Mathematical Biosciences and Engineering

MBE, 20(1): 241–268.
DOI: 10.3934/mbe.2023011
Received: 14 August 2022
Revised: 04 September 2022
Accepted: 05 September 2022
Published: 30 September 2022

http://www.aimspress.com/journal/MBE

Research article

GCS-YOLOV4-Tiny: A lightweight group convolution network for multi-stage fruit detection

Mei-Ling Huang* and Yi-Shan Wu

Department of Industrial Engineering & Management, National Chin-Yi University of Technology, Taichung, Taiwan

* **Correspondence:** Email: huangml@ncut.edu.tw.

Abstract: Fruits require different planting techniques at different growth stages. Traditionally, the maturity stage of fruit is judged visually, which is time-consuming and labor-intensive. Fruits differ in size and color, and sometimes leaves or branches occult some of fruits, limiting automatic detection of growth stages in a real environment. Based on YOLOV4-Tiny, this study proposes a GCS-YOLOV4-Tiny model by (1) adding squeeze and excitation (SE) and the spatial pyramid pooling (SPP) modules to improve the accuracy of the model and (2) using the group convolution to reduce the size of the model and finally achieve faster detection speed. The proposed GCS-YOLOV4-Tiny model was executed on three public fruit datasets. Results have shown that GCS-YOLOV4-Tiny has favorable performance on mAP, Recall, F1-Score and Average IoU on Mango YOLO and Rpi-Tomato datasets. In addition, with the smallest model size of 20.70 MB, the mAP, Recall, F1-score, Precision and Average IoU of GCS-YOLOV4-Tiny achieve 93.42 ± 0.44, 91.00 ± 1.87, 90.80 ± 2.59, 90.80 ± 2.77 and 76.94 ± 1.35%, respectively, on *F. margarita* dataset. The detection results outperform the state-of-the-art YOLOV4-Tiny model with a 17.45% increase in mAP and a 13.80% increase in F1-score. The proposed model provides an effective and efficient performance to detect different growth stages of fruits and can be extended for different fruits and crops for object or disease detections.

Keywords: object detection; YOLOV4-Tiny; SE; SPP; group convolution

1. Introduction

With the advancement of science and technology, deep learning has gradually become the mainstream of artificial intelligence, and it has developed well in various fields. Among them, object detection, which detects the position and category of the target, has become more popular in recent years. The traditional image classification or image segmentation classifies the category of the image, but it cannot locate the position of the category in the image. Object detection is more complex and difficult. Existing object detection methods are divided into two-stage detection algorithms that pay more attention to detection accuracy and one-stage detection algorithms that advocate detection speed.

In the traditional object detection method, the first proposed method is a two-stage detection algorithm. The R-CNN model proposed by Girshick et al. [1] in 2014 is the initial work of the two-stage detection algorithm. Through the screening of Selective Search [2], multiple regions of interest are selected and input to the deep learning model AlexNet [3] to extract features. Then, the extracted features are used in the Support Vector Machine (SVM) for classification, and finally a bounding box is used to predict the location of the region of interest.

Due to the emergence of R-CNN, deep learning has developed advanced technologies in object detection, and many scholars have proposed modified algorithms based on R-CNN. Girshick et al. [4] proposed the Fast R-CNN model by combining R-CNN and SPP [5]. It can adapt to the advantages of different spatial pooling layers and solve the limitation of not using fixed-size images to increase the accuracy of detection to 70%. Ren et al. [6] proposed the Faster R-CNN model, which uses the Region Proposal Network (RPN). Through end-to-end training, the Faster R-CNN can share the convolution features of the two during training, which synchronously classify the original frame of interest and greatly improve the detection time. Li et al. [7] proposed the Feature Pyramid Network (FPN) to solve the general problem that the detection positions of many object detection algorithms are located in the top layer of the entire model network. The FPN has a top-down network architecture, which improves the accuracy of the target detection model and has become the basic technology for many subsequent extended models.

There are many related studies using two-stage object detection methods. Ghosh [8] proposed a new gait recognition method in 2022, using a modified Faster R-CNN to detect whether the pedestrians in the video are carrying objects. The proposed model used Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BLSTM) to identify the pattern of gait and was tested on four public gait datasets (OU-LP-Bag, OUTD-B, OULP-Age and CASIA-B). The research results show that the use of Faster R-CNN and BLSTM has better results, and the accuracy can reach 97.42%. Chen et al. [9] proposed a Faster GG R-CNN model by combining Genetic Algorithm (GA) and Faster R-CNN to detect textile defects in complex backgrounds. Performances were compared with three object detection models, including Faster R-CNN, MsDet and YOLOV3 in 2022. Faster GG R-CNN achieved a mAP of 94.57%, which was the highest among them all. In order to observe the wear state caused by mechanical equipment, Miao et al. [10] used industrial cameras to capture wear images of different types (Normal, Adhesion, Abrasive, Corrosion), and they modified the Faster R-CNN model by replacing the original ResNet backbone with the VGG16 network and adding FPN to improve the detection ability. Results were compared with YOLOV3 and SSD object detection models. The proposed Faster R-CNN model shows the best results, with a Precision of 99.25%, Recall of 99.00% and F-measure of 99.00%.

Cui et al. [11] created an image dataset of highway ground penetrating radar (GPR), which is an

underground survey device that detects road thickness. This study uses Faster R-CNN to detect GPR images to protect the safety of highway traffic and uses ResNet-101 as the main backbone of the network. The Average Precision (AP), Precision and Recall were 88.13, 97.70 and 89.13%, respectively. The results show that the proposed model detects the underground layer of expressways intelligently, and it identifies the images of GPR automatically. In addition, a method based on Faster R-CNN was proposed to detect Lung Nodules using the public database LIDC-IDRI [12]. The CT images marked by doctors were amplified and labeled, and ZF [13] and VGG16 were selected as the backbone models. Results were compared with Faster R-CNN, R-CNN and Fast R-CNN. The model with VGG16 demonstrated the best AP, of 91.20%, which is 8.80, 22.80 and 15.80% higher than APs from Faster R-CNN, R-CNN and Fast R-CNN, respectively. Yang et al. [14] proposed the MT-Faster R-CNN model in 2020, which modified Fast R-CNN with multi-task learning on the KITTI dataset. This model generates 2D and 3D results based on a single vehicle image at the same time, which helps automatic vehicle driving. The above studies of the two-stage algorithms have shown good results in object detection. However, because the model consists of the regional proposal network and the classification network, it often causes longer detection time. In addition, due to the larger size of the model, it needs more advanced equipment.

The one-stage detection algorithm eliminates the RPN in the two-stage, and it can directly detect the category of the object and regress the bounding box, thereby reducing the detection time. The most representative algorithm is You Only Look Once (YOLO). Redmon et al. [15] proposed YOLOV1 in 2016 and used GoogleNet [16] as the backbone network, by extracting bounding boxes from images and directly predicting coordinate locations and categories. YOLOV2 was proposed in 2017 [17] by adding the RPN method proposed by Faster R-CNN. YOLOV2 uses the anchor box function, and it upgrades the original 224 × 224 resolution to 448 × 448, which greatly improve the detection speed. The weakness is that the results are not favorable on predicting small size objects. YOLOV3 [18] modified the backbone network GoogleNet to the DarkNet-53 of the ResNet model [19] and added FPN to predict different features to improve the detection of small objects. Focusing on the improvement of parameter quantity and accuracy, Bochkovskiy et al. [20] proposed YOLOV4 in 2020 by combining the DarkNet-53 backbone network with CSPNet, proposed by Wang et al. [21]. The new backbone CSP DarkNet-53 reduces the computational complexity and memory cost, increases the accuracy and can use GPUs for model training and testing.

Many scholars have applied the YOLO series in various fields. For example, based on YOLOV4-Tiny model, Lin et al. [22] proposed a method using the K-median to identify and find the appropriate anchor box for the end images of bundled logs. The proposed model used three prediction heads and connected each head with SPP to extract small targets. Results show that the Precision, Recall and F1-Score were 93.97, 94.91 and 95.00%, respectively. Kumar et al. [23] modified the YOLOV4-Tiny model and proposed ETL-YOLOV4 to detect masks by modifying the backbone network, adding a dense SPP network and using Mish as the activation function, adding Mosaic and CutMix images to increase the training performance. The mAPs of the proposed ETL-YOLOV4 model evaluated on FMD and MOXA open datasets reached 67.64 and 65.14%, respectively. The performance of the proposed model outperforms YOLOV3 and YOLOV4-Tiny models. Wang et al. [24] proposed the DSE-YOLO model using pointwise convolution and dilated convolution and adding exponentially enhanced binary cross-entropy (EBCE) and double enhanced mean squared error (DEMSE) loss functions to detect smaller fruits and distinguish different growth stages of fruits accurately. The results show that the mAP, F1-Score, and the parameter size on detection of multi-stage strawberry

fruit images were 86.58, 81.59 and 224.39 MB, respectively. Compared with Faster R-CNN, SSD300, SSD512, YOLOV3, YOLOV4 and YOLOV5, DSE-YOLO achieves a balance between accuracy and number of parameters.

Su et al. [25] proposed the YOLO-LOGO model by combining YOLOV5-L6 and Local Global (LOGO) on breast cancer detection. In the model, YOLOV5-L6 locates the tumor in the breast cancer image and then uses LOGO for segmentation. The F1-Socre and IoU were 74.52 and 69.37% on the CBIS-DDSM dataset and 69.37 and 61.09% on the INBreast dataset. Wu et al. [26] proposed the FMD-YOLO model by combining Res2Net and Im-Res2Net-101 to extract features on mask detection. The FMD-YOLO model achieved APs of 92.00 and 88.40% on two open datasets, and it dominated eight object detection models, including Faster R-CNN, Faster R-CNN with FPN, YOLOV3, YOLOV4, RetinaNet, FCOS, EfficientDet and HRNet. Wang et al. [27] proposed the LDS-YOLO model in 2022 to identify images of dead trees taken by drones, which confirms the area of dead trees in order to replant new trees in time. The LDS-YOLO model introduced SPP to increase the detection of smaller targets in UAV images. Considering it is to be combined with UAVs, depthwise separable convolution is used to reduce the model size to 7.60 MB. A similar application was found in Zhao et al. [28], using depthwise separable convolution, and the proposed model reduced the model size to 11 MB while maintaining an accuracy of 95.47% on detection of abnormal fish behavior just in time.

The existing YOLO models consider the trade-off between detection speed and accuracy for real-time detection. Some of them have applied SPP, added SE or used hybrid models to enhance the detection accuracy. There are many related YOLO applications on fruit detection. For example, Tian et al. [29] combined the DenseNet with YOLOV3 to detect growth stages of young, growing and mature apples effectively. Mirhaji et al. [30] used transfer learning on YOLOV2, YOLOV3 and YOLOV4 to detect oranges under different lighting conditions, and they used regression analysis to predict the number of oranges. In addition, YOLO-Tomato models detected tomatoes in a complex environment [31]. Modified YOLOV4 models detected diseases in fruits under challenging environments [32,33]. In order to be more applicable to real-time fruit detection, the size of the model is expected to be small. A modified DenseNet-fused YOLOV4 detected growth stages of mango under a complex environment efficiently [34]. In order to identify grapes in complex backgrounds accurately, Li et al. [35] modified the YOLOV4-Tiny model and added an attention module (Squeeze-and-Excitation) to improve the detection ability on hidden grapes. Furthermore, the depth-wise separable convolution module is used to reduce the number of parameters to improve the real-time performance. Li et al. [36] modified the YOLOV4-Tiny model in 2021 to detect green peppers in complex backgrounds, and they added a multi-scale Adaptive Spatial Feature Fusion (ASFF) to enhance the detection ability for green peppers in small scale. However, using low-dimensional feature maps to increase the feature information of small targets will increase background noise and deteriorate the accuracy of object detection. Therefore, a new channel attention module, the Convolutional Block Attention Module (CBAM), is introduced to solve this problem.

The Faster R-CNN consists of the RPN and the classification network, which prolongs the detection speed and cannot perform real-time detection for high-resolution fruit images. YOLO eliminates the RPN to detect the category of objects and regress the bounding box efficiently. Based on the advantage of fast detection speed of YOLO facilitating real-time detection, this study chooses and modifies the one-stage detection model YOLO for fruit image detection at different growth stages.

The current study focuses on efficient and effective detection of fruit growth stages. Based on YOLOV4-Tiny, this study proposes a one-stage detection model, GCS-YOLOV4-Tiny, to examine

fruits with different sizes in a complex environment. For example, there were fruits occulted by leaves or branches, fruits with small sizes, fruits in poor light or multiple fruits in one image. This study modifies the backbone network CSP DarkNet-53-Tiny in YOLOV4-Tiny. The proposed GCS-YOLOV4-Tiny model uses DIOUS-Non Maximum Suppression and chooses a k-means clustering algorithm to find a suitable anchor frame when detecting the object. In addition, the proposed model adds SE between CBL blocks to combine features of different resolutions and finally improves the performance. To enhance feature diversity, two SPP modules, which avoid the distortion caused by image scaling and fuse local and global features, were added before the full connection layer. Furthermore, the proposed model selects group convolution with fewer computational resources to reduce the model size greatly. With the smallest model size of 20.70 MB, the detection results outperform the state-of-the-art YOLOV4-Tiny model with a 17.45% increase in mAP and a 13.80% increase in F1-score. The proposed model provides an effective and efficient performance to detect different growth stages of fruits, which is beneficial for real-time detection.

The rest of the paper is organized as follows. Section 2 expresses the related studies on SE, SPP, group convolution and YOLOV4-Tiny. Section 3 introduces the datasets and the details of the proposed GCS-YOLOV4-Tiny model. Section 4 presents the experiment results on three open datasets. Finally, Section 5 summarizes the proposed model and suggests future research.

2. Related works

2.1. YOLOV4-Tiny

Wang et al. [37] proposed a one-stage detection YOLOV4-Tiny model, in 2020, which is a simplified version of YOLOV4. This model maintains the detection accuracy with a faster detection speed. The best advantage is that the model uses fewer parameters, performs instant detection and has the feasibility to integrate with embedded devices. The following scholars have used the YOLOV4-Tiny model in different fields. Zhang et al. [38] used drones to capture images of ripe strawberries, immature strawberries and flowers. Based on the YOLOV4-Tiny model, the proposed RTSD-Net possesses fewer convolution layers and faster detection speed to benefit the development of robotic harvesting of strawberries. Yao et al. [39] proposed a modified YOLOV4-Tiny model for the detection of real-time traffic signs. Using the backbone network CSP-DarkNet-53-Tiny of the YOLOV4-Tiny model, the proposed model presents two feature layers of different scales to detect smaller objects.

The ECA-Net model combines the main feature extraction of the YOLOV4-Tiny model with the attention mechanism to improve the feature extraction ability by modifying the YOLOV4-Tiny model to shoot images of insulators in high-voltage transmission lines with an aerial camera. The ECA-Net model improves the model accuracy from 81.01 to 91.19%, and the model size is 24.90 MB, which is suitable for embedded devices and reduces the work time of line transportation and inspection personnel [40]. Zhang et al. [41] used a modified YOLOV4-Tiny model in 2021 to identify the regions where the dials and indicators are located in a water meter image. The output prediction network in the YOLOV4-Tiny model is added as feature maps of three different scales, and the improved YOLOV4-Tiny detects the dial area of the image with high confidence and identifies the type of dial correctly. Li et al. [35] added an SE module to the YOLOV4-Tiny model to improve the performance on the detection of covered grapes. In addition, the depthwise separable convolution module is used to reduce the model size for the real-time detection of grapes by robots. In order to detect green peppers

in complex backgrounds, Li et al. [36] added an adaptively spatial feature fusion (ASFF) pyramid to YOLOV4-Tiny to detect small images of green pepper. The accuracy is as high as 96.91%, and the model size is 30.90 MB.

2.2. SE

SE consists of Squeeze and Excitation [42]. In recent years, many scholars have used the SE method to improve the performance of proposed models, and they have demonstrated good results in various fields. Wang et al. [43] proposed a SAR-U-Net network in 2021 to segment liver CT images automatically by adding SE in each convolutional unit of the U-Net encoder to self-adjust to learn image features and suppress irrelevant regions in a segmentation task. Adding a bottleneck block in the SE module to balance the nonlinear representation ability of the two fully connected layers and improve the detection ability, Ma et al. [44] proposed a MaSE-ResNeXt model for solving the rock slice image classification problem. The SE-ResNeXt model proposed by Khan et al. [45] in 2021 is to solve the recognition problem of Bengali handwritten composite characters by fusing channel spatial information and inter-channel dependencies through SE within local receptive fields. The SECNN model detects five diseases of pepper leaves with fewer parameters of 5.40 MB, and it achieves good results in the pepper leaf disease dataset in 2022 [46]. Huang et al. [47] proposed SESPNets for ship detection based on SE for optical remote sensing images. Alsarhan et al. [48] integrated SE modules into graph convolutional networks to obtain discriminative channel-wise features of the input feature matrix by highlighting the important features to enhance the recognition accuracy.

2.3. SPP

SPP was proposed by He et al. [5] in 2014 to solve the fixed size of the input image in two-stage objection detection RCNN, which causes some images to be deformed due to clipping. This could be solved by adding SPP after the convolutional layer and cutting the input image into multi-sizes to connect with the fully connected layer. Yee et al. [49] proposed a DeepScene model on scene classification via incorporating SPP into CNN to enable the multi-size training of the model. Prasetyo et al. [50] proposed a wing convolutional layer to enhance feature diversity and modified SPP to Tiny-SPP for reduction of computational resources on detection of fish eye, tail and body. The SPP-LSTM-NET combined SPP and LSTM network to predict PM 2.5 concentration, and it achieved better results than that of the traditional LSTM [51].

2.4. Group convolution

Group convolution, first applied to the AlexNet model in 2012, was applied to split the network for execution on two GPUs. The method is to group the input feature maps and convolve each group of feature maps separately. Dividing the convolution into G groups, the parameter amount of this layer is reduced to the original 1/G. Scholars combined or modified group convolution in different fields. For example, Li et al. [52] proposed a classification method based on Interleaved Group Convolutions (IGCs) in 2019 to detect crop image datasets captured by sensors. IGCs shorten the training time of the model without reducing the classification accuracy, especially for training samples with long time series. Yang et al. [53] proposed a lightweight group convolutional network (LGCN) for single image

super-resolution (SISR) in 2019. Group convolution reduces the number of parameters of LGCN and gathers local information on SISR images gradually. In addition, the enhanced super-resolution group CNN (ESRGCNN) uses a 6-layer group augmented convolution block to enhance the representation of low-frequency features to improve the performance and speed of SISR [54].

3. Materials and methods

3.1. Datasets

1) The Mango YOLO dataset was created by Koirala et al. [55]. This dataset has only one category of mango. The images were taken in a dark environment, and the image size is 512 × 512 pixels. The dataset contains 1730 images in total.
2) The Rpi-Tomato dataset contains tomato images of four different maturity levels (green, red, light red and red), and the number of images is 257 in total [56].
3) The *F. margarita* dataset contains images of three different growth stages (mature, immature and growing) of *F. margarita* [57]. Some images include more than one stage. Images are claasified into seven categories: (a) mature, (b) immature, (c) growing, (d) mature and immature, (e) mature and growing, (f) immature and growing and (g) mature, immature and growing. The original number of images is 1031, and data augmentation increases the total number of images to 6,617.

Table 1. Three datasets.

Dataset	Class	Training	Validation	Test	Total
Mango YOLO	Mango	1300	130	300	1730
Rpi-Tomato	Green	40	19	9	257
	Turning	30	19	15	
	Light Red	30	19	12	
	Red	30	19	13	
F. margarita	Mature	2520		39	2559
	Immature	1064		16	1080
	Growing	406		6	412
	Mature and Immature	441		7	448
	Mature and Growing	1295		20	1315
	Immature and Growing	196		3	199
	Mature, Immature and Growing	595		9	604

Table 1 displays the numbers of images, and Figure 1 shows examples of images for the three datasets. images (a1)–(a4) are from the Mango YOLO dataset; images (b1)–(b4) are examples of green, red, light red and red tomatoes from the Rpi-Tomato dataset; images (c1)–(c7) show examples of *F. margaritas* from the *F. margarita* dataset. Images in the Mango YOLO dataset exhibit occluded mango images in a low light environment; datasets of Rpi-Tomato and *F. margarita* represent scenes with natural images in real environments. Images of the above three datasets are used to test and evaluate the performance of object detection for the proposed GCS-YOLOV4-Tiny model.

Figure 1. Examples of three datasets.

3.2. GCS-YOLOV4-Tiny

The main purpose of this research is to propose a lightweight object detection model for real-time detection with favorable detection accuracy. The proposed GCS-YOLOV4-Tiny model is tested and evaluated in three different public datasets. The following describes the proposed model in detail.

1) Based on the studies in Section 2.1, YOLOV4-Tiny demonstrates the detection accuracy with a faster detection speed. The proposed GCS-YOLOV4-Tiny model uses DIOUS- Non Maximum Suppression (DIOU- NMS) [58] to replace NMS and chooses the k-means clustering algorithm to find a suitable anchor frame when detecting the object. This study modifies the backbone network CSP DarkNet-53-Tiny in YOLOV4-Tiny, as shown in Figure 3.

Non maximum suppression (NMS) is commonly used in object detection models to solve the problem of multiple prediction frames around the predicting targets. The NMS removes the redundant frames using the Intersection over Union (IoU) metric. This study uses distance IoU (DIOU) [58] to consider the overlap area and the distance between two central points of bounding boxes. The formula is as follows:

$$R_{DIoU} = \frac{p^2(b, b^{gt})}{c^2} \tag{1}$$

where R_{DIoU} is the penalty for the predicted box and the target box, b and b^{gt} are the central points of the predicted box and the target box, p is the distance, c is the diagonal length of the smallest enclosing box covering the two boxes, and d is the distance of central points of two boxes. Figure 2 illustrate the boxes and distances.

$$S_i = \begin{cases} S_i \cdot IoU - R_{DIoU}(M, B_i) < \varepsilon, \\ 0 \cdot IoU - R_{DIoU}(M, B_i) \geq \varepsilon, \end{cases} \tag{2}$$

S_i is the classification score; M presents the predicted box with the highest classification score; B_i is removed by simultaneously considering the IoU and the distance between central points of two boxes; ε is the NMS threshold.

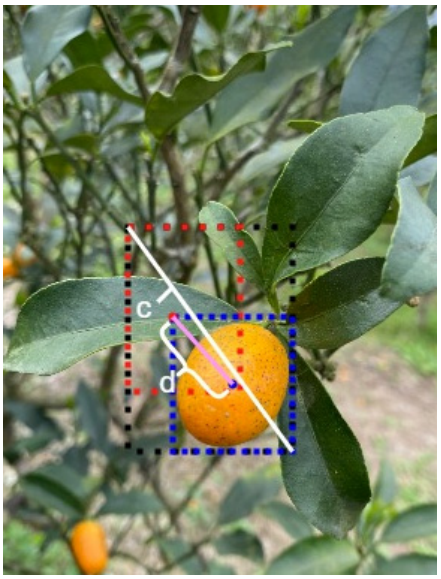


Figure 2. Distance of DIoU.

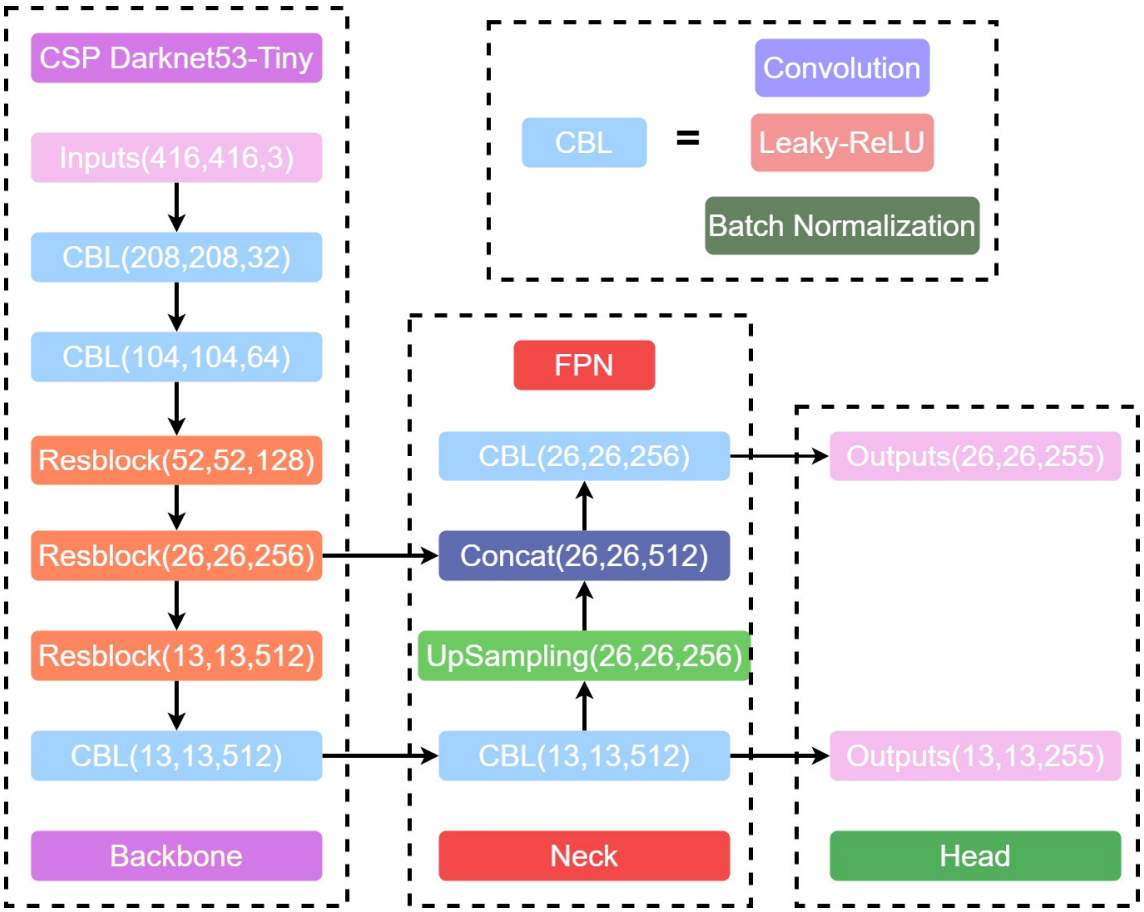
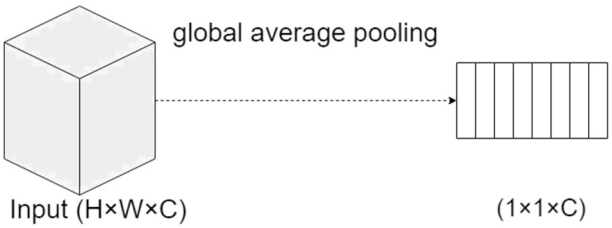
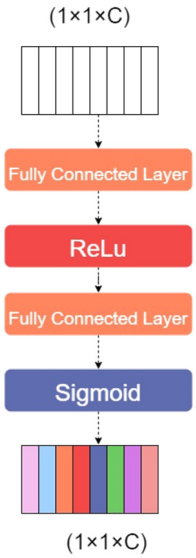


Figure 3. Architecture of YOLOV4-Tiny.

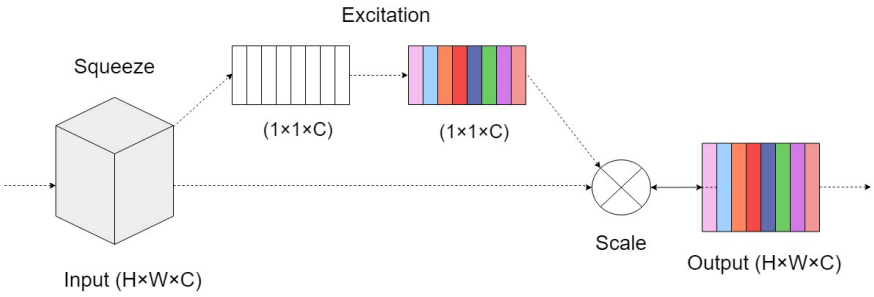
2) Based on the studies in Section 2.2, it can be found that SE learns the features of the image and fuses the features of different channels automatically on image classification, image segmentation and object detection. Figure 4 shows the architecture of squeeze, excitation and SE. The proposed GCS-YOLOV4-Tiny model adds SE between the CBL block with 304 × 304 × 32 and the CBL block with 152 × 152 × 64 to combine features of different resolutions, which finally improves the performance of the model.
3) The SPP module avoids the distortion problem caused by image scaling and fuses local and global features. The proposed GCS-YOLOV4-Tiny model adds two SPPs before the full connection layer to enhance feature diversity. Figure 5 represents the architecture of the proposed SPP.
4) To speed up the training time, instead of using traditional convolution, the proposed GCS-YOLOV4-Tiny model selects group convolution. As shown in Figure 6, the left-hand side describes the traditional convolution. The feature maps of input and output are 12 and 6, respectively. Using group convolution, the feature maps of input and output are 12 and 3, respectively, in the right-hand side of Figure 6. Group convolution requires fewer computational resources and reduces model size greatly.



(a) Squeeze.



(b) Excitation.



(c) Architecture of SE.

Figure 4. Architecture of squeeze, excitation, and SE.

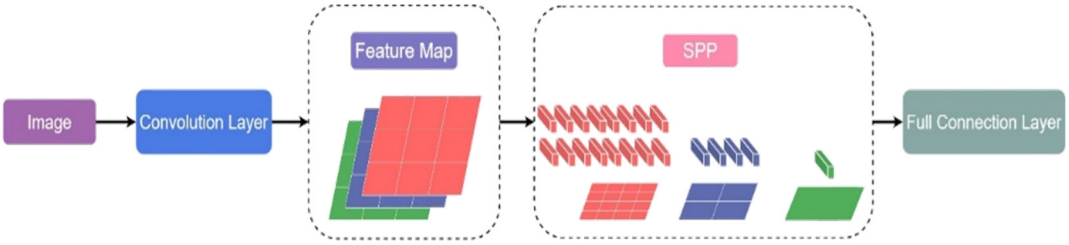


Figure 5. Architecture of SPP.

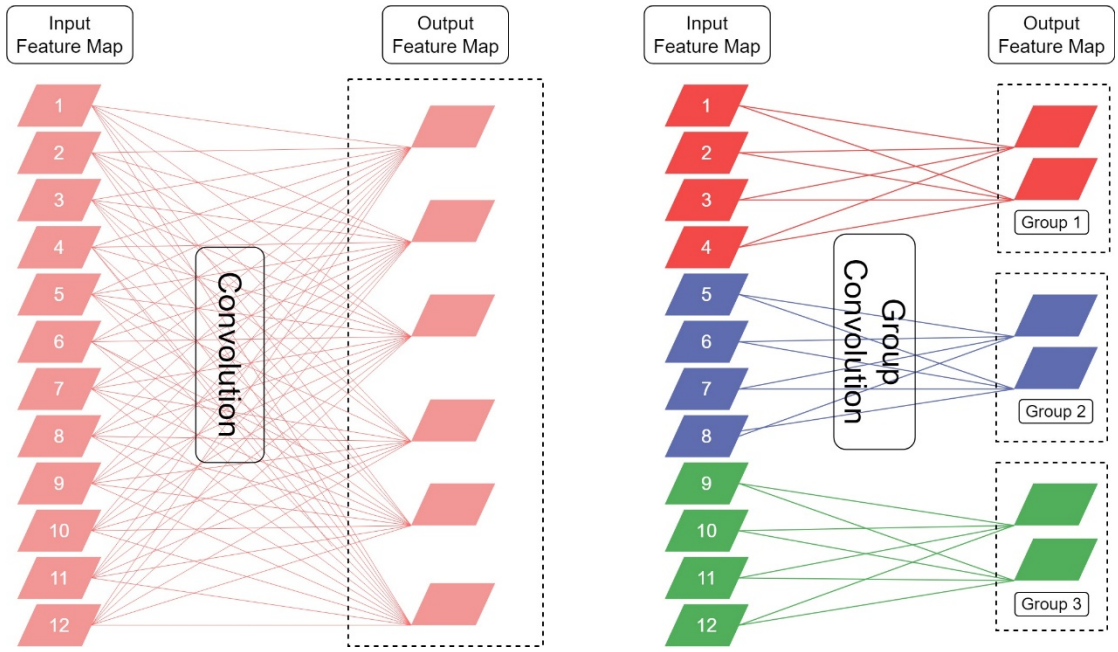


Figure 6. Traditional convolution and group convolution.

5) In summary, the proposed GCS-YOLOV4-Tiny model (a) selects YOLOV4-Tiny as backbone, (b) uses DIOUS- Non Maximum Suppression, (c) adds two SE blocks, (d) adds two SPPs and (e) replaces traditional convolution by group convolution. Figure 7 displays the architecture of the proposed GCS-YOLOV4-Tiny model.

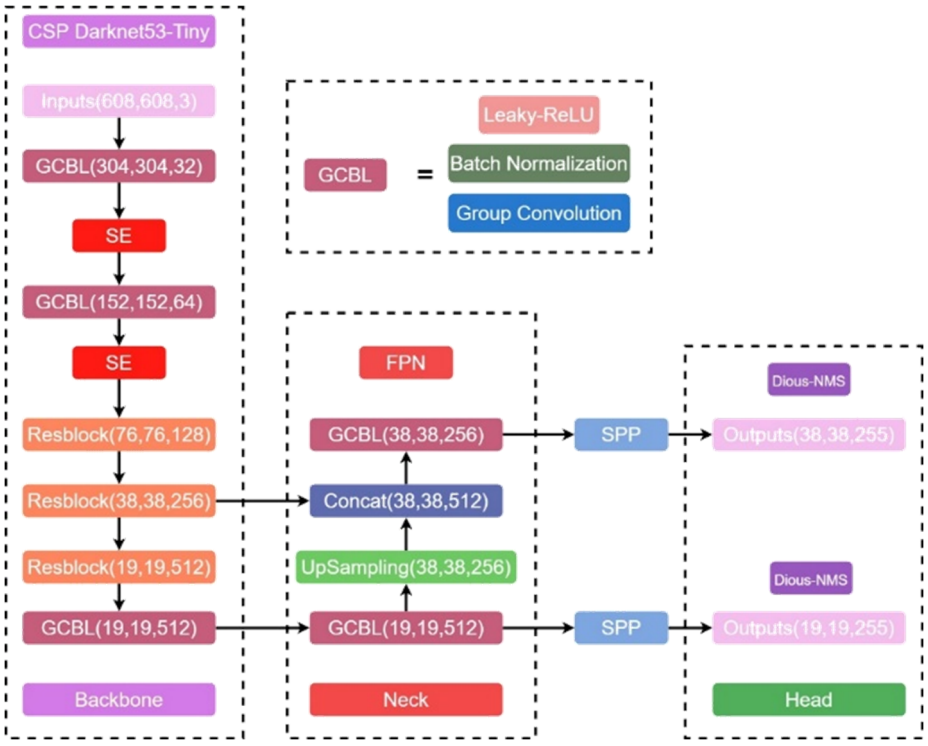


Figure 7. Architecture of GCS-YOLOV4-Tiny.

3.3. Evaluation metrics

To evaluate the performance of the proposed GCS-YOLOV4-Tiny model, Eqs (3)–(8) illustrate the commonly used indices, including Precision, Recall, F1-Score, Accuracy Precision (AP), mean Average Precision (mAP) and Intersection over Union (IoU).

Precision represents the number of positive class predictions that actually belong to the positive class.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Recall represents the number of positive class predictions made out of all positive examples in the dataset.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

F1-Score combines Precision and Recall.

$$\text{F1} - \text{Score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

TP (true positive) represents the number of positive categories that are correctly classified as positive; FP (false positive) represents the number of negative categories that are incorrectly classified as positive; FN (false negative) refers to the number of positive categories that are incorrectly classified as negative.

Accuracy Precision (AP) is the area of the curve contained in the precision and recall, and it represents the accuracy of the detection.

$$\text{AP} = \sum_{K=1}^{M} Precision(K) \triangle Recall(K) \tag{6}$$

mAP represents the average of Accuracy Precision.

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

where M is the number of images, and K is the number of categories.

Intersection over Union (IoU) is the fraction of the Area of Overlap divided by the Area of Union.

$$\text{IoU} = \frac{Area\ \ of\ \ Overlap}{Area\ \ of\ \ Union} \tag{8}$$

where the Area of Overlap (gray box in Figure 9) is the intersection between the predicted bounding box (red box in Figure 8) and the ground-truth bounding box (blue box in Figure 8), and the Area of Union (gray box in Figure 10) is the area encompassed by both the predicted bounding box and the ground-truth bounding box.

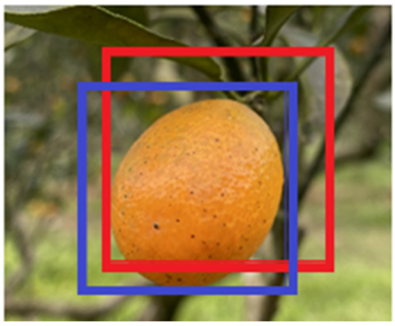


Figure 8. Ground-truth and prediction.

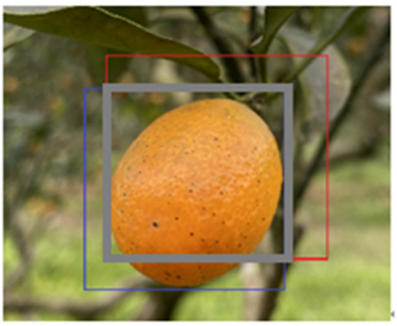


Figure 9. Area of Overlap.

Figure 10. Area of Union.

4. Experiments and results

Three datasets, Mango YOLO, Rpi-Tomato and *F. margarita*, were used in this study. The input image size was adjusted to 608 × 608 for GCS-YOLOV4-Tiny with a batch size of 64 epoch count of 6000 learning rate of 0.001 and decay rate of 0.0005. The equipment used in the experiment is an Intel(R) Core (TM) i7-8700 @ 3.20 GHz CPU, NVIDIA GeForce RTX 2080. The whole experiments were performed using Python 3.8 [Python Software Foundation, Fredericksburg, Virginia, USA].

4.1. Ablation experiment

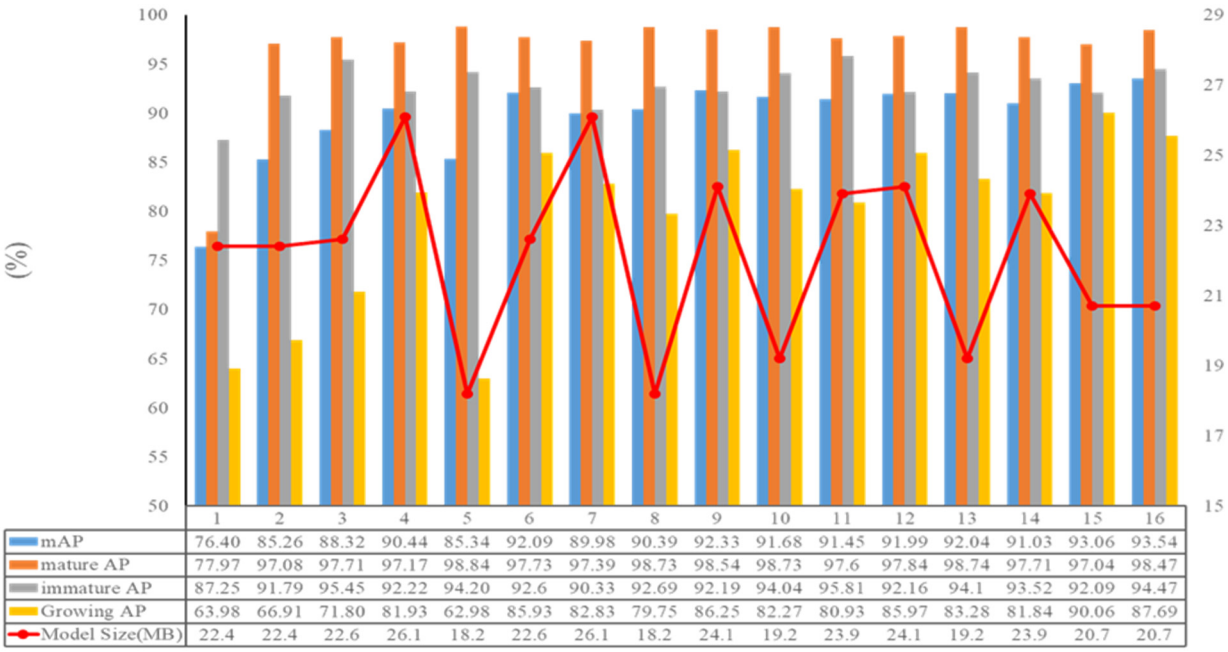
To compare the performances of using DIOU-NMS, adding SE, adding SPP, or using group

convolution in YOLOV4-Tiny on the *F. margarita* dataset, this study conducted sixteen experiments and evaluated the performance of the proposed GCS-YOLOV4-Tiny model, as shown in Table 2. The APs for mature, immature and growing groups were 77.97, 87.25 and 63.98%, respectively, in the original YOLOV4-TINY model (experiment #1). The lowest AP occurs in the growing group for each of the experiments. The APs for mature and immature groups were above 90% for all experiments except experiment #1. The mAP was 76.40% in the original YOLOV4-TINY model (experiment #1), and it increased to 85.26% when using DIOU-NMS (experiment #2). The mAP values of most of the experiments (experiments #4, #9, #11, #12, #14, #15 and #16) when adding SPP were greater than 90%. The highest mAP was 93.54%, when using DIOU-NMS and group convolution and adding SE and SPP into YOLOV4-TINY (experiment #16). It is worth noting that the AP of the growing group has been greatly enhanced from 63.98 to 87.69%, which makes a substantial contribution to mAP.

Table 2. Ablation experiment of GCS-YOLOV4-Tiny.

	YOLOV4-Tiny	DIOU-NMS	SE	SPP	Group Conv.	mAP (%)	mature AP (%)	immature AP (%)	growing AP (%)
1	X					76.40	77.97	87.25	63.98
2	X	X				85.26	97.08	91.79	66.91
3	X		X			88.32	97.71	95.45	71.80
4	X			X		90.44	97.17	92.22	81.93
5	X				X	85.34	98.84	94.20	62.98
6	X	X	X			92.09	97.73	92.60	85.93
7	X	X		X		89.98	97.39	90.33	82.83
8	X	X			X	90.39	98.73	92.69	79.75
9	X		X	X		92.33	98.54	92.19	86.25
10	X		X		X	91.68	98.73	94.04	82.27
11	X			X	X	91.45	97.60	95.81	80.93
12	X	X	X	X		91.99	97.84	92.16	85.97
13	X	X	X		X	92.04	98.74	94.10	83.28
14	X	X		X	X	91.03	97.71	93.52	81.84
15	X		X	X	X	93.06	97.04	92.09	90.06
16	X	X	X	X	X	93.54	98.47	94.47	87.69

Figure 11 presents the results of the ablation study including the APs for mature, immature and growing groups, mAP and model size for each experiment. The minimum model size is 18.20 MB, from experiment #5, which is the original YOLOV4-TINY model using group convolution. Unfortunately, the mAP of experiment #5 is not favorable. The model size for experiment #16, with the highest mAP, is 20.7 MB. Therefore, experiment #16 is the final architecture of the proposed GCS-YOLOV4-Tiny model.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
mAP	76.40	85.26	88.32	90.44	85.34	92.09	89.98	90.39	92.33	91.68	91.45	91.99	92.04	91.03	93.06	93.54
mature AP	77.97	97.08	97.71	97.17	98.84	97.73	97.39	98.73	98.54	98.73	97.6	97.84	98.74	97.71	97.04	98.47
immature AP	87.25	91.79	95.45	92.22	94.20	92.6	90.33	92.69	92.19	94.04	95.81	92.16	94.1	93.52	92.09	94.47
Growing AP	63.98	66.91	71.80	81.93	62.98	85.93	82.83	79.75	86.25	82.27	80.93	85.97	83.28	81.84	90.06	87.69
Model Size(MB)	22.4	22.4	22.6	26.1	18.2	22.6	26.1	18.2	24.1	19.2	23.9	24.1	19.2	23.9	20.7	20.7

Figure 11. Ablation experiment of GCS-YOLOV4-Tiny.

4.2. Results on Mango YOLO

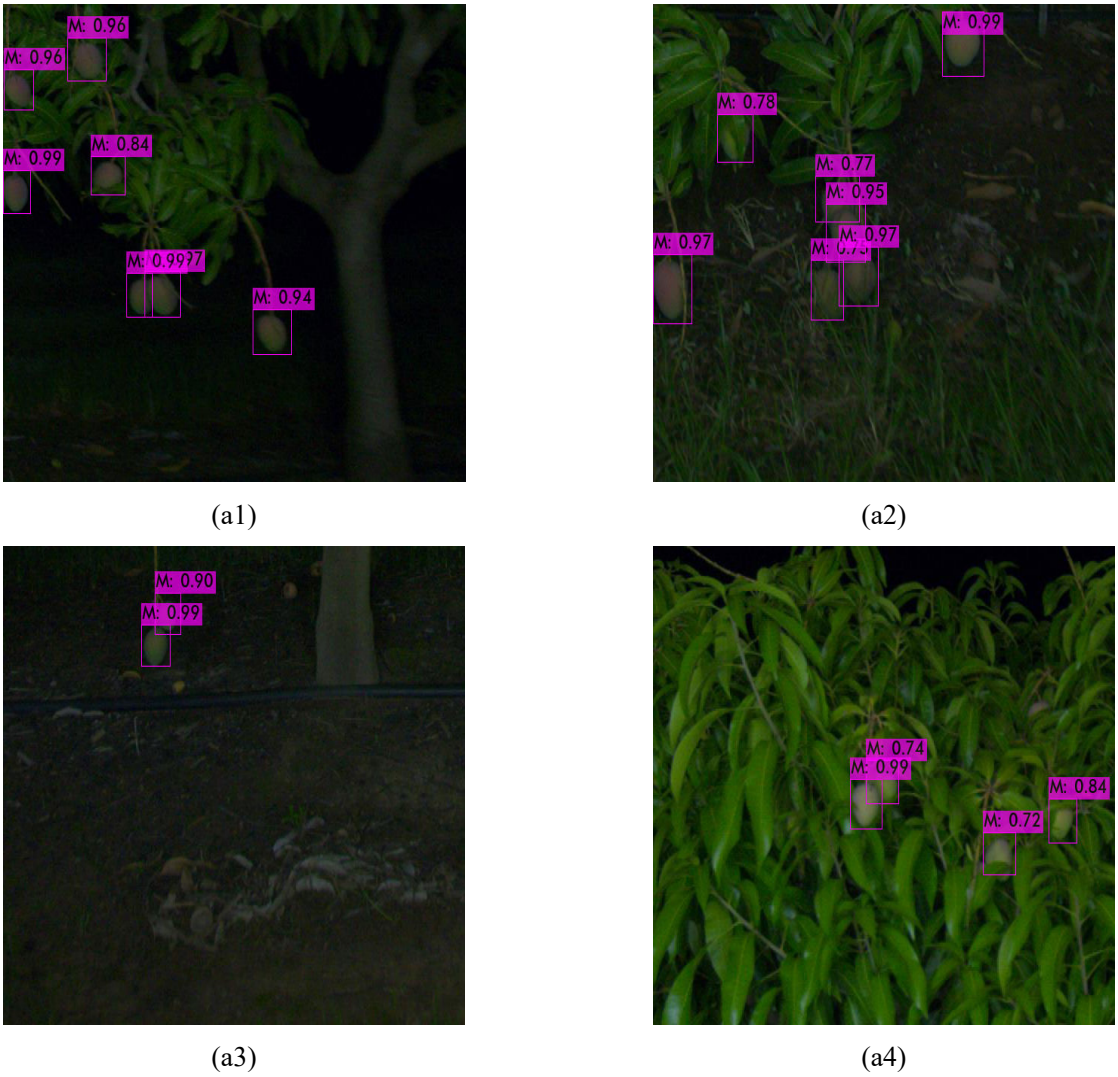
The proposed GCS-YOLOV4-Tiny model was first evaluated using the Mango YOLO dataset, which has only one category. The performance values of GCS-YOLOV4-Tiny, including AP, Recall, F1-Score, Precision and Average IoU are 91.91, 79.00, 88.00, 98.00 and 81.10%, respectively. Table 3 compares the performances of the related studies using the same dataset.

Table 3. Results of Mango YOLO dataset.

Study	Model	AP	Recall	F1-Score	Precision	Average IoU
Koirala et al. (2019) [55]	MangoYOLO	81.15%	–	–	98.86%	–
Kateb et al. (2021) [59]	FruitDet	81.50%	–	–	99.18%	–
Kateb et al. (2021) [59]	YOLOV3	80.90%	–	–	98.60%	–
Kateb et al. (2021) [59]	YOLOV4	81.20%	–	–	98.62%	–
Bochkovskiy et al. (2020) [20]	YOLOV4-Tiny	72.78%	55.00%	71.00%	100.00%	80.71%
This study	GCS-YOLOV4-Tiny	91.91%	79.00%	88.00%	98.00%	81.10%

The proposed GCS-YOLOV4-Tiny achieves the highest AP, while the best Precision is using YOLOV4-Tiny. Figure 12 displays detection results from the proposed GCS-YOLOV4-Tiny model.

The proposed model detects most of the mangos, even though some of them are covered by leaves.



(a1) (a2)

(a3) (a4)

Figure 12. Examples of GCS-YOLOV4-Tiny on Mango YOLO dataset.

4.3. Results on Rpi-Tomato

The second dataset used in this study is Rpi-Tomato, with tomato images of four different maturity levels (green, red, light red and red). Related studies include Moreira et al. [56], which applied SSD MobileNet v2, YOLOV4 and HSV Color Space. This study executed YOLOV4-Tiny and the proposed GCS-YOLOV4-Tiny. Results are shown in Table 4. Among the four classes, the lowest AP occurred in the "Turning" class, while the highest AP comes from the "Light Red" class in both YOLOV4-Tiny and GCS-YOLOV4-Tiny models. YOLOV4-Tiny and GCS-YOLOV4-Tiny models outperform SSD MobileNet v2, YOLOV4 and HSV Color Space on most of the performance indices.

The GCS-YOLOV4-Tiny model results are similar to the YOLOV4-Tiny model results on the Rpi-Tomato dataset. Figure 13 displays detection results from the proposed GCS-YOLOV4-Tiny model for four classes in the Rpi-Tomato dataset. Figure 13 (b1)–(b4) present green, red, light red and red classes, respectively.

Table 4. Results of Rpi-Tomato dataset.

Study	Model	Class	AP	Recall	F1-Score	Precision	Average IoU
Moreira et al. (2022) [56]	SSD MobileNet v2	Green	62.70%	70.09%	65.93%	77.27%	–
		Turning		55.88%		59.38%	–
		Light Red		40.82%		60.61%	–
		Red		84.00%		80.77%	–
Moreira et al. (2022) [56]	YOLOV4	Green	68.87%	84.66%	74.16%	85.38%	–
		Turning		67.65%		70.77%	–
		Light Red		59.18%		76.32%	–
		Red		64.00%		88.89%	–
Moreira et al. (2022) [56]	HSV Color Space	Green	68.10%	98.31%	70.93%	98.24%	–
		Turning		63.24%		50.00%	–
		Light Red		42.86%		58.33%	–
		Red		68.00%		89.47%	–
Bochkovskiy et al. (2020) [20]	YOLOV4-Tiny	Green	85.85%	75.00%	73.00%	72.00%	58.78%
		Turning	52.74%	78.00%	53.00%	40.00%	32.83%
		Light Red	98.33%	92.00%	92.00%	92.00%	72.59%
		Red	93.75%	94.00%	97.00%	100.00%	85.24%
This study	GCS-YOLOV4-Tiny	Green	71.31%	71.00%	77.00%	85.00%	65.59%
		Turning	65.82%	67.00%	67.00%	67.00%	49.22%
		Light Red	99.36%	88.00%	74.00%	100.00%	81.17%
		Red	81.25%	81.00%	90.00%	100.00%	77.22%

(b1)

(b2)

(b3)

(b4)

Figure 13. Examples of GCS-YOLOV4-Tiny on Rpi-Tomato dataset.

4.4. Results on F. margarita

YOLOV3, YOLOV3-Tiny, YOLOV4, YOLOV4-Tiny and GCS-YOLOV4-Tiny models were tested using the *F. margarita* dataset [57] with five-fold cross validation. Results from the proposed GCS-YOLOV4-Tiny are shown in Table 5. The AP values for mature, immature and growing were 98.34 ± 0.75, 93.72 ± 1.53 and 87.80 ± 2.11%, respectively, and the mAP was 93.42 ± 0.44%. The Recall, F1-score, Precision, and Average IoU of GCS-YOLOV4-Tiny were 91.00 ± 1.87, 90.80 ± 2.59, 90.80 ± 2.77 and 76.94 ± 1.35%, respectively.

Table 5. Results of GCS-YOLOV4-Tiny.

Fold	mAP	mature AP	immature AP	growing AP	Recall	F1-Score	Precision	Average IoU
1	93.91%	98.00%	93.22%	90.51%	90.00%	90.00%	90.00%	77.52%
2	93.41%	99.48%	96.02%	84.72%	94.00%	95.00%	95.00%	76.07%
3	93.54%	98.47%	94.47%	87.69%	91.00%	91.00%	92.00%	78.95%
4	92.71%	98.31%	92.45%	87.37%	89.00%	88.00%	88.00%	75.51%
5	92.85%	97.43%	92.42%	88.69%	91.00%	90.00%	89.00%	76.66%
Average	93.42%	98.34%	93.72%	87.80%	91.00%	90.80%	90.80%	76.94%
STDV	0.44%	0.75%	1.53%	2.11%	1.87%	2.59%	2.77%	1.35%

Table 6 records the training time for each fold in the GCS-YOLOV4-Tiny model. The average training time for each fold is 3hr 34min.

Table 6. Training time of GCS-YOLOV4-Tiny.

Fold	Training time
1	3hr 34min
2	3hr 33min
3	3hr 36min
4	3hr 33min
5	3hr 35min
Total	17hr 51min
Average	3hr 34min

Table 7 compares the performance metrics for models of YOLOV3, YOLOV3-Tiny, YOLOV4, YOLOV4-Tiny and the proposed GCS-YOLOV4-Tiny models. ANOVA evaluates the differences among models. The proposed GCS -YOLOV4-Tiny model scores the highest mAP of 93.42%. The same results were found for mature AP, immature AP, growing AP, Recall, F1-Score, Precision and Average IoU. Notably, the proposed GCS-YOLOV4-Tiny model largely improved the AP to 87.80% in the growing group, as the values were around 60–70% in the other four models.

Figure 14 uses box plots to illustrate the performances of the above five models. Obviously, the proposed GCS-YOLOV4-Tiny model achieves higher averages with smaller standard deviations and dominates the other four models in all metrics.

Table 7. Performance comparisons of five models.

Model	mAP	mature AP	immature AP	growing AP	Recall	F1-Score	Precision	Average IoU
GCS-YOLOV4-Tiny	93.42 ± 0.44	98.34 ± 0.75	93.72 ± 1.53	87.80 ± 2.11	91.00 ± 1.87	90.80 ± 2.59	90.80 ± 2.77	76.94 ± 1.35
YOLOV3	83.67 ± 1.16	86.68 ± 1.26	93.33 ± 1.27	70.98 ± 3.36	80.60 ± 2.70	81.20 ±2.59	81.80 ± 3.42	68.82 ± 3.29
YOLOV3-Tiny	80.54 ± 1.27	85.74 ± 0.93	90.95 ± 1.04	64.95 ± 4.48	78.80 ± 2.17	78.60 ±1.52	78.00 ± 1.22	64.31 ± 1.21
YOLOV4	83.90 ± 1.84	93.10 ± 1.93	91.10 ± 1.09	67.51 ± 4.03	88.20 ± 2.17	81.00 ± 1.41	75.00 ± 2.35	63.66 ± 2.15
YOLOV4-Tiny	75.97 ± 0.83	78.46 ± 0.43	86.67 ± 0.58	62.76 ± 2.57	72.80 ± 0.84	77.00 ± 1.87	81.20 ±3.77	68.45 ± 4.44
P	0.00**	0.00**	0.00**	0.00*	0.00**	0.00**	0.00*	0.00*

*p<0.05; **p<0.01

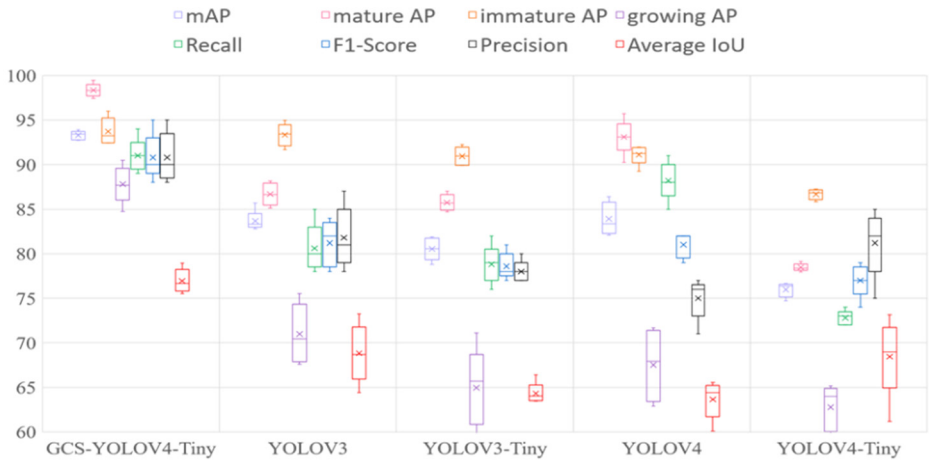


Figure 14. Box plots of the five models.

In addition to the performance metrics in Table 8, this study compares the average training times, Billion Float Operations (BFLOPs) and model sizes among the five models in Table 8. YOLOV4 has the longest average training time of 14hr 26min, the highest BFLOPs of 127.26 and the maximum model size of 244 MB. The average training time of GCS-YOLOV4-Tiny is 3hr35min, which is a little bit longer than that of YOLOV4-Tiny. A similar situation occurs for BFLOPs. Although YOLOV4-Tiny has a shorter average training time and fewer BFLOPs than those of GCS-YOLOV4-Tiny, the model size of GCS-YOLOV4-Tiny is 20.70 MB, which is smaller than that of YOLOV4-Tiny (22.40 MB). Figure 15 plots the mAP values and model sizes for the five models. Obviously, the proposed model achieves the highest mAP of 93.42% with the smalles model size of 20.70 MB. Figure 16–20 show detection examples on the *F. margarita* dataset by YOLOV3, YOLOV3-Tiny, YOLOV4, YOLOV4-Tiny, and the proposed GCS-YOLOV4-Tiny models.

Table 8. The average training times, BFLOPs and model sizes.

Model	Average Training Time	BFLOPs	Model Size (MB)
GCS-YOLOV4-Tiny	3hr 35min	10.20	20.70
YOLOV3	5hr 02min	65.32	234.00
YOLOV3-Tiny	3hr 44min	5.45	33.10
YOLOV4	14hr 26min	127.26	244.00
YOLOV4-Tiny	3hr 28min	6.79	22.40

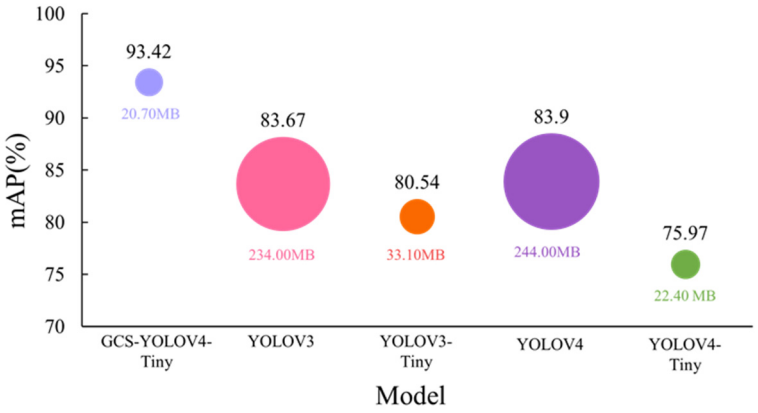


Figure 15. The mAP values and model sizes for five models.

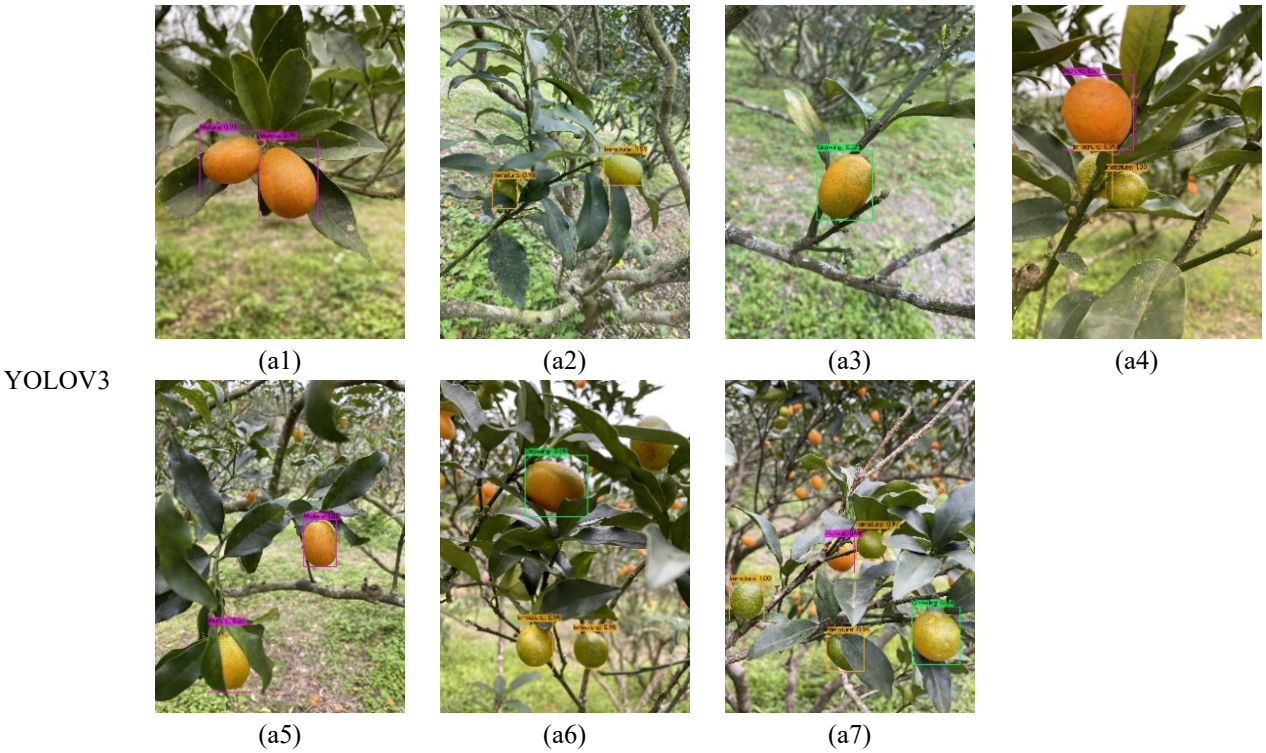


Figure 16. Detection examples by YOLOV3.

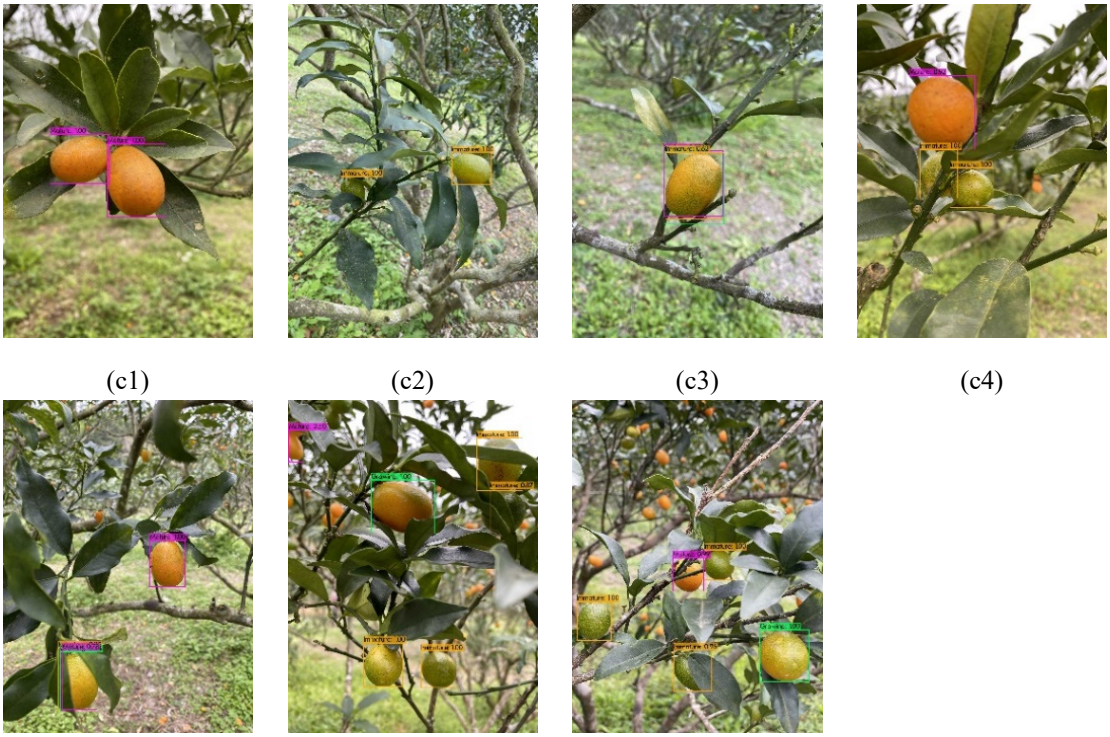
YOLOV3-Tiny

(b1) (b2) (b3) (b4)

(b5) (b6) (b7)

Figure 17. Detection examples by YOLOV3-Tiny.

YOLOV4



(c1) (c2) (c3) (c4)

(c5) (c6) (c7)

Figure 18. Detection examples by YOLOV4.

YOLOV4-Tiny


(d1)


(d2)


(d3)


(d4)


(d5)


(d6)


(d7)

Figure 19. Detection examples by YOLOV4-Tiny.

GCS-YOLOV4-Tiny

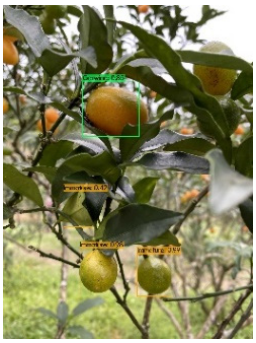

(e1)


(e2)


(e3)


(e4)


(e5)


(e6)


(e7)

Figure 20. Detection examples by GCS-YOLOV4-Tiny.

5. Conclusions

Based on YOLOV4-Tiny, this study proposes a one-stage detection model, GCS-YOLOV4-Tiny, to examine fruits with different sizes in a complex environment. The proposed GCS-YOLOV4-Tiny model uses DIOUS-NMS and adds SE and SPP modules. Furthermore, the group convolution was applied to reduce model size greatly. With the smallest model size of 20.70 MB, the detection results outperform the state-of-the-art YOLOV4-Tiny model with a 17.45% increase in mAP and a 13.80% increase in F1-score on the *F. margarita* dataset. The proposed model provides an effective and efficient performance to detect different growth stages of fruits, which is beneficial for real-time detection.

This study mainly selects the one-stage detection YOLO algorithm and focuses on construction of a lightweight network to perform real-time detection on fruit growth stages. The two-stage detectors RCNN, SSD and mask-RCNN were not compared in this study. Furthermore, due to the limitation of hardware equipment, this research did not use the latest YOLO version. Although the performance of the proposed model is favorable, there is the possibility to modify the architecture of the proposed model or of using the latest network version to achieve better performance in the future.

Acknowledgments

The research was partially funded by the National Science and Technology Council of Taiwan, R.O.C. (Research Grant Project number MOST 111-2221-E-167-007-MY3).

Conflict of interest

The authors declare there is no conflict of interest.

Ethics Statement

This study did not conduct experiments involving humans and animals.

References

1. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2014), 580–587. https://doi.org/10.1109/CVPR.2014.81
2. J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, Selective search for object recognition, *Int. J. Comput. Vision*, **104** (2013), 154–171. https://doi.org/10.1007/s11263-013-0620-5
3. H. Jiang, C. Zhang, Y. Qiao, Z. Zhang, W. Zhang, C. Song, CNN feature based graph convolutional network for weed and crop recognition in smart farming, *Comput. Electron. Agric.*, **174** (2020), 105450. https://doi.org/10.1016/j.compag.2020.105450
4. R. Girshick, Fast R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169

5. K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in *Computer Vision – ECCV 2014*, **8691** (2014), 346–361. https://doi.org/10.1007/978-3-319-10578-9_23
6. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE T. Pattern Anal.*, **39** (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
7. X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, R. Chen,et al., Feature pyramid networks for object detection, in *2019 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, (2019), 1500–1504. https://doi.org/10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217
8. R. Ghosh, A Faster R-CNN and recurrent neural network based approach of gait recognition with and without carried objects, *Expert Syst. Appl.*, **205** (2022), 117730. https://doi.org/10.1016/j.eswa.2022.117730
9. M. Chen, L.Yu, C. Zhi, R. Sun, S. Zhu, Z. Gao, et al., Improved faster R-CNN for fabric defect detection based on gabor filter with genetic algorithm optimization, *Comput. Ind.*, **134** (2022), 103551. https://doi.org/10.1016/j.compind.2021.103551
10. D. Miao, W. Pedrycz, D. Ślezak, G. Peters, Q. Hu, R. Wang, Rough Sets and Knowledge Technology, in *RSKT: International Conference on Rough Sets and Knowledge Technology*, (2014), 364–375. https://doi.org/10.1007/978-3-319-11740-9
11. F. Cui, M. Ning, J. Shen, X. Shu, Automatic recognition and tracking of highway layer-interface using Faster R-CNN, *J. Appl. Geophys.*, **196** (2022), 104477. https://doi.org/10.1016/j.jappgeo.2021.104477
12. Y. Su, D. Li, X. Chen, Lung Nodule Detection based on Faster R-CNN Framework, *Comput. Meth. Prog. Bio.*, **200** (2021), 105866. https://doi.org/10.1016/j.cmpb.2020.105866
13. M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, preprint, arXiv：1311.2901
14. W. Yang, Z. Li, C. Wang, J. Li, A multi-task Faster R-CNN method for 3D vehicle detection based on a single image, *Appl. Soft Comput. J.*, **95** (2020), 106533. https://doi.org/10.1016/j.asoc.2020.106533
15. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 779–788. https://doi.org/10.1109/CVPR.2016.91
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 1–9. https://doi.org/10.1109/CVPR.2015.7298594
17. J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 6517–6525. https://doi.org/10.1109/CVPR.2017.690
18. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, preprint,arXiv:1804.02767
19. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90

20. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, preprint, arXiv: 2004.10934
21. C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, I. H. Yeh, CSPNet: A new backbone that can enhance learning capability of CNN, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (2020), 1571–1580. https://doi.org/10.1109/CVPRW50498.2020.00203
22. Y. Lin, R. Cai, P. Lin, S. Cheng, A detection approach for bundled log ends using K-median clustering and improved YOLOv4-Tiny network, *Comput. Electron. Agr.*, **194** (2022), 106700. https://doi.org/10.1016/j.compag.2022.106700
23. A. Kumar, A. Kalia, A. Kalia, ETL-YOLO v4: A face mask detection algorithm in era of COVID-19 pandemic, *Optik.*, **259** (2022), 169051. https://doi.org/10.1016/j.ijleo.2022.169051
24. Y. Wang, G. Yan, Q. Meng, T. Yao, J. Han, B. Zhang, DSE-YOLO: Detail semantics enhancement YOLO for multi-stage strawberry detection, *Comput. Electron. Agr.*, **198** (2022), 107057. https://doi.org/10.1016/j.compag.2022.107057
25. Y. Su, Q. Liu, W. Xie, P. Hu, YOLO-LOGO: A transformer-based YOLO segmentation model for breast mass detection and segmentation in digital mammograms, *Comput. Meth. Prog. Bio.*, **221** (2022), 106903. https://doi.org/10.1016/j.cmpb.2022.106903
26. P. Wu, H. Li, N. Zeng, F. Li, FMD-Yolo: An efficient face mask detection method for COVID-19 prevention and control in public, *Image Vision Comput.*, **117** (2022), 104341. https://doi.org/10.1016/j.imavis.2021.104341
27. X. Wang, Q. Zhao, P. Jiang, Y. Zheng, L. Yuan, P. Yuan, LDS-YOLO: A lightweight small object detection method for dead trees from shelter forest, *Comput. Electron. Agr.*, **198** (2022), 107035. https://doi.org/10.1016/j.compag.2022.107035
28. S. Zhao, S. Zhang, J. Lu, H. Wang, Y. Feng, C. Shi,et al., A lightweight dead fish detection method based on deformable convolution and YOLOV4, *Comput. Electron. Agr.*, **198** (2022), 107098. https://doi.org/10.1016/j.compag.2022.107098
29. Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, Z. Liang, Apple detection during different growth stages in orchards using the improved YOLO-V3 model, *Comput. Electron. Agr.*, **157** (2019), 417–426. https://doi.org/10.1016/j.compag.2019.01.012
30. H. Mirhaji, M. Soleymani, A. Asakereh, S. Abdanan Mehdizadeh, Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions, *Comput. Electron. Agr.*, **191** (2021), 106533. https://doi.org/10.1016/j.compag.2021.106533
31. M. O. Lawal, Tomato detection based on modified YOLOv3 framework. *Sci. Rep.*, **1447** (2021). https://doi.org/10.1038/s41598-021-81216-5
32. A. M. Roy, R. Bose, J. Bhaduri, A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. *Neural Computing and Applications*, **34** (2022), 3895–3921. https://doi.org/10.1007/s00521-021-06651-x
33. A. M. Roy, J. Bhaduri, A Deep Learning Enabled Multi-Class Plant Disease Detection Model Based on Computer Vision. *AI*, **2**(2021), 413-428. https://doi.org/10.3390/ai2030026
34. A. M. Roy, J. Bhaduri, Real-time growth stage detection model for high degree of occultation using DenseNet-fused YOLOv4, *Comput. Electron. Agr.*, **193** (2022), 106694. https://doi.org/10.1016/j.compag.2022.106694

35. H. Li, C. Li, G. Li, L. Chen, A real-time table grape detection method based on improved YOLOv4-tiny network in complex background, *Biosyst. Eng.*, **212** (2021), 347–359. https://doi.org/10.1016/j.biosystemseng.2021.11.011
36. X. Li, J. D. Pan, F. P. Xie, J. P. Zeng, Q. Li, X. J. Huang, et al., Fast and accurate green pepper detection in complex backgrounds via an improved Yolov4-tiny model, *Comput. Electron. Agr.*, **191** (2021), 106503. https://doi.org/10.1016/j.compag.2021.106503
37. C. Y. Wang, A. Bochkovskiy, H. Y. M. Liao, Scaled-YOLOv4: Scaling Cross Stage Partial Network, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 13024-13033. https://doi.org/10.1109/CVPR46437.2021.01283
38. Y. Zhang, J. Yu, Y. Chen, W. Yang, W. Zhang, Y. He, Real-time strawberry detection using deep neural networks on embedded system (rtsd-net): An edge AI application, *Comput. Electron. Agr.*, **192** (2022), 106586. https://doi.org/10.1016/j.compag.2021.106586
39. Y. Yao , L. Han, C. Du, X. Xu, X. Xu,Traffic sign detection algorithm based on improved YOLOv4-Tiny, *Signal Process.-Image*, **107** (2022), 116783. https://doi.org/10.1016/j.image.2022.116783
40. G. Han, M. He, F. Zhao, Z. Xu, M .Zhang, L.Qin, Insulator detection and damage identification based on improved lightweight YOLOv4 network, *Energy Rep.*, **7** (2021), 187–197. https://doi.org/10.1016/j.egyr.2021.10.039
41. Q. Zhang, X. Bao, B. Wu, X. Tu, Y. Jin, Y. Luo, et al., Water meter pointer reading recognition method based on target-key point detection, *Flow Meas. Instrum.*, **81** (2021), 102012. https://doi.org/10.1016/j.flowmeasinst.2021.102012
42. J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-Excitation Networks, *IEEE T. Pattern Anal.*, **42** (2020), 2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372
43. J. Wang, P. Lv, H. Wang, C. Shi, SAR-U-Net: Squeeze-and-excitation block and atrous spatial pyramid pooling based residual U-Net for automatic liver segmentation in Computed Tomography, *Comput. Meth. Prog. Bio.*, **208** (2021), 106268. https://doi.org/10.1016/j.cmpb.2021.106268
44. H. Ma, G. Han, L. Peng, L. Zhu, J. Shu, Rock thin sections identification based on improved squeeze-and-Excitation Networks model, *Comput. Geosci.*, **152** (2021), 104780. https://doi.org/10.1016/j.cageo.2021.104780
45. M. M. Khan, M. S. Uddin, M. Z. Parvez, L. Nahar, A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition, *J. King Saud Univ.-Com.*, **34** (2022), 3356–3364. https://doi.org/10.1016/j.jksuci.2021.01.021
46. B. N. Naik, R. Malmathanraj, P. Palanisamy, Detection and classification of chilli leaf disease using a squeeze-and-excitation-based CNN model, *Ecol. Inform.*, **69** (2022), 101663. https://doi.org/10.1016/j.ecoinf.2022.101663
47. G. Huang, Z. Wan, X. Liu, J. Hui, Z. Wang, Z. Zhang, Ship detection based on squeeze excitation skip-connection path networks for optical remote sensing images, *Neurocomputing*, **332** (2019), 215–223. https://doi.org/10.1016/j.neucom.2018.12.050
48. T. Alsarhan, U. Ali, H. Lu, Enhanced discriminative graph convolutional network with adaptive temporal modelling for skeleton-based action recognition, *Comput. Vis. Image Und.*, **216** (2022), 103348. https://doi.org/10.1016/j.cviu.2021.103348
49. P. S. Yee, K. M. Lim, C. P. Lee, DeepScene: Scene classification via convolutional neural network with spatial pyramid pooling, *Expert Syst. Appl.*, **193** (2022),116382. https://doi.org/10.1016/j.eswa.2021.116382

50. E. Prasetyo, N. Suciati, C. Fatichah, Yolov4-tiny with wing convolution layer for detecting fish body part, *Comput. Electron. Agr.*, **198** (2022), 107023. https://doi.org/10.1016/j.compag.2022.107023
51. J. Li, G. Xu, X. Cheng, Combining spatial pyramid pooling and long short-term memory network to predict PM2.5 concentration, *Atmos. Pollut. Res.*, **13** (2022), 101309. https://doi.org/10.1016/j.apr.2021.101309
52. Z. Li, G. Zhou, T. Zhang, Interleaved group convolutions for multitemporal multisensor crop classification, *Infrared Phys. Techn*, **102** (2019), 103023. https://doi.org/10.1016/j.infrared.2019.103023
53. A. Yang, B. Yang, Z. Ji, Y. Pang, L. Shao, Lightweight group convolutional network for single image super-resolution, *Inf. Sci.*, **516** (2020), 220–233. https://doi.org/10.1016/j.ins.2019.12.057
54. C. Tian, Y. Yuan, S. Zhang, C. Lin, W. Zuo, D. Zhang, Image super-resolution with an enhanced group convolutional neural network, *Neural Networks*, **153** (2022), 373-385. https://doi.org/10.1016/j.neunet.2022.06.009
55. A. Koirala, K. B. Walsh, Z. Wang, C. McCarthy, Deep learning–Method overview and review of use for fruit detection and yield estimation, *Comput. Electron. Agr.*, **162** (2019), 219–234. https://doi.org/10.1016/j.compag.2019.04.017
56. G. Moreira, S. A. Magalhães, T. Pinho, F. N. DosSantos, M. Cunha, Benchmark of Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato, *Agronomy*, **12** (2022), 356. https://doi.org/10.3390/agronomy12020356
57. M. L. Huang, Y. S. Wu, A dataset of fortunella margarita images for object detection of deep learning based methods, *Data Brief*, **38** (2021), 107293. https://doi.org/10.1016/j.dib.2021.107293
58. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: Faster and better learning for bounding box regression, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999
59. F. A. Kateb, M. M. Monowar, M. A. Hamid, A. Q. Ohi, M. F. Mridha, FruitDet: Attentive feature aggregation for real-time fruit detection in orchards, *Agronomy*, **11** (2021), 2440. https://doi.org/10.3390/agronomy11122440