IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 June 2023, accepted 23 July 2023, date of publication 28 July 2023, date of current version 9 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299595

RESEARCH ARTICLE

Detection Mature Bud for Daylily Based on Faster R-CNN Integrated With CBAM

JUNHUI FENG[1,2], XUERONG ZHAO[1], TINGYU ZHU[1], TAO LI[1], ZHICHAO QIU[1], AND ZHIWEI LI[2,3]

[1]College of Agricultural Engineering, Shanxi Agricultural University, Taigu 030801, China
[2]Dryland Farm Machinery Key Technology and Equipment Key Laboratory of Shanxi Province, Taigu 030801, China
[3]College of Information Science and Engineering, Shanxi Agricultural University, Taigu 030801, China

Corresponding author: Zhiwei Li (lizhiweitong@163.com)

This work was supported in part by the Datong City and Shanxi Agricultural University Municipal Cooperative Scientific Research Project under Grant 2020HXDTHH05, in part by the Sub-Project of the National Key Research and Development Program of China under Grant 2021YFD1600301-4, and in part by the Earmarked Fund for Modern Agro-Industry Technology Research System.

ABSTRACT The daylily (*Hemerocallis citrina* Baroni) is rich in not only nutrition ingredients but also functional components, and the edible part is the flower, not containing its pedicel. The primary challenge in developing a robotic daylily harvester is recognizing mature bud in the unstructured and uncertain environment. The objective of this study is to propose an accurate detection model. *Hemerocallis citrina* cv. 'DatongHuanghua' variety is used in this study. We initially adopt VGG16, VGG19, ResNet50, ResNet101 and ResNet152 as the backbones of Faster R-CNN respectively to build different detection models. The experimental results show that VGG19 and ResNet50 are two best-performing models in the corresponding VGGNet and ResNet, and the Average Precision (AP) of VGG19 is 90.18%, while ResNet50 is 88.35%. Based on these, we further integrate Convolutional Block Attention Module (CBAM) in Faster R-CNN with three different integration modes: plugging CBAM behind Conv5_x of VGG19 and ResNet50 respectively, as well as between every two "bottleneck" blocks of ResNet50. The comparison demonstrate plugging CBAM between every two blocks of ResNet50 is the best integration mode, and the corresponding detection model has a 2.22% highest increase in AP. Therefore, we empirically validate the performance of detection model for daylily mature bud based on Faster R-CNN integrated with CBAM.

INDEX TERMS Daylily, faster R-CNN, detection, attention mechanism, CBAM.

I. INTRODUCTION

The daylily (*Hemerocallis citrina* Baroni) is a perennial herb and belongs to *Liliaceae Hemerocallis* [1]. It is rich in not only nutrition ingredients such as carbohydrate, protein, vitamin and carotene [2], but also functional components such as polyphenols, polysaccharides and flavonoids, which have the activities of preventing cardiovascular disease, anti-hyperlipidemia, anti-diabetics, anti-tumor, anti-depression, anti-aging and improving immunity [3]. At present, the daylily is commercially produced in China, Malaysia, Japan, Indonesia and Madagascar, while China accounts for more than 98% of the overall daylily planting areas. The daylily harvesting is a labor-intensive operation. Taking the daylily in the Datong County, Shanxi Province (*Hemerocallis citrina* cv. 'DatongHuanghua') as an example, which has continuously won various awards at provincial, national and international agricultural products fairs, the daylily harvest season lasts from late June to early August, and during the peak season, the harvesting takes 10 to 12 hours every day and should be accomplished before 8:00 in the morning [4]. Considering the poor harvest conditions, the heavy labor intensity, the low production efficiency and the high labor cost, it is necessary to increase the mechanization in daylily harvest. In fact, our research team have developed 4HF-6 hanging and 4HF-2 crawling auxiliary daylily harvesters, which are shown in Figure1. Although the auxiliary daylily harvester can improve the working conditions and reduce labor intensity, it does not change the status quo of manual harvesting. Therefore, we aim to further develop a robotic daylily harvester.

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyu Zhou.

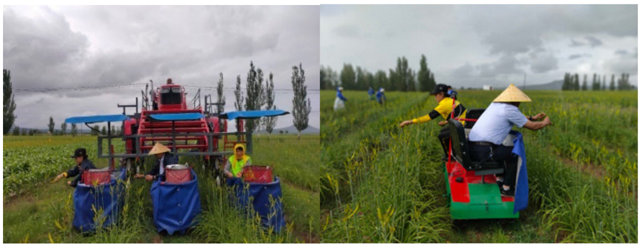

FIGURE 1. The 4HF-6 hanging (left) and 4HF-2 crawling (right) auxiliary daylily harvesters developed by our research team.

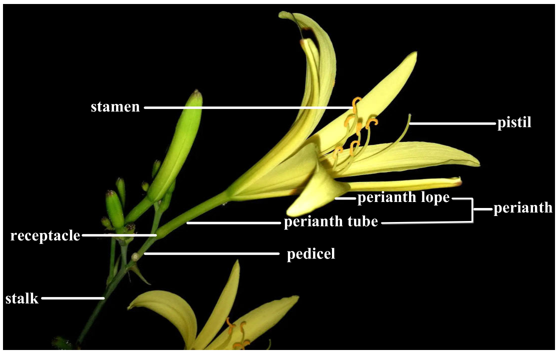

FIGURE 2. The flower structure of Hemerocallis citrina cv. 'DatongHuanghua.'

As is known, a typical fruit or vegetable harvesting robot contains vision system, harvesting manipulator, end-effector and motion system [5]. The vision system plays an important role in autonomous harvesting and makes mature fruits or vegetables recognization intelligently possible [6].

In the daylily harvest period, the buds are randomly located on the stalk, and the weather conditions continuously change due to clouds, sun direction and the wind that moves plants. This paper aims to address all these issues and achieve accurate and robust detection of daylily.

The daylily flower consists of gynoecium, androecium, perianth, receptacle and pedicel [7]. In the daylily family, different varieties have different forms in terms of color, shape and size. Figure 2 shows the flower structure of *Hemerocallis citrina* cv. 'DatongHuanghua'. The pedicel's length is 0.2cm-0.5cm. The receptacle is a small pad swelled at the pedicel's tip. The perianth is composed of one perianth tube and six perianth lopes. The length of perianth tube is 3cm-4cm, and perianth lope is 9cm-12cm. The gynoecium is one pistil, which consists of stigma, style and ovary. The androecium is composed of six stamens, which consists of filament and anther.

The edible part of daylily is the flower, not containing its pedicel. The time of daylily harvesting comes when its bud is full, firm and bursting, meanwhile, the three seams, which are formed by outer three perianth lopes surrounding inner three perianth lopes, are very conspicuous. These visible features are the bases of further detection, and we call the daylily flower ready to be detected or harvested as mature bud. If the harvesting were ahead of time, the bud you got would be without a full size, and of light weight and poor quality and color (greenish or greenish-yellow, not golden-yellow). However, if the harvesting time were delayed, the flower would open and the pollen grains in anthers would disseminate, and the daylily would be considered as flowered one and lose its commodity value. In Figure 3, the daylily in the red box is the flowered one and it is no need to harvest, while the daylily in the blue box has not developed into a full size so it should not be harvested. And, the daylily in the green box belongs to the mature bud so it is seen as the detection objective.

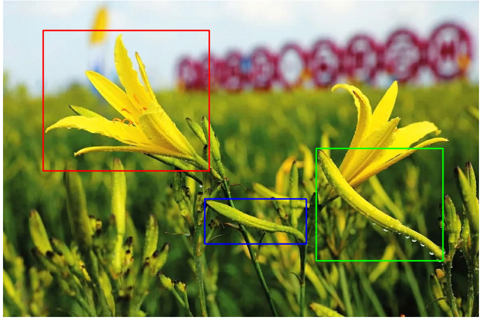

FIGURE 3. The daylily in different development for Hemerocallis citrina cv. 'DatongHuanghua.'

The problem of localizing mature bud with a bounding box and classifying it into a specific category as shown in Figure 3 belongs to object detection, which is the object level task. While there is a pixel level task addressing localization and classification simultaneously, which refers to instance segmentation [8]. Instance segmentation aims to yield further instance masks based on pixels. It may provide much more information about daylily than object detection, such as the contour of mature bud. However, instance segmentation has a high requirement for resolution as pixel level information relies on much more details, while object detection focuses more on object level features [9]. It is not easy to guarantee all the images in the daylily image dataset high resolution ones, let alone those captured during the process of harvesting by the robotic daylily harvester. Therefore, we applied object detection instead of instance segmentation to detect mature bud for daylily.

Object detection based on deep learning uses Convolutional Neural Network (CNN) instead of traditional detectors [10], and can be divided into CNN based on one-stage detectors and two-stage detectors. One-stage detectors can detect all objects in a one-step inference, such as YOLO (You Only Look Once) [11], SSD (Single Shot MultiBox Detector) [12], RetinaNet [13], CornerNet [14], Center-Net [15], DETR [16] and so on. While two-stage detectors follow a coarse-to-fine process and can easily attain a high precision, such as R-CNN (Region based CNN) [17], SPPNet (Spatial Pyramid Pooling Networks) [18], Fast R-CNN (Fast Region based CNN) [19], Faster R-CNN (Faster Region

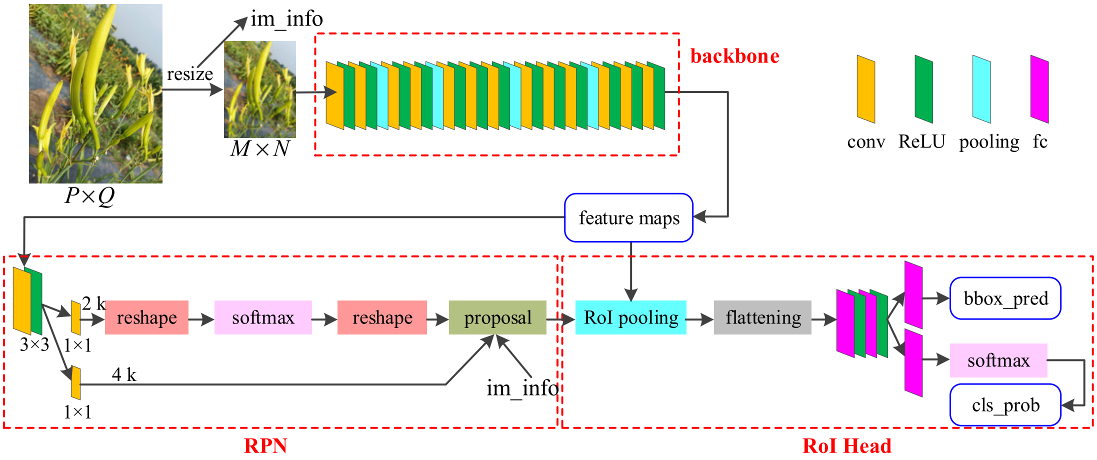

FIGURE 4. The architecture of Faster R-CNN. The backbone of Faster R-CNN is the modified VGG16, which is the typical VGG16 with its last max-pooling layer, fully-connected layers and softmax layer removed.

based CNN) [20], FPN (Feature Pyramid Networks) [21] and so on.

With the development of modern agriculture, boosting the performance of precision and efficiency remains a huge challenge for the fruit or vegetable harvesting robot. The precision of object detection is primary for the harvesting success rate. The harvesting efficiency depends on the detection and location speed of the vision system, the movement efficiency of the harvesting manipulator and the complexity of harvesting action for the end-effector [22], while the latter two are the key factors that restricts the efficiency of current harvesting robot. The speed of most object detection models based on CNN can match to the movement of harvesting manipulator and the action of the end-effector.

To achieve accurate and robust detection of mature bud for daylily in the unstructured and uncertain environment, the object detection method based on Faster R-CNN is applied, which is a two-stage detector and has a relatively higher accuracy, moreover, can meet need of the speed of harvesting manipulator and end-effector.

II. FASTER R-CNN WITH DIFFERENT BACKBONES

We initially use Faster R-CNN as the object detection model to detect mature bud for daylily.

A. FASTER R-CNN

The architecture of Faster R-CNN is as Figure 4, which can be divided into three parts: backbone, RPN and RoI (Region of Interest) Head.

The input image of arbitrary size is isotropically scaled to a fixed-size image, and then processed through sequential operations of features extraction, region proposals generation, features classification and bounding-box regression. In Figure 4, assuming that the size of the input image is $P \times Q$ and the isotropically-scaled image is $M \times N$, the re-scaling method is as follows:

$$\text{scale} = \begin{cases} \frac{\text{t\_size}}{\min(P,Q)} & \textit{if } \frac{\text{t\_size}\cdot\max(P,Q)}{\min(P,Q)} \leq \text{max\_size} \\ \frac{\text{max\_size}}{\max(P,Q)} & \textit{otherwise} \end{cases} \tag{1}$$

$$\begin{cases} M = P \cdot \text{scale} \\ N = Q \cdot \text{scale} \end{cases} \tag{2}$$

where t_size is the target minimum of height and width in the re-scaled image, max_size is the upper limit of the maximum for height and width in the re-scaled image, and scale is the scale factor, therefore im_info in Figure 4 is (M, N, scale).

The backbone extracts feature maps based on CNN, which includes a sequence of convolutions, ReLU and pooling operations. In Figure 4, the CNN adopts VGG16, with its last max-pooling layer, three fully-connected layers and the softmax layer removed, therefore, through 13 convolutional layers, 13 ReLU layers and 4 pooling layers, the extracted feature maps have a fixed-size $M/16 \times N/16$ in 512 channels.

In the RPN, there is an $n \times n$ convolutional layer followed by two sibling 1×1 convolutional layers: a box-classification layer for estimating each box is object or not-object, and a box-regression layer for outputting the coordinates of boxes. Finally, the RPN outputs a set of rectangular region proposals, each with 2 scores that estimate probability of object / not-object. Because the box-classification layer and the box-regression layer are both based on 1×1 convolution operation, it is equal to predict k region proposals and output 4 k coordinates of the k boxes at each sliding-window location, and the boxes are called anchors.

In the RoI Head, RoI pooling is applied independently into each RoI in the feature map so that the fixed-size (pooled_w, pooled_h, 512) feature vectors are produced, where pooled_w and pooled_h are usually set to 7 respectively. Following this, the 512-d feature vectors are flattened to 1-d vectors. Through fully connected layers, the network outputs two vectors per RoI: softmax probabilities and per-class bounding-box regression offsets [19].

B. BACKBONE ALGORITHM SELECTION

The function of backbone in the Faster R-CNN is extracting feature maps. In order to extract dominant visual features of the mature bud for daylily, backbone algorithm is selected through detection experiments.

We initially chose the typical CNNs: VGGNet [23] and ResNet [24] to be the backbone of Faster R-CNN. VGGNet won the first and the second places in the localization and classification tracks respectively on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2014, and ResNet secured the first place on the ILSVRC 2015 classification, localization and detection tasks. For VGGNet, VGG16 and VGG19 are the two best-performing models on the 1000-category ImageNet dataset [23]. For ResNet, ResNet50, ResNet101 and ResNet152 are more accurate than the rest [24]. Therefore, VGG16, VGG19, ResNet50, ResNet101 and ResNet152 are selected to be the backbone of Faster R-CNN respectively to build different detection models. However, these CNNs must be modified. For VGG16 and VGG19, the last max-pooling layer, fully-connected layers and softmax layer are removed. For ResNet50, ResNet101 and ResNet152, the average pooling layer, fully-connected layer and softmax layer are removed, moreover, the convolutional layer with stride 2 is replaced by the one with stride 1 in the Conv5_x. The modified CNNs are in Figure 5.

Because there are four max-pooling layer with stride 2 in VGG16 and VGG19, and there are three convolutional layers with stride 2 and one max-pooling layer with stride 2 in ResNet50, ResNet101 and ResNet152, these backbones are trained from input images, which are actually the re-scaled. (M, N).images in Figure 4, to output feature maps with a size $(M/16, N/16)$.

C. EXPERIMENTS AND RESULTS

1) DAYLILY IMAGE DATASET

We took 2248 images with multi-angle, multi-scale and multi-development daylily flowers under different weather conditions in the daylily harvest season of 2021 and 2022 at the standardized organic daylily planting base in Yunzhou District, Datong City, Shanxi Province and Hemerocallis resource garden of Shanxi Agricultural University. The images have different resolutions, such as 4608 × 2128 pixels, 3264 × 2448 pixels, 2592 × 1728 pixels, 2338 × 1080 pixels, 1280 × 720 pixels, 640 × 480 pixels. Furthermore, to avoid the overfitting problem, we performed data augmentation [25] through horizontal flip, vertical flip, rotation, brightness enhancement, brightness reduction, adding Poisson noise [26], blurring and sharpening operation, so the image data increased by 8 times, in other words, the daylily image dataset contained 20556 images. In addition, we used LableImg tool for data annotation to build a pre-labelled daylily image dataset. To validate daylily detection models, we scattered the images randomly and segmented them into training sets, verification sets and test sets with a ratio of 6:2:2.

TABLE 1. Hardware and software configurations.

Parameter	Value
Computer series model	AMAX ServMax™ PSC-HB2X
CPU	Intel Xeon E5-2680
GPU	1 NVIDIA Quadro K2200 and 2 NVIDIA Tesla K40c
OS	Windows7
Development language	Python3.5
Development environment	PyCharm
Deep learning architecture	TensorFlow1.4
Extension packages	OpenCV, NumPy, Matplotlib, and et al.

TABLE 2. Network training hyper-parameters.

Hyper-parameter	Value
optimizer	Nesterov Momentum
weight_decay	0.0005
learning_rate	0.001
momentum	0.9
max_iterations	40000
batch_size	512
rpn_negative_overlap	0.3
rpn_positive_overlap	0.7
roi_pooling_size	7

2) EXPERIMENTAL SETTINGS

In our study, the experimental hardware and software configurations are shown in Table 1, and the network training hyper-parameters [27] are shown in Table 2. In Table 2, rpn_negative_overlap and rpn_positive_overlap are the thresholds for training RPN to estimate that the anchor is positive, negative, or neither positive nor negative, and the estimation method is as follows:

$$\text{anchor} = \begin{cases} \text{positive} & \text{if IoU>rpn\_positive\_overlap} \\ \text{negative} & \text{if IoU<rpn\_negative\_overlap} \end{cases} \tag{3}$$

$$\text{IoU} = \frac{\text{anchor} \cap \text{ground truth}}{\text{anchor} \cup \text{ground truth}} \tag{4}$$

where ground truth is the labeled objective through data annotation [28].

3) EVALUATION CRITERIA

Object detection model can be evaluated by the criteria of Precision (P), Recall (R), Precision-Recall curve (P-R curve), Average Precision (AP) and so on, the formulas are as shown in Equations (5)-(7) respectively:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{AP} = \int_0^1 P(R)\text{d}R \approx \sum_{k=1}^{N} P(k)\Delta R(k) \tag{7}$$

where, TP is the amount of daylily mature bud regions which are actually true and properly classified, FP is the amount of image regions which are classified as daylily mature bud but actually not, FN is the amount of image regions which

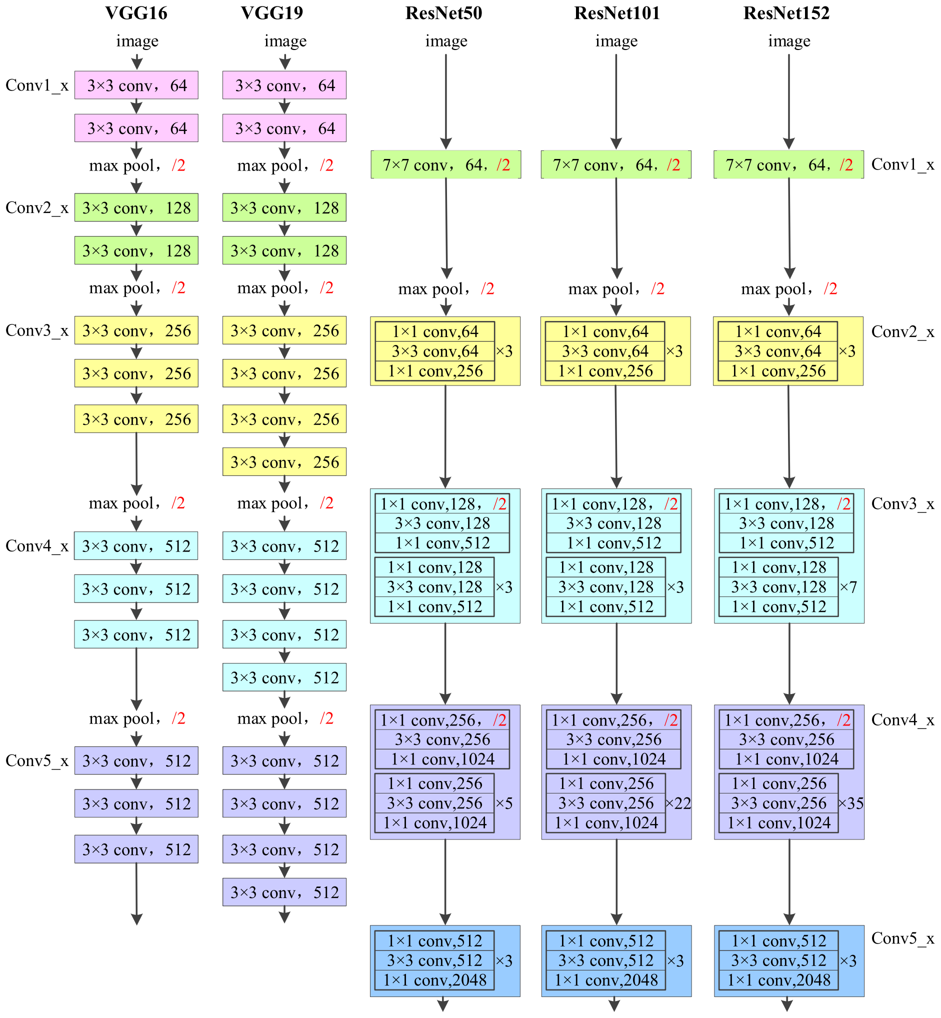

FIGURE 5. Different backbones of Faster R-CNN.

are actually true daylily mature bud but not be classified correctly. *P* indicates the proportion of accurate detection results in the total detection results. *R* indicates the proportion of properly detected mature bud regions in all the actual mature bud regions. AP is a comprehensive evaluation criterion about Precision and Recall, furthermore, AP is the area under P-R curve, which is drawn with *P* as the vertical axis and *R* as the horizontal axis.

We adopted AP as the primary criterion and *R* as the auxiliary to quantitatively evaluate detection model and made a qualitative analysis with P-R curve.

4) RESULTS AND DISCUSSION

The modified VGG16, VGG19, ResNet50, ResNet101 and ResNet152 are applied as backbones of Faster R-CNN respectively to build different detection models. The test results are observed and analyzed as shown in Table 3 and Figure 6. To describe briefly, Faster R-CNN with VGG16 as backbone will be shortened to VGG16 later, and the rest will be in the same manner. For VGGNet, VGG19 outperforms VGG16 on both AP and *R*; with Recall changing in the P-R curves, Precision of VGG19 is almost always better than VGG16. While for ResNet, ResNet50, ResNet101 and

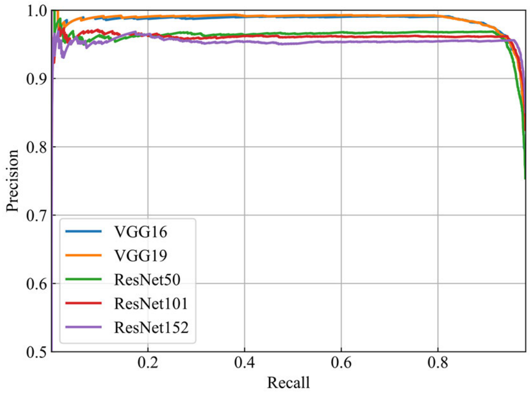

FIGURE 6. P-R curves in experiments for Faster R-CNN with different backbones.

TABLE 3. Detection results in experiments for Faster R-CNN with different backbones.

backbone	AP (%)	R (%)	Parameters	Rate (s/im)
VGG16	90.07	97.80	27.32M	0.472
VGG19	90.18	97.85	28.38M	0.532
ResNet50	88.35	98.08	5.63M	0.541
ResNet101	87.93	98.13	9.43M	0.617
ResNet152	87.51	98.15	12.56M	0.668

ResNet152 have an increasing order of R, but a decreasing order of AP, and a decreasing order of Precision when Recall changes from about 20% to 95% in the P-R curves. In term of primary evaluation criterion AP, VGG19 and ResNet50 are optimal and selected to be integrated with dual attention mechanisms afterwards.

To thoroughly show the performance of different detection models, both parameters and rate are listed in Table 3.

On the basis of Faster R-CNN with different backbones, the visualization results are shown in Figure 7.

III. FASTER R-CNN INTEGRATED WITH CBAM

In the computer vision, attention mechanism imitates human attention property to selectively focus on salient parts in the image and ignore those irrelative. There are four kinds of attention mechanisms: channel attention, spatial attention, temporal attention and branch attention [29]. Channel attention is applied to select the channels of CNN which can efficiently extract primary features of the image (e.g., SENet [30]). Spatial attention focuses on the selection of spatial region in the image which is helpful to extract import regions of the image (e.g., STN [31]). Temporal attention is used for dynamic temporal selection in the video processing (e.g., GLTR [32]). Branch attention is utilized for branch selection in the multi-branch structure (e.g., SKN [33]). In our study, channel attention and spatial attention will be integrated with Faster R-CNN to enhance the detection accuracy of mature bud for daylily.

A. CBAM

Convolutional Block Attention Module (CBAM) is a simple yet effective attention module for feed-forward CNNs, which is a channel and spatial attention hybrid proposed by Woo et al. in 2018 [34]. Its architecture is shown in Figure 8.

There are two sequential sub-modules: channel and spatial. After channel attention module, the channel-refined feature map is extracted; further after spatial attention module, the final refined feature map is achieved. Given the input feature map $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$, max-pooling and average-pooling operations are simultaneously conducted for aggregating spatial information, therefore two different spatial context descriptors are generated: $\boldsymbol{F}^{\boldsymbol{c}}_{\text{max}}$ and $\boldsymbol{F}^{\boldsymbol{c}}_{\text{avg}}$. Both descriptors are then forwarded to a shared multi-layer perceptron (MLP). Following this, element-wise summation is applied to merge the output feature vectors. Through the sigmoid function, 1D channel attention map $\boldsymbol{M_c}$ $(\boldsymbol{F})$ $(\boldsymbol{M_c}(\boldsymbol{F}) \in \mathbb{R}^{C\times 1\times 1})$ is achieved, which can be computed as:

$$\begin{aligned}\boldsymbol{M_c}(\boldsymbol{F}) &= \sigma(\text{MLP}(\text{max\_pool}(\boldsymbol{F})) + \text{MLP}(\text{avg\_pool}(\boldsymbol{F})))\\ &= \sigma(\text{MLP}(\boldsymbol{F}^{\boldsymbol{c}}_{\text{max}}) + \text{MLP}(\boldsymbol{F}^{\boldsymbol{c}}_{\text{avg}})) \qquad (8)\end{aligned}$$

where σ denotes the sigmoid function. We calculate the channel-refined feature map $\boldsymbol{F}'$ using element-wise multiplication of $\boldsymbol{F}$ and $\boldsymbol{M_c}$ $(\boldsymbol{F})$, which are as follows:

$$\boldsymbol{F}' = \boldsymbol{F} \otimes \boldsymbol{M_c}(\boldsymbol{F}) \qquad (9)$$

The channel-refined feature map $\boldsymbol{F}'$ is the input of spatial attention module. Max-pooling and average-pooling operations are sequentially applied along the channel axis to focus on informative regions, therefore two 2D feature maps are generated: $\boldsymbol{F}^{\boldsymbol{s}}_{\text{max}}$ and $\boldsymbol{F}^{\boldsymbol{s}}_{\text{avg}}$. They are then concatenated to input convolutional layer, whose filter size is 7 × 7. Through the sigmoid function, 2D spatial attention map $\boldsymbol{M_s}$ $(\boldsymbol{F})$ $(\boldsymbol{M_s}(\boldsymbol{F}) \in \mathbb{R}^{1\times H\times W})$ is generated, which can be calculated as:

$$\begin{aligned}\boldsymbol{M_s}(\boldsymbol{F}) &= \sigma(f^{7\times 7}([\text{max\_pool}(\boldsymbol{F}); \text{avg\_pool}(\boldsymbol{F})]))\\ &= \sigma(f^{7\times 7}([\boldsymbol{F}^{\boldsymbol{s}}_{\text{max}}; \boldsymbol{F}^{\boldsymbol{s}}_{\text{avg}}])) \qquad (10)\end{aligned}$$

$\boldsymbol{M_s}$ $(\boldsymbol{F})$ is element-wise multiplied with the channel-refined feature map $\boldsymbol{F}'$, so the final refined feature map $\boldsymbol{F}''$ is as follows:

$$\boldsymbol{F}'' = \boldsymbol{F}' \otimes \boldsymbol{M_s}(\boldsymbol{F}') \qquad (11)$$

Because CBAM is a lightweight and general attention module, it can be seamlessly integrated into Faster R-CNN for pushing more accurate results based on the preceding detection models.

B. INTEGRATION MODE

The previous chapter has concluded that Faster R-CNN with VGG19 and ResNet50 as the backbone respectively are two best-performing models in the corresponding VGGNet and ResNet, and they will be integrated with CBAM to boost the accuracy of their base networks in this chaper. However, integration modes may affect the overall performance.

Woo et al. [34] integrated CBAM with ResBlocks in ResNet50 for ImageNet-1K classification dataset and they plugged CBAM between every two ResBlocks. Xu and Ma [35] designed three CBAM integration modes in the crack detection model for asphalt pavement based on Faster R-CNN with ResNet50 as the backbone: plugging CBAM behind Conv1_x of ResNet50 (as depicted in Figure 4), behind Conv5_x, and behind both Conv1_x and Conv5_x, and the experiments showed that plugging CBAM behind Conv5_x of ResNet50 performed best. She et al. [36] proposed CBAM Faster R-CNN with VGG16 as the backbone to detect esophageal cancer in the barium meal angiography, and tested seven CBAM integration modes: plugging CBAM behind Conv3_x of VGG16, behind Conv4_x, behind Conv5_x, behind Conv3_x and Conv4_x, behind Conv3_x and Conv5_x, behind Conv4_x and Conv5_x, and behind

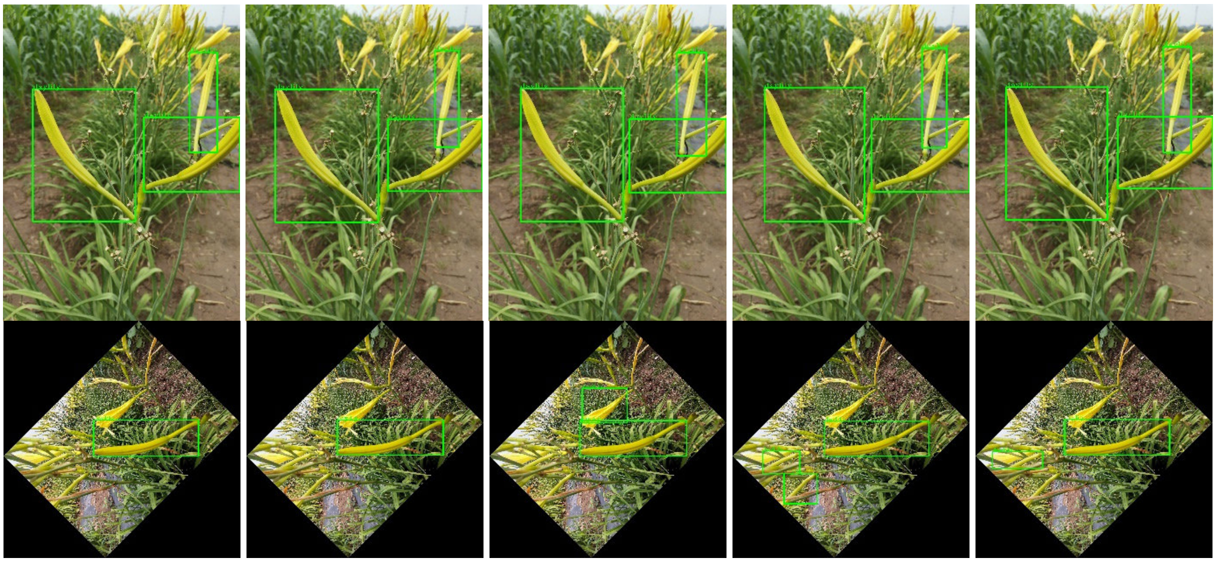

FIGURE 7. Detection results based on Faster R-CNN with different backbones. The five columns are sequentially based on VGG16, VGG19, ResNet50, ResNet101 and ResNet152 from left to right. For the input image in the first row, there are three mature buds in fact, and all the detection models with different backbones detect correctly. For the input image in the second row, there are only one mature bud actually, the models of VGG16 and VGG19 detect correctly, however, the model of ResNet50 detects an extra flowered daylily, the model of ResNet101 detects two extra perianth tubes, and the model of ResNet152 detects one extra perianth tube.

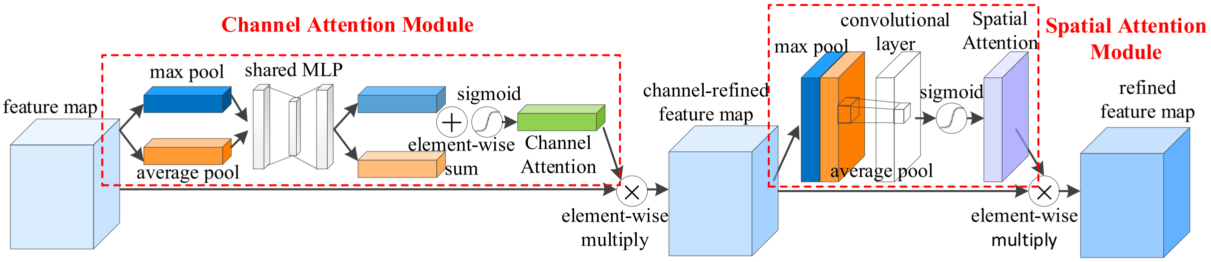

FIGURE 8. The architecture of CBAM. The Channel Attention Module and Spatial Attention Module are in order.

Conv3_x, Conv4_x and Conv5_x, and eventually the experiments concluded that plugging CBAM behind Conv5_x of VGG16 performed best.

Different integration modes of CBAM have different effects on the feature extraction by CNN in the channel dimension and spatial dimension. Therefore, we compare three integration modes for the two detection models: plugging CBAM behind Conv5_x of VGG19, behind Conv5_x of ResNet50, and between every two "bottleneck" blocks of ResNet50. The daylily image dataset, experimental settings and evaluation criteria remain the same with the previous chapter.

C. EXPERIMENTS AND RESULTS

The VGG19 plugged CBAM behind Conv5_x, ResNet50 plugged CBAM behind Conv5_x and ResNet50 plugged CBAM between every two "bottleneck" blocks are applied as backbones of Faster R-CNN respectively to build different detection models, which are respectively shortened to VGG19-CBAM5, ResNet50-CBAM5 and ResNet50-CBAM-blocks. The test results are summarized as shown in Table 4, in which the statistics of VGG19 and ResNet50 in Table3 are listed to compare the performance between detection models integrated with CBAM and the baselines.

TABLE 4. Comparison of different CBAM integrated modes.

backbone	AP (%)	R (%)	Parameters	Rate (s/im)
VGG19	90.18	97.85	28.38M	0.532
VGG19-CBAM5	90.43	97.85	28.43M	0.547
ResNet50	88.35	98.08	5.63M	0.541
ResNet50-CBAM5	88.69	97.80	5.84M	0.549
ResNet50-CBAM-blocks	90.57	97.04	9.66M	0.595

In Table 4, we observe that the detection models with CBAM outperform all the baselines on AP, although their R s are not improved. Compared with VGG19, VGG19-CBAM5 has an equal R and a 0.25% increase in AP. In addition, Figure 9 depicts their P-R curves, where when Recall changes from 10% to 97.87%, VGG19-CBAM5 has a higher Precision than VGG19. In term of ResNet50, ResNet50-CBAM5 has a 0.28% decrease in R and a 0.34% increase in AP, while

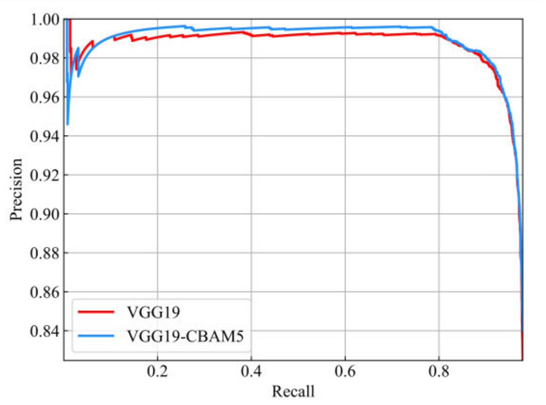

FIGURE 9. P-R curves in experiments for VGG19 integrated with CBAM or not in Faster R-CNN.

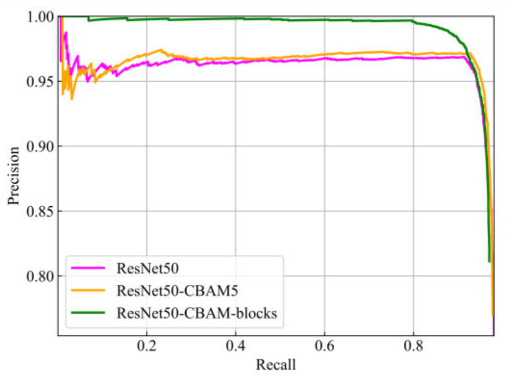

FIGURE 10. P-R curves in experiments for ResNet50 integrated with CBAM or not in Faster R-CNN.

ResNet50-CBAM-blocks has a 1.04% decrease in *R* and a 2.22% increase in AP. As depicted in Figure 10, with Recall changing in the P-R curves, ResNet50-CBAM-blocks has almost always the best Precision.

Given the qualitative analysis of different CBAM integration modes, we present the P-R curves of VGG19-CBAM5, ResNet50-CBAM5 and ResNet50-CBAM-blocks in Figure 11. It's clear that ResNet50-CBAM-blocks outperform than the other two on Precision during most of Recall varying period. What is more important, the AP of ResNet50-CBAM-blocks is the best. Therefore, plugging CBAM between every two "bottleneck" blocks of ResNet50 is the best integration mode.

Based on VGG19-CBAM5, ResNet50-CBAM5 and ResNet50-blocks, the visualization results are shown in Figure 12.

We consequently propose the model of detection mature bud for daylily based on Faster R-CNN integrated with CBAM, whose backbone is ResNet50 plugged CBAM between every two "bottleneck" blocks. On the basis of detection model, the gathered images and results of daylily are shown in Figure 13. In addition, the average detection time of our model is 0.595s per image, which is not particularly fast, but can match sufficiently the speed of harvesting manipulator and end-effector.

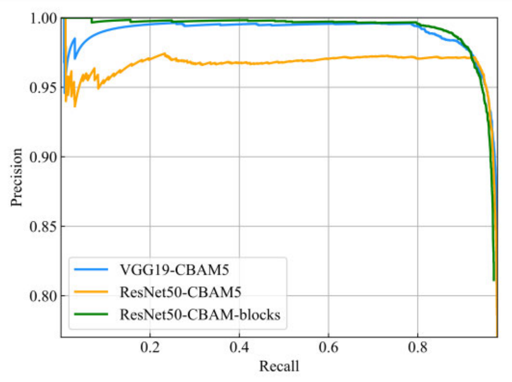

FIGURE 11. P-R curves in experiments for different integration modes of CBAM in Faster R-CNN.

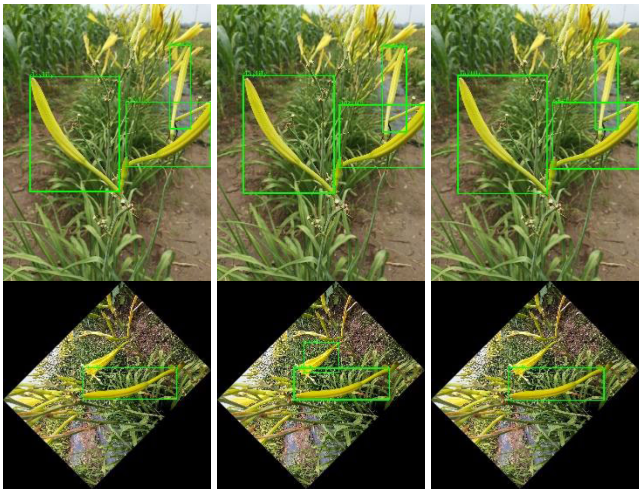

FIGURE 12. Detection results based on different CBAM integrated modes. The three columns are sequentially based on VGG19-CBAM5, ResNet50-CBAM5, and ResNet50-CBAM-blocks from left to right. The input images are the same with Figure 7. For the first row, all the detection results are correct. In the second row, the models of VGG19-CBAM5 and ResNet50-CBAM-blocks detect correctly, however, the model of ResNet50-CBAM5 detects an extra flowered daylily, which is the same error in the model of ResNet50 in the second row of Figure 7.

TABLE 5. Comparison of different object detection methods.

architecture	backbone	AP (%)	Parameters	Rate (s/im)
Faster R-CNN	ResNet50-CBAM-blocks	90.57	9.66M	0.595
YOLOv3	Darknet-53	80.29	62.12M	0.020
YOLOv4	ResNet50	87.34	57.61M	0.024
YOLOv4	CSPDarknet-53	88.78	58.33M	0.023

We further perform experiments based on YOLOv3 and YOLOv4 to compare with our method. Using Average Precision as the evaluation criterion, the experimental results are summarized in Table 5. We can clearly see that the AP of our methods is obviously higher than YOLOv3 with Darknet-53 as backbone, YOLOv4 with ResNet50 as backbone and YOLOv4 with CSPDarknet-53 as backbone.

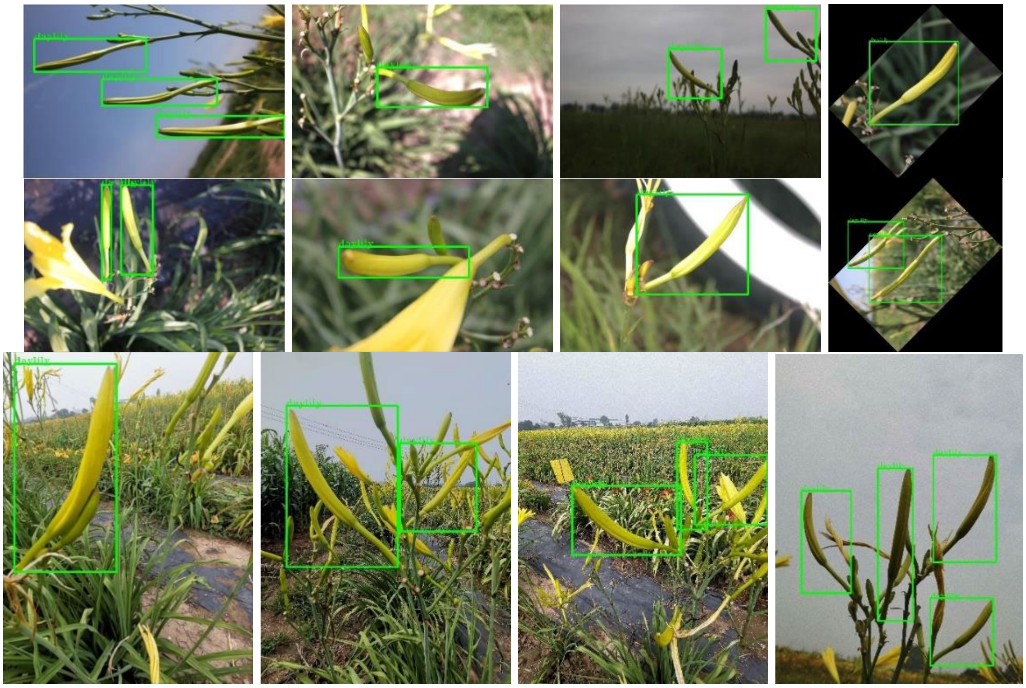

FIGURE 13. Detection results. We select some daylily images taken under different weather conditions, with different shooting angles and different resolutions, and some images generated through data augmentation.

IV. CONCLUSION

To achieve excellent accuracy of mature bud detection for daylily in the unstructured field environment, VGG16, VGG19, ResNet50, ResNet101 and ResNet152 with high recognition accuracy in the VGGNet and ResNet are modified to be the backbones of Faster R-CNN respectively to build different detection models, and the results show that VGG19 and ResNet50 are two best-performing models in the corresponding VGGNet and ResNet. Furthermore, we integrated CBAM with VGG19 and ResNet50 in three modes: plugging CBAM behind Conv5_x of VGG19, behind Conv5_x of ResNet50, and between every two ‘‘bottleneck’’ blocks of ResNet50, and empirically verify that the detection models with CBAM outperform all the baselines on AP, and plugging CBAM between every two blocks of ResNet50 is the best integration mode.

REFERENCES

[1] C. Ma, ‘‘The studies on reproductive biology of Hemerocallis lilio-asphodelus Linn,’’ M.S. thesis, Dept. Landscape Archit., Northeast Forestry Univ., Harbin, Heilongjiang, China, Jun. 2014.

[2] J. Mao, ‘‘A review on nutritional value and process technology of *Hemerocallis citrina* Baroni,’’ *J. Anhui Agri. Sci.*, vol. 36, no. 3, pp. 1197–1198, Aug. 2008, doi: 10.3969/j.issn.0517-6611.2008.03.147.

[3] X. Qin, L. Zhang, Y. Wen, J. Li, and Y. Zhang, ‘‘Advances in research on nutritional activities of important functional components of *Hemerocallis citrina* Baroni,’’ *Food Res. Develop.*, vol. 43, no. 5, pp. 204–209, Mar. 2022, doi: 10.12161/j.issn.1005-6521.2022.05.030.

[4] D. Jiao, ‘‘Cultivation and harvesting processing technology of *Hemerocallis citrina* cv. ‘DatongHuanghua,’’’ *Agri. Technol. Equip.*, no. 16, pp. 39–42, Aug. 2013, doi: 10.3969/j.issn.1673-887X.2013.08.019.

[5] Y. Edan, S. F. Han, and N. Kondo, ‘‘Automation in agriculture,’’ in *Springer Handbook of Automation*. Berlin, Germany: Springer, 2009, pp. 1095–1128, doi: 10.1007/978-3-540-78831-7_63.

[6] Y. Zhao, L. Gong, Y. Huang, and C. Liu, ‘‘A review of key techniques of vision-based control for harvesting robot,’’ *Comput. Electron. Agricult.*, vol. 127, pp. 311–323, Sep. 2016, doi: 10.1016/j.compag.2016.06.022.

[7] E. J. Bidlack and H. J. Shelley, ‘‘Stern’s introductory plant biology,’’ in *Agricola*, vol. 3, M. Lange, Ed., 12th ed. New York, NY, USA: McGraw-Hill, 2011, pp. 128–129.

[8] A. M. Hafiz and G. M. Bhat, ‘‘A survey on instance segmentation: State of the art,’’ *Int. J. Multimedia Inf. Retr.*, vol. 9, no. 3, pp. 171–189, Jul. 2020, doi: 10.1007/s13735-020-00195-x.

[9] S. Wang, Y. Gong, J. Xing, L. Huang, C. Huang, and W. Hu, ‘‘RDSNet: A new deep architecture for reciprocal object detection and instance segmentation,’’ in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12208–12215, doi: 10.1609/aaai.v34i07.6902.

[10] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, ‘‘Object detection in 20 years: A survey,’’ 2019, *arXiv:1905.05055*.

[11] R. Joseph, S. Divvala, R. Girshick, and A. Farhadi, ‘‘You only look once: Unified, real-time object detection,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ‘‘SSD: Single shot MultiBox detector,’’ 2015, *arXiv:1512.02325*.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ‘‘Focal loss for dense object detection,’’ 2017, *arXiv:1708.02002*.

[14] H. Law and J. Deng, ‘‘CornerNet: Detecting objects as paired keypoints,’’ 2018, *arXiv:1808.01244*.

[15] X. Zhou, D. Wang, and P. Krähenbühl, ‘‘Objects as points,’’ 2019, *arXiv:1904.07850*.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, ‘‘End-to-end object detection with transformers,’’ 2020, *arXiv:2005.12872*.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘‘Region-based convolutional networks for accurate object detection and segmentation,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

[18] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Spatial pyramid pooling in deep convolutional networks for visual recognition,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[19] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[22] Z. Wang, Y. Xun, Y. Wang, and Q. Yang, "Review of smart robots for fruit and vegetable picking in agriculture," *Int. J. Agricult. Biol. Eng.*, vol. 15, no. 1, pp. 33–54, 2022, doi: 10.25165/j.ijabe.20221501.7232.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[25] S. Yao, "Regularizing deep neural networks by ensemble-based low-level sample-variances method," M.S. thesis, Dept. Comput. Sci., Tianjin Univ., Tianjin, China, Nov. 2019.

[26] C.-A. Deledalle, F. Tupin, and L. Denis, "Poisson NL means: Unsupervised non local means for Poisson noise," in *Proc. IEEE Int. Conf. Image Process.*, Hong Kong, Sep. 2010, pp. 801–804, doi: 10.1109/ICIP.2010.5653394.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[28] B. Pande, K. Padamwar, S. Bhattacharya, S. Roshan, and M. Bhamare, "A review of image annotation tools for object detection," in *Proc. Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC)*, Salem, India, May 2022, pp. 976–982, doi: 10.1109/ICAAIC53929.2022.9792665.

[29] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," 2021, *arXiv:2111.07624*.

[30] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.

[31] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2015, *arXiv:1506.02025*.

[32] J. Li, S. Zhang, J. Wang, W. Gao, and Q. Tian, "Global-local temporal representations for video person Re-Identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 3957–3966, doi: 10.1109/ICCV.2019.00406.

[33] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," 2019, *arXiv:1903.06586*.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," 2018, *arXiv:1807.06521*.

[35] K. Xu and R. Ma, "Crack detection of asphalt pavement based on improved Faster-RCNN," *Comput. Syst. Appl.*, vol. 31, no. 7, pp. 341–348, Mar. 2022, doi: 10.15888/j.cnki.csa.008594.

[36] Y. She, J. Gao, X. Min, S. Xu, A. Pan, and W. Lan, "Detection of esophageal cancer based on CBAM R-CNN," *J. South-Central Univ. Nat., Natural Sci. Educ.)*, vol. 40, no. 6, pp. 631–638, Dec. 2021, doi: 10.12130/znmdzk.20210613.

JUNHUI FENG was born in Qingxu, Shanxi, China, in 1988. She received the Ph.D. degree in agricultural engineering from Shanxi Agricultural University, Taigu, Shanxi, in 2023. Her current research interests include agricultural information detection and control.

XUERONG ZHAO was born in Jinzhong, Shanxi, China, in 2000. She is currently pursuing the M.S. degree in agricultural engineering with the College of Agricultural Engineering, Shanxi Agricultural University, Taigu, Shanxi. Her current research interest includes image processing.

TINGYU ZHU was born in Linfen, Shanxi, China, in 1999. She is currently pursuing the M.S. degree in agricultural engineering with the College of Agricultural Engineering, Shanxi Agricultural University, Taigu, Shanxi. Her current research interest includes intelligent agricultural equipment.

TAO LI was born in Changzhi, Shanxi, China, in 1997. He is currently pursuing the M.S. degree in agricultural engineering with the College of Agricultural Engineering, Shanxi Agricultural University, Taigu, Shanxi. His current research interests include agricultural information monitoring and control technology.

ZHICHAO QIU was born in Chaoyang, Liaoning, China, in 1998. She is currently pursuing the M.S. degree in agricultural engineering with the College of Agricultural Engineering, Shanxi Agricultural University, Taigu, Shanxi, China. Her current research interests include agricultural information detection and control technology.

ZHIWEI LI was born in Taigu, Shanxi, China, in 1969. He received the Ph.D. degree in agricultural resources and utilization from Nanjing Agricultural University, Nanjing, Jiangsu, China, in 2005. He is currently a Professor with the College of Agricultural Engineering, Shanxi Agricultural University. His research interests include precision agriculture and intelligent agricultural equipment.