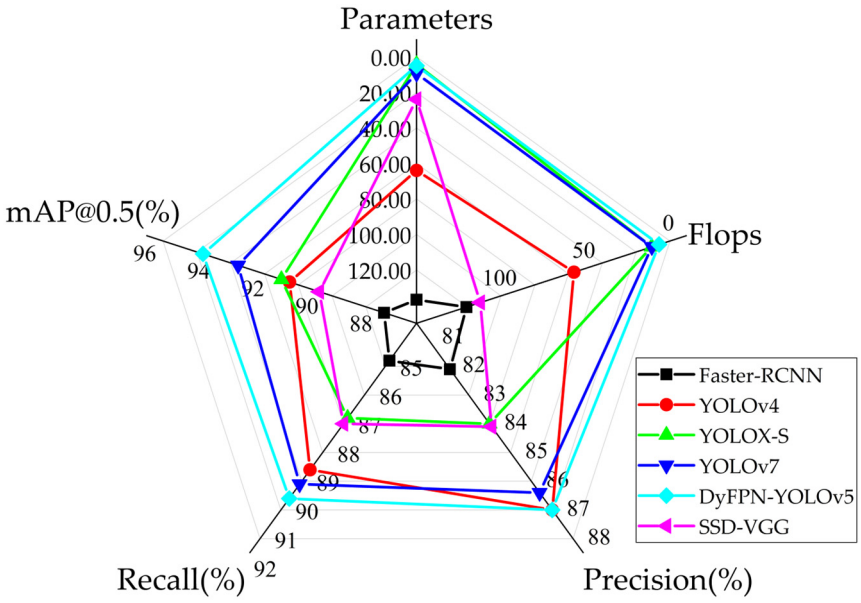

Figure 12. Performance comparison of different detection algorithms.

3.4. Analysis of Tracking and Counting Results

In this study, the trained FPNDyH-YOLOv5 combined with Bytetrack is used to achieve real-time detection and counting of Chinese flowering cabbage buds. The FPNDyH-YOLOv5 model is exported in ONNX (Open Neural Network Exchange) format, and after several experimental optimizations, the NMS (Non-Maximum Suppression) IoU threshold was set to 0.45, and the confidence threshold was set to 0.25; Bytetrack parameters include a tracking threshold of 0.2 and a matching threshold of 0.5. To validate the effectiveness of this method, five video segments were randomly selected from the test dataset. The videos had a resolution of 1080 × 1920 and ran at 30 frames per second. Each video segment contained vegetable beds approximately 0.8 m wide and 2.0 m long. The number and total of plants

at different maturities were manually counted and compared with the algorithm's results. The results are shown in Table 6. The counting accuracy for the ripe, growing, and over-ripe categories was found to be 90.4%, 80.0%, and 95.1%, respectively. The lower counting accuracy of growing is mainly attributed to the small size of the objects in this category, making them challenging to distinguish from the background. The counting accuracy of the ripe category is moderate. The count of the over-ripe category is higher than manual counting, primarily due to false detections of the ripe category by the detector. Overall, this method can achieve a relatively good counting of Chinese flowering cabbage buds.

Table 6. Counting result.

Video	Manual Counting				Algorithm Counting			
	Ripe	Growing	Over-Ripe	Total	Ripe	Growing	Over-Ripe	Total
1	87	37	21	145	79	30	20	129
2	70	42	14	126	64	35	16	115
3	68	36	15	119	61	27	14	102
4	73	35	20	128	64	26	22	112
5	76	40	18	134	70	34	20	124
Total	374	190	88	652	338	152	92	582
Average counting accuracy (%)					90.4	80	95.1	88.5

4. Discussion

In this study, the primary objective is category-based counting, and the accurate detection and classification of the detector are crucial for achieving correct counts [43]. To this end, this study has focused on improving the detector. The incorporation of three types of attention mechanisms has resulted in noticeable improvements in the performance of the detector. The spatial-aware attention module plays a significant role [34]. This is mainly because the deformable convolution enhances the deformation representation ability of the model [44]. It is invariant to spatial transformation and aggregates the features of each scale at the same spatial position, fully integrating the information of each scale. However, the improvement for the "growing" category is not substantial, with a precision of only 86.9%. This is mainly due to the immature flower buds being concealed among the plant's stems and leaves and their very small size, making them observable only from a vertical overhead view. This has also resulted in the lowest accuracy for the "growing" category during tracking and counting. Recognizing that relying solely on algorithmic improvements may be insufficient to address this issue, in our next step, we consider utilizing the height difference information between flower buds and leaves. We will use an RGB-D camera to acquire depth images for 4D input model training [26]. By incorporating depth information, we anticipate that the model will be better equipped to detect and distinguish "growing" flower buds even when they are partially obscured by plant structures or have a smaller visible profile. This enhancement should contribute to achieving more accurate and reliable counting results for Chinese flowering cabbage buds [45,46].

During the tracking and counting process, this study did not rely on maximum ID but rather designed a method based on the central region that considers the actual scenario. This method counts the Chinese flower cabbage buds only when the target moves to the central region of the image (i.e., in a vertical overhead view), to some extent ensuring counting accuracy. However, it is important to note that this method has some limitations. Specifically, the width of the line cannot be adaptively adjusted to match different movement speeds. In other words, if a target moves too fast, with a displacement large enough to skip across the count lines between consecutive frames, it can lead to inaccurate counting. In addition, the distance between the two lines will also affect the counting. Theoretically, the smaller the distance, the more accurate the counting will be. This is because the Kalman filter in the Bytetrack is a linear model, and its premise is that the target is constant speed. This means that as the displacement of the target decreases, the

probability of ID change decreases, but if the distance between the two lines is too small, it becomes difficult to adapt to faster speeds. In general, it is necessary to experimentally select appropriate parameters to balance speed and accuracy, and within a certain range of movement speeds, our method remains effective.

In future research, we plan to explore the development of a lightweight model suitable for deployment on edge computing devices [33]. This would enable real-time tracking and counting of Chinese flowering cabbage buds in practical agricultural settings, further enhancing the applicability of our approach.

5. Conclusions

This study achieved real-time non-destructive detection of Chinese flowering cabbage maturity before harvest. Introducing spatial-aware, scale-aware, and task-aware attention mechanisms into YOLOv5, we proposed the FPNDyH feature fusion method, named FPNDyH-YOLOv5. The improved algorithm demonstrated impressive precision, recall, and mAP@0.5 scores of 86.5%, 90.1%, and 93.9%, respectively, surpassing the original algorithm. Based on the ablation experiment, we visualized the feature maps after the three attention mechanisms were applied. It can be intuitively seen that the improved model pays more attention to the small target. Compared to other models, it has fewer parameters and computations, with superior performance in all metrics. Utilizing the trained detection model combined with the Bytetrack algorithm, we designed a central region counting method for real-time tracking and counting of various targets. In test videos, the counting accuracy for each category was 90.4%, 80%, and 95.1%, respectively. The results indicate the effectiveness of the method in detecting and counting Chinese flowering cabbage buds at different maturity stages. This study also provides a non-manual solution for timely harvest assessment of other crops, contributing to advancements in agricultural practices.

Author Contributions: Conceptualization, K.Y.; methodology, K.Y.; software, K.Y.; validation, K.Y., Q.W., Y.M. and Y.L.; formal analysis, K.Y.; investigation, K.Y., Q.W. and Y.M.; resources, Z.Z.; data curation, Q.W.; writing—original draft preparation, K.Y.; writing—review and editing, Z.Z.; visualization, K.Y.; supervision, Z.Z.; project administration, Z.Z.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of this study from the State Key Research Program of China (Grant No. 2022YDF2001901-01), the Guangdong Provincial Department of Agriculture's Modern Agricultural Innovation Team Program for Animal Husbandry Robotics (Grant No. 2019KJ129), and the Special Project of Guangdong Provincial Rural Revitalization Strategy in 2020 (YCN (2020) No. 39) (Fund No. 200-2018-XMZC-0001-107-0130).

Data Availability Statement: Data are contained within the article.

Acknowledgments: Thank you to the management personnel of QiLin Farm for providing us with an experimental site for our research.

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

References

1. Kleiber, T.; Kleiber, T.; Liu, W.; Liu, Y. A review of progress in current research on Chinese flowering cabbage (*Brassica campestris* L. ssp. chinensis var. utilis Tsen et Lee). *J. Elem.* **2021**, *26*, 149–162. [CrossRef]
2. Hongmei, X.; Kaidong, Z.; Linhuan, J.; Yuanjie, L.; Wenbin, Z. Flower bud detection model for hydroponic Chinese kale based on the fusion of attention mechanism and multi-scale feature. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 161–168. [CrossRef]
3. Gil, M.; Tudela, J.; Martínez-Sánchez, A.; Luna, M. Harvest maturity indicators of leafy vegetables. *Stewart Postharvest Rev.* **2012**, *8*, 1–9. [CrossRef]
4. Yiyu, J.; Shuo, W.; Lina, Z.; Yu, T. Maturity classification using mechanical characteristics of hydroponic lettuce. *Trans. Chin. Soc. Agric. Eng.* **2023**, *39*, 179–187. [CrossRef]
5. Cakmak, H. 10-Assessment of fresh fruit and vegetable quality with non-destructive methods. In *Food Quality and Shelf Life*; Galanakis, C.M., Ed.; Academic Press: New York, NY, USA, 2019; pp. 303–331. [CrossRef]

6. Mahanti, N.K.; Pandiselvam, R.; Kothakota, A.; Ishwarya, S.P.; Chakraborty, S.K.; Kumar, M.; Cozzolino, D. Emerging non-destructive imaging techniques for fruit damage detection: Image processing and analysis. *Trends Food Sci. Technol.* **2022**, *120*, 418–438. [CrossRef]
7. Antonelli, M.G.; Zobel, P.B.; Durante, F.; Raparelli, T. Development of an Automated System for the Selective Harvesting of Radicchio. *Int. J. Auto. Technol.-Jpn.* **2017**, *11*, 415–424. [CrossRef]
8. Birrell, S.; Hughes, J.; Cai, J.Y.; Iida, F. A field-tested robotic harvesting system for iceberg lettuce. *J. Field Robot.* **2020**, *37*, 225–245. [CrossRef]
9. Cakmak, H.; Sogut, E. Imaging Techniques for Evaluation of Ripening and Maturity of Fruits and Vegetables. In *Nondestructive Quality Assessment Techniques for Fresh Fruits and Vegetables*; Pathare, P.B., Rahman, M.S., Eds.; Springer Nature: Singapore, 2022; pp. 35–59. [CrossRef]
10. *NY/T 1647-2008*; Guangzhou Ministry of Agriculture Vegetable and Fruit Quality Supervision, I.A.T.C. Grades and Specifications of Flowering Chinese Cabbage. Industry Standards-Agricultural: Beijing, China, 2008.
11. Xiao, F.; Wang, H.; Li, Y.; Cao, Y.; Lv, X.; Xu, G. Object Detection and Recognition Techniques Based on Digital Image Processing and Traditional Machine Learning for Fruit and Vegetable Harvesting Robots: An Overview and Review. *Agronomy* **2023**, *13*, 639. [CrossRef]
12. Mamat, N.; Othman, M.F.; Abdoulghafor, R.; Belhaouari, S.B.; Mamat, N.; Hussein, S.F.M. Advanced Technology in Agriculture Industry by Implementing Image Annotation Technique and Deep Learning Approach: A Review. *Agriculture* **2022**, *12*, 1033. [CrossRef]
13. Yu, X.; Fan, Z.; Wang, X.; Wan, H.; Wang, P.; Zeng, X.; Jia, F. A lab-customized autonomous humanoid apple harvesting robot. *Comput. Electr. Eng.* **2021**, *96*, 107459. [CrossRef]
14. Hameed, K.; Chai, D.; Rassau, A. Texture-based latent space disentanglement for enhancement of a training dataset for ANN-based classification of fruit and vegetables. *Inf. Process. Agric.* **2023**, *10*, 85–105. [CrossRef]
15. Lin, G.; Tang, Y.; Zou, X.; Cheng, J.; Xiong, J. Fruit detection in natural environment using partial shape matching and probabilistic Hough transform. *Precis. Agric.* **2020**, *21*, 160–177. [CrossRef]
16. Septiarini, A.; Sunyoto, A.; Hamdani, H.; Kasim, A.A.; Utaminingrum, F.; Hatta, H.R. Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features. *Sci. Hortic.* **2021**, *286*, 110245. [CrossRef]
17. Bhargava, A.; Bansal, A. Classification and grading of multiple varieties of apple fruit. *Food Anal. Method.* **2021**, *14*, 1359–1368. [CrossRef]
18. Hua, X.; Li, H.; Zeng, J.; Han, C.; Chen, T.; Tang, L.; Luo, Y. A Review of Target Recognition Technology for Fruit Picking Robots: From Digital Image Processing to Deep Learning. *Appl. Sci.* **2023**, *13*, 4160. [CrossRef]
19. Akkem, Y.; Biswas, S.K.; Varanasi, A. Smart farming using artificial intelligence: A review. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105899. [CrossRef]
20. Darwin, B.; Dharmaraj, P.; Prince, S.; Popescu, D.E.; Hemanth, D.J. Recognition of Bloom/Yield in Crop Images Using Deep Learning Models for Smart Agriculture: A Review. *Agronomy* **2021**, *11*, 646. [CrossRef]
21. Amjoud, A.B.; Amrouch, M. Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE Access* **2023**, *11*, 35479–35516. [CrossRef]
22. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R. BiFormer: Vision Transformer with Bi-Level Routing Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 10323–10333. [CrossRef]
23. Li, Y.; Huang, Q.; Pei, X.; Chen, Y.; Jiao, L.; Shang, R. Cross-Layer Attention Network for Small Object Detection in Remote Sensing Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2148–2161. [CrossRef]
24. Li, R.; Wu, Y. Improved YOLO v5 Wheat Ear Detection Algorithm Based on Attention Mechanism. *Electronics* **2022**, *11*, 1673. [CrossRef]
25. Chen, Y.; Zheng, L.L.; Peng, H.X. Assessing Pineapple Maturity in Complex Scenarios Using an Improved Retinanet Algorithm. *Eng. Agric.* **2023**, *43*, e20220180. [CrossRef]
26. Rong, J.; Zhou, H.; Zhang, F.; Yuan, T.; Wang, P. Tomato cluster detection and counting using improved YOLOv5 based on RGB-D fusion. *Comput. Electron. Agric.* **2023**, *207*, 107741. [CrossRef]
27. Wang, Z.; Walsh, K.; Koirala, A. Mango Fruit Load Estimation Using a Video Based MangoYOLO-Kalman Filter-Hungarian Algorithm Method. *Sensors* **2019**, *19*, 2742. [CrossRef] [PubMed]
28. Li, Y.; Ma, R.; Zhang, R.; Cheng, Y.; Dong, C. A Tea Buds Counting Method Based on YOLOv5 and Kalman Filter Tracking Algorithm. *Plant Phenomics* **2023**, *5*, 30. [CrossRef] [PubMed]
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. Computer Vision & Pattern Recognition. *arXiv* **2016**, arXiv:1506.02640.
30. Park, H.; Yoo, Y.; Seo, G.; Han, D.; Yun, S.; Kwak, N. C3: Concentrated-Comprehensive Convolution and its application to semantic segmentation. *arXiv* **2018**. [CrossRef]
31. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In Proceedings of the the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9197–9206.

32. Thuan, D. *Evolution of Yolo Algorithm and Yolov5: The State-of-the-Art Object Detection Algorithm*; Oulu University of Applied Sciences: Oulu, Finland, 2021.
33. Li, S.; Zhang, S.; Xue, J.; Sun, H. Lightweight target detection for the field flat jujube based on improved YOLOv5. *Comput. Electron. Agric.* **2022**, *202*, 107391. [CrossRef]
34. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Lu, Y.; Zhang, L. Dynamic Head: Unifying Object Detection Heads with Attentions. *arXiv* **2021**. [CrossRef]
35. Han, Z.; Fang, Z.; Li, Y.; Fu, B. A novel Dynahead-Yolo neural network for the detection of landslides with variable proportions using remote sensing images. *Front. Earth Sci.* **2023**, *10*, 1077153. [CrossRef]
36. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic ReLU. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XIX 16. Springer: Berlin/Heidelberg, Germany, 2020; pp. 351–367. [CrossRef]
37. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-Object tracking by associating every detection box. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–21. [CrossRef]
38. Hsu, W.-Y.; Lin, W.-Y. Adaptive Fusion of Multi-Scale YOLO for Pedestrian Detection. *IEEE Access* **2021**, *9*, 110063–110073. [CrossRef]
39. Liu, Y.; Lu, N.; Shieh, P.; Sun, C. Combination of a Self-Regulation Module and Mobile Application to Enhance Treatment Outcome for Patients with Acne. *Medicina* **2020**, *56*, 276. [CrossRef]
40. Mahdavi-Hormat, A.; Menhaj, M.B.; Shakarami, A. An effective Reinforcement Learning method for preventing the overfitting of Convolutional Neural Networks. *Adv. Comput. Intell.* **2022**, *2*, 34. [CrossRef]
41. Lawal, O.M.; Huamin, Z.; Fan, Z. Ablation studies on YOLOFruit detection algorithm for fruit harvesting robot using deep learning. *Iop Conf. Ser. Earth Environ. Sci.* **2021**, *922*, 12001. [CrossRef]
42. Liu, Y.; Sun, P.; Wergeles, N.; Shang, Y. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Appl.* **2021**, *172*, 114602. [CrossRef]
43. Yang, J.; Ge, H.; Yang, J.; Tong, Y.; Su, S. Online multi-object tracking using multi-function integration and tracking simulation training. *Appl. Intell.* **2022**, *52*, 1268–1288. [CrossRef]
44. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
45. Sun, Q.; Chai, X.; Zeng, Z.; Zhou, G.; Sun, T. Noise-tolerant RGB-D feature fusion network for outdoor fruit detection. *Comput. Electron. Agric.* **2022**, *198*, 107034. [CrossRef]
46. Li, Y.; He, L.; Jia, J.; Lv, J.; Chen, J.; Qiao, X.; Wu, C. In-field tea shoot detection and 3D localization using an RGB-D camera. *Comput. Electron. Agric.* **2021**, *185*, 106149. [CrossRef]