YOLO-pest: A real-time multi-class crop pest detection model

Shifeng Dong[a,b], Jie Zhang*[a], Fenmei Wang[a,b], Xiaodong Wang[a,b]
[a]Institute of Intelligent Machines, Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China; [b]Scinece Island Branch, Graduate School of USTC, Hefei 230026, China

ABSTRACT

Crop pest control is one of the important tasks for crop yield. However, multi-class pests and high similarity in appearance bring challenges to precision recognition of pests. In recent years, deep-learning based algorithms in object detection have achieved an excellent results, such as the YOLO detector, which can balance accuracy and speed. YOLO performs well in detecting normal size objects, but has low precision in detecting small objects. The accuracy decreases notably when dealing with pest data set, which have large-scale changes and multi-class. To solve the detection problem of multi-scale pest, we propose a detector named YOLO-pest based on YOLOv4 to improve the performance of pest detection. Our approach includes using lite but efficient backbone mobileNetv3 and lite fusion feature pyramid network. The improved detector significantly increased accuracy while remaining fast detection speed. Experiments on the constructed Croppest12 data set show that our improved algorithm outperforms other compared methods.

Keywords: Pest detection, YOLO, deep learning, feature pyramid network

1. INTRODUCTION

Crop pests have a significant impact on crop yields and the agricultural economy. To solve the pest problem, it is necessary to distinguish the pest categories and apply precise medication to control them. However, there are so many pest categories and high similar morphology in appearance that non-agricultural specialists are not able to distinguish between them. The traditional method to recognition pests mainly relies on experiences. It is inaccurate and labor-intensive, thus will affect the precision pest control work. Therefore, it is essential to propose a new method to detect multi-class crop pests in real-time and accurately.

In recent years, with the development of deep convolutional neural networks (DCNNs), object detection has made great achievements[1-2]. The DCNN based object detection algorithm can automatically extract the features of pests, eliminating the subjective factor of manual feature extraction[3], and thus can accurately identify the species and number of pests. However, most of the existing recognition methods are designed for generic images collected on the Internet as training data sets[4-5]. On the basis of this, some significant progress has been made in object detectors using common datasets. Among these methods, two-stage methods are more popular for pest detection due to their high detection accuracies, such as Faster R-CNN[1], R-FCN[6], and Cascade R-CNN[7]. One-stage methods are less time-consuming because it has a simple network, but lose accuracy. One-stage method has YOLO[2,8,9], SSD[10], RetinaNet[11] and so on. However, there is still a big gap in the practical application of pest detection.

In this paper, a crop pest detection framework YOLO-pest is proposed. YOLO-pest uses Mobilenetv3 to replace the YOLOv4 backbone network to significantly reduce the number of parameters, and proposes a lite-FPN architecture. We built a pest image dataset named Croppest12 containing several forms of 12 common crop pests. YOLO-pest achieves 70.07% mAP on the Croppest12 dataset, which is only 2.4 AP lower than the YOLOv4 method. The model size is only 46.9M, which is 198.8M less than YOLOv4.

2. METHOD

In this paper, we proposed YOLO-pest mainly improved in two aspects, one is to replace the YOLOv4[12] backbone network with Mobilenetv3[13], and the other one is to design the FPN-lite. The network framework is shown in Figure 1.

*zhangjie@iim.ac.cn

International Conference on Computer Application and Information Security (ICCAIS 2021), edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12260, 1226003 · © The Authors. Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2637467

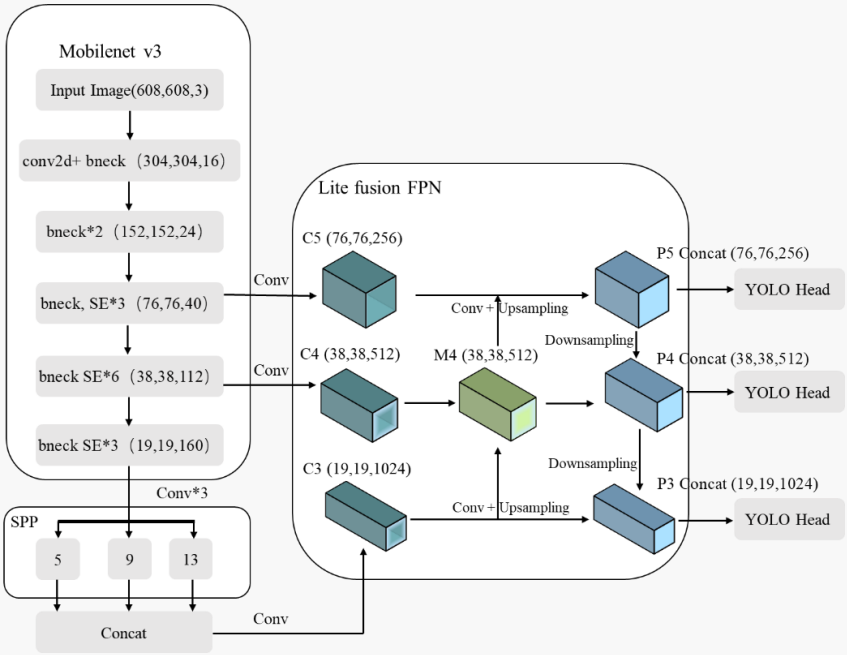

Figure 1. YOLO-pest framework.

2.1 Backbone

MobileNet[13] uses the reverse residual module of linear bottleneck to improve feature extraction based on adopting deeply separable convolution. The images are fed into the backbone feature extraction network, which uses the bneck structure. SE[14] indicates that the attention mechanism is added to this layer. Bneck structure is used to up-dimension the input feature map first and then perform deep separable convolution, while squeeze-and-excite attention module is added to balance the weights of each channel of the feature map.

In the backbone network, the h-swish activation function modified by the swish activation function was used. Equation (1) is the swish activation function.

$$swish(x) = x \cdot \sigma(\kappa x) \tag{1}$$

$$\sigma = \frac{1}{1+e^{-x}} \tag{2}$$

where x is input, κ is the hyper-parameter used to adjust the slope of the activation function, σ is the sigmoid function. h-swish uses the ReLU6 activation function to optimize the $\sigma(\kappa x)$ in swish.

$$hswish(x) = x \cdot \frac{\mathrm{Re}\,LU6(x+3)}{6} \tag{3}$$

$$\mathrm{Re}\,LU6(x) = \min(\max(0, x), 6) \tag{4}$$

The use of ReLU6 activation allows limiting the input x to between 0 and 1, thus replacing the function of the sigmoid function. At the same time, h-swish reduces the number of activation functions in the bneck structure to 16 while maintaining the same accuracy of 32 activation functions using swish, reducing the complexity of the network.

2.2 Lite fusion feature pyramid network

Since the pests have different scales of targets, the single-scale convolution kernel cannot adapt to multi-angle and multi-scale changing pictures. Thus we need the feature pyramid network architecture[15]. The FPN shallow layer has a larger resolution and contains clearer location information, the deep layer features contain rich semantic information, and the feature layers at different scales contain different feature information and are more adaptable to objects of different sizes.

To solve the problem that multi-scale pests reduce model accuracy, a lightweight multi-layer fusion module is constructed for the feature pyramid network, which is shown in Figure 2. The feature map of size 52×52 is first downsampled using a 2×2 averaging pooling layer. This allows feature fusion operations to provide shallow visual information and preserve more detailed features. Second, the feature map of size 13×13 is upsampled to size 26×26. This feature map has high-level semantic information and contains global object information. In the end, three feature maps of size 26×26 are concat into one feature map. The size of 13×13 and 26×26 feature maps are upsampled to generate 52×52 and 13×13 additional feature maps then combined them to feature pyramids.

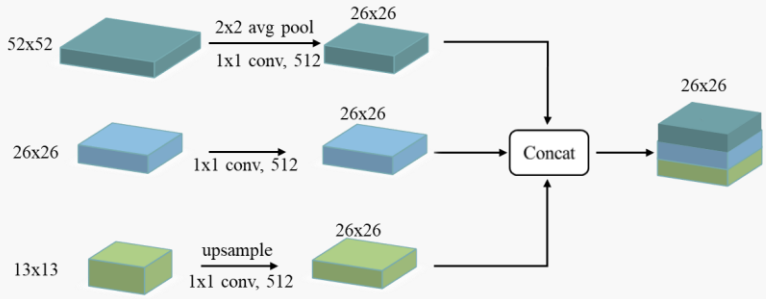

Figure 2. The architecture of the proposed lite fusion module.

3. EXPERIMENTS

3.1 Experiment platform and dataset

Experiment platform. In this paper, the model is trained on Ubuntu 18.04 operating system, using PyTorch framework, Intel Core I7-10700 CPU, NVIDIA TITAN RTX GPU (24GB), CUDA10.0, CuDNN7.6, Python3.7 software environment. The input images are resized to 640×640, the training batch is set to 16, the initial learning rate is 0.001, the IoU threshold is set to 0.5, and all models are trained for 100 epochs according to these parameters.

Dataset. We collected 11,130 pest images with the resolution of 1944×2592 using under-light pest image acquisition equipment. The images are annotated by agricultural experts with the pest bounding boxes and classes and the labeling open source software is LabelImg. Based on this, we built a dataset named Croppest12. Table 1 shows the pest names and their corresponding instance numbers, the average height and width of the pests in that category. As can be seen in Table 1, the pest instances range from 198 to 18,463. There are 8 categories of pests with instances less than 1,000. And the width and height of the pest boxes are mostly less than 100 pixels, which is still small compared to the 1944×2592 resolution image.

Table 1. Statistics of Croppest12.

Classes	Pest name	Instances	Average width (pixels)	Average height (pixels)
AS	Agrotis segetum	815	80.8	63.2
AT	Agrotis tokionis	243	97.1	76.3
AE	Agrotis exclamationis	285	90.1	69.5
XC	Xestia c-nigrum	362	78.3	60.6
HO	Holotrichia oblita	446	70.2	54.9
HP	Holotrichia parallela	5186	66.3	52.0
AC	Anomala corpulenta	18463	60.6	47.7
GO	Gryllotalpa orientalis	3237	119.8	92.1
PC	Pleonomus canaliculatus	228	69.1	54.4
AS	Agriotes subrittatus	3615	44.8	35.8
MC	Melanotus caudex	437	41.5	32.2
SF	Spodoptera frugiperda	198	53.5	41.6

3.2 Evaluation metrics

To evaluate the performance of the algorithm in this paper, some evaluation metrics such as Precision (P) and Recall (R) are used to quantitatively evaluate the model, which is calculated in the form shown in equation (5). TP, TN, and FP denote the number of targets that are correct, targets that are incorrect, and undetected, respectively.

$$\begin{cases} precision = \dfrac{TP}{TP + FP} \\ recall = \dfrac{TP}{TP + FN} \\ AP = \displaystyle\int_0^1 p(r)dr \\ mAP = \dfrac{1}{C}\displaystyle\sum_{j=1}^{C} AP_j \end{cases} \tag{5}$$

Average Precision (AP) is used to evaluate the performance of the model on the test set. The multi-category detection results are usually measured using mean Average Precision (mAP), it calculates based on the shape of the PR curve. In equation (5), C is the number of classes. In addition, we measure the speed of the detection algorithm in terms of the number of images processed per second (FPS).

3.3 Experiment results

As shown in Table 2, we compare the number of model parameters, mAP, and FPS of Faster R-CNN, SSD, and YOLOv3 while keeping the training parameters consistent. Compared with Faster RCNN, the average precision of YOLO-pest is 5 mAP higher, but in terms of inference speed, YOLO-pest is 40 FPS faster than Faster R-CNN, which meets the real-time detection requirement. Table 3 shows the scientific names of each pest category, as well as the number of instances. There is also the AP of different methods for each category of pests, and it can be seen that our method surpasses other methods for almost all pest categories.

Table 2. Performances of different models.

Method	Input image size	Backbone	Params (M)	FPS	mAP (%)
Faster R-CNN	1280×800	ResNet-50	41.25	22.3	65.16
YOLOv3	416×416	Darknet53	62.3	54.2	62.22
SSD	512×512	VGG16	36.04	38.7	62.17
YOLO-pest (ours)	608×608	MobileNet v3	46.9	62.5	70.07

Table 3. Performances of single class pest.

Classes	Pest name	Instances	AP (%)		
			SSD	YOLOv3	Ours
AS	Agrotis segetum	815	40.1	34.8	47.5
AT	Agrotis tokionis	243	45.2	54.8	68.2
AE	Agrotis exclamationis	285	68.9	56.3	68.9
XC	Xestia c-nigrum	362	50.9	53.3	62.8
HO	Holotrichia oblita	446	51.0	55.9	58.6
HP	Holotrichia parallela	5186	84.1	81.0	85.2
AC	Anomala corpulenta	18463	91.4	87.7	93.7
GO	Gryllotalpa orientalis	3237	92.7	93.1	94.0
PC	Pleonomus canaliculatus	228	54.0	50.6	61.7
AS	Agriotes subrittatus	3615	70.0	71.8	76.0
MC	Melanotus caudex	437	41.4	57.7	61.7
SF	Spodoptera frugiperda	198	56.4	49.7	62.6

Table 4 shows the results of the ablation experiments performed on YOLO-pest. Mobilenetv3 in the table represents replacing the CSPDarknet53 backbone network in YOLOv4 with the Mobilenetv3 structure. Lite-fusion FPN represents replacing the PANet FPN in YOLOv4 with the Lite-fusion FPN structure fusion network. The ablation experiments compare the module parameters, FPS, and mAP under various structure combinations, respectively. It can be seen that the number of parameters of the original YOLOv4 is 245.7M, and the model size further decreases to only 47.6M after replacing CSPDarknet53 with Mobilenetv3 on this model, but the mAP also decreases to 68.6%, and it can be concluded that if only PANet FPN is replaced with Lite-fusion FPN structure, the model size is almost unchanged, but the mAP is increased by 2.6 mAP, indicating that Lite-fusion FPN can indeed improve the module with almost no effect on the model size. Finally, by combing both Mobilenetv3 and Bi-FPN-Lite structures, the mAP of the algorithm increases to 70.1%, and the model size decreases to only 46.9M, which ensures a high mAP even though the model size and number of parameters are significantly reduced. We report the detection result in Figure 3. Our method can accurately detect most pest targets, which meets the needs practical application requirements.

Table 4. Ablation experiment.

Method	Params (M)	FPS	mAP (%)
YOLOv4	245.7	45.3	72.5
YOLOv4+ Mobilenetv3	47.6	59.6	68.6
YOLOv4+Lite-fusion FPN	244.5	51.8	71.2
YOLO-pest (Mobilenetv3+ Lite-FPN)	46.9	62.5	70.1

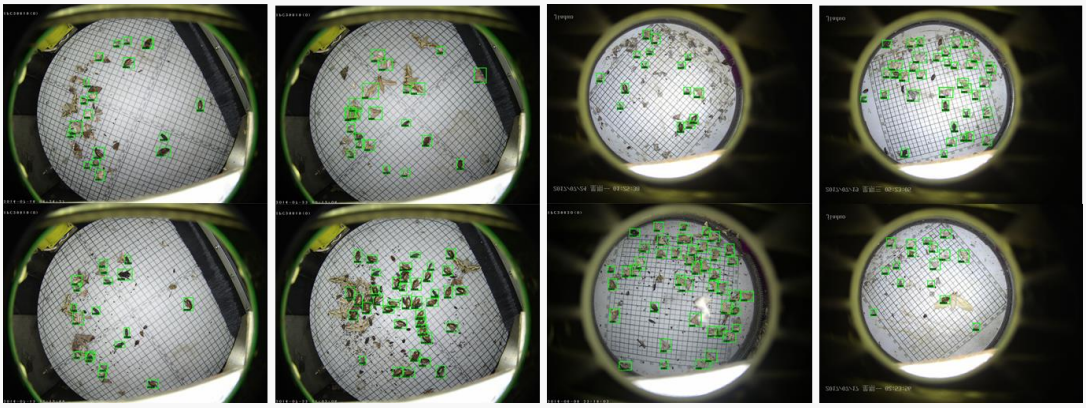

Figure 3. Some detection results of our model on Croppest12.

4. CONCLUSION

To address the problems of complex network structure and redundant computational parameters of existing detection algorithms, a lightweight pest detection method is proposed, which can achieve efficient real-time detection at multiple scales and objects. Based on YOLOv4, the problem of large model size is alleviated by replacing the backbone feature extraction network, then the feature pyramid is improved to enhance the expression of semantic feature and location information. The experimental results show that the YOLO-pest method parameters are smaller than other mainstream algorithms, and can balance detection accuracy and speed, which has good engineering application value.

REFERENCES

[1] Ren, S., He, K., Girshick, R. and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., 1137-1149 (2017).

[2] Redmon, J and Farhadi, A., "Yolov3: An incremental improvement," arXiv:1804.02767, (2018).

[3] Tetila, E. C., Machado, B. B. and Astolfi, G., "Detection and classification of soybean pests using deep learning with UAV images," Computers and Electronics in Agriculture, 105836, (2020).

[4] Wang, F., Jiang, M. and Qian, C., "Residual attention network for image classification," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 6450-6458 (2017).

[5] Xie, Q., Luong, M. T. and Hovy, E., "Self-training with noisy student improves ImageNet classification," arXiv:1911.04252, (2020).

[6] Dai, J., Li, Y., He, K. and Sun, J., "R-FCN: Object detection via region-based fully convolutional networks," Annual Conf. on Neural Information Processing Systems, 379-387 (2016).

[7] Cai, Z. and Vasconcelos, N., "Cascade R-CNN: Delving into high quality object detection," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 6154-6162 (2018).

[8] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., "You only look once: Unified, real-time object detection," arXiv:1506.02640, (2016).

[9] Redmon, J. and Farhadi, A., "YOLO9000: Better, faster, stronger," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 6517-6525 (2017).

[10] Liu, W., Anguelov, D. and Erhan, D., "SSD: Single shot multibox detector," European Conf. on Computer Vision (ECCV), 21-37 (2016).

[11] Lin, T. Y., Goyal, P., Girshick, R. B. and Dollár, P., "Focal loss for dense object detection," IEEE Inter. Conf. on Computer Vision, 2999-3007 (2017).

[12] Bochkovskiy, A., Wang, C. Y. and Liao, H. Y. M., "YOLOv4: Optimal speed and accuracy of object detection arXiv:2004.10934, (2020).

[13] Howard, A., Sandler, M. and Chu, G., "Searching for mobilenetv3," IEEE Inter. Conference on Computer Vision, 1314-1324 (2019).

[14] Hu, J., Shen, L., Albanie, S., Sug, N. and Wu, E., "Squeeze-and-excitation networks," IEEE Trans. Pattern Anal. Mach. Intell., 2011-2023 (2020).

[15] Lin, T. Y., Dollar, P., Girshick, R,, He, K., Hariharan, B. and Belongie, S., “Feature pyramid networks for object detection,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 936-944 (2017).