An Apricot Detection Algorithm in Complex Environments Based on Improved YOLOv7

Qiang Guo, Chi Ma*, and Hui Hu

***Abstract*—Apricot detection is a prerequisite for counting and harvesting tasks. Existing algorithms face challenges in adapting to the impacts of complex environmental factors such as lighting variations, shadows, dense foliage, and the uneven distribution of samples in mechanized apricot harvesting. This paper proposes an enhanced model, YOLOv7-DC, based on YOLOv7, to address these challenges. YOLOv7-DC preprocesses diverse apricot tree samples to accommodate real-world harvesting detection scenarios. To improve model inference speed and detection accuracy, the detection network is redesigned with a new feature fusion method. DCNv2 is embedded within the efficient layer aggregation network (ELAN), and PConv is introduced to replace conventional convolutions, reducing the parameter impact of DCNv2. The training process incorporates the CBAM attention mechanism to enhance spatial and channel information. The ConvMixer architecture captures spatial and channel relationships transmitted to the detection head through the attention mechanism, improving the model's detection accuracy for each specific classification sample. Experimental results show that YOLOv7-DC maintains good detection speed and recognition rates across various classification tasks. The improved model achieves a 6.2% increase in average detection accuracy compared to previous algorithms, with a 13% reduction in model parameters. YOLOv7-DC is better suited for handling imbalanced samples and complex environmental scenarios.**

***Index Terms*—Apricot biloba detection, YOLOv7, Attention mechanism, Feature fusion.**

I. Introduction

APRICOT harvesting recognition has significant application value in the agricultural sector. With the continuous research and development of its efficacy in industries such as healthcare and cosmetics, the demand for apricot is steadily increasing. Currently, apricot harvesting is primarily done manually. Apricot trees have a short growth cycle, and the work environment is challenging, making them susceptible to weather conditions. To improve the efficiency of apricot tree harvesting and reduce labor costs, mechanized harvesting has become an inevitable trend. The primary task of mechanized processing is to apply detection and recognition to apricot trees, with the application of computer vision technology laying the foundation for subsequent counting and harvesting. In the research on mechanized apricot harvesting, many scholars have conducted preliminary explorations. In the field of computer vision, Kumari [1] and others utilized image processing techniques [2] for early automatic detection of leaves to achieve detection and classification of apricot harvesting objects. Triki [3] and others proposed a deep learning-based method called Deep Leaf, using an improved state-of-the-art instance segmentation method (Mask R-CNN [4]) to detect and pixel-segment apricot trees. Deep Leaf can accurately detect each fruit in plant samples and measure relevant morphological features.

Additionally, many studies have applied YOLO series algorithms to apricot detection. The initial version of YOLO introduced the concept of transforming the object detection problem into a regression problem. Subsequent versions introduced features such as Anchor Boxes, Darknet-19 network structure, and multi-scale training to support multi-category detection. YOLOv3 made further improvements by adopting a deeper Darknet-53 network structure and a strategy of detecting at different scales. YOLOv4 introduced numerous technological innovations, including the CIOU loss function, PANet, SAM, CSPDarknet53, etc., enhancing detection performance and speed. YOLOv5, developed by Ultralytics, is easier to use and train. However, our research found that the earlier versions of YOLO still face performance bottlenecks in apricot detection due to the limitations of their detection range and are not directly applicable to apricot detection. Khan [5] and others proposed a new technique to improve the accuracy of detecting apricot and branch structures. This method first obtains the basic branch structure and then applies image processing algorithms to improve the accuracy of the predicted branch structure before detecting apricot trees [6]. There are also studies that utilize the Hungarian (Munkres) algorithm to match branch images of apricot trees from each day with those from the previous and following days. This is done to identify the optimal matches between fruit in a plant image and the corresponding fruit in the same plant from another day. Ideally, this matching process groups identical fruits together, allowing the exclusion of fruits detected from erroneous data. However, these methods have some shortcomings in recognition tasks in complex environments. Harvesting recognition in complex environments requires large-scale and diverse training datasets. The datasets used in the mentioned studies for harvesting recognition may be relatively small or lack a sufficient number of complex samples, limiting the model's generalization ability in real complex scenarios. Additionally, in complex environments, target objects may be occluded, posing a challenge to recognition performance. Most of the previous research has not adequately addressed occlusion scenarios, resulting in decreased model performance in occluded scenes. In order to overcome the aforementioned challenges, this study employs YOLOv7 as the base detection model for apricot tree harvesting recognition tasks in complex environments. However, during the research process, it was observed that

Manuscript received March 29, 2024; revised October 31, 2024. This paper is supported by Foundation on Guangdong Educational Committee under the Grant No. 2022ZDZX4052, 2021ZDJS082.

Qiang Guo is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing. AnShan 114051, China (e-mail: 254755139@qq.com).

Chi Ma is an associate professor of the School of Computer Science and Engineering, Huizhou University, Huizhou 516007, China (corresponding author to provide phone: 18641250800; e-mail:machi@hzu.edu.cn)

Hui Hu is a lecturer of the School of Computer Science and Engineering, Huizhou University, Huizhou 516007, China (e-mail:machi@hzu.edu.cn).

using YOLOv7 alone might not fully exploit crucial feature information. To enhance the perceptual ability towards key features, the CBAM attention mechanism is introduced. The CBAM attention mechanism [7] is a method designed to enhance the perceptual ability of convolutional neural networks by adaptively adjusting the weights of feature maps, allowing the network to focus more on features contributing to the classification task. In this paper, we integrate the CBAM attention mechanism into the network structure of YOLOv7 to improve apricot tree harvesting recognition performance in complex environments. Additionally, we employ the strategy of attention feature fusion [8]. This technique involves weighting different regions of the image, focusing more attention on key areas, thereby enhancing the accuracy and robustness of apricot recognition. The attention feature fusion strategy aims to improve the discriminative ability for apricot by placing more emphasis on critical regions.

II. Algorithm Description

The YOLO algorithm, as a representative of one-stage object detection algorithms, is based on deep neural networks for object recognition and localization [9]. It operates at high speed, making it suitable for real-time systems. YOLOv7 is currently the most advanced algorithm in the YOLO series, surpassing its predecessors in terms of accuracy and speed [10]. YOLOv7 introduces the concept of model reparameterization into the network architecture, which was originally seen in REPVGG. The label assignment strategy utilizes YOLOv5's cross-grid search and YOLO's matching strategy. A new network architecture is proposed in YOLOv7, focusing on efficiency. YOLOv7 introduces a training method for auxiliary heads with the main purpose of increasing accuracy by adding training costs without affecting inference time. The auxiliary head is only present during the training process. YOLO is analyzed in detail, breaking it down into seven modules: CBS module, CBM module, REP module, MP module, ELAN module, Upsample module, and SPPCSPC module.

Regarding the CBS module, as seen in Figure 1, it is composed of a Conv layer (Convolutional layer), a BN layer (Batch Normalization layer), and a Silu layer (Activation function) [11]. The Silu activation function is a variant of the swish activation function, and their formulas are as follows:

$$silu(x) = x \cdot sigmoid(x) \tag{1}$$

$$swish(x) = x \cdot sigmoid(\beta x) \tag{2}$$

CBM module and CBS module are fundamentally similar. The REP module is divided into two parts: a training module (train) and an inference module (deploy) [12]. The training module has three branches: the top branch is a 3x3 convolution for feature extraction, the middle branch is a 1x1 convolution for smoothing features, and the final branch is an Identity, which does not perform convolution but is directly moved over [13]. Finally, these branches are added together. In the inference module, it includes a 3x3 convolution with a stride of 1, which is transformed from the reparameterization of the training module. The MP module has two branches for downsampling. The first branch undergoes a max-pooling operation for downsampling, followed by a 1x1 convolution to change the number of channels. The second branch first undergoes a 1x1 convolution to change the number of channels, then passes through a 3x3 convolution kernel with a stride of 2, which is also used for downsampling. Finally, the results of the first and second branches are added together, resulting in a super downsampling outcome.

ELAN module is an efficient network structure, as shown in Figure 2. It controls the shortest and longest gradient paths, allowing the network to learn more features and exhibit stronger robustness. Regarding the ELAN-W module, it is very similar to the ELAN module, with a slight difference in the selected output quantity in the second branch. The UPSample module is an upsampling module that uses nearest-neighbor interpolation. SPP aims to increase the receptive field, enabling the algorithm to adapt to different resolution images. It achieves this by obtaining different receptive fields through max-pooling [14]. The CSP module first divides the features into two parts. One part undergoes conventional processing, while the other part undergoes SPP structure processing. Finally, these two parts are merged together, reducing half of the computational load, resulting in increased speed and improved accuracy.

III. Improvement strategies

Apricot images collected in the wild may introduce interference due to complex backgrounds, impacting the detection performance of the YOLOv7 model based on convolutional neural networks. The attention mechanism, derived from studies on human vision, concentrates attention on important areas of an image while discarding irrelevant regions in computer vision [15]. This enables the neural network to focus on a subset of features. Therefore, the improvement strategy in this paper integrates the CBAM attention mechanism into the YOLOv7 network, directing the model's attention more towards the apricot itself rather than the background environment. The CBAM module combines channel attention mechanism and spatial attention mechanism, enhancing the feature representation and target localization accuracy of the YOLOv7 model by applying attention weighting to both the channel and spatial dimensions of the feature map. The ConvMixer architecture is employed in the detection head to enhance the performance of small target detection. The ConvMixer in the detection head helps capture spatial and channel relationships conveyed through the attention mechanism to improve the model's effectiveness. To address irregular apricot sizes and the difficulty in extracting shape features in complex environments leading to subsequent detection errors, a feature fusion module is introduced in the backbone network by embedding DCNv2 in the efficient layer-aggregation network ELAN. This enhances the capability to extract apricot shape features. Simultaneously, to mitigate the decrease in detection speed caused by adding the feature fusion module, PConv is used to replace the conventional convolution in the backbone network, effectively reducing the model's computational load and improving detection speed. The overall architecture of the improved model is depicted in Figure 3.

A. CBAM module

To address challenges encountered during mechanical harvesting of apricot, including variations in lighting conditions,

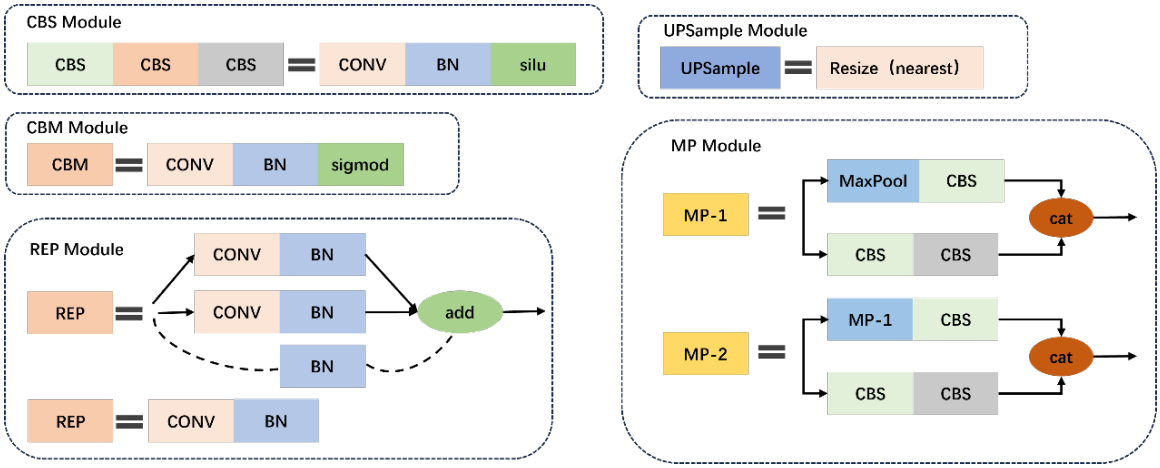

Fig. 1: Detailed Introduction to Partial Modules of YOLOv7

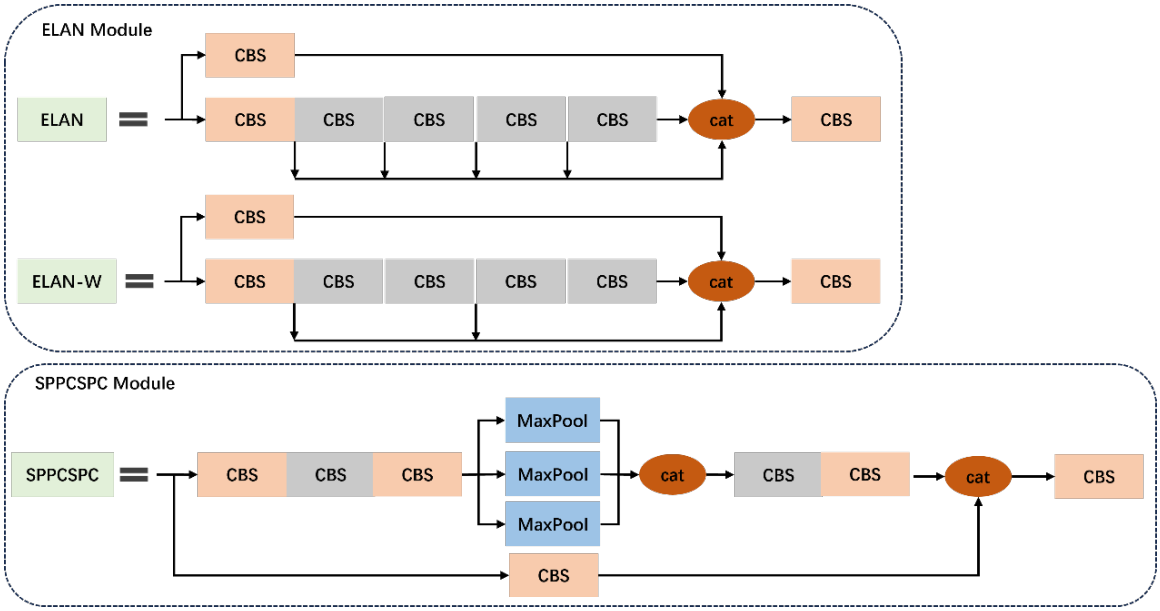

Fig. 2: Detailed introduction of partial modules of YOLOv7

the presence of shadows, and the uneven distribution of samples within dense foliage, the improved YOLOv7 algorithm with enhanced attention mechanisms incorporates three CBAM modules in the backbone network. This addition aims to enhance the network's feature extraction capabilities. CBAM (Convolutional Block Attention Module) is an attention mechanism module designed to boost the receptive field and channel attention of convolutional neural networks. The CBAM module consists of two sub-modules: the Spatial Attention Module (SAM) and the Channel Attention Module (CAM).

The SAM structure is illustrated in Figure 4, aiming to enhance the model's perception of important regions by learning the spatial correlations in image space [16]. It first applies average pooling and max pooling to the features and concatenates the resulting feature maps. Subsequently, a convolution operation is employed on the concatenated feature map to generate the final spatial attention feature map. Finally, the learned spatial attention weights are applied to the original feature map to obtain a feature representation enhanced with spatial attention.

The CAM structure is shown in Figure 5, aiming to enhance the model's perception of important features by learning the inter-channel correlations. It initially performs max pooling and average pooling operations in the channel dimension to obtain the max-pooled feature map and average-pooled feature. In our network architecture, the feature map processed through the convolutional network is first input into the channel module. The Channel Attention Module utilizes the relationships between the channel attention of features to generate channel attention information, primarily employed to determine the focus points in the input image. The calculation formula is as follows:

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= (W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \end{aligned} \tag{3}$$

σ represents the sigmoid function, W0 and W1 denote the shared weights of the MLP (Multi-Layer Perceptron), AvgPool and MaxPool signify average and max pooling operations, respectively. During the computation, to integrate the average-pooled feature F and max-pooled feature FE, they are forwarded to an information-sharing layer network composed of multiple perceptron (MLP) layers and hidden

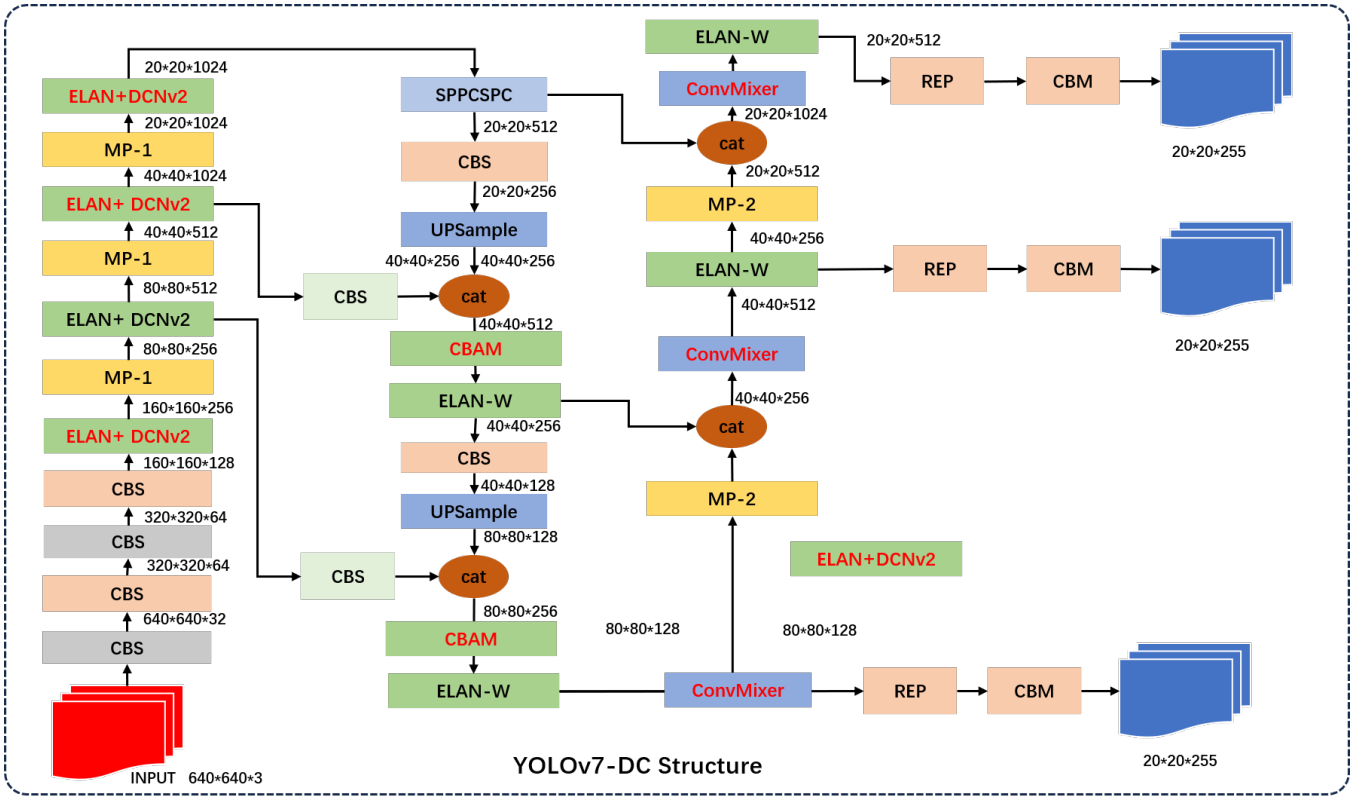

Fig. 3: YOLOv7-DC Structure

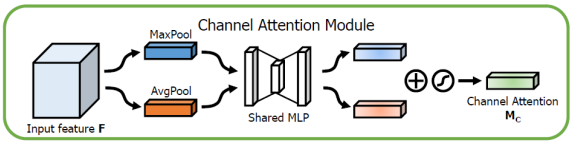

Fig. 4: SAM structure

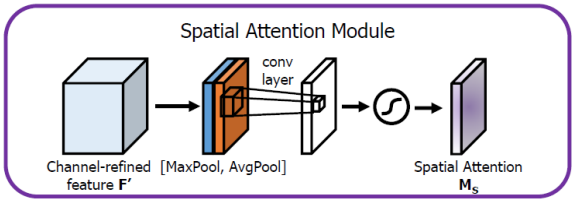

Fig. 5: CAM structure

layers for information integration. This process generates a channel attention map, and after processing each channel feature descriptor through the shared network, the feature vectors are combined using the SUM operation to obtain the output. The feature map processed by the Channel Attention Module then enters the Spatial Attention Module. The Spatial Attention Module utilizes the spatial relationships of feature points to generate a spatial attention map, primarily focusing on the attention points determined by the Channel Attention Module. The calculation formula for complementary information and channel attention is as follows:

$$M_S(F) = \sigma(f^{7\times7}[AvgPool(F); MaxPool(F)]) = \sigma(f^{7\times7}[F^x_{avg}; F^x_{max}]) \quad (4)$$

Among them, 7 denotes the convolution operation with a filter size of 7x7. In the process of spatial computation, the module in the spatial attention map operates by first performing pooling along the channel axis to generate two pooled features. These two features are then concatenated to produce an effective feature descriptor. Subsequently, a standard convolution operation is applied to generate the spatial attention map. Through a series of multi-layer perceptrons to learn channel attention weights, capturing global correlations between channels to obtain a feature representation enhanced by channel attention. By combining SAM and CAM together, the CBAM module can simultaneously consider inter-channel correlations and image spatial correlations. The application of the CBAM module in YOLOv7 enables the model to better capture detailed features of targets, while improving the detection accuracy of targets in complex backgrounds and occlusion situations, further enhancing the detection performance and application effectiveness of YOLOv7. For the basic convolution block in YOLOv7, the CBAM module is added after it. This way, each basic convolution block undergoes processing by SAM and CAM to obtain weights for spatial information and inter-channel correlations. This attention mechanism allows the network to adaptively focus on important areas and features in the image, suppressing interference from noise and redundant features. In the improved YOLOv7 model in this paper, the detection head uses the ConvMixer architecture to enhance the performance of small target detection. The ConvMixer in the prediction head helps capture spatial and channel relationships passed to the prediction head features, which are discovered through deep convolution and pointwise convolution in ConvMixer. The pointwise convolution in ConvMixer also enhances the detection capability of the prediction head for small objects since it processes information at the individual data point level. Additionally, unlike traditional convolutional neural

networks, ConvMixer maintains the integrity of the input structure throughout the entire mixer layer, making it well-suited for the prediction head of the apricot detection architecture. The architecture of the ConvMixer layer is shown in Figure 6. The ConvMixer module itself consists of deep convolution (specifically grouped convolution, where the number of groups equals the number of channels) followed by pointwise convolution (1x1 convolution). After each convolution, there is the GELU activation function and activated BatchNorm (batch normalization). The expressions for the content of Figure 6 can be formulated as follows:

$$z_l' = BN(\sigma\{ConvDepthwise(z_{l-1}))\}) + z_{l-1} \quad (5)$$

$$z_{l-1} = BN(\sigma\{ConvDepthwise(z_l'))\}) \quad (6)$$

B. Feature fusion module

The Efficient Layer Aggregation Network (ELAN) in YOLOv7 is an efficient network architecture that uses a residual structure to aggregate feature maps obtained from each intermediate layer for feature fusion in the final layer. It can extract feature maps obtained from different convolutions and features from different channels. Simultaneously, the DCNv2 (Deformable Convolutional Networks version 2) is embedded into the ELAN network to form the DCNv2+ELAN module. In the DCNv2+ELAN module, continuous use of DCNv2 convolution allows feature extraction at different scales and receptive fields, merging them to capture details and structural information at different levels, thereby enhancing the accuracy and robustness of feature extraction. The structure of the DCNv2+ELAN module is illustrated in Figure 7. The LeakyReLU activation function is used instead of the original ReLU function, modifying all negative weights to non-zero slopes, expanding the range of the ReLU function, and reducing the issue of apricot misdetection caused by environmental factors.

The DCNv2+ELAN module enhances the feature extraction capability of the backbone network but introduces increased computational load and memory access during detection, leading to an increase in model detection time. As apricot detection is a real-time process with high speed requirements, aiming to deploy the model on resource-constrained mobile operation devices, PConv (Point-wise Convolution) is used instead of conventional convolution. This substitution reduces the model's computational load while improving the hardware device's floating-point operations per second, achieving lower detection latency. Figure 8 provides a detailed explanation of the conventional convolution and DCNv2 convolution processes, described by formulas 3-5.

$$L = \frac{f}{F} \quad (7)$$

$$C = h \times w \times k^2 \times c^2 \quad (8)$$

$$M = h \times w \times 2c \times k^2 \times c^2 \quad (9)$$

In equation (7), it illustrates the relationship between detection latency (L), model computational load (f), and hardware floating-point operations per second (F). Equations (8) and (9) respectively calculate the computational load (C) and the memory access count (M) during the convolution process, where h and w represent the height and width of the input feature map. c is the number of channels in the convolution, and k is the size of the convolution kernel. As indicated by equation (7), detection speed is related to both the model's computational load and the device's floating-point operations per second. The former leads to an extension of model computation time, and the latter frequent memory access can cause a decline in hardware processing performance. Therefore, larger model computational load and frequent memory access can result in increased latency. Considering that the number of channels (c) in PConv is 1/4 of conventional convolution, as per equation (8), the computational load of PConv is only 1/16 of conventional convolution, significantly reducing the model's computational load. As indicated by equation (9), the memory access count of PConv is approximately 1/4 of conventional convolution, leading to an improvement in hardware performance by around four times. Thus, PConv reduces its computational load and memory access count, optimizing both model and hardware performance, reducing latency, and enhancing the model's detection speed.

IV. Experiment and Analysis

A. Experimental Setup

This paper uses the acfr-fruit dataset to meet the diversity requirements for apricot tree recognition in complex environments and the authenticity of planting scenes. The dataset is captured at different time periods and angles, collecting apricot images with varying levels of density, lighting conditions, and occlusion. It comprises 620 apricot tree images, with 935 fully annotated instances. In terms of both quantity and quality, it holds a significant advantage among similar datasets. Examples of samples from the dataset are illustrated in Figure 9.

In this experiment, the acfr-fruit dataset is divided into training and validation sets. The split is based on an 8:2 ratio, considering the clarity of sample targets and the quality of data. The training set consists of 520 samples, and the validation set consists of 100 samples, totaling 620 samples with 1 class of target objects. The dataset is processed according to the YOLOv7 annotation format. The experimental environment configuration includes the Windows 10 operating system, an RTX 3060 graphics card, a 4-core CPU, 32GB of memory, a 1000GB system disk, CUDA version 2.0.1, and Python language environment version 3.10. The total number of iterations for this experiment is 300, with a batch size set to 24. The dimensions for each detection head are 256, 512, and 1024. Evaluation metrics include commonly used metrics in object detection tasks: Precision (P), Recall (R), mean Average Precision at IoU 50 (mAP50), and the model's parameter count. Among these, P represents the average accuracy of the model's detection results, R represents the model's recall rate, mAP50 indicates the mean Average Precision at IoU 50, and a higher mAP50 value corresponds to better overall model performance on detection data. A lower parameter count results in a smaller model size. To comprehensively verify the effectiveness of each module in the experimental operation, multiple ablation experiments, analysis, and comparative experiments are conducted. The experimental results are compared with and without the addition of the CBAM fusion attention mechanism and feature fusion module.

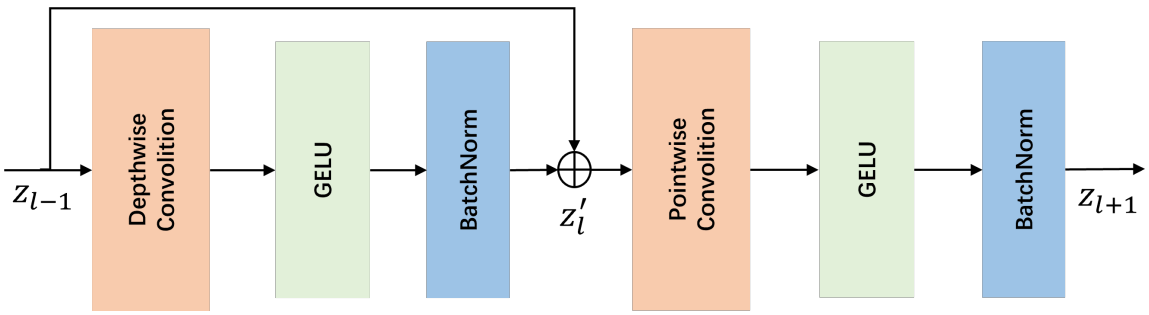

Fig. 6: ConvMixer structure

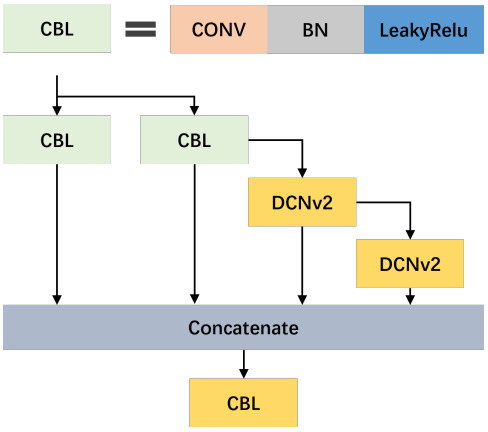

Fig. 7: Details of DCNv2 + ELAN module

Fig. 8: Convolution process of conventional convolution (a) and DCNv2(b)

Fig. 9: An example of the dataset.

B. Ablation Experiments

In the ablation experiments, two modules are selected for investigation to better study the impact of different modules on the performance of the YOLOv7 model. Ablation experiments are conducted on the CBAM, ConvMixer, DCNv2, and PConv modules. CBMA and ConvMixer are collectively referred to as attention mechanism modules, while DCNv2 and PConv are combined as feature fusion modules for separate testing. The ablation experiments, as illustrated, involve comparing neural network models utilizing different modules. Recall and precision are used as evaluation criteria, effectively demonstrating the effectiveness of the YOLOv7 model.

In Table 1, it can be observed that the improved model achieves an accuracy of 0.755 and a recall of 0.712. It's worth noting that, with the addition of the PConv module, the model's accuracy has slightly decreased. This is attributed to the primary purpose of the PConv module, which is to reduce the model's parameter count. We believe that the precision loss here is only 0.006, but it contributes to a better balance between speed and accuracy. Therefore, we choose to retain this module for subsequent experiments.

C. Comparative Experiments

In the comparative experimental evaluation of the model, we selected YOLOv3, FastRCNN, YOLOv5, SSD, and YOLOv7 for comparative experiments, evaluating precision, recall, and mAP. The number of labels is 935, and the selected image size is 640*640. The experimental results are shown in Table 2.

In Table 2, analyzing the mAP values, the best-performing model is YOLOv7-DC proposed in this paper, with a mAP value of 0.758, which is a 6.2% improvement compared to the original YOLOv7 model. The worst-performing model is FastRCNN, with a mAP value of only 0.523. This is because FastRCNN, as a classic two-stage model, is not sensitive to the occlusion situations and the imbalance of samples in the apricot dataset. In the evaluation of recall and precision, YOLOv7-DC still maintains good performance compared to other algorithms, especially in the evaluation of precision, showing a 7.6% improvement over the original model. However, there is not a significant increase in recall. Nevertheless, this still demonstrates the effectiveness of the model. We present detailed data comparison curves in Figure 6, showing precision, recall, F1, and mAP, which better illustrate the difference between the improved model and the original model.

We present detailed data from the experimental process in Figure 10, illustrating the variations of real data and test data during the training process. Additionally, we provide a detailed display of the experimental results obtained at each iteration.

Finally, the model, trained based on the evaluation of its parameter count and inference speed. The size of the model's parameters and fps values are used as evaluation metrics. Experimental results prove the effectiveness of retaining PConv, further validating our assumption of balancing speed and accuracy. The trained weights also justify the previously mentioned slight decrease in accuracy by 0.006. In Table

TABLE I: Comparison of Ablation Study

	YOLOv7	Attention mechanism module		Feature fusion module		accuracy	Recall
		CBAM	ConvMixer	DCNv2	PConv		
YOLOv7-1	✓					0.679	0.708
YOLOv7-2	✓	✓				0.718	0.674
YOLOv7-3	✓	✓	✓			0.725	0.650
YOLOv7-4	✓			✓		0.702	0.715
YOLOv7-5	✓			✓	✓	0.696	0.711
YOLOv7-6	✓	✓	✓	✓		0.742	0.698
YOLOv7-7(ours)	✓	✓	✓	✓	✓	0.755	0.712

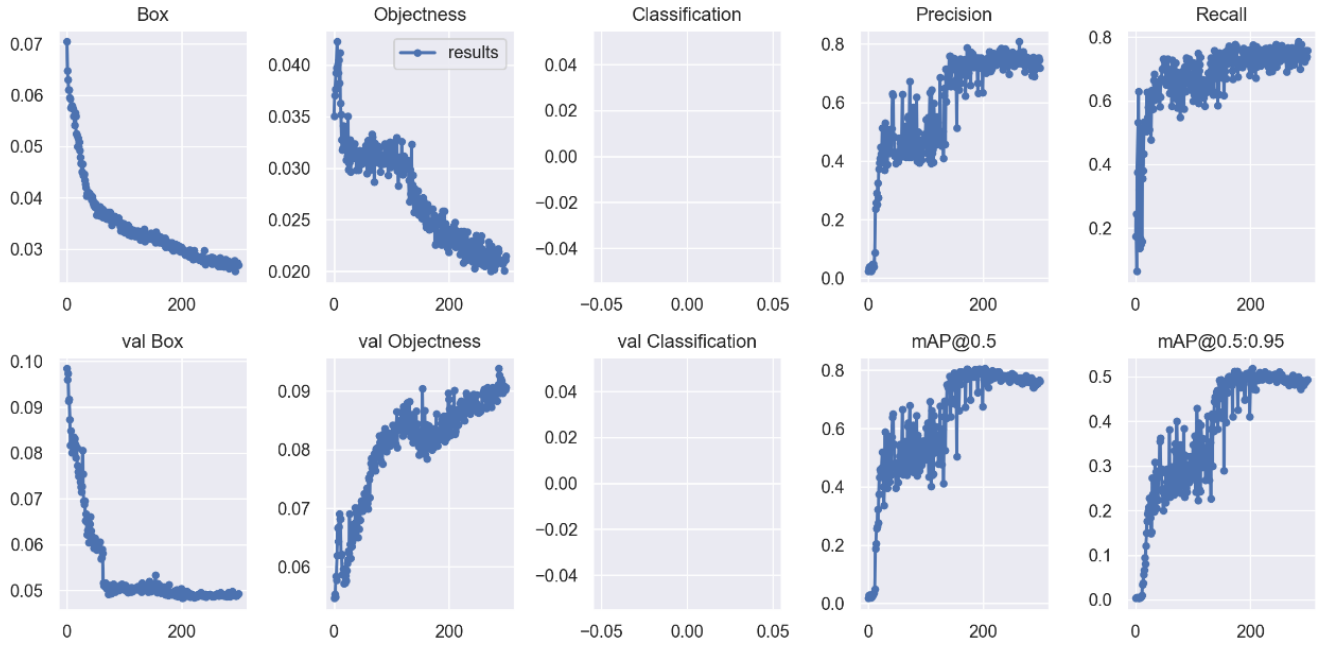

Fig. 10: Demonstration of the training process

TABLE II: Comparison of Ablation Study

	Labels	Size	P	R	Map@.5
YOLOv3	935	640*640	0.502	0.631	0.578
Fast RCNN	935	640*640	0.513	0.511	0.523
YOLOv4	935	640*640	0.532	0.521	0.536
YOLOv5	935	640*640	0.602	0.614	0.623
SSD	935	640*640	0.521	0.645	0.592
YOLOv7	935	640*640	0.679	0.708	0.696
YOLOv7-DC(ours)	935	640*640	0.755	0.712	0.758

TABLE III: Model parameters and inference speed

	Parameters/kb	FPS
YOLOv3	120519	39
YOLOv4	182421	65
YOLOv5	104106	47
YOLOv7	115231	79
YOLOv7-DC£¨ours£©	100251	93

4, it can be observed that YOLOv7-DC performs the best, demonstrating significant advantages in both parameter count and FPS.

D. Visual Result Analysis

The confusion matrix is an essential tool for evaluating the performance of object detection models. It helps the model better focus on important features during training and prediction. The figure 12 displays the parameters recognized by the improved YOLOv7 model and the original model in the confusion matrix.

It can be observed that the results obtained by the YOLOv7-DC model are 0.72, 1.00, 0.28, while the original model's results are only 0.70, 0.00, 0.26. The improved YOLOv7 model may perform better in apricot detection. In this experiment, we showcase the visualized results, indicating that YOLOv7-DC can more accurately identify apricot harvesting targets in complex environments. The figure below demonstrates the recognition performance in situations with complex backgrounds and dense tree branches.

In Figure 11, the detection results of 9 images in complex backgrounds are showcased, and the model continues to maintain good detection performance. Particularly in the fifth image, the model exhibits a low rate of false negatives. However, in the 8th image, the apricot in the bottom left corner is not detected, likely due to multiple factors such as lighting and color. The best performance is observed in the first image, achieving nearly 100% detection accuracy. Finally, we show the recognition effect of one set of images. This is

Fig. 11: Detection effect display

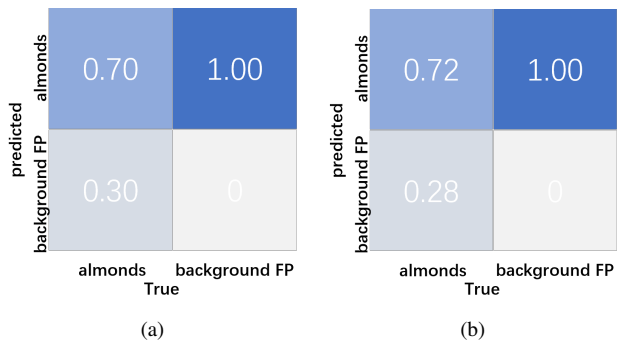

Fig. 12: Comparison with confusion matrix. (a) YOLOv7. (b) YOLOv7-DC.

shown in Figure 13. In order to show that YOLOv7-DC can more accurately identify apricot picking targets in complex environments, Figure 14 shows the recognition effects of different algorithms in the case of complex backgrounds and dense branches. Four images are detected each time. In Figure 14, the first column shows the detection effect of YOLOv3, the second column shows the detection effect of YOLOv5, the third column shows the detection effect of YOLOv7, and the last column shows the detection effect of YOLOv7-DC. In Figure 14, the detection effects of different algorithms on four images with complex backgrounds are shown in total, and the YOLOv7-DC model still maintains a good detection effect. Especially in the third image, we can see that the miss rate of the model is not high. In the fourth image, the apricot in the lower left corner is not detected, which is caused by multiple factors such as illumination and color. The best performance is the second image, which achieves almost 100% detection. Compared with the detection results of the other three models, the performance of the model can be seen more intuitively. In summary, YOLOv7-DC maintains excellent performance across various evaluations, especially when facing complex scenarios, demonstrating robust detection capabilities with the improved algorithm.

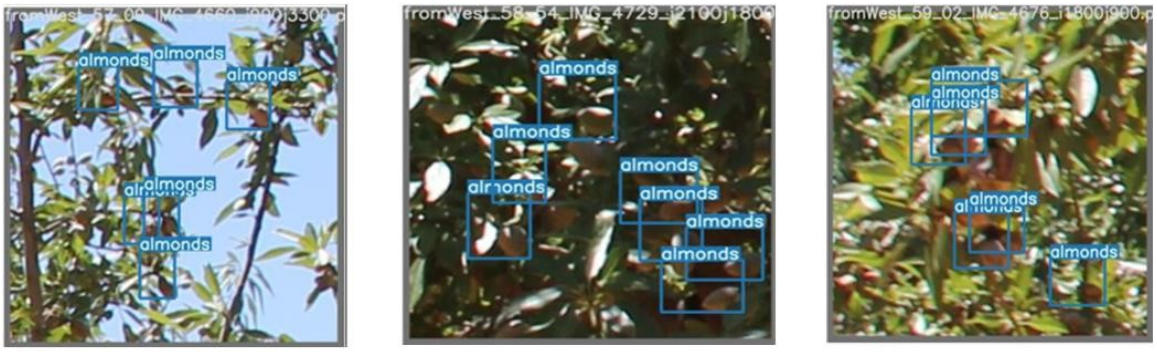

Fig. 13: Comparison of complex background detection effects

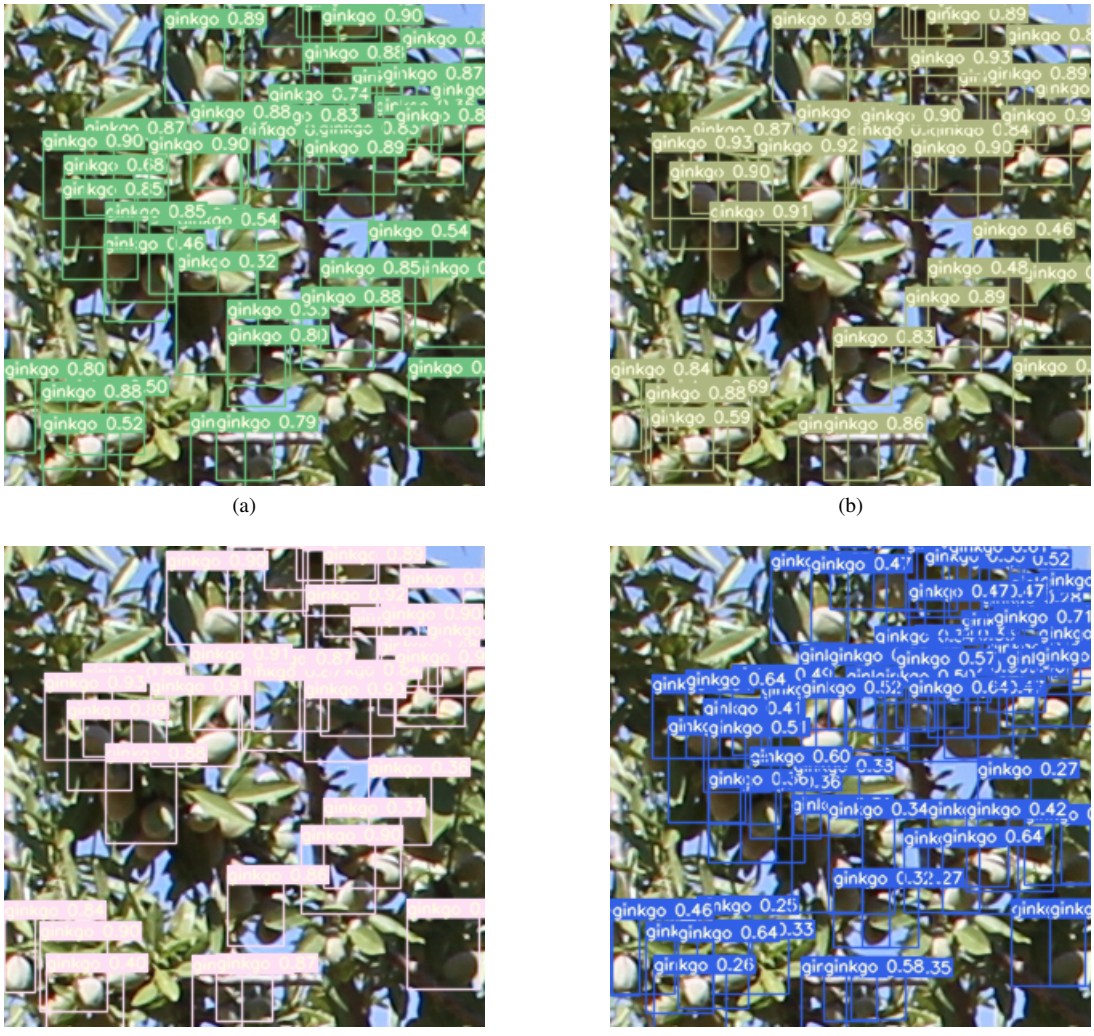

Fig. 14: Detection effect display with others. (a) YOLOv3. (b) YOLOv5. (c) YOLOv7. (d) YOLOv7-DC.

V. Conclusion

This paper introduces an enhanced model based on YOLOv7, which improves detection performance by integrating the CBAM attention mechanism and feature fusion module. The improved model, after training, demonstrates good detection accuracy and speed on various apricot tree samples. It not only enhances the average detection accuracy for each sample category but also significantly improves the average accuracy for imbalanced sample categories. Apricot tree detection and recognition hold important application potential in agriculture and health care. However, challenges posed by complex environmental factors such as lighting variations, shadows, dense foliage, and uneven sample distribution hinder the desired performance in apricot detection and recognition, impacting subsequent processing tasks. Previous research has not adequately addressed these factors, resulting in unstable model performance in real-world scenarios. YOLOv7-DC provides an effective solution, serving as a reference for future research. In the practical harvesting of apricot, detection and recognition are just the

initial steps. Subsequent research will primarily focus on aspects such as counting and analyzing apricot maturity.

REFERENCES

[1] Kumari C U, Vignesh N A, Panigrahy A K, et al. Fungal disease in cotton leaf detection and classification using neural networks and support vector machine, *Entropy*, vol.8, no.10, pp. 3-0200073, 2019.

[2] Chitradevi B, Srimathi P. An overview on image processing techniques, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 6466-6472, 2014.

[3] Triki A, Bouaziz B, Gaikwad J, et al. Deep leaf: Mask R-CNN based leaf detection and segmentation from digitized herbarium specimen images, *Pattern Recognition Letters*, vol. 150, pp. 76-83, 2021.

[4] He K, Gkioxari G, et al. Mask R-CNN, *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2961-2969, 2017.

[5] Khan N A, Lyon O A S, Eramian M, et al. A novel technique combining image processing, plant development properties, and the Hungarian algorithm, to improve leaf detection in Maize, *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 74-75, 2020.

[6] Chen L. Topological structure in visual perception, *Science*, vol. 218, no. 4573, pp. 699-700, 1982.

[7] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module, *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3-19, 2018.

[8] Dai Y, Gieseke F, Oehmcke S, et al. Attentional feature fusion, *Proceedings of the IEEE/CVF winter Conference on Applications of Computer Vision*, pp. 3560-3569, 2021.

[9] Girshick R, et al. Fast R-CNN, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448, 2015.

[10] Yongzhong Fu, Liang Qiu, Xiao Kong, et al. Deep Learning Based Online Surface Defect Detection Method for Door Trim Panel, *Engineering Letters*, vol. 32, no. 5, pp. 939-948, 2024.

[11] Wu M, Yue H, Wang J, et al. Object detection based on RGC Mask R-CNN, *IET Image Processing*, vol. 14, no. 8, pp. 1502-1508, 2020.

[12] Kang Tan, Linna Li, and Qiongdan Huang, Image Manipulation Detection Using the Attention Mechanism and Faster R-CNN, *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1261-1268, 2023.

[13] Shao Yanhua, Zhang Duo, Chu Hongyu, et al. A review of YOLO target detection based on deep learning, *Journal of Electronics and Information*, vol. 44, no. 10, pp. 3697-3708, 2022.

[14] Tan L, Huangfu T, Wu L, et al. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification, *BMC medical informatics and decision making*, vol. 21, no. 324, pp. 1-11, 2021.

[15] Y. Y. Ding, and L. Wang, Research on the Application of Improved Attention Mechanism in Image Classification and Object Detection, *IAENG International Journal of Computer Science*, vol. 50, no.4, pp. 1174-1182, 2023.

[16] Xiao Y, Yin H, Wang S H, et al. TReC: Transferred ResNet and CBAM for detecting brain diseases, *Frontiers in Neuroinformatics*, vol. 15, no. 71, pp. 781551, 2021.