AgriEngineering

MDPI

Article

Lightweight Improved YOLOv5s-CGhostnet for Detection of Strawberry Maturity Levels and Counting

Niraj Tamrakar [1,2], Sijan Karki [1], Myeong Yong Kang [3], Nibas Chandra Deb [1], Elanchezhian Arulmozhi [1], Dae Yeong Kang [3], Junghoo Kook [3] and Hyeon Tae Kim [1,*]

[1] Department of Biosystems Engineering, Institute of Smart Farm, Gyeongsang National University, Jinju 52828, Republic of Korea; niraj@gnu.ac.kr (N.T.)
[2] Nepal Telecom, Nepal Doorsanchar Co., Ltd., Kathmandu 44600, Nepal
[3] Department of Smart Farm, Gyeongsang National University (Institute of Smart Farm), Jinju 52828, Republic of Korea
* Correspondence: bioani@gnu.ac.kr; Tel.: +82-55-772-1896

Abstract: A lightweight strawberry detection and localization algorithm plays a crucial role in enabling the harvesting robot to effectively harvest strawberries. The YOLO model has often been used in strawberry fruit detection for its high accuracy, speed, and robustness. However, some challenges exist, such as the requirement for large model sizes, high computation operation, and undesirable detection. Therefore, the lightweight improved YOLOv5s-CGhostnet was proposed to enhance strawberry detection. In this study, YOLOv5s underwent comprehensive model compression with Ghost modules GCBS and GC3, replacing modules CBS and C3 in the backbone and neck. Furthermore, the default GIOU bounding box regressor loss function was replaced by SIOU for improved localization. Similarly, CBAM attention modules were added before SPPF and between the up-sampling and down-sampling feature fusion FPN–PAN network in the neck section. The improved model exhibited higher mAP@0.5 of 91.7% with a significant decrement in model size by 85.09% and a reduction in GFLOPS by 88.5% compared to the baseline model of YOLOv5. The model demonstrated an increment in mean average precision, a decrement in model size, and reduced computation overhead compared to the standard lightweight YOLO models.

Keywords: CBAM; ghost module; loss function; strawberry; YOLOv5

Citation: Tamrakar, N.; Karki, S.; Kang, M.Y.; Deb, N.C.; Arulmozhi, E.; Kang, D.Y.; Kook, J.; Kim, H.T. Lightweight Improved YOLOv5s-CGhostnet for Detection of Strawberry Maturity Levels and Counting. *AgriEngineering* **2024**, *6*, 962–978. https://doi.org/10.3390/agriengineering6020055

Academic Editor: Travis Esau

Received: 7 March 2024
Revised: 29 March 2024
Accepted: 4 April 2024
Published: 9 April 2024

1. Introduction

Strawberries are one of the most cherished fruits in the world. The popularity of the fruit is attributed to its nutritional value, vibrant texture, succulent taste, and sweet aroma. These distinct characteristics position strawberries as vital cash crops and contribute significantly to worldwide production. Within the Asia Pacific (APAC) region, South Korea was the second-highest strawberry production country, with an annual production of 192.85 thousand metric tons in 2021 [1]. With massive strawberry production, the production process is labor-intensive and expensive, particularly during harvesting. Harvesting is predominantly carried out by the farmer and seasonal manual labor. Studies reveal that harvesting costs constitute a substantial portion, ranging from 73–80%, of the total strawberry cultivation and production expense [2]. Furthermore, the harvesting period is short and requires timely harvesting; otherwise, fruit will be overripe and soft. This makes fruit prone to mechanical damage and fungal infection. Therefore, harvesting robots are of utmost importance in addressing labor shortages, minimizing harvesting costs, and increasing fruit shelf-life. In recent years, researchers have focused on designing and implementing prototype strawberry harvesting robots. The harvesting robots consist of an autonomous driving mechanism, a manipulator to pick ripe strawberries, and a vision system to navigate, recognize, and localize ripe strawberries in real field environments.

Thus, for proper resource management in harvesting robots, a primary requirement is a lightweight and efficient machine vision algorithm.

Traditionally, harvesting robots use computer vision systems with machine learning algorithms to extract features such as the shape, size, color, and texture of targeted strawberries. The extracted features are used with spatial information to identify the location of strawberries and ripeness levels. Xu et al. [3] proposed detection of strawberries in two steps, first detecting strawberry-like regions from HSV color information, then processing its HoG descriptor using an HoG/SVM classifier. Lim et al. [4] suggested an algorithm to detect separate strawberries based on their color threshold. Similarly, Karki et al. [5] utilized different machine learning models to identify strawberry ripeness levels using different color spaces (RGB, HLS, CIELab, and YCbCr) and biometrical characteristics. The results showed that a feed-forward artificial neural network (ANN) with CIELAB color space achieved the highest accuracy. While studies have utilized image attributes with machine learning to predict fruits' physical and chemical characteristics, their methods have several limitations. These include susceptibility to noise, the requirement for a controlled experimental setup, specialized equipment, reliance on trained experts to handcraft features, and time-consuming processes. In addition, when these machine learning models are implemented in open fields, they provide poor performance due to differences in lighting, occlusion, and complex plant canopies.

In contrast to traditional machine learning, convolution-neural-network-based deep learning (CNN-DL) has made significant progress in image processing. The ability to extract features of target objects automatically, resilience toward noise, high generalization, and robustness have allowed CCN-DL to be widely utilized in various agricultural applications. This includes plant disease classification, crop/plant recognition, flower/fruit counting, harvesting, and yield prediction. Narayanan et al. [6] introduced a hybrid CNN architecture designed to classify four banana diseases, achieving a remarkable performance accuracy of 99%. Similarly, Yu et al. [7] proposed Mask-RCNN, a two-stage detection model, for instance segmentation of strawberry fruit and identification of a picking point for harvesting robot. Additionally, Liu et al. [8] utilized a circular bounding box in place of a rectangular bounding box to reduce box loss and minimize Intersection over Union (IoU) to accurately detect tomatoes in You Look Only Once (YOLO) version 3, a one-stage model. Although the CNN-DL algorithm has achieved notable performance, it has a larger network size, higher computational cost, and slower response time. This requires a lot of resources in terms of memory, computation operation, and power to run CNN-DL and also hinders the real-time response. These characteristics challenge modifying present CNN-DL architecture to realize the lightweight CNN-DL algorithm for harvesting robots.

Recently, studies have focused on developing lightweight CNN-DL models with smaller network sizes, fewer weight parameters, and lesser computation effort. The lightweight model is realized using techniques such as low-rank decomposition of dense convolution kernels, model pruning, knowledge distillation, and parameter quantification [9]. Wang et al. [10] proposed an improved YOLOv3 model with detailed semantic features extraction to classify different fruit maturities and a modified loss function to detect small-size fruits. The detection model displayed improvement in mAP and F1-score by 9.83% and 6.49%, respectively, over the YOLOv5 base model. He et al. [11] suggested a YOLOv4-tiny model to detect different maturity levels of strawberries that outperformed the standard YOLOv4 model with an increment in mAP by 5.77% and a decrease in inference time of 51.01 ms. Ge et al. [12] developed an improved YOLOv5s model named YOLO-DeepSort that replaces the original backbone with a shufflenetv2 and added an attention module to better identify and count flowers, green tomatoes, and red tomatoes during growth phases. The improved model outperformed base YOLOv5 in precision in identifying flower, green, and red tomatoes by 17%, 2%, and 2.3%, respectively, with a decreased model size of 10.5 MB. Fang et al. [13] suggested an improved YOLOv3 model to detect ginger shoots and seeds in real-time, which significantly compresses network size and improves inference time by pruning redundant channels and network layers. The

results suggest a decrement in model size by 87.2%, an increase in detection speed by 85%, and mAP lagging slightly by 0.1%. Feng et al. [14] proposed knowledge distillation as a means of knowledge transfer method to initiate the training weight from a large pre-trained network for lesser training time and faster convergence. In prevalent research work, YOLO one-stage models are widely adopted due to their high accuracy, small size, and high-speed detection compared to two-stage models. Therefore, in this research, YOLOv5s was taken as the base model for improvement. In YOLOv5, most target detection models emphasize modifying the backbone network to improve model parameters such as model accuracy, size, processing requirement, and inference time. However, a notable research gap exists in addressing the comprehensive network compression, introducing better loss function, and addition of attention modules for performance enhancement, appreciable model size, and decrement in computation requirement.

Therefore, the primary goal of this research is to modify the YOLOv5s object detection model for better overall network compression, reduced computational requirement, increased accuracy, and improved real-time detection. These characteristics would contribute to developing a lightweight object detection model for ripe strawberries using resource-constrained harvesting robots. Hence, the specific research objectives for this study are to:

1. Prepare a home-grown greenhouse strawberry dataset.
2. Improve YOLOv5s by replacing base CBS and C3 modules on the backbone and neck using the Ghost module, modify box loss regressor functions, and add an attention module.
3. Compare the improved model named YOLOv5s-CGhostnet with standard lightweight YOLO models (YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, and YOLOv5s6) and relative research works.

2. Materials and Methods

2.1. Plant Materials and Image Acquisition

Strawberry cultivar Seolhyang (*Fragaria* × *ananassa Duch.*) seedlings were transplanted and grown in the greenhouse at Gyeongsang National University, Smart Farm Systems Laboratory, Jinju, Republic of Korea from September 2022 to March 2023. The greenhouse's temperature, humidity, and carbon dioxide levels were monitored using a MCH 383SC sensor system manufactured by Lutron Electronics, Taipei, Taiwan, and controlled by a Farm Link controller system (Farm Link v 3.0 UBN, Jinju, South Korea). Bio plus compost soil served as the growing medium for five hundred strawberry plants cultivated in a bench-type configuration with five rows. The composition of Bio plus consists of cocopeat, peat moss, perlite, and zeolite in proportions of 68.7%, 11.0%, 11.0%, and 9.0%, respectively. Hoagland solution was provided, and the plants were irrigated daily with a volume ranging from 20 to 50 mL per plant throughout their growth period.

RGB images were acquired from November 2022 to January 2023 as strawberries entered the first batch of fruiting stage. This stage consists of fruits divided into mature, transition, and immature according to the maturity level, as illustrated in Figure 1a–c. The maturity level of the strawberry was categorized based on the amount of redness on the strawberry. In the study, Basak et al. [15] categorized the maturity level of strawberries according to different growth stages: whiting, turning red, three-quarter red, bright red, and dark red. In reference to that study, the ripeness level of strawberries was categorized as white-green to green as immature, turning red up to three-quarter red as transition, and above three-quarters red to dark red as mature. The RGB camera used for image acquisition was the Sony Cyber-shot DSC-RX100 VII, equipped with optical image stabilization and autofocus. The captured images had a 5472 × 3648 pixels resolution, with exposure set in auto mode.

(**a**) (**b**) (**c**)

Figure 1. Three different strawberry maturity levels: (**a**) Mature; (**b**) Transition; (**c**) Immature.

The generalization ability and robustness of deep learning models increase with diversity in the training images. Therefore, strawberry images were captured under various lighting conditions at different times of the day (morning = 5000, midday = 22,000, and noon = 12,000 lumens, on average), and the distance between the camera and the target object was kept within 40 to 60 cm since the row spacing between the beds was one meter. Additionally, images were acquired with dense fruition, occlusion caused by plant canopy, overlapping strawberry fruits with different maturity and sizes, and variations in camera orientation.

2.2. Dataset Construction and Preprocessing

In total, 2000 images were acquired to prepare a homegrown dataset; each image consists of five to eight strawberry samples, on average, having different maturity levels. Specifically, samples with immature, transition, and mature stages were 5000, 1250, and 4200, respectively. According to the defined maturity criteria, the acquired images were annotated using bounding boxes with annotation software Label-studio version 1.8.0 as ground truth. The label dataset images were divided into training, test, and validation sets, with a ratio of 80:10:10. The validation dataset was used to cross-validate the trained models.

Furthermore, to facilitate better feature extraction, dataset enhancement, and model robustness and performance, labeled images were augmented by employing techniques following the guidance provided in the Ultralytics documentation under 'Train Setting'—'Augmentation settings and hyperparameters' [16]. As indicated for the YOLO series, the data augmentation configuration file (hyp.scratch-low.yaml) was set with the following augmentation parameters: HSV augmentation (hsv_h: 0.015, hsv_s: 0.7, hsv_v: 0.4), rotation (degrees: 15), translation (translate: 0.1), scaling (scale: 0.5), flipping (flip left and right fliplr: 0.5), mix-up (mixup: 0.8), and mosaic augmentation (mosaic: 1.0). By introducing controlled variation in training data, the model gained generality by handling different types of image input.

2.3. Strawberry Detection Methodologies

2.3.1. YOLOv5 Object Detection Network

YOLOv5 is a prevalent one-stage target detection model proven for its small size, quick training time, and fast detection speed [17]. Depending upon the YOLO model configuration set for the depth of the network and channel width, the model has five variants: YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extreme large). According to the model's varying network depth and feature width, model size, performance, and detection speed also vary. Therefore, YOLOv5 models serve as benchmark models for developing lightweight strawberry detection algorithms for harvesting robots.

YOLOv5s architecture comprises an input, backbone, neck, and detection head, as shown in Figure 2a. The input block includes mosaic data augmentation [18], adaptive image scaling [3], and adaptive anchor frame [19]. Mosaic data augmentation blends multiple images with random image transformation functions like cropping and scaling. This process enhances dataset diversity, particularly addressing small targets, and con-

tributes to the accelerated training speed of the network. Adaptive image scaling ensures optimal handling of varying image sizes. Similarly, an adaptive anchor frame functions to dynamically calculate the optimal anchor frame value within the training sets during each training iteration.

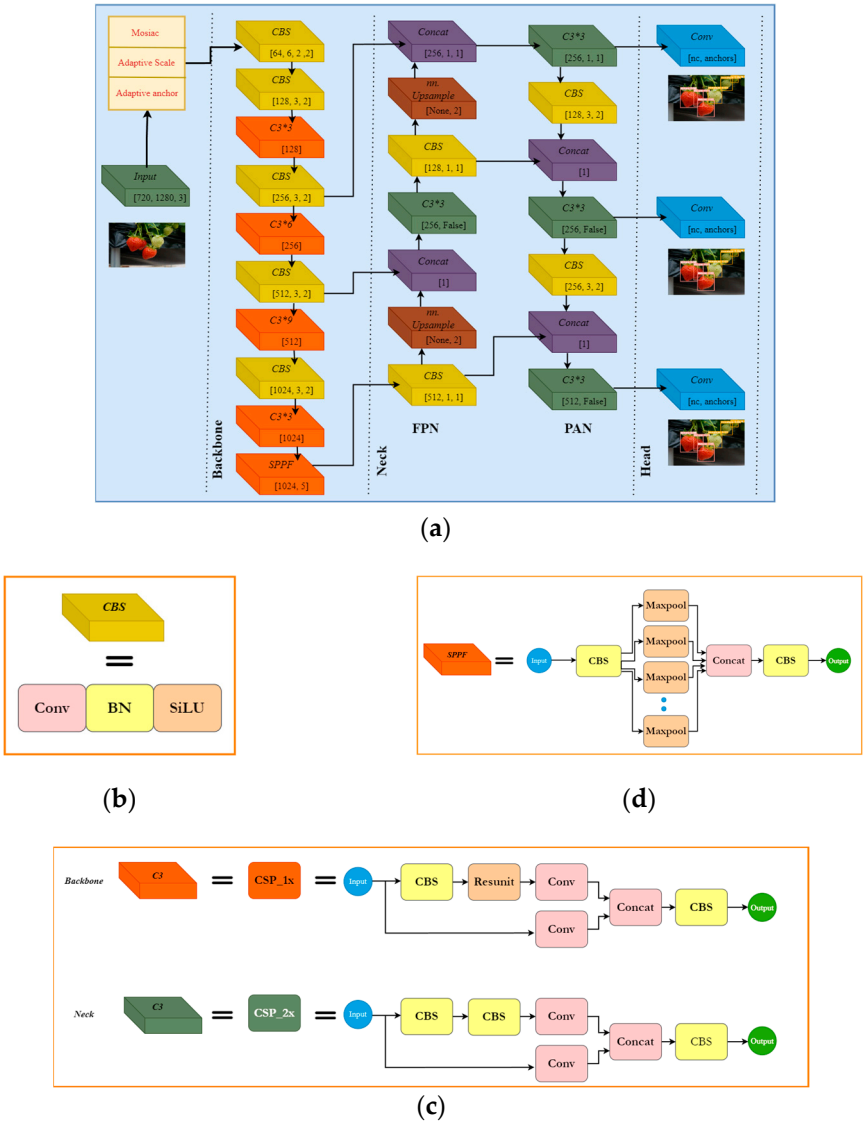


Figure 2. YOLOv5s structure and component model: (**a**) Architecture with input data augmentation, backbone, neck, and head section; (**b**) CBS module; (**c**) C3 module; (**d**) SPPF module.

Following the input is the backbone block, which serves as a primary block. The function of the block is to convert the original input image into multiple feature maps that are vital for subsequent object detections. The backbone structure includes convolution block (CBS), Cross-Stage Partial layer (C3), and Spatial Pyramid Pooling Function (SPPF) modules. Figure 2b shows the CBS module; it is the basic module used in a convolution neural network comprised of the convolution layer, batch normalization (BN) layer, and Sigmoid weighted linear unit (SiLU) activation function. The activation function is a non-linear unit providing non-linear transformation capability to neural networks. The BN layer function is to normalize the convolution layer. This helps to accelerate the training process, improves model generalizability, and makes the model invariant to input initialization.

The C3 module fundamentally increases the network depth and receptive field, abstracting the global information related to the object and enhancing the extraction of features, as shown in Figure 2c. The C3 module is represented as CSP1_x in the backbone,

whereas CSP2_x is in the neck block. CSP1_x consists of a Resnet unit, a transition layer, and a dense layer. The Resnet unit is composed of two convolutional layers and a shortcut connection. The transition layer is a convolutional layer that reduces the number of channels by half. The dense layer is a concatenation of the output of the transition layer and the shortcut connection. CSP2_x is like CSP1_x; the only difference is that two CBL units replace the Resnet unit.

The SPPF module functions fundamentally as a pooling unit that applies different sizes of receptive fields to capture features of different scales from the same input image, as shown in Figure 2d. This helps the SPPF module to achieve spatial and positional invariance of input data and improve the network's recognition ability for different scales of target objects.

The neck block enhances object detection accuracy through multi-scale feature fusion using the feature pyramid network + pyramid aggregation network (FPN–PAN) structure. It employs up-sampling and down-sampling operations to create a multi-scale feature pyramid, merging features from different levels for improved localization. The top–down and bottom–up segments, respectively, combine features through up-sampling and convolutional layers, reinforcing localization information. The head block is the final module, consisting of three different detectors to detect large, medium, and small target objects using grid-based anchors.

2.3.2. Improved YOLOv5s-CGhostnet Network

In this study, YOLOv5s was customized to improve performance and named YOLOv5s-CGhostnet, as shown in Figure 3. Similar architecture compression with the attention module was observed in vehicle detection technology [20]. The YOLOv5s architecture implements the lightweight Ghost module and channel block attention module (CBAM) to accomplish a lightweight and efficient model to localize and detect different mature strawberry fruits in a greenhouse environment.

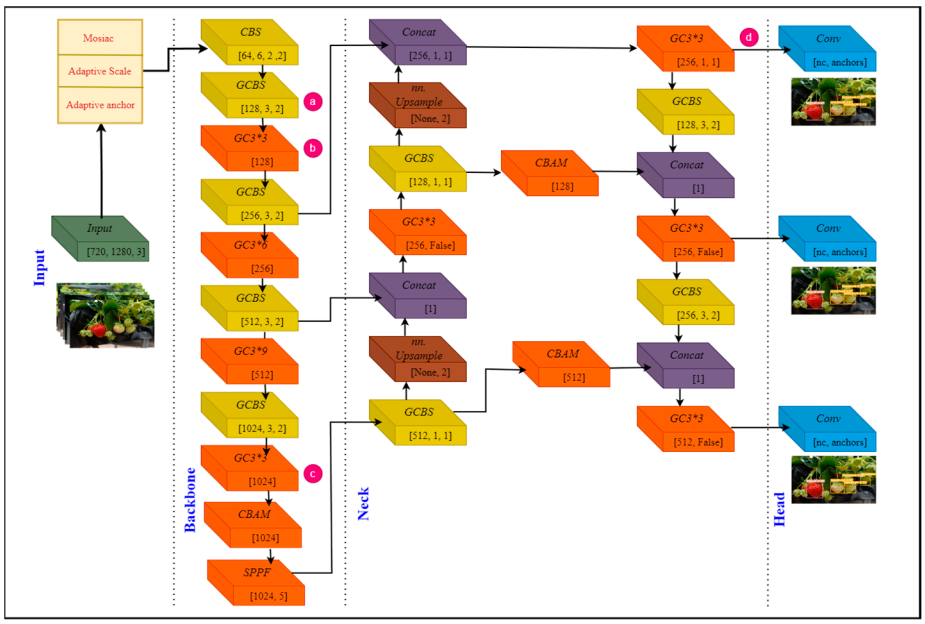


Figure 3. The proposed YOLOv5s-CGhostnet network with Ghostnet module: Ghost operation (GCBS), GhostBottleneck, and Ghost C3(GC3). The labels a, b, c, and d denote the network layers where feature maps were visualized in Section 3.1.

Ghostnet Network

This study implemented the Ghost module as a lightweight neural network [21]. The principle of the Ghost module is based upon depth-wise convolution (DWC) operation, which serves as a cheap linear transformation. This significantly reduces the complexity of

operations and increases speed. Figure 4a illustrates the realization of the Ghost module operation for generating the final 'ghost feature map' from the input feature map that resembles the baseline feature map as obtained from ordinary convolution. The generation of a 'ghost features map' involves the serialization of two operations. In the first operation, an ordinary convolution operation is performed in input features with fewer filters to generate a smaller number of intrinsic feature maps. Then, the features map obtained from the first section is further transformed using the 's' number of cheap linear operations to generate intermediate ghost feature maps. Finally, a 'ghost feature map' that resembles the output of an ordinary convolution is obtained by concatenating the first and second section feature maps.

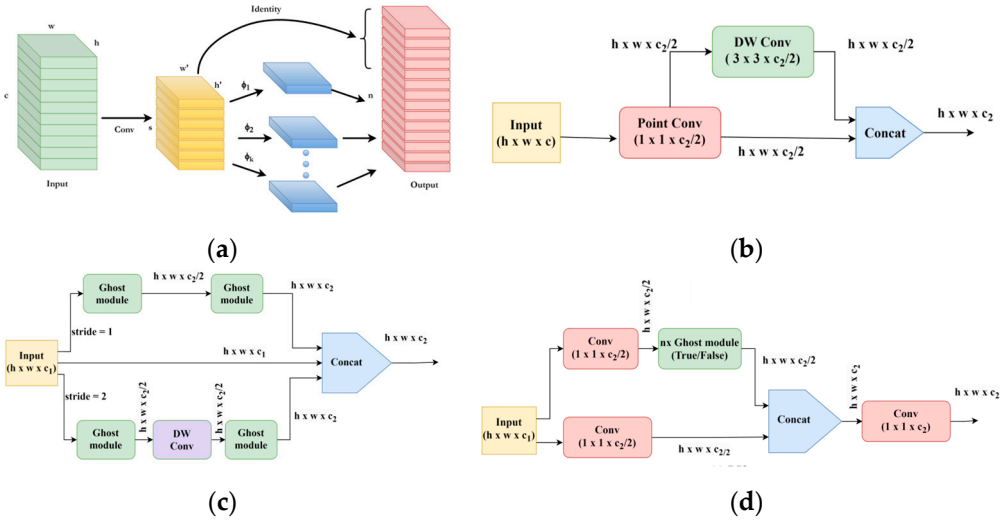


Figure 4. Ghost network: (**a**) Ghost operation; (**b**) Ghost module; (**c**) Ghost Bottleneck; (**d**) GC3.

Operation details regarding ordinary convolution and Ghost module with parameter and FLOPS saving can be calculated as (Equations (1)–(5)):

$$\text{Input feature map tensor} = h \times w \times c \tag{1}$$

$$\text{Output feature map tensor} = h' \times w' \times n \tag{2}$$

In ordinary convolution:

$$\text{No. of Flops operation} = n \times h' \times w' \times c \times k \times k \tag{3}$$

In Ghost convolution:

$$\text{No. of Flops operation} = \frac{n}{s} \times h \times w \times c \times k \times k + (s-1) \times \frac{n}{s} \times h' \times w' \times d \times d \tag{4}$$

From Equations (3) and (4):

$$\text{Theoretical ratio of ordinary convolution to Ghost module convolution} \approx s \tag{5}$$

where, h = height of input feature map, w = width of the input feature map, c = no of the channel, k = size of filter kernel, h' = height of output feature map, w' = width of the output feature map, n = no of filters, d = size of the kernel for linear cheap operations with similar magnitude as k, and s = the number of cheap transformation operations.

Equation (5) suggests that the floating-point operation related to Ghost modules is 's' times more efficient than ordinary convolution operations. Similarly, the number of parameters needed for operation also resembles FLOPs. Therefore, it can also be approximated to be decreased with 's' ratio. This shows computational superiority in the computation

cost of the Ghost module. Therefore, Ghost operation (GCBS), Ghost Bottleneck, and Ghost C3 (GC3) are structured based on the Ghost module, as shown in Figure 4b–d. These new structures can reduce the computation costs and compress the model size by replacing the CBS and C3 structures in the original YOLOv5s model.

CBAM

The mechanism of the attention module in computer vision is inspired by human vision. This allows one to selectively focus on a specific target object against all the unimportant objects and backgrounds. The CBAM is a widely used lightweight attention module due to its ability to adaptively recalibrate features maps by capturing both channel-wise and spatial-wise attention, enabling more effective and context-aware feature extraction. The structure of CBAM contains two sequential blocks stacked one after another, first the channel attention module (CAM) and then the spatial attention module (SAM), as shown in Figure 5 [22]. The CBAM attention module provides "refined feature maps" from "input intermediate feature maps" in convolution neural networks. The feature maps, input, and output in convolution layers are represented as tensors with 3-dimensional metrics c × h × w. The operation of the CBAM module produces refined feature maps as mentioned in the following calculations (Equations (6)–(11)).

$$\text{Input Intermediate feature map}: \ F \in R^{C \times H \times W} \tag{6}$$

$$\text{Output feature map after first block}: \ F' = M_c(F) \otimes F \tag{7}$$

$$\text{CA}: \ M_c(F) = \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1(W_0(F^c_{max}))\right) \tag{8}$$

$$\text{Input feature map on sec ond block}: \ F' \tag{9}$$

$$\text{Final output feature map}: \ F'' = M_s(F') \otimes F' \tag{10}$$

$$\text{SA}: \ M_s(F) = \sigma\left(f^{7\times7}\left[F^s_{avg}; F^s_{max}\right]\right) \in R^{H \times W} \tag{11}$$

where σ = sigmoid activation function, W_1, W_0 = weight of multilayer perception, F^c_{avg}, F^c_{max} = channel average polled feature and max polled feature, F^s_{avg} , F^s_{max} = spatial average polled feature and max pooled feature, and $f^{7\times7}$ = filter size of 7 × 7

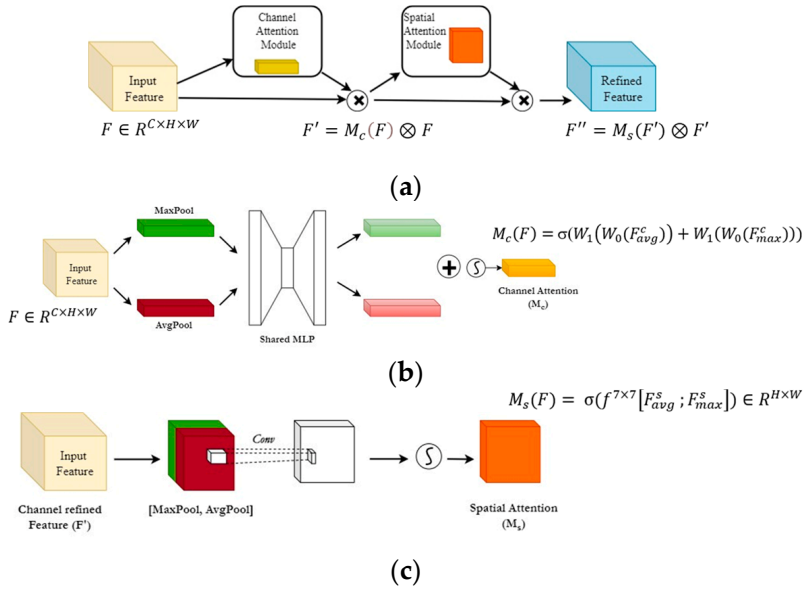


Figure 5. CBAM: (**a**) Structure; (**b**) CA module; (**c**) SA module.

SIOU Loss Function

In YOLOv5s, loss functions consist of three losses: confidence loss (I_{obj}), classification loss (I_{cls}), and position loss of the target box represented by prediction box loss (I_{box}). The loss equation is as follows:

$$\text{Loss} = I_{obj} + I_{cls} + I_{box} \quad (12)$$

In YOLOv5s, the Generalized Intersection over Union (GIoU) loss function is the default bounding box regression loss (I_{box}) function [23]. Compared to the IoU loss function regressor, GIoU solves differentiability issues when the prediction box and target box do not intersect, i.e., IoU = 0, and when the intersection of two prediction boxes has the same size and same IoU. However, GIoU cannot solve the problem with the prediction box being inside the target box, and in this circumstance, GIoU performance is the same as IoU. Therefore, the SCYLLA-IoU (SIoU) loss function regressor was implemented in this study to improve the I_{box} loss function, since the SIoU loss function takes account of angle cost, distance cost, shape cost, and IoU cost collectively to calculate a more precise bounding box localization [24].

2.4. Test Platform and Parameter Setting

The models employed in the study were trained, validated, and inferred using a computer with an Intel Core i7-10700 CPU @ 2.90 GHz, 16 GB RAM, graphic card, NVIDIA GeForce RTX 2060 with dedicated 6 GB memory, and Microsoft Windows 10 Pro operating system. The widely used PyTorch framework facilitated the training and inference of object detection models. Transfer learning and fine-tuning strategies were applied to expedite model training. During the model training phase, all models' iteration batch size was set to 16 with a default decay coefficient of 0.0005, initial learning rate of 0.01, and total number of iterations of 100.

2.5. Evaluation of Model Performance

This experiment uses the following metrics to evaluate the performance of object detection models: precision (P), recall (R), F1-Score, and mean average precision (mAP@0.5). Simultaneously, the network parameters, model size, and detection speed were used to analyze the network architecture, model size, and inference time.

$$Precision\ (P) = \frac{True\ Positive}{True\ Postive + False\ Positive} \quad (13)$$

$$Recall\ (R) = \frac{True\ Positive}{True\ Postive + False\ Negative} \quad (14)$$

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precison + Recall} \quad (15)$$

$$Mean\ Avg.\ Precision = \frac{\sum_1^N Avg.\ Precision\ of\ kth\ class}{True\ Number\ of\ k\ class\ (N)} \quad (16)$$

Precision and recall assess the model's proficiency in detecting and localizing fruit classes. The F1-score provides a balanced assessment of the model's classification performance by harmonizing precision and recall. Additionally, map@0.5 calculates the average precision at different confidence levels for predicted bounding boxes, considering an IoU threshold more significant than 0.5 with the ground truth. In Equations (13)–(16), True Positive represents correctly detected strawberry classes, False Positive accounts for falsely detected strawberry classes, and False Negative indicates the number of missed detections.

3. Results and Discussion

The improved YOLOv5s-CGhostnet proposed by this study has a series of modifications and incorporated an attention module into the baseline YOLOv5s model. These enhancements aimed to improve the strawberry fruit detection model's performance, op-

timize size, and reduce computational requirements. To assess the effectiveness of these modifications, YOLOv5s-CGhostnet was compared with several lightweight YOLO detection models, namely YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, and YOLOv5s6.

3.1. Test Results on Fruit Detection

The overall performance of each algorithm was evaluated using metrics, including precision, recall, F1-score, and mAP@0.5, as presented in Table 1. Notably, the improved YOLOv5s-CGhostnet demonstrated superior detection accuracy compared to all other models, achieving F1-score and mAP@0.5 values of 90.1% and 91.7%, respectively. Compared to YOLOv5s, YOLOv5s-CGhostnet improved 0.12% in F1-score and 0.98% in mAP@0.5. Furthermore, the improved YOLOv5s-CGhostnet achieved the highest accuracy with 88.1% precision and 87.7% recall. The results illustrate the enhanced performance of YOLOv5s-CGhostnet regarding overall detection accuracy.

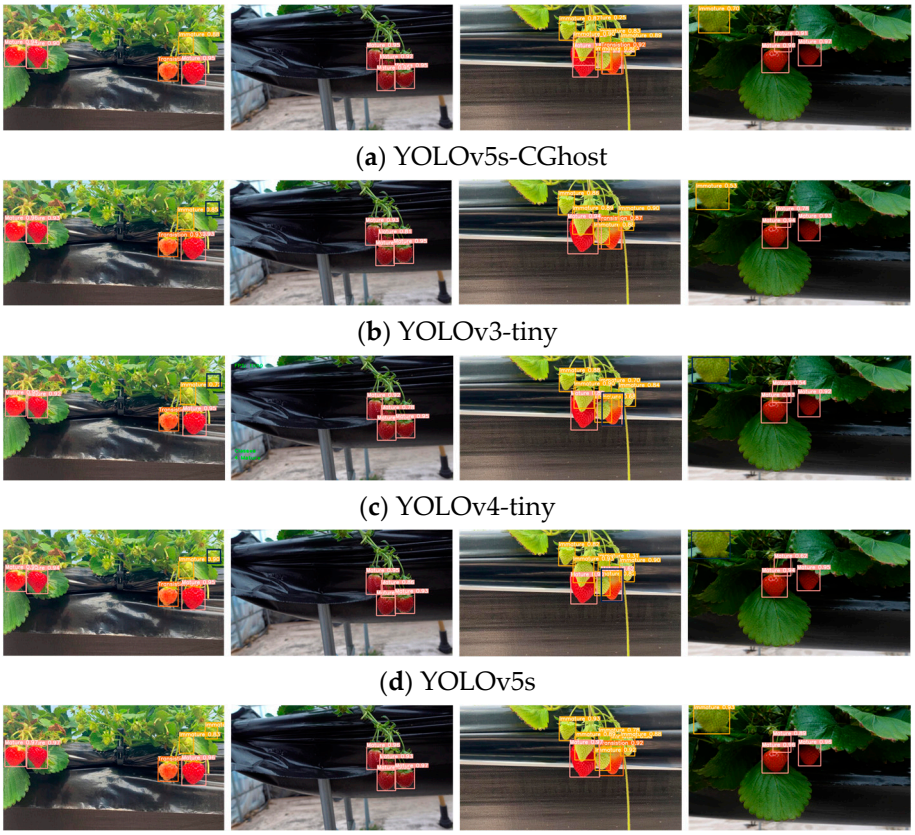
Table 1. The comparative results of the different algorithms.

Algorithm	Precision (%)	Recall (%)	F1-Score (%)	mAP@0.5 (%)
YOLOv3-tiny	85.7	84.9	85.3	90.1
YOLOv4-tiny	83	83.7	83.4	89.8
YOLOv5s6	86.3	86.9	86.6	91.1
YOLOv5s	87.4	85.8	86.6	90.7
YOLOv5s-CGhostnet	88.1	87.7	87.9	91.7

The target detection performance of YOLOv5s-CGhostnet against lightweight standard models is presented in Figure 6. The YOLOv5s-CGhostnet model exhibited high confidence in identifying various ripeness classes under diverse conditions, including intense and low-light images, dense fruition with different sizes, and occlusion levels. In contrast, the models YOLOv3-tiny, YOLOv4-tiny, and YOLOv5s reveal instances of missing fruit detection and false positives, indicated by black boxes as shown in Figure 6b–d. These issues significantly impacted the overall performance and accuracy of these detection models. The confidence level in strawberry fruit detection was notably low. Similarly, YOLOv5s6 exhibited a detection and confidence level comparable to the improved YOLOv5s-CGhostnet model. This can be attributed to the higher depth and width of the model network. In summary, the YOLOv5s-CGhostnet model presents a viable solution to challenges faced by other lightweight models, addressing issues such as low confidence in detection, missing fruit, false positives, and occlusions in the detection of various classes of fruits.

While the evaluation of a deep learning model's performance through metrics is crucial, obtaining a comprehensive understanding of the inner workings, often concealed within the 'black box', is equally essential for effective optimization and refinement. In recent years, researchers have adopted feature visualization techniques, converting features from various convolution and network output layers into visual representations to uncover and understand the complexity of these models. In order to demonstrate the efficacy of the YOLOv5s-CGhostnet algorithm, four randomly selected greenhouse strawberry images underwent feature visualization at different stages of the network as labeled a, b, c, and d in Figure 3, and depicted in Figure 7.

During the initial phase, the layers extracted shallow features that detect the target strawberries and the features and texture of the background environment consisting of plant canopy. However, as the depth of the layer increases, the feature maps become more abstracted, focusing on retaining and enhancing the strawberry fruit information with high dimensional deep semantic features. This high dimensional feature map is further passed through the FPN–PAN network to find the spatial location of the targeted strawberries in the image. The localization of features can be seen in the last feature map visualization, Figure 7d. The feature visualization exhibits that the improved model has higher feature extraction capability and refines the feature information, focusing on the desired classes of strawberries. This further justifies the detection model's accuracy and effectiveness.



(**a**) YOLOv5s-CGhost

(**b**) YOLOv3-tiny

(**c**) YOLOv4-tiny

(**d**) YOLOv5s

(**e**) YOLOv5s6

Figure 6. Comparison of the strawberry fruit detection results in various conditions with different algorithms.

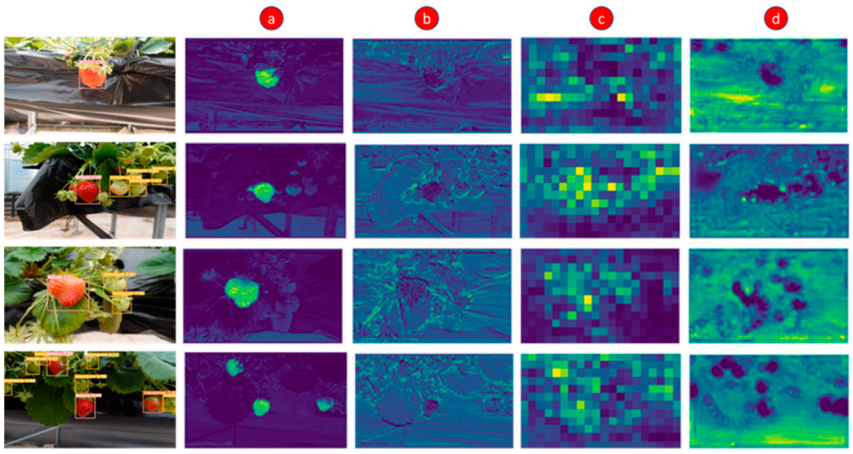


Figure 7. Strawberry detection feature maps. Label a, b, c, and d are different stages of network depicted in Figure 3.

3.2. Analysis of Class-Wise Strawberry Fruit Detection Algorithms

The detection results across various maturity levels of strawberries highlight the model's proficiency in extracting and distinguishing target features. Table 2 presents the mAP@0.5 for class-wise fruit detection of different algorithms, affirming the model's overall accuracy. Notably, the proposed YOLOv5s-CGhostnet model exhibited superior performance, achieving the highest mAP@0.5 for mature, transition, and immature classes at 97.2%, 86.8%, and 91.2%, respectively. The model improvements over YOLOv5s for mature, transition, and immature classes are 0.1%, 2.8%, and 0.6%, respectively. However, it trails slightly behind YOLOv5s6, which attains a 97.4% mAP for mature fruits.

Table 2. The comparative results on mAP@0.5 for different algorithms on different strawberry maturity classes.

Maturity Classes	YOLOv3-Tiny	YOLOv4-Tiny	YOLOv5s	YOLOv5s6	YOLOv5s-CGhostnet
Mature	96.4	96.5	97.1	97.4	97.2
Transition	85.7	83.4	84	85.2	86.8
Immature	88.2	89.5	91.1	90.6	91.2

The slightly lower detection accuracy for the immature class than the mature class can be attributed to the fact that mature strawberries have similar sizes and textures. However, immature fruits vary in size and texture due to different growth stages. In addition, the small size of immature fruits poses challenges in distinguishing them from the background plant canopy. Furthermore, annotating the mature strawberries is easier, but immature fruit at the budding stages was not annotated and was not included in the dataset. Similarly, the transient maturity class exhibits the least mAP value due to the wide feature space between immature and mature classes, compounded by the limited number of data samples in the dataset. A similar observation was observed in the studies of SDNet [25] and DSE-YOLO model [10].

3.3. Analysis of Model Size and Computational Cost Requirement of Different Algorithms

In detection algorithms, the model's lightweight is crucial beyond excelling in target detection and accuracy of the model, as it influences the ease of implementation in devices with limited resources. Therefore, the model parameters, model size, and computational requirements for the improved YOLOv5s-CGhostnet were compared comprehensively against standard YOLO models. Table 3 summarizes network parameters, memory size, and GFLOPS for each model.

Table 3. The comparative result on the network parameters, memory size, and computation requirement (GFLOPS) for different algorithms.

Models	Parameter (10^6)	Model Size (MB)	GFLOPS
YOLOv3-tiny	8.8	17.02	13.2
YOLOv4-tiny	6.06	11.3	16.2
YOLOv5s6	7.2	14.03	16.4
YOLOv5s	12.6	24.47	16.8
YOLOv5s-CGhostnet	6.02	9.4	9.8

Among all the models considered, YOLOv5s-CGhostnet exhibited the most efficient utilization of resources, requiring the lowest parameters, memory size, and computational operation with values of 6.02 × 106, 9.4 MB, and 9.8 GFLOPS, respectively. This marks a substantial improvement over the reference model, YOLOv5s, with reductions in model parameters, memory size, and computational operation of 46.18%, 49.25%, and 67.35%, respectively. The improvement of YOLOv5s-CGhostnet against models YOLOv3-tiny, YOLOv4-tiny, and YOLOv5s6 on model size reductions was 81.06%, 20.21% and 160.32%, respectively, whereas model GFLOPS reduction was 34.69%, 65.31%, and 71.43%, respectively. This remarkable compression underscores the model's efficiency and suitability for deployment in resource-constrained environments. This thorough evaluation reinforces YOLOv5s-CGhostnet as an optimal choice for strawberry detection, demonstrating exceptional efficiency in model size and computational demands.

3.4. Ablation Experiment

Ablation study serves as a methodology to interpret the specific contributions of various components integrated into the original model. This approach involves systematically adding and removing individual components to evaluate their impact on the overall performance of the improved model. In our investigation, we applied ablation study techniques

to the YOLOv5s model, introducing modifications such as dataset enhancement (DEH) and replacing baseline detection model modules with Ghostnet modules. Specifically, the standard CBS and C3 modules in the backbone and neck section were replaced by the GCB and GC3 block, respectively. The SIoU box loss function replaced the GIoU box loss function. Further experimentation includes the addition of the CBAM before the SPPF module and placing CBAM in sections between FPN and PAN at the neck block of YOLO. This exploration aimed to identify the optimal improved architecture termed YOLOv5s-CGhostnet. The detailed structures of the models for the ablation experiment are outlined in Table 4.

Table 4. Model structures of ablation experiments.

Models Structure	Input	Backbone			Neck		CBAM + SIOU
	DEH	CBS	C3	SPPF	CBS	C3	
YOLOv5s	Yes	CBS	C3	SPPF	CBS	C3	-
YOLOv5s-Ghostnet without DEH (A1)	No	GCBS	GC3	SPPF	GCBS	GC3	-
YOLOv5s-Ghostnet with DEH (A2)	Yes	GCBS	GC3	SPPF	GCBS	GC3	-
YOLOv5s-CGhostnet (A3)	Yes	GCBS	GC3	SPPF	GCBS	GC3	With CBAM
YOLOv5s-CGhostnet with SIOU (A4)	Yes	GCBS	GC3	SPPF	GCBS	GC3	With CBAM + SIOU

Furthermore, Table 5 presents the quantitative results for each structure, evaluating the performance metrics related to mAP@0.5, size, GFLOPS, and inference time.

Table 5. Results of ablation experiments.

Performance/Model	Yolov5s	YOLOVv5s-Ghostnet without DEH (A1)	YOLOv5s-Ghostnet with DEH (A2)	YOLOv5s-CGhostnet (A3)	YOLOv5s-CGhostnet with SIOU (A4)
mAP@0.5	0.907	0.824	0.849	0.895	0.917
Size (MB)	14.1	7.8	7.8	9.1	9.4
GFLOPS (10^6)	15.8	8.1	8.1	9.1	9.8
Inference (ms)	3.4	3.4	3.9	4.1	5

The results show that the A1 model structure underwent replacing all the network structures of the YOLOv5s benchmark model with Ghostnet without dataset enhancement. This resulted in a reduction in the model's accuracy, mAP@0.5, by 9.15%. However, there was a significant reduction in the model's size and computation operation by 44.68% and 95.06%. In the A2 structure, data enhancement was introduced to enhance the dataset through flip, scale, mosaic, and mix-up augmentation. This did not affect the model size and computation operation, but the prediction accuracy mAP@0.5 value increased by 0.25. In the A3 structure, the CBAM was introduced to the A2 model; this further improved the model accuracy mAP@0.5 value by 0.046 with further increments of model size and computation operation by 13.26% and 10.99%, respectively. Finally, introducing A4 with loss function SIoU further enhanced the model accuracy mAP@0.5 value by 0.022 with slight increments in size and computation operation by 3.19% and 7.14%, respectively. The final improved model A4, YOLOv5s-CGhostnet, surpassed the baseline model YOLOv5s accuracy mAP@0.5 by 0.01 with a significant reduction in model size and computational operation and a comparable inference time of 5 ms, which is enough for the real-time detection of the strawberry fruit. These results justify the data enhancement, modification of structure, loss function, and addition of attention module for the improved models.

3.5. Real-Time Detection and Counting

In this study, the improved YOLOv5s-CGhostnet was analyzed on performance metrics and compared with the standard detection model and recent relevant studies, ablation experiments, and feature visualizations. However, the quantification information is also essential to determine yield and helps to count the number of ripe fruits in the image. Therefore, to identify and quantify different classes, real-time detection and counting of the fruits is visualized using an autonomous program developed by modifying inherent capability of detect.py source code in YOLOv5, as recommended in a prior study [25]. The result of fruit detection and counting is shown in Figure 8. The figure shows the frame rate in the top left corner, reaching an average of 64 frames per second (FPS), and fruit counts in the bottom left corner for the respective image frame. This scouting method can estimate the strawberry yields and probable harvesting cycle.

Figure 8. Real-time detection and class-wise strawberry counting.

3.6. Comparison with Relevant Studies

Our aim in improving the YOLOv5s-CGhostnet network was to design a lightweight model that could perform real-time detection with comparable accuracy in a field strawberry cultivation environment. To achieve this, we analyzed several existing studies on strawberry detection and localization models for comparison: SDNet, DSE-YOLO, and YOLOv8 fused with LW-Swin Transformer. SDNet was chosen for its high accuracy in object detection with a YOLOx model, NAM attention module, and latest loss function. DSE-YOLO was chosen for its emphasis on compact model size using pointwise and dilated convolution to extract semantic features in horizontal and vertical dimensions. Finally, YOLOv8 with fused LW-Swin Transformer was taken to represent the application of recent advancements in object detection. It is worth noting that the models' architectures and the field environment used in those studies varied, making it challenging to attain the same level of accuracy on our dataset. Nonetheless, we can analyze these models based on their accuracy, network size, and frame rate. Table 6 shows the performance metric mAP@0.5, model size, and FPS of three strawberry detection models with our model YOLOv5s-CGhostnet. Among the compared object detection models, SDNet [25] stands out with the highest mAP@0.5 of 94.26, showcasing commendable accuracy. However, its larger model size at 54.6 MB and a moderate FPS of 30.5 are of concern. Conversely, the DSE-YOLO model [10] exhibits the lowest accuracy with a mAP@0.5 of 86.58, accompanied by a smaller model size of 22.39 MB but a slower FPS of 18.20. YOLOv8, incorporating the fused LW-Swin Transformer [26], demonstrates competitive performance with a mAP@0.5 of 94.4 and a significantly reduced model size of 13.42 MB, albeit at a lower FPS of 19.23. Noteworthy is our improved model, YOLOv5s-CGhostnet, which shines with its lightweight model size of 9.4 MB and a remarkable FPS of 64, despite not securing the highest accuracy, registering a mAP@0.5 of 91.7. The study shows that the improved model YOLOv5s-CGhostnet has comparable accuracy to previous research studies and significant

advantages in model size and real-time detection, suggesting a viable model to implement on resource-constrained devices.

Table 6. Comparative studies with recent research works.

Model	mAP@0.5	Model Size (MB)	FPS
SDNet	94.26	54.6	30.5
DSE-YOLO	86.58	22.39	18.20
YOLOv8 with fused LW-Swin Transformer	94.40	13.42	19.23
YOLOv5s-CGhostnet (Ours)	91.7	9.4	64

4. Conclusions

The present study successfully exhibits an improved YOLOv5 model named YOLOv5s-CGhostnet for detecting and localizing strawberries at different maturity stages. The base YOLOv5 model had a comprehensive change in structure in the backbone and neck part with the replacement of CBS and C3 with the Ghost module, supplemented by CBAM and SIoU box loss function. The model performance was evaluated with a prepared dataset and compared with standard lightweight YOLO models. Furthermore, an ablation study was done to justify data enhancement, replacing the base module with the Ghost module, adding SIOU box loss function, and integrating the attention module. The results show that YOLOv5s-CGhostnet achieved the highest performance score with 91.7% mAP@0.5, 5 MB model size, and 9.8 GFLOPS of computation requirement. The comprehensive performance evaluation and comparison with the standard lightweight model show that the improved model has a compact structure, lower complexity, higher detection accuracy, and real-time detection capability. The inference time of 5 ms was slightly higher than the YOLO tiny model and YOLOv5s models; however, the achieved inference time is enough for real-time detection purposes. In future enhancements, the focus would be to deploy the improved model to resource-constrained devices such as Raspberry Pi and develop the platform for scouting and harvesting robots. Additionally, the transformer model for vision processing would be explored using images acquired with an RGB depth camera to enhance further detection algorithms with compact size, improved computation operation, high accuracy, and real-time operation. Specifically, transformer-based architectures will be investigated to determine whether they can outperform YOLOv5s-CGhostnet for strawberry detection on a resource-constrained device, employing model pruning techniques with parameter quantification for size reduction.

Author Contributions: Conceptualization, Methodology, Formal analysis, Writing—Original Draft, Investigation, Data curation, Software, Visualization, N.T.; Investigation, Resources, and Writing—Review and Editing, S.K.; Data curation and Validation, N.C.D., E.A. and D.Y.K.; Investigation and Software, J.K. and M.Y.K.; Resources, Writing—Review and Editing, Supervision, Project administration, and Funding acquisition, H.T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been financially supported by the National Research Council of Science and Technology (NST) grant from the Korean government (MSIT) (CRC23041-000).

Data Availability Statement: The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request, and source code for the YOLOv5s-CGhostnet is available at https://github.com/NirajGNU/YOLOv5s-CGhostnet (accessed on 5 March 2024).

Conflicts of Interest: Author Niraj Tamrakar was employed by the company Nepal Telecom, Nepal Doorsanchar Co., Ltd. The remaining authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. FAO. *Strawberry Production in The Asia-Pacific Region in 2021, by Country or Territory (in 1,000 Metric Tons)*; FAO: Rome, Italy, 2023.
2. Wu, F.; Guan, Z.; Garcia-Nazariega, M. *Comparison of Labor Costs between Florida and Mexican Strawberry Industries*; FE1023, 12/2017; EDIS: Zamora, Mexico, 2018.
3. Jiang, W.; Xu, H.; Chen, G.; Zhao, W.; Xu, W. An Improved Edge-Adaptive Image Scaling Algorithm. In Proceedings of the 2009 IEEE 8th International Conference on ASIC, Changsha, China, 20–23 October 2009; pp. 895–897.
4. Lim, Y.W.; Lee, S.U. On the Color Image Segmentation Algorithm Based on the Thresholding and the Fuzzy C-Means Techniques. *Pattern Recognit.* **1990**, *23*, 935–952. [CrossRef]
5. Karki, S.; Basak, J.K.; Paudel, B.; Deb, N.C.; Kim, N.-E.; Kook, J.; Kang, M.Y.; Kim, H.T. Classification of Strawberry Ripeness Stages Using Machine Learning Algorithms and Colour Spaces. *Hortic. Environ. Biotechnol.* **2023**, *65*, 337–354. [CrossRef]
6. Narayanan, K.L.; Krishnan, R.S.; Robinson, Y.H.; Julie, E.G.; Vimal, S.; Saravanan, V.; Kaliappan, M. Banana Plant Disease Classification Using Hybrid Convolutional Neural Network. *Comput. Intell. Neurosci.* **2022**, *2022*, 9153699. [CrossRef] [PubMed]
7. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit Detection for Strawberry Harvesting Robot in Non-Structural Environment Based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
8. Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3. *Sensors* **2020**, *20*, 2145. [CrossRef] [PubMed]
9. Zhou, X.; Liu, H.; Shi, C.; Liu, J. Chapter 3—Model Design and Compression. In *Deep Learning on Edge Computing Devices*; Zhou, X., Liu, H., Shi, C., Liu, J.B.T.-D.L., Eds.; Elsevier: Amsterdam, The Netherlands, 2022; pp. 39–58, ISBN 978-0-323-85783-3.
10. Wang, Y.; Yan, G.; Meng, Q.; Yao, T.; Han, J.; Zhang, B. DSE-YOLO: Detail Semantics Enhancement YOLO for Multi-Stage Strawberry Detection. *Comput. Electron. Agric.* **2022**, *198*, 107057. [CrossRef]
11. He, Z.; Karkee, M.; Zhang, Q. Detecting and Localizing Strawberry Centers for Robotic Harvesting in Field Environment. *IFAC-PapersOnLine* **2022**, *55*, 30–35. [CrossRef]
12. Ge, Y.; Lin, S.; Zhang, Y.; Li, Z.; Cheng, H.; Dong, J.; Shao, S.; Zhang, J.; Qi, X.; Wu, Z. Tracking and Counting of Tomato at Different Growth Period Using an Improving YOLO-Deepsort Network for Inspection Robot. *Machines* **2022**, *10*, 489. [CrossRef]
13. Fang, L.; Wu, Y.; Li, Y.; Guo, H.; Zhang, H.; Wang, X.; Xi, R.; Hou, J. Using Channel and Network Layer Pruning Based on Deep Learning for Real-Time Detection of Ginger Images. *Agriculture* **2021**, *11*, 1190. [CrossRef]
14. Feng, J.; Yu, C.; Shi, X.; Zheng, Z.; Yang, L.; Hu, Y. Research on Winter Jujube Object Detection Based on Optimized Yolov5s. *Agronomy* **2023**, *13*, 810. [CrossRef]
15. Basak, J.K.; Madhavi, B.G.K.; Paudel, B.; Kim, N.E.; Kim, H.T. Prediction of Total Soluble Solids and PH of Strawberry Fruits Using RGB, HSV and HSL Colour Spaces and Machine Learning Models. *Foods* **2022**, *11*, 2086. [CrossRef] [PubMed]
16. Ultralytics 2023 Train. Available online: https://docs.ultralytics.com/modes/train/ (accessed on 4 December 2023).
17. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Kwon, Y.; Michael, K.; Changyu, L.; Fang, J.; Skalski, P.; Hogan, A. *Ultralytics/Yolov5: V6. 0-YOLOv5n'Nano'models, Roboflow Integration, TensorFlow Export, OpenCV DNN Support*; Zenodo: Geneva, Switzerland, 2021.
18. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
19. Gao, M.; Du, Y.; Yang, Y.; Zhang, J. Adaptive Anchor Box Mechanism to Improve the Accuracy in the Object Detection System. *Multimed. Tools Appl.* **2019**, *78*, 27383–27402. [CrossRef]
20. Dong, X.; Yan, S.; Duan, C. A Lightweight Vehicles Detection Network Model Based on YOLOv5. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104914. [CrossRef]
21. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
22. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
23. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. *arXiv* **2019**, arXiv:1902.09630.
24. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.

25. An, Q.; Wang, K.; Li, Z.; Song, C.; Tang, X.; Song, J. Real-Time Monitoring Method of Strawberry Fruit Growth State Based on YOLO Improved Model. *IEEE Access* **2022**, *10*, 124363–124372. [CrossRef]
26. Yang, S.; Wang, W.; Gao, S.; Deng, Z. Strawberry Ripeness Detection Based on YOLOv8 Algorithm Fused with LW-Swin Transformer. *Comput. Electron. Agric.* **2023**, *215*, 108360. [CrossRef]