Machine Vision and Applications (2022) 33:76
https://doi.org/10.1007/s00138-022-01310-0

ORIGINAL PAPER

A global activated feature pyramid network for tiny pest detection in the wild

Liu Liu[1] · Rujing Wang[2] · Chengjun Xie[2] · Rui Li[2] · Fangyuan Wang[1] · Long Qi[3]

Received: 28 June 2020 / Revised: 28 February 2022 / Accepted: 23 March 2022

Abstract
Small object detection techniques have been developed for decades, but one of key remaining open challenges is detecting tiny objects in wild or nature scenes. While recent works on deep learning techniques have shown a promising potential direction on common object detection in the wild, their accuracy and robustness on tiny object detection in the wild are still unsatisfactory. In this paper, we target at studying the problem of tiny pest detection in the wild and propose a new effective deep learning approach. It builds up a global activated feature pyramid network on convolutional neural network backbone for detecting tiny pests across a large range of scales over both positions and pyramid levels. The network enables retrieving the depth and spatial intension information over different levels in the feature pyramid. It makes variance or changes of spatial or depth-sensitive features in tiny pest images more visible. Besides, a hard example enhancement strategy is also proposed to implement fast and efficient training in this approach. The approach is evaluated on our newly built large-scale wide tiny pest dataset containing 27.8K images with 145.6K manually labelled pest objects. The results show that our approach perform well on pest detection with over 71% mAP, which outweighs other state-of-the-art object detection methods.

Keywords Convolutional neural network · Tiny pest detection · Global activated feature pyramid network · Hard example enhancement

1 Introduction

Due to the success of deep learning techniques, the study of general object detection has been well developed in computer vision community, where it also brings benefits to many subsequent small object detection applications, like detection of tiny face [1], traffic light [2], pedestrians [3], insects [4], tomato fruit [5], etc. But comparing with normal object detection, tiny object detection remains an open problem due to a challenging fact that many discriminative details and features of tiny object are small, blurred, hidden and short of sufficient details. These pose a fundamental dilemma that it is hard to distinguish small object from generic clutter in the background. The diversity and complexity of scenes in the wild cause a variety of variation or noisy factors to increase the difficulty of applying typical generic object detection techniques. Taking the example of pest detection in the wild, detection accuracy of tiny pests is impacted by many factors like lighting illumination, location of pests and distribution of pests as shown in Fig. 1. In this paper, we target at studying tiny pest detection in the wild and explore an efficient deep learning-based solution towards this problem.

✉ Rujing Wang
rjwang@iim.ac.cn

Liu Liu
liuliu66@mail.ustc.edu.cn

Chengjun Xie
cjxie@iim.ac.cn

Rui Li
lirui@iim.ac.cn

Fangyuan Wang
wfy710@mail.ustc.edu.cn

Long Qi
qilong@scau.edu.cn

1 University of Science and Technology of China, Hefei 230026, China

2 Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China

3 South China Agricultural University, Guangzhou 510642, China

Published online: 10 August 2022

Recently, deep learning techniques have brought significant progress in various image processing based tasks especially on 2D object detection, in which numerous researches attract attention in computer vision community such as SSD [6], Faster R-CNN [7], feature pyramid network (FPN) [8] and other extended variants of these approaches [9–11]. Among these novel methods, Faster R-CNN proves to be a powerful alternative in object detection task comparing with the other one-shot CNN approaches, which shows great success in PASCAL VOC [12] and Microsoft COCO [13] object detection challenges. This might be contributed by the coarse-to-fine learning strategy in Faster R-CNN as well as its variants that adopt region-of-interest (RoI) pooling to focus on local regions of input images. But targeting at small object detection, feature pyramid structure [8] has become an increasingly popular module deployed in Faster R-CNN architecture, which achieve a superior performance on detecting objects with small sizes by two novel designs: multi-scale feature maps and top-down connection between neighbour levels. The former enables the model to detect specific-scale objects on corresponding feature level and the latter promotes the integration of contextual information. Therefore, for small object detection in deep learning area, the recently emerging works usually choose FPN as feature extraction network [14,15].

However, due to the applicable gap between generic object detection and real-world intelligent agriculture tasks, despite there are some works proposed on dealing with plant relevant issues [16,17], these deep learning approaches are often intractable for practical pest monitoring systems. The reasons come from many in-field difficulties that brings unsatisfied performance on current computer vision techniques: (1) Wild environment might increase difficulty in feature extraction where the visual information is unavoidably confused with the clutter background. (2) The model might not be sensitive to recognizing the categories of detected tiny pests due to the limited inter-class distance in feature space. (3) Some of pest species usually tend to gather into clique in agricultural crops so the density distribution makes the model insensitive to parse the gathered tiny pests in images.

For overcoming above obstacles, we attempt to propose a new effective deep learning approach with parallel attention mechanism in feature pyramid network towards tiny pest detection in the wild. The motivation of our idea is to first build up a feature pyramid structure on CNN backbone, and then propose a global activated module (GAM) for retrieving depth and spatial attention over different levels in the pyramid network. The extracted depth and spatial attention information will generate weights to re-balance the feature pyramid network. Differing with existing work [18,19] in adoption of serial mode attention mechanism in object detection, we design our GAM in a parallel mode so that it could extract channel and position attention avoiding their mutual influence to each other. Also compared to Faster R-CNN, the adjusted network will enable variance or changes of spatial or depth-sensitive features in images more visible in the pooling layers. This property will allow some missing features of tiny pests in pooling layers in one level to be re-detected by many pyramid levels. It will help deal with some obstacles of tiny pest detection in the wild, like overlapping, different density distribution and tiny size.

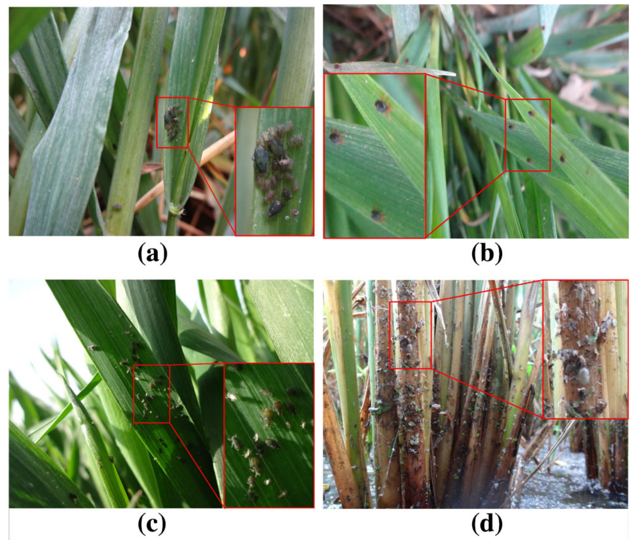


Fig. 1 The problem of tiny pest detection in the wild scene is affected by many factors, **a** Dense or clustered distribution of pests. **b** Sparse distribution of pests. **c** Influence of sunlight illumination. **d** Background noise

Following this idea, we present a two-stage Global activated feature pyramid network (GaFPN)-based CNN approach towards tiny pest detection in the wild as shown in Fig. 2. In the first stage, the feature pyramid structure is built on CNN backbone for multi-scale feature extraction; a GAM is designed to extract and fuse depth and spatial activation information from feature pyramid. In the second stage, a region proposal network (RPN) is applied for providing region proposals and fully connected layers, which are adopted for pest classification and position regression. During the training phase, a hard example enhancement (HEE) strategy is proposed into our method for implementing efficient training that mines hard examples for fast gradient descent. We build CNN backbone with more layers in shallow blocks and fewer layers in deep blocks aiming to mine more valuable semantic information while maintaining sufficient object information for localization. The feature maps outputted from shallow blocks participate more in boxes finetuning. The approach is evaluated over our newly published large-scale pest detection specific image dataset containing 27.8K raw images with 145.6K manually labelled pest objects. The image data were collected in the wild field using mobile camera over 7 years. The experimental results show

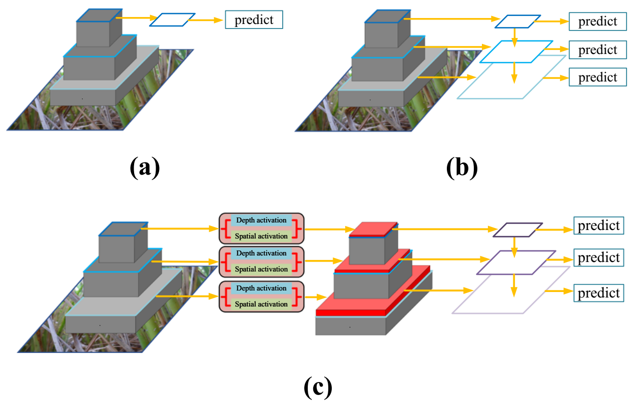


Fig. 2 **a** Faster R-CNN using single feature map, which is suitable for common object detection in wild but not well on tiny object detection. **b** FPN building a feature pyramid with multi-scale levels, which can observe some missing information over small tiny objects in single feature map. **c** Our approach augmenting global activated module (GAM) for extracting and fusing depth and spatial activation in feature pyramid network, which allows this network sensitive to variations or changes of depth and spatial features of tiny objects in the images

that our approach achieves over mAP of 71%, which outweighs other state-of-the-art methods.

The major contributions of this paper are as follows:

- A novel Global activated module (GAM) is proposed in feature pyramid network for tiny object detection task. It can help recognize and extract discriminative features of tiny objects, then accommodate large variations and changes of distribution of tiny objects over images.
- A two-stage CNN-based solution by integrating the GaFPN method is presented towards tiny pest detection in the wild. It is feasible to deal with pest images from the wide environment with good accuracy and efficiency.
- A comprehensive and in-depth experimental evaluation on practical industry level large-scale wild tiny pest dataset (7 years over 27k images) is provided for verifying the proposed approach. The results show that our approach delivers an average 71% accuracy over 6 types of pest detections, which outweighs the state-of-the art approaches.
- A domain-specific dataset containing more than 27K images and 145K annotated pests is built up and publicly available. The pest image dataset produces an initial state-of-the-art benchmark for further deep study and evaluation. We believe these advances will facilitate future research and applications in tiny object detection in the wild.

2 Related work

Classic Object Detection Prior arts in general object detection in the wild usually copy with object deformation [20]. These approaches are based on extracting discriminative and efficient features for general object detection, like Adaboost's face detector [21], histogram of oriented gradients (HOG) [22] or covariance features [23] for pedestrian detection. Additionally, the deformable part model (DPM) [24] has been proposed by accelerating coarse-to-fine search of possible locations. The object recognition methods like spatial pyramid matching (SPM) [25,26] are presented to detect objects with large deformations. These approaches are capable of handling roughly rigid objects, but not good in detecting tiny object instance with less deformable shapes. Towards tiny object detection, many hand-engineered features such as colour [27], shape [28], texture [29] and high-level features like SIFT [30] are widely used. But selecting feasible features is laborious and insufficient to represent all aspects of the targeted tiny objects. In this work, we aim to solve the feasible feature selection for tiny object detection by an automatic way.

Deep Learning for Object Detection The emergence of deep learning-based computer vision approaches become to be of great benefit to modern image classification and object detection tasks. Neural network proves to own the superior capacities on extracting powerful enough feature representation for 2D static images. Besides, CNN-based object detectors are also robust to learn invariant descriptors for various object categories [31]. For example, two-stage approaches [7,8] utilize dense sliding window on extracted feature maps from input image to search the potential object regions with low-level cues and then focus on these local regions for correct identification. During this region proposal process, proper criterion (also known as loss function) would also guide the CNN backbone to pay more attention on the object areas. During this region proposal process, proper criterion (also known as loss function) would also guide the CNN backbone to pay more attention on the object areas. Therefore, the end-to-end training strategy could bring automatic object detection capability [32]. On the other hand, considering objects with various sizes, SSD [6] and DSSD [33] perform to find instances at multilayers. In order to improve the detection accuracy, RetinaNet [34] introduces a novel loss function named focal loss to alleviate the imbalance risk of background–foreground samples. Besides, feature pyramid structure is also employed in RetinaNet to consider both large-scale and small-scale objects. Although these methods have shown great accuracy beyond what were achieved by traditional signal processing methods, they might be intractable for our targeted wild pest detection task due to some inherent difficulties in in-field environment in agriculture applications. Therefore, current object detectors require task-specific improvement for practical pest monitoring applications.

Tiny Object Detection Generally, there are two types of approaches in addressing tiny object detection task when employing deep learning techniques. The first one is that building novel tiny object specific modules into Faster R-CNN style architectures. These methods improve the general object detection models into tiny object detection task and make it work well. Gao et al. attempt to explore more effective methods on faster R-CNN and obtain 71.53 score on MS COCO tiny AP [35]. Gong et al. propose a novel concept named fusion factor to control information from the various layers in feature pyramid [36]. The other one is using coarse-to-fine strategy that first search the object cluster regions, then crop the clusters from the original image and detect tiny objects [37–40]. For tiny pest detection, Li et al. propose a tiny object detector that jointly connects two general detection models [41]. Besides, Du et al. present a novel cluster region proposal network and a density merging and partition module for a precise cluster finding. Compared with coarse-to-fine architectures, faster R-CNN style models train the object detection model by an end-to-end manner that can save a large amount of time consumption. On the other hand, coarse-to-fine methods usually work well on the scenes where tiny objects gather into group but not perform better on sparse object environment. Therefore, in this work, we follow the faster R-CNN style object detection architectures to present our method.

3 Materials and methods

3.1 Approach overview

We give a brief description on the proposed CNN-based deep learning architecture as shown in Fig. 3. In the first stage, it relies on the traditional CNN backbone, where our proposed new module GAM is aggregated on each convolutional block for screening and activating depth and spatial information from feature maps outputted by each block. The multi-scale image features extracted from GAM are used to rebuild the feature maps. This design has two considerations: (1) Sufficient shallow layers enable mining more valuable semantic features for classification. (2) The bottom layers with high spatial information are fully utilized for avoiding some features vanish in deep block.

In the second stage, similar to Fast R-CNN, we use RPN module connecting to each level so that regions were extracted in different spatial scales. Considering that most pests are tiny objects in the wild images and their distributions might be dense or sparse, smaller and denser anchors as well as high initial image resolution are significant for accurate pest localization. Thus, fully connected layers for classification and box regression are capable of allowing the pest regions being discriminated. Finally, HEE strategy is presented to help our method for fast and efficient training.

3.2 Global activated module (GAM)

The inspiration of GAM is under two key observations from sample tiny pest data. The first observation is that the pest is usually very tiny object with a size less 1% of whole images. In that case, through a deep learning architecture, they will become even smaller and insensitive with each pooling layer. Thus, we have to build up feature maps from CNN layers with lots of convolution kernels, in which each kernel is designed to extract the corresponding type of feature. As we know, the feature information of tiny pest might vanish in deeper CNN layers and convolutional operation own a defect of limited receptive field. From a feature pyramid perceptive, these feature maps from extracting these depth values, it is possible to detect some tiny objects with discriminated features.

The second observation from sample pest data is that there are large amount of variation of pest distribution and sizes in the image dataset. These factors can be categorized as a dimension of 'spatial'. Eventually, different kernel aims to detect different shapes of objects. Ideally, pests from the same category have shown the similar shapes in images because of tiny character, which are different from generic objects that being represented with great posture differences. Thus, by extracting the value of this spatial information, it is possible to reduce the influence of variation of pest distribution and sizes in training process. Motivated by these two observations, we propose a novel module global activated module (GAM) for activating the depth and spatial information after each output from CNN block to boost the representational power of feature maps. Figure 4 shows an intuitive framework of our GAM in the backbone.

The GAM consists of two parts. In the first part of depth activation module (the upper part), the 3D feature map with

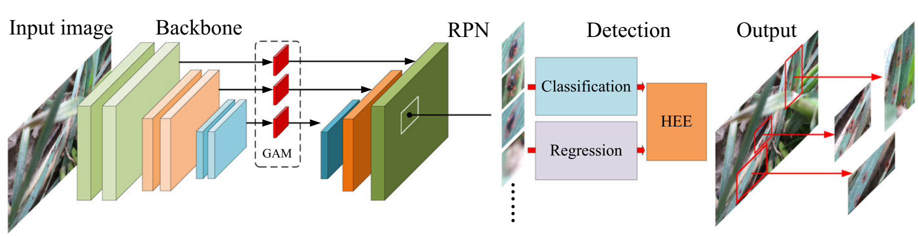


Fig. 3 The proposed architecture for tiny pest detection in the wild

extracted by CNN block is input into an extra global pooling layer which takes average from the whole feature maps in each channel to generate a lower-dimensional (1D) feature with shape of $1 \times 1 \times C$, in which the averaged value represents the global feature for every channel. Then, we apply two groups of convolutional layers with nonlinear activation ReLU following. The 1D feature vector is mapped into (0,1) area by adopting Sigmoid function so the output with shape of $1 \times 1 \times C$ is named as depth activation factors. The output of depth activation module is the broadcast element-wise product of the original input 3D feature map and the 1D depth activation factors. Finally, feature map is activated in depth level and the potential tiny object can be effectively detected in these activated channels.

The lower part in Fig. 4 is spatial activation module, whose operations are similar with depth activation module. Specifically, the spatial activation branch is a segmentation-like branch, in which the supervised mask is obtained by fulfilling 1 into the ground truth positions and 0 into the background areas. In this part, the input 3D feature map ($W \times H \times C$) is processed by an extra 'global convolution' operation containing a convolutional layer kernel with only 1 filter. The output is a 2D feature map with shape of ($W \times H \times 1$), which could ensure the spatial activation vector is learned in spatial level by a supervised auxiliary loss. The spatial activation factors are generated through two convolutional layers with a nonlinear activation ReLU and Sigmoid function. At last, the learned spatial activation factor is fed into exponential operation and then dot with the input 3D feature maps in each position rather than naïve multiplication. In this way, it could maintain more context information while highlight the object information. In this way, our spatial activation could enhance the feature maps in pest objects area and diminish the opposition. Finally, the output of GAM is the sum of two activated feature maps, and the feature pyramid will be built upon the output of GAM.

3.3 Loss function

Loss function is the criterion for training process in deep learning approaches. Our loss function is defined as the sum of the cross-entropy loss and the box regression loss:

$$
\begin{aligned}
L(s,t) &= \sum_i l(s,t) \\
&= \sum_i -\log(s_{c_i^*}) + \lambda[c_i^* > 0]\text{smooth}_{L1}(t_i - t_i^*)
\end{aligned} \tag{1}
$$

where $s_{c_i^*}$ denotes the predicted score class c_i^* while t_i and t_i^* denote $\{t_x, t_y, t_w, t_h\}$ of bounding boxes. $\lambda[c_i^* > 0]$ indicates that we only consider the boxes of non-background (if $c_i^* = 0$ the box is background). This loss function is the sum of the classification loss and the box regression loss.

3.4 Hard example enhancement (HEE)

Considering that our targeted pest dataset is large-scale and contains multiple species of insect, it is necessary to design a strategy for speeding up the training process in the deep learning architecture. Motivated by the idea of online hard example mining (OHEM) [42], we design a hard example enhancement strategy (HEE) for mining hard examples of current mini-batch during the training. Usually there is only one image in each mini-batch, the role of HEE is to enhance the weights of hard examples during training for quickly reaching convergence and saving the iteration numbers.

In the HEE strategy, we proposed a modulating factor α to strengthen the neural network to fit more on hard examples. The loss function is reshaped to up-weight these hard examples:

$$L^*(s,t) = \alpha L(s,t) = \alpha \sum_i l(s,t) \tag{2}$$

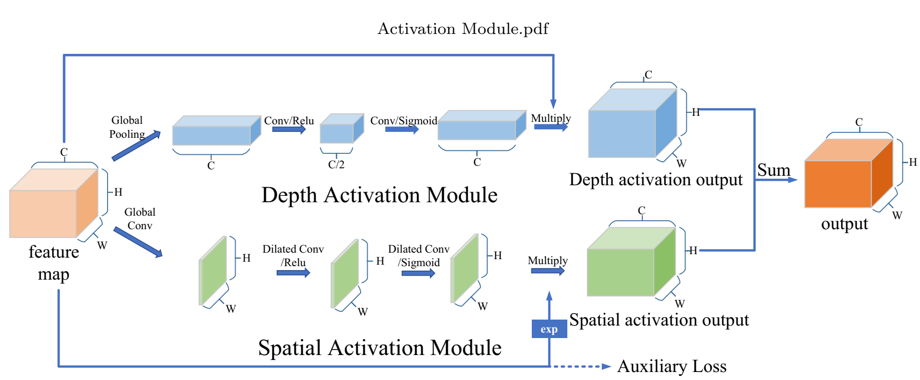


Fig. 4 Illustration of global activated module (GAM) structure. For each feature level from standard FPN, we build two parallel branches to extract different types of activation information separately. The depth activation vector is trained by unsupervised learning while the spatial activation matrix is trained by an auxiliary loss. Final feature map is the sum of these two branches' outputs

in which α_i denotes the weighting factor of the ith example in the last training epoch:

$$\alpha_i{}^{(p)} = 1 + \frac{l^{(p-1)}(s,t)}{\frac{1}{N}L^{(p-1)}(s,t)} \tag{3}$$

where superscripts p and $p-1$ represent the pth and $p-1$th training epoch. Intuitively, the weighting factor α_i enhances the loss contribution from hard examples and weakens it from easy ones. This factor will help our method fast and efficient convergence. The second stage of HEE is to reduce the number of training iterations for easy examples. During training, easy examples are learned well by network at early epoch so the loss would be close to 0 at the next few epochs. In this case, the network learns only few things since the gradient is becoming to almost a value of 0, in which results in wasteful learning. In order to remove these easy examples epoch by epoch, we first perform a warm training phase on a small part of training set so that our network is feasible to the new task. In the $p-1$th training epoch, we compute the losses of all the N training examples and remove the 10% of those with the smallest losses and the network is trained based on the rest examples. After 5 epochs, the amount of training samples is 0.5N that is only half of the original training set. Thus, we restart training with the original examples again to guarantee that our network is still able to fit the easy samples. In this case, the examples with small losses and gradients are deleted to save computation cost. Combined with the modulating factor α in loss function, the hard examples are better fully learned with less training time. It makes our model more fit more to large-scale dataset.

4 Dataset set-up of tiny pest in the wild

To our best knowledge, while there exist some open insect databases released such as IP102 [43], no existing large-scale datasets that cover multi-class pests in the wild or nature environments are released for study yet. Thus, we attempt to establish a research-based large-scale dataset for tiny insect detection in the wild.

Our dataset is built up from our previous industrial-oriented projects over 7 years containing 27,905 images with 145,672 manual labelled pests in 6 classes. The images are captured in wheat and rice field. For image acquisition, we use CCD camera with 4 mm focal length with an aperture of f/3.3 to make sure that the captured images are high resolution (1440 × 1080).

Then the images are annotated with several labels as well as bounding boxes by agricultural experts to guarantee the professionalism of these annotations. The statistical details of our dataset are shown in Table 1. Note that the sizes of pests would not be at most 1.5% at the whole image because we aim to detect tiny pest. Moreover, to investigate the environmental impact factors, we first manually split the validation subset into 4 typical challenges including dense distribution, sparse distribution, sunlight illumination influence and large background noise, as shown in Table 2. Some typical samples of our dataset are shown in Fig. 5.

5 Experimental results and discussion

5.1 Experiment set-up

On our dataset, we build some experiments to evaluate the performance of our approach. We compare the performance of our method with two types of object detection methods: generic detection methods and tiny object detection methods. The generic detection methods we choose are Faster R-CNN [7], RetinaNet [34], Yolov3 [11] and FCOS [44], which are representative works for two-stage detector, one-state detector, high-speed detector and anchor-free detector, respectively. The tiny object detection methods are ClusDet [39], DMNet [40]. DCTDet [45] and Li et al.'s method [41], in which ClusDet and DMNet are designed for tiny objects while DCTDet and Li et al's method focus on tiny pest detection in agriculture. All convolutions in CNN architectures are performed in a unit consisting of convolution layer, batch normalization [46] and ReLU. Besides, we double the depth of shallow layers and halve those in deep layers to ensure the extraction of sufficient semantic information before the position information disappears. In the second stage, RPN architecture is referenced by Faster R-CNN and we set the half of original anchor size because pests to be detected are tiny objects so most of our region proposals are provided by RPN in shallow blocks. We implemented our codes on PyTorch framework and run on two GeForce GTX TITAN X GPUs. The momentum SGD [47] is chosen as our optimizer with momentum 0.9 and mini-batch size is 2. As to learning rate policy, 'step' strategy is applied in gradient descent, in which we initialize learning rate to 0.001 and the learning rate will be divided by 10 per 1 epoch.

In terms of model evaluation, Precision, Recall and Average Precision (AP) [12] are three major evaluation metrics for this pest detection task. Specifically, given the IoU (Intersection over Union) threshold, we could decide the detected bounding box is correct or not. Based on the judgement, the Precision and Recall for the input image are defined as:

$$\text{Precision} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FP}} \tag{4}$$

$$\text{Recall} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}} \tag{5}$$

Table 1 Statistics on two subsets of our dataset with training subset and validation subset

Pest name	size (%)	Training		Validation		All	
		#images	#objects	#images	#objects	#images	#objects
WM	0.089	11,505	54,423	1278	6095	12,783	60,518
SA	0.086	5230	26,385	588	2844	5818	29,229
SG	0.075	5997	34,262	656	3819	6653	38,081
RP	0.100	697	1634	72	133	769	1767
SW	1.512	2901	2980	303	320	3,204	3300
RPH	0.148	1084	11,352	121	1425	1205	12,777
Total		25,114	131,036	2791	14,636	27,905	145,672

For each class, the number of images (containing at least one insect the class), the number of objects is shown in this table. Note that because single image may contain objects of several classes, the totals shown in the '#images' columns are not simply the sum of the corresponding columns. (WM: Wheat mite, SA: Sitobion avenae, SG: Schizaphis graminum, RP: Rhopalosiphum padi, SW: Sticky worm, RPH: Rice planthopper)

Table 2 Statistic on validation subset of our dataset split in 4 typical challenges

Pest name	Dense		Sparse		Sunlight		Noise	
	#images	#objects	#images	#objects	#images	#objects	#images	#objects
WM	178	1745	1064	4205	214	760	56	324
SA	236	2036	241	632	211	935	31	121
SG	210	2296	310	1199	209	1014	69	700
RP	17	46	55	87	25	49	15	27
SW	–	–	–	–	105	109	236	253
RPH	34	694	53	548	5	34	119	1396
Total	583	694	1627	548	708	2901	499	2821

Note that single image may contain more than one type of challenges, the totals shown in the '#images' columns are not simply the sum of the corresponding columns. Besides, the pest SW of class #5 is not concerned to distribution challenges both dense and sparse since most of their images contain only 1 pest

Table 3 Multiclass tiny pest detection results comparison

Method	Backbone	Pest name							
		WM	SA	SG	RP	SW	RPH	mean	s/img(GPU)
Generic detection methods									
Faster R-CNN [7]	ResNet50	61.63	51.26	53.38	74.74	89.89	66.18	66.18	0.039
Faster R-CNN [7]	ResNet50+FPN	63.20	57.11	58.16	79.59	90.10	66.22	69.06	0.065
RetinaNet [34]	ResNet50+FPN	57.61	51.85	51.89	75.65	88.72	63.52	64.87	0.053
Yolov3 [11]	DartkNet53	57.58	50.95	51.47	74.89	89.01	64.38	64.71	0.024
FCOS [44]	ResNet50+FPN	61.89	52.33	54.08	76.98	90.07	68.35	67.28	0.048
Tiny object detection methods									
ClusDet [39]	ResNet50+FPN	63.98	58.62	58.96	83.20	90.12	68.37	70.54	0.218
DMNet [40]	ResNet50+FPN	65.46	60.25	61.24	85.10	90.12	69.85	72.00	0.268
DCTDet [45]	ResNet50+FPN	66.38	61.14	61.56	85.08	90.10	70.49	72.44	0.289
Li et al. [41]	ResNet50+FPN+DarkNet53	68.06	63.98	63.87	86.47	90.07	72.48	74.16	0.396
Our method									
Ours	Inception+GAM	60.01	58.71	54.29	76.84	88.18	60.87	66.48	0.037
Ours	Inception+GaFPN	61.50	60.14	57.15	79.35	88.53	63.06	68.29	0.054
Ours	ResNet50+GAM	63.96	61.51	57.86	85.14	90.07	68.17	71.07	0.041
Ours	ResNet50+GaFPN	64.75	61.87	59.30	85.34	90.07	69.60	71.82	0.069

The AP is computed over IoU 0.5

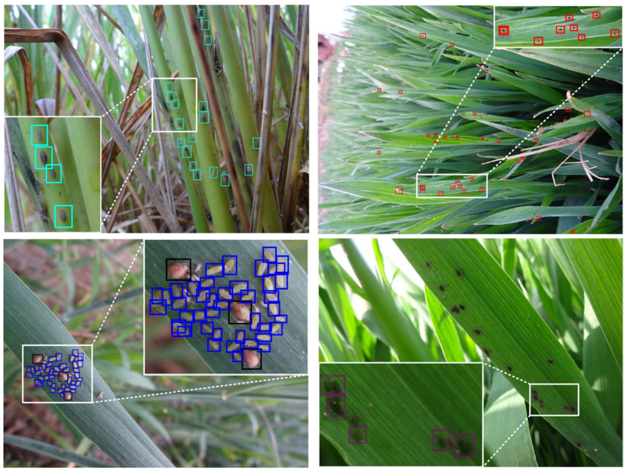


Fig. 5 Typical samples in our dataset

where the #TP, #FP and #FN indicate the number of true positives, false positives and false negatives. Therefore, Precision measures the detection accuracy of the input pest images while a higher Recall points out lower misdetections. In addition to IoU threshold, we can also obtain various couples of Precision and Recall by selecting different confident threshold during pest detection. In this way, Precision–Recall curve could be defined with these sequential Precision and Recall sets, in which we introduce AP as a high-level evaluation metric for our pest detection task that is defined as the integration of Precision–Recall curve:

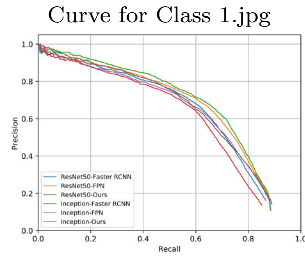
$$AP = \int_0^1 \text{Precision} d \text{Recall} \tag{6}$$

Finally, we average the AP values in all the pest categories K to obtain mean AP (mAP) as the evaluation metric for the whole dataset in our work:

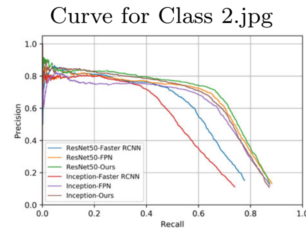
$$mAP = \frac{1}{K} \sum\nolimits_{c=1}^{K} AP(c) \tag{7}$$

5.2 Wild tiny pest detection results

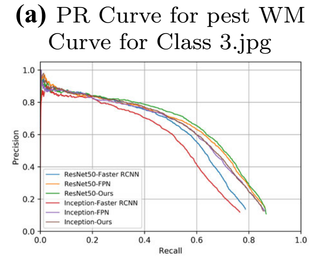
Table 3 presents the final wild multiclass tiny pest detection results. We compare our method with two types of object detection approaches. The first one is generic detection methods. Table 3 shows that our method with the proposed GAM module and HEE training strategy can achieve much higher mAP than the current generic detection models, which obtain 5.64, 2.76, 6.95, 7.11, 4.54 improvement compared to the baselines. This is because object features of tiny pest become vanishing in deep layers so that it is hard for Faster R-CNN to find them. Among all of the approaches, the best detection performance occurs in our method using ResNet50 as backbone which achieves mAP with 71.82%. The second one is



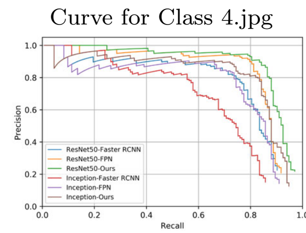
(a) PR Curve for pest WM



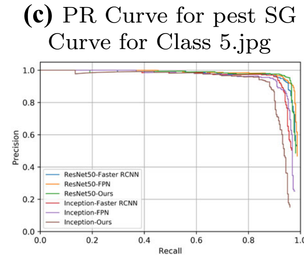
(b) PR Curve for pest SA



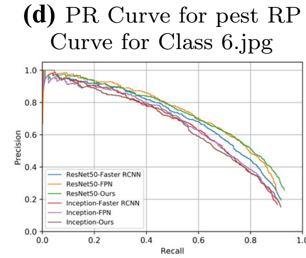
(c) PR Curve for pest SG



(d) PR Curve for pest RP



(e) PR Curve for pest SW



(f) PR Curve for pest RPH

Fig. 6 Precision-recall curve for classes with our method compared with other state-of-the-art methods with different CNN backbones

tiny object detection methods. As it can be seen, coarse-to-fine style approaches seem to be more suitable for tiny pest detection task, with 70.54, 72.00, 72.44 and 74.16 mAP performance. However, although our method might be slightly worse than these methods, we can obtain much higher inference speed compared with them. In detail, our method can spend only 0.069 seconds for one image. Overall, our method perform better than the other object detection approaches.

There is an obvious difference with performance over different types of pests. Specifically, pest SW owns the most obvious feature that all the models could discriminate and localize it well while SA and SG are two difficult species to detect with lower AP. This might be explained by that pest SW holds the largest size in images (1.51%) compared to other pests whose sizes are much smaller (less than 1%), which improves difficulty to localize them. Similarly, pest RPH also is detected well in GaFPN due to its relatively large size (0.148%).

Data volume is another significant factor affecting model performance. Our method obtains higher detection accuracy on pest of class WM who holds huge number of training examples with more than 64.75% AP even it is one of the tiny objects. In spite of the limited number of training examples, pest RP seems to be simply found in our dataset with a high precision of 85.34% because its large difference in colour and surroundings and our method could dramatically improve localization precision of this pest compared with other methods.

5.3 Precision–recall analysis

As for Precision–Recall (PR) analysis, we evaluate the PR curves comparing our method to Faster R-CNN and FPN in six classes shown in Fig. 6. It indicates that precision keeps a high value with the recall increasing. Our method with ResNet50 and Inception obtains a greater precision and recall compared with other architectures. It means our method effectively reduces false positives rate and misdetections rate. But pest SA is relatively difficult to attain a high precision while maintaining recall in FPN method, in which the PR curve starts to drop drastically at the recall of 0.6, but our method keeps a little bit more.

Yet, our method does not get a significant improvement in pest SW. It is probably because Faster R-CNN could categorize and localize it well. PR curves for class SA and RP represent that it is hard to obtain a high recall value but could get satisfied precision so these two curves signify. Our method could make sure that almost all the detected pests of these two classes are correct. It improves most on class #4 that obtains much higher recall than FPN methods.

5.4 Challenge analysis

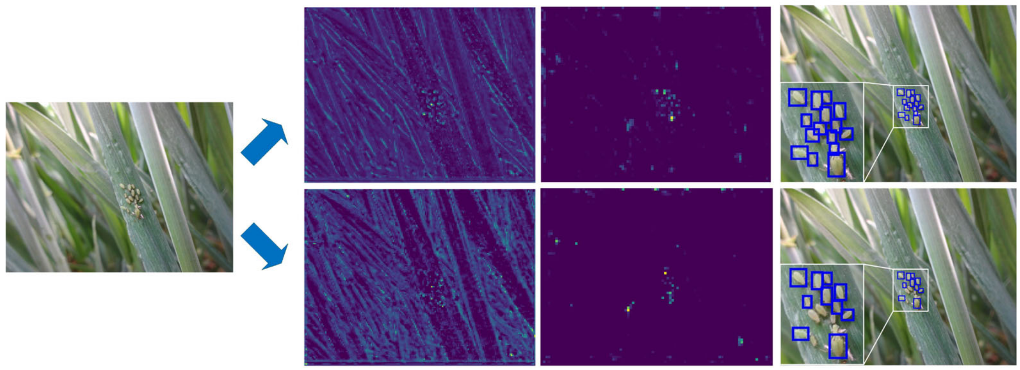
Table 4 illustrates the proposed approach performs with four typical challenges dense, sparse, sunlight and noise compared to approaches. Note that the $\mathrm{mAP}_{\mathrm{dense}}$ and $\mathrm{mAP}_{\mathrm{sparse}}$ are computed on 5 classes excluding pest SW while others are evaluated on 6 classes. The results show that our method using ResNet50 outperform Faster R-CNN and other generic detection methods on our pest detection in 4 typical challenges. For the most influential factor 'background noise,' our method has an improved mAP over 6% to Faster R-CNN. This shows that GAM module effectively helps to filter the background noise in spatial level. In addition, under the denser anchor boxes employed in our method, pests with dense distribution is better detected.

For visualizing the effect of GAM module in our method, we generate feature maps with four challenges outputted by our method and FPN using ResNet50 backbone in Fig. 7. Note that only parts of feature maps are presented here due to the limited space and these feature maps are extracted from shallow block to deep block. The first observation is that feature maps in shallow block are strongly influenced by background, in which those in FPN become to be more severe while our method can filter out the environment to some extent. Second, with the layer going deeper, the pest in feature maps is learned better and lighter points appear in potential pest locations. It is found that the feature maps in GaFPN diminish the highlights of non-objects and focus more attention on pest regions with lighter activation points. Specifically, in Fig. 7b and d, FPN cannot effectively detect

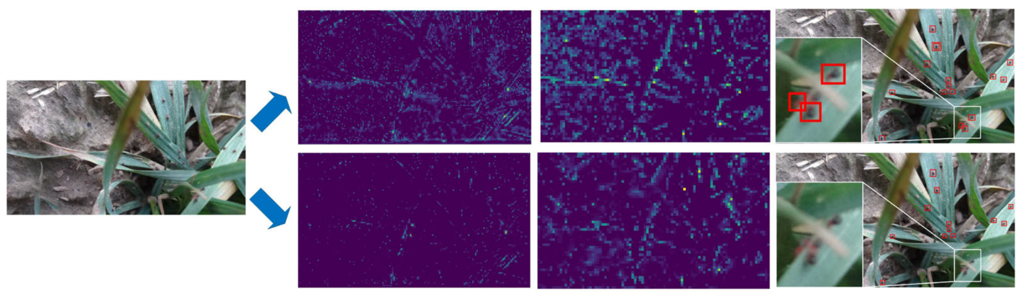
Table 4 Detection results comparison on 4 typical challenges

Method	Backbone	$\mathrm{mAP}_{\mathrm{dense}}$	$\mathrm{mAP}_{\mathrm{sparse}}$	$\mathrm{mAP}_{\mathrm{sunlight}}$	$\mathrm{mAP}_{\mathrm{noise}}$
Generic detection methods					
Faster R-CNN [7]	ResNet50	59.76	64.40	62.93	59.29
Faster R-CNN [7]	ResNet50+FPN	64.68	65.88	66.07	61.47
RetinaNet [34]	ResNet50+FPN	58.03	56.74	61.17	57.84
Yolov3 [11]	DartNet53	57.48	55.84	59.42	55.49
FCOS [44]	ResNet50+FPN	61.94	61.99	63.87	58.46
Tiny object detection methods					
ClusDet [39]	ResNet50+FPN	65.84	58.96	70.62	60.04
DMNet [40]	ResNet50+FPN	68.02	61.75	72.63	61.29
DCTDet [45]	ResNet50+FPN	69.85	63.48	74.36	63.92
Li et al. [41]	ResNet50+FPN+DarkNet53	70.64	66.97	76.32	65.15
Our method					
Ours	Inception+GAM	61.37	63.71	65.49	60.27
Ours	Inception+GaFPN	64.34	64.87	66.91	62.60
Ours	ResNet50+GAM	64.81	65.94	69.19	63.08
Ours	ResNet50+GaFPN	66.75	68.58	70.53	65.26

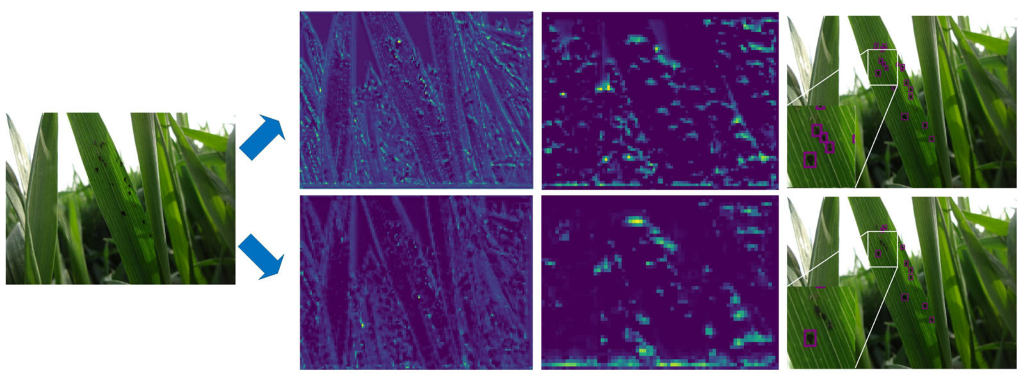
The AP is computed over IoU 0.5



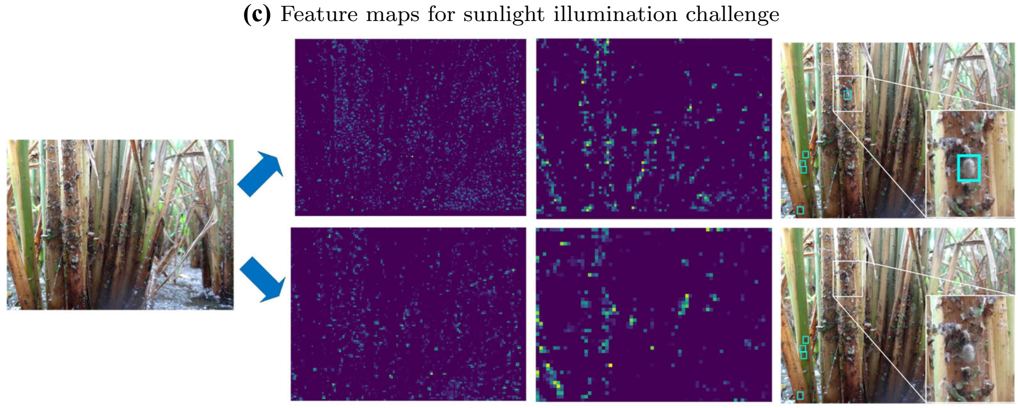
(a) Feature maps for dense distribution challenge



(b) Feature maps for sparse distribution challenge



(c) Feature maps for sunlight illumination challenge



(d) Feature maps for background noise challenge

Fig. 7 Comparison of feature visualization for four major challenges in our dataset. The first rows represent feature maps outputted from 3 blocks (shallow to deep) of our method with ResNet50 backbone while the second rows are those from FPN with ResNet50. Light points in these feature maps indicate the potential pest locations predicted by network

Table 5 Ablation study on our method

Method	Backbone	GAM	HEE	mAP
Ours	ResNet50+FPN			69.06
Ours	ResNet50+FPN	✓		71.16
Ours	ResNet50+FPN		✓	70.48
Ours	ResNet50+FPN	✓	✓	71.82

We investigate the effect of GAM module and HEE training strategy

Table 6 Quantitative results on the VisDrone and TinyPerson benchmark

Benchmark	Method	Backbone	AP	s/img(GPU)
VisDrone [48]	ClusDet [39]	ResNeXt101 [50]+FPN	53.2	0.234
	RRNet [51]	Stacked Hourglass [52]	55.8	0.327
	DMNet [40]	ResNeXt101 [50]+FPN	49.3	0.294
	DCTDet [45]	ResNeXt101 [50]+FPN	56.6	0.316
	Ours	ResNeXt101 [50]+GaFPN	46.7	0.096
TinyPerson [49]	FCOS [44]	ResNet50+FPN	29.25	0.042
	RetinaNet [34]	ResNet50+FPN	43.32	0.055
	Adaptive FreeAnchor [53]	ResNet50+FPN	51.23	0.051
	Faster R-CNN [7]	ResNet50+FPN	53.48	0.060
	Ours	ResNet50+GaFPN	56.04	0.067

most of the pests when influenced by sparse distribution and large background noise. Although our method with GAM might light up lots of noises, it could maintain pest information while FPN become losing some pest locations. This is in line with the conclusion that GaFPN is more able to fit tiny targets pest and leads to better detection performance. The final detection demonstration results are shown in the right columns. Our method achieves multiclass tiny pest detection in wild scene under various distributions of both dense and sparse in Fig. 7a and b. It also overcomes the challenges with large sunlight shadow and background noise influence and obtain a better detection performance compared to FPN in Fig. 7c and d.

5.5 Ablation study

The key idea of our method is to develop a global activated module for high-quality feature extraction as well as a HEE training strategy. To further investigate the effect of the proposed modules, we present the ablation study in Table 5. As it can be observed, the GAM module helps a lot on our method to accurately detect tiny pests, which achieves 2.10 mAP improvement compared with the baseline that does not employ GAM module. On the other hand, HEE can help our model to quickly find the hard examples and enhance the training values of our training data. On the experiments, within HEE, our method can get further 0.66 mAP improvement. Therefore, we can conclude that our GAM and HEE are both useful component for our method in tiny pest detection task.

5.6 Generalization capacity

Our method is not picky about the type of objects being detected. Thus, we also validate the detection performance of our method on the other popular tiny object detection benchmarks. Here we choose Visdrone [48] and TinyPerson [49] as validation datasets. We compare our method on the baselines provided from original Visdrone competition leaderboard and TinyPerson article. The experimental results are shown in Table 6. As it could be seen, on Visdrone benchmark, our methods hold the advantage of high inference speed for tiny object detection with only 0.096 seconds for one image. In addition, on another tiny object benchmark TinyPerson, we could also find that within GAM and HEE modules contribute to an obvious improvement on AP metric. This could conclude that our method shows to be a good way to detect objects in a finer level.

5.7 Limitations and improvements

Despite that our approach shows better performance than other popular object detectors, there are two limitations for future study. One is that our method gets the lowest mAP when the input image is in noisy background. It is probably because the appearance of noise factor is very similar to pest regarding the size or shape. It is hard to distinguish them from spatial or depth attention extracted from feature pyramid. The other point is that our method may be suitable to some certain tiny object detection case in the wild, with properties of severe clustering or overlapping distribution. Future work will lie in discovering a finer feature extraction and detection architecture and also employ the tiny pest detection models into real-world applications.

6 Conclusion

We have presented a global activated feature pyramid network (GaFPN)- based CNN approach for tiny pest detection in wild scene. Our approach extracts depth and spatial information from feature pyramids of pest images to make discriminative features of tiny pests more activated in deep learning architecture. We evaluate the approach in large-scale wild tiny pest dataset over 27k images. The results show that our approach delivers an average 71% accuracy and outweighs two state-of-the-art methods. In future we will extend this approach in more general tiny object detection cases in the wild.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 31401293, and in part by the Major Special Science and Technology Project of Anhui Province under Grant 201903a06020006.

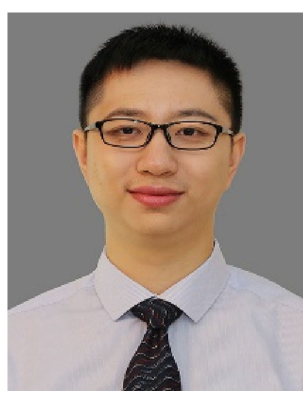
Declaration

Conflicts of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

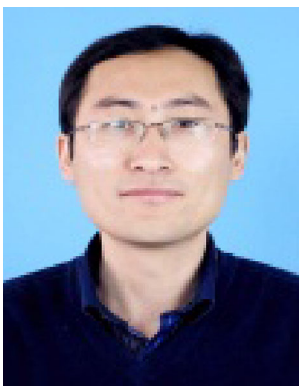
1. Bai, Y., Zhang, Y., Ding, M., Ghanem, B.: Finding tiny faces in the wild with generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 21–30 (2018)
2. Philipsen, M.P., Jensen, M.B., Møgelmose, A., Moeslund, T.B., Trivedi, M.M.: Traffic light detection: A learning algorithm and evaluations on challenging dataset. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, IEEE, pp. 2341–2345 (2015)
3. Ma, Z., Yu, L., Chan, A.B.: Small instance detection by integer programming on object density maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3689–3697 (2015)
4. Wadhai, M., Gohokar, V., Khaparde, A.: Agriculture pest detection using video processing technique. In: 2015 International Conference on Information Processing (ICIP), IEEE, pp. 544–547 (2015)
5. Sun, J., He, X., Wu, M., Wu, X., Shen, J., Lu, B.: Detection of tomato organs based on convolutional neural network under the overlap and occlusion backgrounds. Mach. Vis. Appl. **31**(5), 31 (2020)
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision, Springer, pp. 21–37 (2016)
7. Ren ,S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
8. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
10. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
11. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018)
12. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vision **88**(2), 303–338 (2010)
13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer, pp. 740–755 (2014)
14. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
15. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
16. Atanbori, J., French, A.P., Pridmore, T.P.: Towards infield, live plant phenotyping using a reduced-parameter cnn. Mach. Vis. Appl. **31**(1), 2 (2020)
17. Kumar, J.P., Dominic, S.: Rosette plant segmentation with leaf count using orthogonal transform and deep convolutional neural network. Mach. Vis. Appl. **31**(1), 6 (2020)
18. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp. 3–19 (2018)
19. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)
20. Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 17–24 (2013)
21. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vision **57**(2), 137–154 (2004)
22. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE, vol. 1, pp. 886–893 (2005)
23. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 1–8 (2007)
24. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 1–8 (2008)
25. Van de Sande, K.E., Uijlings, J.R., Gevers, T., Smeulders, A.W.: Segmentation as selective search for object recognition. In: 2011 International Conference on Computer Vision, IEEE, pp. 1879–1886 (2011)
26. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)

27. Zhu, L.Q., Zhang, Z.: Auto-classification of insect images based on color histogram and glcm. In: 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, vol. 6, pp. 2589–2593 (2010)
28. Yaakob, S.N., Jain, L.: An insect classification analysis based on shape features using quality threshold artmap and moment invariant. Appl. Intell. **37**(1), 12–30 (2012)
29. Qing, Y., Jun, L., Qj, L., Gq, D., Bj, Y., Hm, C., Jian, T.: An insect imaging system to automate rice light-trap pest identification. J. Integr. Agric. **11**(6), 978–985 (2012)
30. Solis-Sánchez, L.O., Castañeda-Miranda, R., García-Escalante, J.J., Torres-Pacheco, I., Guevara-González, R.G., Castañeda-Miranda, C.L., Alaniz-Lumbreras, P.D.: Scale invariant feature approach for insect monitoring. Comput. Electron. Agric. **75**(1), 92–99 (2011)
31. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., et al.: Large scale distributed deep networks. In: Advances in Neural Information Processing Systems, pp. 1223–1231 (2012)
32. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
33. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
34. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
35. Gao, C., Tang, W., Jin, L., Jun, Y.: Exploring effective methods to improve the performance of tiny object detection. In: European Conference on Computer Vision, Springer, pp. 331–336 (2020)
36. Gong, Y., Yu, X., Ding, Y., Peng, X., Zhao, J., Han, Z.: Effective fusion factor in fpn for tiny object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1160–1168 (2021)
37. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 951–959 (2017)
38. Pang, J., Li, C., Shi, J., Xu, Z., Feng, H.: R2-cnn: Fast tiny object detection in large-scale remote sensing images. arXiv preprint arXiv:1902.06042 (2019)
39. Yang, F., Fan, H., Chu, P., Blasch, E., Ling, H.: Clustered object detection in aerial images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8311–8320 (2019)
40. Li, C., Yang, T., Zhu, S., Chen, C., Guan, S.: Density map guided object detection in aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 190–191 (2020)
41. Li, R., Wang, R., Xie, C., Liu, L., Zhang, J., Wang, F., Liu, W.: A coarse-to-fine network for aphid recognition and detection in the field. Biosys. Eng. **187**, 39–52 (2019)
42. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–769 (2016)
43. Wu, X., Zhan, C., Lai, Y.K., Cheng, M.M., Yang, J.: Ip102: A large-scale benchmark dataset for insect pest recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
44. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
45. Du, J., Liu, L., Li, R., Jiao, L., Xie, C., Wang, R.: Towards densely clustered tiny pest detection in the wild environment. Neurocomputing (2022)
46. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
47. Bottou, L.: Stochastic gradient descent tricks. In: Neural networks: Tricks of the trade, Springer, pp. 421–436 (2012)
48. Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., Nie, Q., Cheng, H., Liu, C., Liu, X., et al.: Visdrone-det2018: The vision meets drone object detection in image challenge results. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)
49. Yu, X., Gong, Y., Jiang, N., Ye, Q., Han, Z.: Scale match for tiny person detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1257–1265 (2020)
50. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 492–1500 (2017)
51. Chen, C., Zhang, Y., Lv, Q., Wei, S., Wang, X., Sun, X., Dong, J.: Rrnet: A hybrid detector for object detection in drone-captured images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
52. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision, Springer, pp. 483–499 (2016)
53. Zhang, X., Wan, F., Liu, C., Ji, R., Ye, Q.: Freeanchor: Learning to match anchors for visual object detection. Adv. Neural Inf Process Syst 32 (2019)



Liu Liu received the B.E. degree in information engineering aerospace information applications from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015, and M.S. degree in advanced computer science from University of Manchester, Manchester, UK, in 2016. He received the Ph.D. degree in pattern recognition and intelligent system with University of Science and Technology of China, Hefei. His current interest is deep learning and computer vision.

Rujing Wang received the B.E. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1987, and M.S. degree in electronic engineering from Dalian University of Technology, Dalian, China, in 1990, and Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 2005. He is currently with the Institute of Intelligent Machinery of the Chinese Academy of Sciences as Professor and Researcher. His main research interests include intelligent agriculture, agricultural internet of things, agricultural knowledge engineering.



Chengjun Xie received the M.S. degree in software engineering from the Hefei University of Technology, Hefei, China, in 2008, and Ph.D. degree in image processing from in the Hefei University of Technology, Anhui, China, in 2011. He is currently working in the Institute of Intelligent Machinery of the Chinese Academy of Sciences as Associate Researcher. His research interests include crop disease and pest image recognition, agricultural big data, and agricultural Internet of Things.

Rui Li received the B.E. degree from the Hebei University of Architecture, China, in 2010, and the M.S. degree in computer applied technology from the Hefei University of Technology, Hefei, China, in 2013. He is currently pursuing the Ph.D. degree in electronic information with the University of Science and Technology of China, Hefei. His current research interests include deep learning and computer vision.

Fangyuan Wang received the B.E. and M.S. degrees in electrical engineering and automation from Hefei University of Technology, Hefei, China, in 2014 and 2017, respectively. He received the Ph.D. degree in pattern recognition and intelligent system with University of science and technology of China, Hefei. His current interest is deep learning and computer vision.

Long Qi received the M.Sc. and Ph.D. degrees from Jilin University, in 2006 and 2009, respectively. He is currently a Professor with the College of Engineering, South China Agricultural University. His research interests include image processing, machine vision, deep learning, simulation, and their applications in agriculture. He has published more than 80 articles and holds numerous patents in his research areas.