**Frontiers in Computing and Intelligent Systems**
ISSN: 2832-6024 | Vol. 5, No. 2, 2023

# Aphid Detection Network from Global to Local

**Hui Zhang [1, 2, *], Xiaoping Yang [1]**

[1] Zhejiang Yuexiu University, Shaoxing Zhejiang, 312000, China
[2] School of Science, Zhejiang University of Science and Technology, Hangzhou Zhejiang, 310023, China
[*] **Corresponding author**: Hui Zhang (Email: 1520939382@qq.com)

**Abstract:** Common aphids on crops are not suitable for general-purpose object detection frameworks due to their small size and the presence of occlusion. Taking this as a starting point, we fully consider the characteristics of aphid targets and propose a network called Overall-Specific Net (OS-Net) for detecting aphid targets in a manner that goes from overall to specific in a single-stage detection network. This network consists of two detection head modules. The first detection head module is responsible for detecting aphids in densely populated areas, while the second module, based on the first module, detects aphids in densely populated areas by deploying denser anchor boxes. Experimental results on our dataset show that the average accuracy can be improved by approximately 5.1% compared to the baseline network.

**Keywords:** Deep Learning; Convolutional Neural Networks; Pest Detection.

## 1. Introduction

Agriculture is the foundation of the national economy, and crop production is crucial for the livelihood of the people. As of the end of 2022, without considering stockpiles, the per capita grain ownership of China's 1.4 billion population reached 475 kilograms, while the internationally recognized food security line is 400 kilograms per capita. The average grain ownership for the global population of 8 billion is 351 kilograms, all of which reflects the significant achievements in the development of China's agricultural industry. Especially in main grains such as rice, wheat, and corn, production and demand are basically matched. Although the total grain production increased by 0.5% in 2022 compared to 2021, the average yield per acre of grain decreased by 1%. This is mainly due to factors such as weather and pests. However, the decrease in production due to weather reasons is uncontrollable, while the prevention of pests can be planned. Pest infestations occur throughout the various stages of crop growth, resulting in incalculable grain losses. Therefore, how to more accurately monitor pest targets and apply pesticides rationally is crucial to protecting crop growth.

Before the widespread application of computer vision, crop pest monitoring mainly relied on field observations by farmers and agricultural professionals. However, this monitoring method has two drawbacks: on the one hand, the observation results are subjective and limited, requiring a high level of knowledge from relevant personnel; on the other hand, this process is relatively labor-intensive, with poor working conditions, leading to a reduction in the number of professionals, making it difficult to meet the needs of large-scale pest monitoring in the country. Therefore, designing a more reasonable and efficient pest monitoring scheme to replace manual monitoring is of great research and practical value.

Smart agriculture is an advanced solution that uses modern information technology to control the production process of agriculture, and one of its main applications is the automated identification of pest targets using artificial intelligence methods. Specifically, the automated identification of pests utilizes convolutional neural networks in deep learning to extract information from image data, then iteratively searches for the position of the target in the images on computer devices and determines the target category. To advance smart agriculture, many scholars have developed high-performance target detection solutions for different pests, some of which can even achieve 100% detection accuracy in laboratory environments. However, applying target detection solutions to practical field pest recognition requires considering more factors such as lighting and occlusion. In addition, for some very small pests in the pest domain, such as aphid targets commonly found on major crops such as corn, wheat, and canola, there are insufficient accuracy and recall rates in detection.

In aphid detection, there are mainly two problems. On the one hand, aphid targets are smaller than most other pest targets, which makes it more difficult to retain aphid target information during feature extraction, as shown in Figure 1. The figure shows the conventional-sized pests and aphid targets on three common crops: corn, canola, and wheat. For example, in the case of corn pests, the length of the sugar beet night larva is approximately 30mm, while the length of the corn aphid adult is about 2.2mm, resulting in a huge difference in target size, which makes it easy to lose information during model training. On the other hand, aphid targets have the characteristic of dense distribution, which poses problems in pest detection, such as occlusion of targets, as shown in Figure 2. The figure shows the distribution of aphid targets on three crops, and when performing detection, the obscured targets are easily mistaken for background areas, leading to a high false negative rate and poor model generalization ability.

## 2. Related Work

With the development of deep learning, computer vision capabilities have gradually enriched and improved. However, in the early stages of computer vision growth, the functionality was somewhat limited. Before the introduction of the R-CNN [3] network, the primary function of computer vision was image classification. Image classification generally involves two processes: first, manually designed feature representations are extracted, and then these representations, which capture abstract information about the

images, are classified using a trained classifier to determine the image category. In the early stages of using computer vision to recognize pests, the task was image classification. In this article, traditional pest detection is divided into two categories based on the detection scene of the target: detection in experimental environments and detection in natural environments.

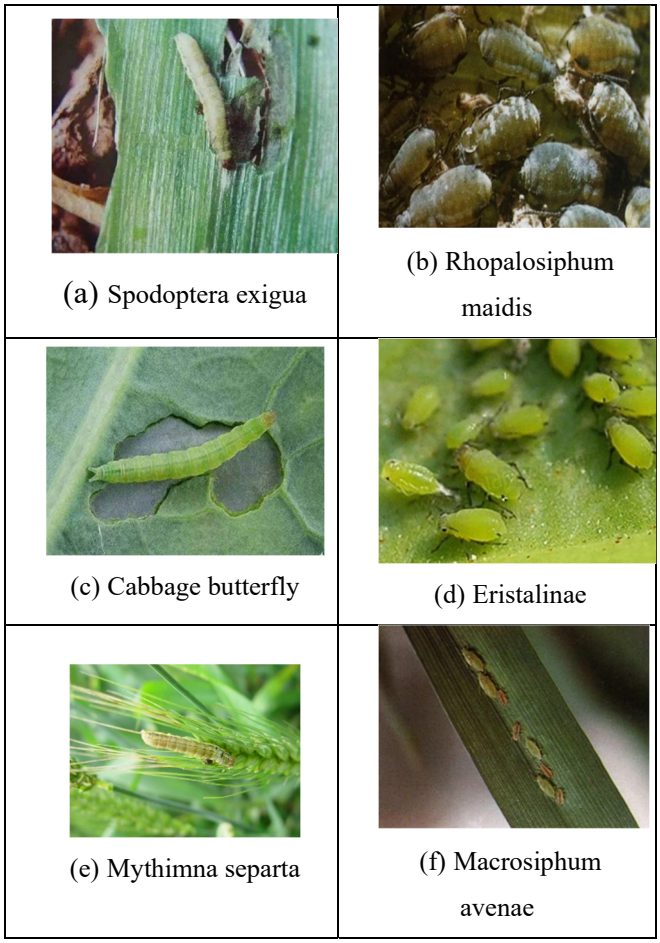


**Figure 1.** Comparison chart of pest sizes

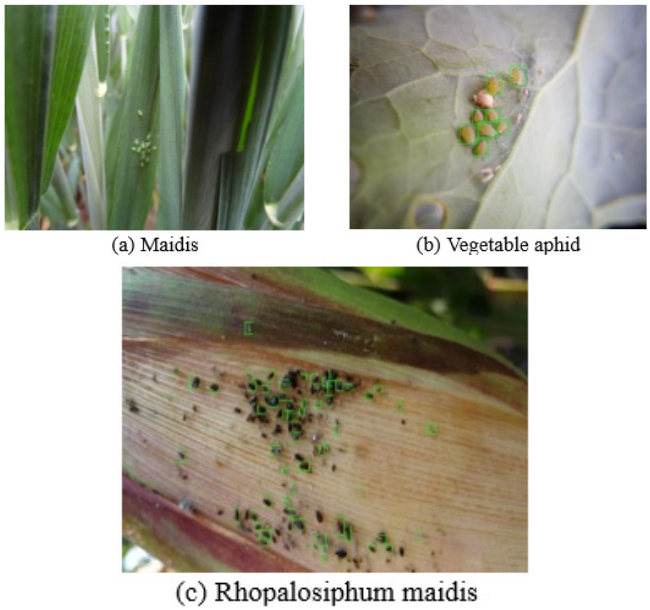


**Figure 2.** Densely distributed small target aphids

Detection in experimental environments refers to capturing field pests by designing attractors through adjustments such as lighting and placing food. Then, image information is collected to create an experimental dataset. Finally, a classifier is trained to achieve artificial intelligence classification of the images in the dataset. The image data collected in this way is non-natural and difficult to be used for practical detection in complex agricultural environments. Shahrul et al. [4] used quality thresholds and moment invariants to extract features of 20 different insects for classifier training and introduced intra-class analysis methods to evaluate the position sensitivity and rotation invariance of images. Wang et al. [5] proposed an insect recognition system based on the family level, classifying and identifying 225 images of 9 common insect orders. With the support of artificial neural networks, they achieved a 93% accuracy rate. To reduce the reliance on human efforts for pest recognition, Qing et al. [6] designed an automated imaging system for identifying light-induced rice field pests. They extracted 156 features including color and shape and trained a support vector machine for classification, achieving a 97.5% accuracy rate. For 6 common field pests, Han Rui-zhen et al. [7] extracted 25 features based on morphology and color and trained a support vector machine classifier, ultimately achieving an 87.4% accuracy rate, providing new ideas for field pest monitoring and control at the time. Li Wen-yong et al. [8] collected 320 images of four pests, including aphids, and extracted RGB and HSV color features from the images. They achieved a 100% recognition accuracy with a multi-class support vector machine. Although these methods achieved high detection accuracy, the images used for training were obtained in experimental setups such as attractors, which lacked the impact of factors such as lighting and occlusion compared to actual field environments. Additionally, the detection methods were not flexible enough and could not be applied in practical applications.

Detection in natural environments refers to capturing pest images directly in the field and training detection solutions on data constructed through manual selection and annotation. Weeks et al. [9] used principal component analysis to construct a classifier for identifying unknown insect specimen images, further demonstrating the tremendous potential of convolutional neural networks in image classification. Cho J et al. [10] designed an automatic identification and detection model based on the size and color features of greenhouse whiteflies, thrips, and other targets. They achieved an accuracy rate of 89.7% for six pests. Solis et al. [11] combined loss algorithms with scale-invariant features to propose a new pest detection algorithm, which achieved an average accuracy of 96.4% for six common pests in greenhouses. They compared the results with manual statistics, demonstrating the effectiveness of the algorithm. Although these methods are more adaptable to complex environments, they still suffer from drawbacks such as slow detection speed and limited detection functionality.

## 3. Network Proposal

To address the challenges of dense distribution and small target size in aphid detection, we made improvements on YOLOv5, as shown in Figure 3.


**Figure 3.** Framework diagram of the aphid detection network

First, we extract features from pest images using a convolutional neural network with an attention mechanism. Then, we perform multi-scale feature fusion on the obtained feature maps. Next, we feed the three different scale feature maps into the detection head for dense area classification and regression. Finally, we send the obtained dense feature regions to the detection head for small target pest detection. Given that most of the target detection frameworks proposed in recent years consist of a backbone network, feature fusion modules, and detection heads, we use this as a basis to provide a detailed description of the improved YOLOv5 network.

The YOLOv5 version 6.1 backbone uses ten convolutional modules for feature extraction, resulting in three different-scale feature maps. The detailed architecture of the network is shown in Figure 4:

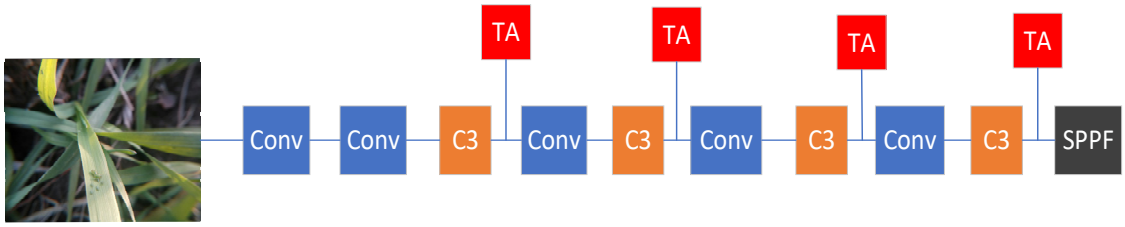


**Figure 4.** Backbone network of YOLOv5.

Specifically, the backbone of YOLOv5, as shown in the gray part in Figure 2, consists of 5 convolutional layers, 4 C3 modules, and an SPPF (Spatial Pyramid Pooling Fusion) layer. The convolutional layers and C3 layers are responsible for feature extraction from images, while the SPPF layer is used to convert feature maps of arbitrary sizes into fixed-size feature vectors. In Figure 2, the red modules represent the introduction of a TA (Triplet Attention) module after each C3 layer. TA is an attention mechanism that combines spatial attention with channel attention, allowing the feature extraction network to selectively focus on target regions containing important information while suppressing other irrelevant information. This reduces the impact of background information on detection results, thereby improving the model's ability to detect small targets. The specific architecture of the TA attention mechanism is shown in Figure 5.

The images in the pest dataset are three-channel images with a tensor shape of [H, W, C], and they are input into the convolutional neural network for feature extraction. As a result, the shape of the tensor changes, with reduced width and height but increased channel numbers, as shown in the input tensor in Figure 3. We feed the input tensor into three parallel branches. The first branch calculates weights for regions of the tensor through pooling, convolution, and activation, and then multiplies these weights with the original tensor to obtain the tensor within the orange region. The second branch rotates the input tensor counterclockwise by 90 degrees along the axis to obtain a tensor in the green region. It calculates weights in the same way as the first branch, applies them, and then rotates the tensor back to its original shape. The third branch is similar to the second branch. The tensors obtained from the three branches are then added and averaged to obtain the output tensor with cross-dimensional interactions between spatial and temporal attention.

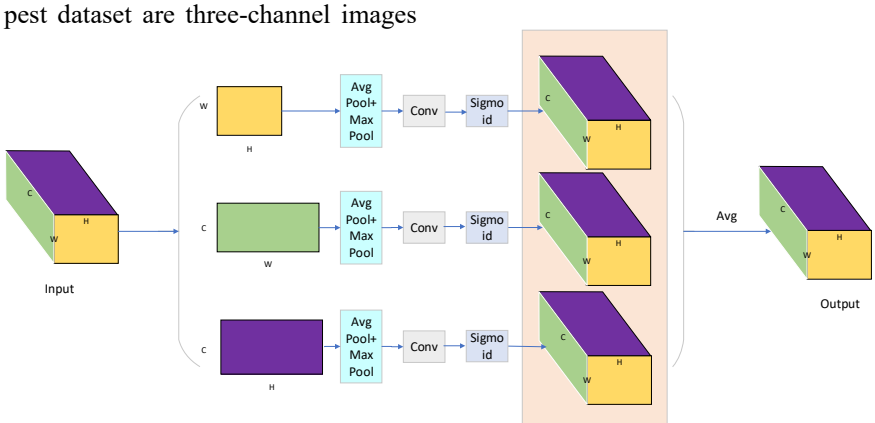


**Figure 5.** TA (Triplet Attention) attention diagram.

To make the information in the feature maps more global in the object detection process, it is common to incorporate feature fusion modules for multi-scale feature fusion. The feature fusion module in YOLOv5 is a network structure that combines FPN (Feature Pyramid Network) and PAN (Path Aggregation Network). Three feature maps with scales of [H/8, W/8], [H/16, W/16], and [H/32, W/32] are obtained from the second and third C3 modules of the backbone

network, as well as the SPPF module. These feature maps are then fed into the feature fusion module, as shown in Figure 6:

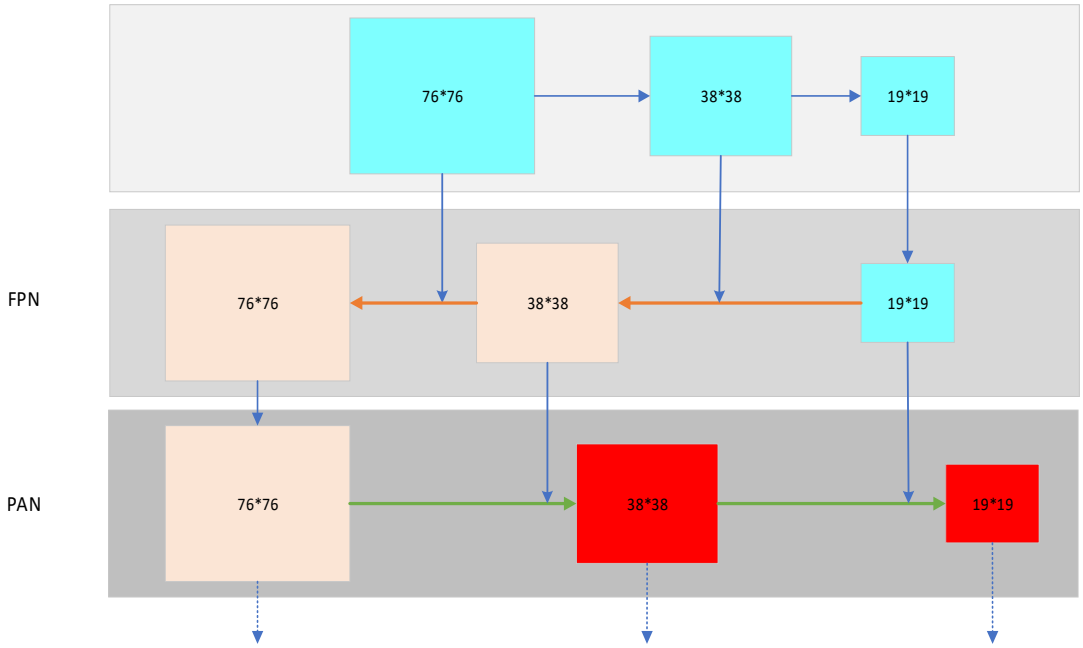


**Figure 6.** Feature fusion module.

The orange arrows represent upsampling operations, the green arrows represent downsampling operations, and the blue dashed arrows represent the feature maps that will be sent to the detection head. In the FPN module, the three feature maps obtained from the backbone network undergo an upsampling and concatenation operation, starting from the top-level feature map and conveying semantic information to the bottom-level feature maps for multi-scale feature fusion. PAN, on the other hand, performs downsampling and concatenation operations to transfer positional information from the bottom-level feature maps to the top-level feature maps, enhancing precision in object localization.

The detection head is the module responsible for classifying and localizing the objects based on the feature map information. In the traditional YOLOv5, the detection head consists of three parts, each searching for targets in feature maps of three different scales. We have added an additional detection head, making the first detection head responsible for locating pests in densely populated areas, and the second detection head for detecting pests within those dense areas. The specific details are shown in Figure 7:

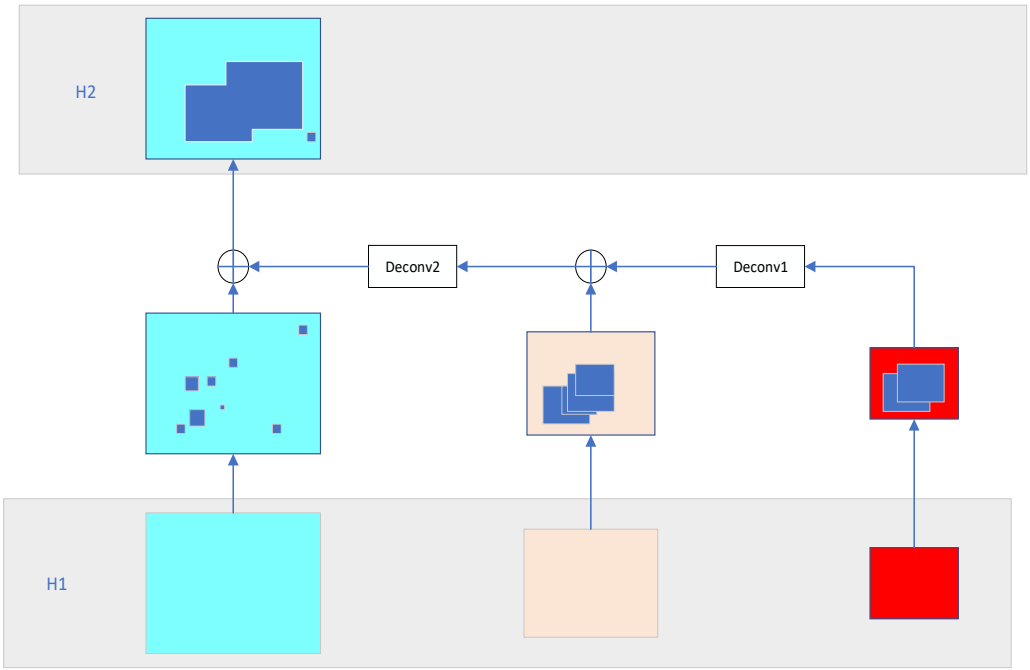


**Figure 7.** Detection head module

The detection head receives three feature maps of different scales obtained from the convolutional neural network and detects object targets based on pre-set anchor boxes. The light blue feature map is used for detecting small targets, the light yellow one for medium-sized targets, and the red small feature map for detecting large targets. Therefore, we obtain pest-dense regions on three feature maps and then fuse the pest distribution information from the three feature maps,

which is sent to the detection head for detailed detection. Specifically, we first perform deconvolution on the smallest-scale feature map and add it to the corresponding positions of the medium-scale feature map. We then perform upsampling on the obtained feature map and add it to the corresponding positions of the large-scale feature map to obtain the final feature map after multi-scale fusion. This process fuses the target information carried by the three different-scale feature maps and yields the dense pest distribution areas, which are then sent to the detection head, achieving detection that focuses only on dense areas and outliers. With the development of multi-task loss functions, this method becomes more practical, and we adjust the weights between the detection head and the use of different loss functions to achieve better results.

To improve convergence speed and final accuracy, we calculate the most suitable anchor box values for the real boxes in the dataset and the dense distribution real boxes through clustering. For each feature map, we assign three anchor boxes with different aspect ratios, with ratios of [116, 90], [156, 198], and [373, 326]. The detection head is responsible for classifying and regressing on one scale of feature map, and we have preset nine anchor boxes on it, with ratios of [5, 6, 8, 14, 15, 11], [10, 13, 16, 30, 33, 23], and [30, 61, 62, 45, 59, 119]. Since the annotated information for dense distribution targets includes both well-separated pest targets with a small proportion in the original image and large and medium-sized true target boxes distributed in clusters, using the default anchor boxes in YOLOv5 for dense area detection is sufficient to produce good detection results. In the only feature map faced by the detection head, pests are concentrated, and using only three anchor boxes may lead to a low recall rate for small target detection. Therefore, we need to lay denser anchor boxes. However, this operation introduces more parameters and computational complexity. To address this issue, we made some improvements in the positive and negative sample matching process.

The positive and negative sample selection scheme for the detection head is consistent with YOLOv5, requiring that the overlap between anchor boxes and real boxes is greater than a threshold and the aspect ratios of the two are not too different. For the detection head, we propose a new matching requirement: we first determine whether the center point of the pre-set anchor box falls within the dense pest area output by the detection head. When it falls within the dense area and meets the two conditions for matching positive samples of the detection head, it is considered a positive sample and the loss function is calculated; otherwise, it will not be matched as a positive or negative sample. Therefore, by first filtering the detection areas, finding the range with dense targets, and then using multiple anchor boxes for multi-target detection, we greatly increase the recall rate of dense pest detection at a relatively low cost.

# 4. Experimental and Dataset

We started training a pest detection network from scratch, making appropriate adjustments to hyperparameters during the process. Our experiments were conducted using the AP dataset, which consists of aphid images collected in agricultural environments, and the composition of the dataset's images is presented in Table 1.

**Table 1.** Composition of Pest Information in the AP Dataset

| Class | Ratito | Number | Object |
|---|---|---|---|
| **Greenbug** | **8.6%** | **426** | **26380** |
| **Aphid** | **6.0%** | **151** | **61050** |
| **Sitobion avenae** | **4.2%** | **34** | **22538** |
| **all** | **------** | **611** | **109968** |

The method proposed in this paper, along with the related comparative experiments, was implemented using the PyCharm integrated development environment and code written using the PyTorch 1.10.2 framework with Python 3.6 API. We used an 18.04 version of the Ubuntu host on a Linux server, equipped with a Tesla T4 GPU processor with 15GB of memory and 48 CPU processors. The training was performed using the SGD algorithm on the T4 processor, running for a total of 500 epochs, with each batch containing 16 images. The initial learning rate was set to a certain value and then reduced by a factor of 10 every 1000 iterations using learning rate decay. We incorporated an attention mechanism into the YOLOv5 backbone network and maintained the FPN+PAN structure. In the detection head section, we used a cascading approach with two detection heads for detection. The input resolution of images was standardized to a certain value.

**Table 2.** Comparison of Experimental Results

| Model | Backbone | AP [0.5:0.95] | AP0.5 | AP0.75 | Input Size |
|---|---|---|---|---|---|
| **Faster-Rcnn** | R101 | 0.190 | 0.305 | 0.190 | 1333×800 |
| **R-Fcn** | R101 | 0.109 | 0.283 | 0.051 | 1333×800 |
| **Cascade-Rcnn** | R101 | 0.183 | 0.306 | 0.178 | 1333×800 |
| **SSD** | Vgg16 | 0.136 | 0.277 | 0.086 | 512×512 |
| **Retinanet** | R101 | 0.174 | 0.298 | 0.161 | 1333×800 |
| **YOLOv5** | Darknet53 | 0.237 | 0.526 | 0.207 | 640×640 |
| **FCOS** | HRNet | 0.154 | 0.309 | 0.126 | 1333×800 |

We conducted comparative experiments by selecting three commonly used two-stage object detection algorithms and three single-stage object detection algorithms, along with one anchor-free method. The selected algorithms include Faster R-CNN, R-FCN, Cascade R-CNN, SSD, RefineDet, YOLOv5, and FCOS.

From Table 2, we can observe that our OS-Net network achieves the best detection results on the AP-A dataset even

at lower image resolutions, further demonstrating the effectiveness of the network architecture. Moreover, compared to the baseline YOLOv5 network, we made changes in two aspects: the inclusion of an attention mechanism in the backbone network and the design of an overall-to-specific detection head. Inserting four TA modules at specified locations in the YOLOv5 backbone network improves the average accuracy by only 0.001, indicating that this improvement strategy has a limited impact on the final results. In other words, the improvement in average accuracy of the OS-Net network primarily stems from the design of the overall-to-specific detection head.

# 5. Conclusion

We proposed the OS-Net network, which is based on YOLOv5 and incorporates our concept of going from coarse to fine to achieve aphid detection. Specifically, we added an additional detection head, where the first head is used to detect densely annotated aphid clusters, and the second component is used to capture more small target aphids within the target regions detected by the first component by matching more anchor boxes. The experiments demonstrate the effectiveness of our approach.

# Acknowledgments

This work was supported in part by a grant from ZY2021002.

# References

[1] Liu J, Wang X. Plant diseases and pests detection based on deep learning: a review[J]. Plant Methods, 2021, 17: 1-18.

[2] Wolfert S, Ge L, Verdouw C, et al. Big data in smart farming–a review [J]. Agricultural systems, 2017, 153: 69-80.

[3] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[4] Yaakob S N, Jain L. An insect classification analysis based on shape features using quality threshold ARTMAP and moment invariant[J]. Applied Intelligence, 2012, 37: 12-30.

[5] Wang J, Lin C, Ji L, et al. A new automatic identification system of insect images at the order level[J]. Knowledge-Based Systems, 2012, 33: 102-110.

[6] Qing Y A O, Jun L V, Liu Q, et al. An insect imaging system to automate rice light-trap pest identification[J]. Journal of Integrative Agriculture, 2012, 11(6): 978-985.

[7] Tian H, Wang T, Liu Y, et al. Computer vision technology in agricultural automation--A review[J]. Information Processing in Agriculture, 2020, 7(1): 1-19.

[8] Al Ohali Y. Computer vision based date fruit grading system: Design and implementation[J]. Journal of King Saud University-Computer and Information Sciences, 2011, 23(1): 29-36.

[9] Weeks P J D, O'Neill M A, Gaston K J, et al. Species–identification of wasps using principal component associative memories [J]. Image and Vision Computing, 1999, 17(12): 861-866.

[10] Cho J, Choi J, Qiao M, et al. Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis [J]. Red, 2007, 346(246): 244.

[11] Solis-Sánchez L O, Castañeda-Miranda R, García-Escalante J J, et al. Scale invariant feature approach for insect monitoring [J]. Computers and electronics in agriculture, 2011, 75(1): 92-99.