BIOSYSTEMS ENGINEERING 213 (2022) 148–161

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

A multi-branch convolutional neural network with density map for aphid counting

Rui Li [a,c], Rujing Wang [a,b,c], Chengjun Xie [a,c,*], Hongbo Chen [a,b,c,**], Qi Long [d], Liu Liu [e], Jie Zhang [a,c], Tianjiao Chen [a,b,c], Haiying Hu [a,c], Lin Jiao [a,b,c], Jianming Du [a,c], Haiyun Liu [a,b,c]

[a] Institute of Intelligent Machines, and Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China
[b] University of Science and Technology of China, Hefei 230026, China
[c] Intelligent Agriculture Engineering Laboratory of Anhui Province, China
[d] College of Engineering, South China Agricultural University, China
[e] Department of Computer Science and Engineering Shanghai Jiao Tong University, China

ARTICLE INFO

Article history:
Received 5 July 2021
Received in revised form
12 October 2021
Accepted 23 November 2021

Keywords:
Density map
Aphid counting
Convolutional neural network
Deep learning

In agriculture, aphids always cause major damage in wheat, corn and rape, which significantly affect the crop yield. Manual aphid counting approaches are often labour-consuming and time-costing for Integrated Pest Management (IPM). In addition, the results of existing aphid counting methods based on computer vision are not satisfactory due to the complex background and the dense distribution. In order to address these problems, a novel multi-branch convolutional neural network (Mb-CNN) with density map for aphid counting is developed in this paper. In this approach, the aphid images are firstly fed into multi-branch convolutional neural networks, which have three branches for extracting the feature maps of different scales. Then, an aphid density map is generated via Mb-CNN, which contains the distribution information of aphids. Finally, the counting of aphids is estimated by using the density map. Experiment results on our dataset demonstrate that our Mb-CNN achieves the performance of 10.22 Mean Absolute Error (MAE) and 12.24 Mean Squared Error (MSE) in the aphid counting, which outweighs the state-of-the-art approaches.

* *Corresponding author*. Institute of Intelligent Machines, and Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China.
** *Corresponding author*. Institute of Intelligent Machines, and Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China.
E-mail addresses: rjwang@iim.ac.cn (R. Wang), cjxie@iim.ac.cn (C. Xie), hbchen@iim.ac.cn (H. Chen), tjchen@iim.ac.cn (T. Chen).
https://doi.org/10.1016/j.biosystemseng.2021.11.020

Nomenclature

Abbreviations

Mb-CNN	multi-branch convolutional neural network
IPM	integrated pest management
CNN	convolutional neural network
MAE	Mean Absolute Error
MSE	mean squared error
mAP	mean average precision
Faster R-CNN	faster R-CNN: towards real-time object detection with region proposal networks
FPN	feature pyramid networks for object detection
YOLO	you only look once: unified, real-time object detection
Cascade R-CNN	cascade R-CNN: delving into high quality object detection
MCNN	multi-column convolutional neural network
FCOS	fully convolutional one-stage object detection
CRNet	crowd counting via cross-stage refinement networks
RPSN	rice planthopper search network
D2C	decoupled two-stage crowd counting and beyond

1. Introduction

Aphids cause damage and lower agricultural yields in wheat, rape and corn. They can build to high population density and wither plants by absorbing their sap. Since the aphids are tiny and clustered, the manual counting is very time-costing and labour-consuming, which affects the investigation efficiency of Integrated Pest Management (IPM) in the field. With the development of computer vision, many scientists used methods of digital image processing to identify and count aphids. Since automatic aphid counting can reduce labour intensity and improve work efficiency.

The premise of the automatic pest counting estimation is accurate detecting. There are a lot of research papers related to pest detection. Elison et al. (2020) presented a method of automatic aphid counting and classification by using machine learning methods. The experimental results showed that this method was reliable and useful to aphid population monitoring studies. Although their methods achieved good results, their monitoring equipment was fixed and could only detect specific pests in a small region. Furthermore, these methods were not suitable for Integrated Pest Management (IPM) due to the limited regions. Thus, agricultural experts could not get an accurate counting of pests from real farm environments. To tackle this issue, Yao et al. (2014) developed a handheld device for easily capturing planthopper images on rice stems and proposed an automatic method for counting rice planthoppers based on image processing. They achieved 85.2% detection rate and 9.6% false detection rate for counting white-back planthoppers. Maharlooei et al. (2017) used image processing techniques to detect and count multi-sized soybean aphids on a soybean leaf. The results showed that images captured with an inexpensive regular digital camera could also provide satisfactory results under high illumination conditions. Liu et al. (2016) developed a method of aphid identification and population monitoring based on digital images. Their method achieved the performance of 86.81% mean identification rate and 8.91% error rates. Although all the above methods achieved satisfied performance, they mainly focused on hand-crafted feature extractors and classifiers, which suffered from several limitations. In the practical application, the complexity of image background, illumination, scales and arbitrary direction are the major challenges in feature extractors and classifiers.

With the domination of convolutional neural network (CNN) and the evidence that feature learning approaches usually perform better than traditional techniques in many computer vision applications, object-detection has witnessed a quantum leap in the performance on benchmark datasets (Lecun et al., 2015). Lots of interest has been shown to the deep learning architectures have emerged, such like Faster-RCNN (Ren et al., 2015), you only look once (YOLO) (Redmon et al., 2016), Feature Pyramid Network (FPN) (Lin et al., 2017), Cascade R-CNN (Cai et al., 2018), and other extended variants of these architectures (Cai et al., 2018 , Tian et al., 2019). There are many detection-based methods applied to pest counting. To monitor the number of stored-grain insects, Shen et al. (2018) developed a method of detection and identification by applying deep neural networks. The results showed that the developed method could detect and identify insects in the stored grain condition, and its mean Average Precision (mAP) reached 88%. To estimate the insect pest grade in nature fields, Ding and Taylor (2016) proposed an automatic detection pipeline based on deep learning for identifying and counting pests in images taken from field traps. The results showed that their method had a promising performance in both quality and quantity. To monitor more kinds of pest, Liu et al. (2019) proposed a region-based end-to-end approach named PestNet for pest localization and recognition based on deep learning and achieved 75.46% mAP. To address the small pest recognition and detection problem in the light trap equipment, Wang, Jiao, et al. (2021) proposed a novel sampling-balanced region proposal network and achieved 78.7% mAP. To improve the tiny pest detection accuracy from sticky trap images, Li et al. (2021) developed a TPest-RCNN, and the determination coefficients reached 99.6% and 97.4% for whitefly and thrips recognition. All the above methods have achieved satisfied performance, but they aimed to detect pests under simple background rather than complex background in field environment. Wang et al. (2020) proposed a novel two-stages mobile vision based cascading pest detection approach (DeepPest) towards large-scale multiple species of pest data. Experimental results showed that DeepPest outperformed state-of-the-art object detection methods in detecting field pests. In order to improve the pest detection accuracy in the field, Li et al. (2019) proposed an effective data augmentation strategy and achieved the pest detection performance of 81.4% mAP. For monitoring the tiny pest, Wang, Wang, et al. (2021), proposed a rice planthopper search network (RPSN) and the experimental results showed that their system performs well on detecting rice planthoppers in non-specific wild environment with recognition recall up to 91% in industrial circumstance. To detect the pests in greenhouse, Karar et al. (2021) developed a new mobile application for agricultural pest detection and recognition. The results showed that their application could monitor five kinds of pests in greenhouse. Although this approach alleviated the complex background challenge to some extent, it was still inefficient in detection due to the ignorance of the target density. Different from other common objects, the estimation of aphid counting by using detection-based methods have many constraints: (1) Aphids are always densely distributed, making the detection very inefficient as shown Fig. 1. (2) The aphids are easily confused with complex background, because their colour very similar to the background. (3) Their arbitrary direction and different scales also make the detection difficult for the reason that the scale invariance and rotation invariance of image features are too weak and insensitive through CNN (Sabour et al., 2017).

To tackle these issues, many researchers have paid their attention to using density maps for estimating objects numbers like crowd counting, wildlife and cell counting. They map the image to a density map instead of using hand-crafted feature. The information (location and distribution) of objects is recorded in the density map. The object counting can be estimated via the density map. To estimate the crowd count accurately, many studies have used CNN to generate density maps to count the population (Chen et al., 2020; Li et al., 2018; Zhang et al., 2016; Xiong et al., 2019). In order to create a model for counting any class of object, Lu et al. (2018) exploited a class-agnostic counting model, this model could count cars, cells, penguins and crowd. The experimental results showed that their method surpassed the state-of-the-art methods especially on cell and crowd counting datasets. Despite the above methods had achieved good performance, these methods could not be applied to aphid counting due to complex background and severely dense distribution. In order to solve these problems, in this paper, we developed the multi-branch convolutional neural network (Mb-CNN) with density map for aphid counting estimation, which could improve the counting accuracy of aphids in densely distributed regions and overlapped areas. The key idea of our method is to use a multi-branch convolutional neural network framework to generate the density maps of aphids for estimating their number.

2. Materials and methods

2.1. *Image collection and preprocessing*

In this paper, we select 1100 images of aphids in dense regions, which are collected under the wild environment. The resolution of these images is 1440 × 1080 pixels taken by CCD (Charge Coupled Device) camera with 4 mm focal length with an aperture of f/3.3. Different from other common pests in field, aphids are often densely distributed in a region. The intuitive features of aphids in dense regions are easily confused with complex background in field environments, because aphids have a similar colour to the leaves or stems. Thus, the density of aphids might affect the automatic counting of aphids. According to the distribution density and number of aphids, in this paper, we divide our dataset into three densities: low density, normal density and high density. In general, if there are 1–30 aphids in an image, this image is classified as low density, 30 to 80 aphids are classified as normal density, and more than 80 are classified as high density. We randomly split the dataset into training subset, validation subset and test subset at the ratio of 8:1:1. The aphid dataset is illustrated in Fig. 2. The statistics of these three subsets are illustrated in Table 1.

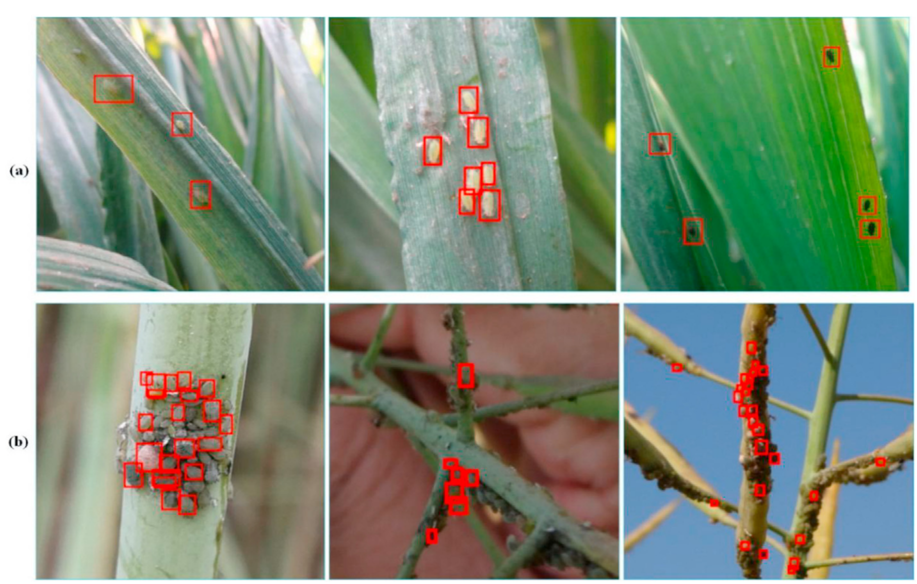

Fig. 1 – (a) The results of aphids in sparse distribution using detection-based method. (b) The results of aphids in dense distribution using detection-based method. The limitation of detection-based methods is that occlusion among aphids in a clustered environment or in a very dense region significantly affects the performance of the detector hence the final estimation accuracy.

Fig. 2 – According to the density distribution and number of aphids, the dataset of aphids is divided into low density, normal density and high density.

After that, we label these images with annotations by using Labelme. Labelme is a graphical image annotation tool, which is written in Python and uses Qt for its graphical interface (https://pypi.org/project/labelme/2.0.1/). When the number of objects is large, dotting (pointing) is a usual way to count objects for humans. Thus, different from bounding-box annotation methods, this paper, we apply a specific annotation method that positions a single dot on each object in a given image. The method not only gives the spatial distribution of the aphids in the given image but also is less labour intensive than the bounding-box annotation. The category name and the location of aphids would be saved to a "json" file. Figure 3 shows some examples of the dotted annotation.

2.2. *Method*

The density-based methods contained CNN backbone, density map generation and counting estimation as shown in Fig. 4. The convolution feature maps are extracted by CNN, and then the density map is generated by utilizing the 1×1 convolution followed by ReLU. Finally, the number of objects would be estimated by using the density map.

2.2.1. *Density map generation*

In the training step, the density maps of aphids generated by training dataset would be used as the ground truth, which is very import for counting aphids. Firstly, we first describe how to generate a density map of an image with labelled aphids.

If there is an aphid at pixel x_p in the image, the image with M aphids annotated can be represented as:

$$F(x) = \sum_{p=1}^{M} \delta(x - x_p) \tag{1}$$

$\delta(x - x_p)$ is a delta function and can be represented as:

$$\delta(x - x_p) = 0 \ , \ x \neq x_p \tag{2}$$

$$\int_a^b \delta(x - x_p) dx = 1, \quad a < x_p < b \tag{3}$$

There is a serious problem in the density map generated via $F(x)$, which will cause the generated density map to be sparse. The sparse density map would result in the loss value of CNN approaching to 0. It is not conducive to count the number of aphids when the aphids are very dense. Therefore, Gaussian kernel $K_s(x)$ is used to convolve with $F(x)$, the convolved density map can be represented as following:

$$G(x) = \sum_{p=1}^{M} \delta(x - x_p) * K_s(x) \tag{4}$$

where $K_s(x)$ is the Gaussian kernel, M is the number of aphids annotated. s is the spread parameter. The process of convolution uses a two-dimensional Gaussian kernel to slide on the density map as shown in Fig. 5.

To generate the density map, we should determine the spread parameter s based on the size of the aphids in the image. However, the size of the aphids is not obvious because of overlapping. It is difficult to find the underlying relationship between the aphid size and the density map. To address this problem, Zhang et al. (2016) proposed an adaptive spread parameter method based on its average distance to its neighbours. They found that the size of object is highly related to the distance of neighbours. If x_p is an aphid in a given image, its k nearest neighbours are denoted as $\{d_1^p, d_2^p, \ldots, d_n^p\}$, $n = 4$, so average distance is $\overline{d_p} = \frac{1}{n}\sum_{j=1}^{n} d_j^p$. Thus, the pixel associated with x_i corresponds to an area on the ground in the scene roughly of a radius proportional to $\overline{d_p}$. The spread parameter can be represented as $s = \beta\overline{d_p}$ with $\beta = 0.3$. The density maps generated by using $G(x)$ are shown in Fig. 6.

Table 1 – Statistics on two subsets for our dataset with training subset, validation subset and test subset.

Density distribution	Threshold	Train subset		Validation subset		Test subset	
		#Images	#Annotation	#Images	#Annotation	#Images	#Annotation
Low-density	[0, 30]	561	3079	77	433	68	447
Normal-density	[31–81]	200	9580	14	716	20	997
High-density	[81,+∞]	119	10,305	19	1573	22	1814
Total		880	22,964	110	2722	110	3258

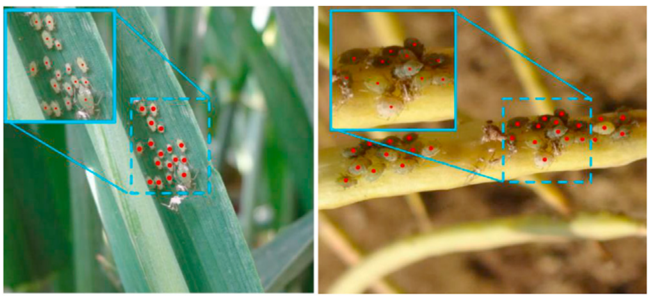

Fig. 3 – The example of the dotted annotations.

2.2.2. *Multi-branch convolutional neural network*

In our aphid dataset, aphids have different scales due to the distance from camera and their ages. In order to estimate the counting of aphids, inspired by Multi-Column Convolutional Neural Network (MCNN) (Zhang et al., 2016) and FPN (Lin et al., 2017), a novel multi-branch convolutional neural network (Mb-CNN) is proposed in this paper. MCNN is a method for estimating the crowd counting, which contains three columns correspond to filters with different sizes for extracting feature maps of different scale. However, the background of aphids is more complex than crowd and aphids are tinier than human heads. Thus, the methods of crowd count estimation might be not suitable for aphid counting. FPN is a top-down architecture with lateral connections is developed for building high-level semantic feature maps at different scales. As shown in Fig. 7, Mb-CNN includes a backbone network and three branch networks. These three branch networks come from different layers of the backbone network, and the convolution kernel sizes of these layers are different, thus, Mb-CNN can be considered as a pyramid structure. The reason for using pyramid structure is that the three branches correspond to filters with receptive fields of different sizes (big, medium and tiny).

The multi-scale feature maps from different branch networks in the backbone network are extracted to adapt to the variety of different scales of aphids. Mb-CNN consists of six convolution layers (Conv1–Conv6), each of which has different sized convolution kernels. In order to reduce the dimension of the feature maps and improve the speed of Mb-CNN, the feature maps from Conv2 are forwarded via a convolution layer (Conv2_1) in Branch_1. The depth of feature maps is reduced from 48 to 24 via the convolution layer of Conv2_1. In order to maintain the structure of pyramid, the size of the convolution kernel in the Conv2_1 remains unchanged. The max pooling (2 × 2) is used to scale the feature maps from Conv2_1 to a size similar to that from Conv4_1 in Branch_2. Similarly, the feature maps from Conv4 and Conv6 are forwarded through Conv4_1 and Conv6_1. Similar to Branch_1, an upsample operation is used in Branch_3 to keep the size uniform with the feature maps from Conv4_1 in

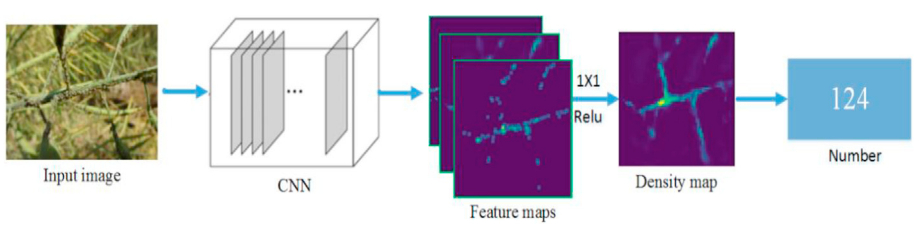

Fig. 4 – The architecture of density-based methods.

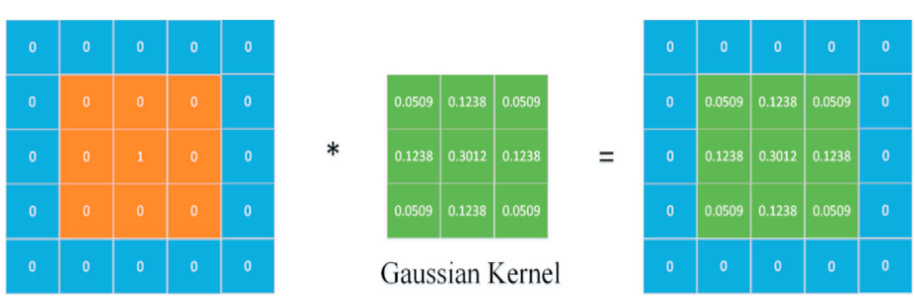

Fig. 5 – The process of density map generation by using Gaussian Kernel.

Fig. 6 – Original images and corresponding crowd density maps obtained by convolving geometry-adaptive Gaussian Kernels.

Branch_2. The feature maps from Branch_1, Branch_2 and Branch_3 are concatenated together to generate the density map.

In the training step, our Mb-CNN is trained by using Euclidean distance to measure the difference between the estimated density map and ground truth map. The loss function could be defined as below:

$$\text{loss} = \frac{1}{2N}\sum_{i=1}^{N}\|F(X_i;\Theta) - GT_i\|_2^2 \tag{5}$$

where *loss* is the loss between ground truth map and estimated density map. N is the number of images in training set, X_i is the input image, Θ is the parameters in Mb-CNN including learning rate, batch-size and epoch. $F(X_i;\Theta)$ is the estimated density map of X_i, GT_i is the ground truth map of X_i.

In the test step, the aphid density map would be generated when the aphid image was feed into Mb-CNN. The number of aphids in the image was the sum of all the pixels of the density map:

$$\text{Num} = \text{round}\left(\sum_{x} F(x)\right)$$

where $F(x)$ is the density map and x is the value of pixel. The final number of aphids is obtained via rounding $F(x)$, round$(\cdot)$ is the is the rounding function.

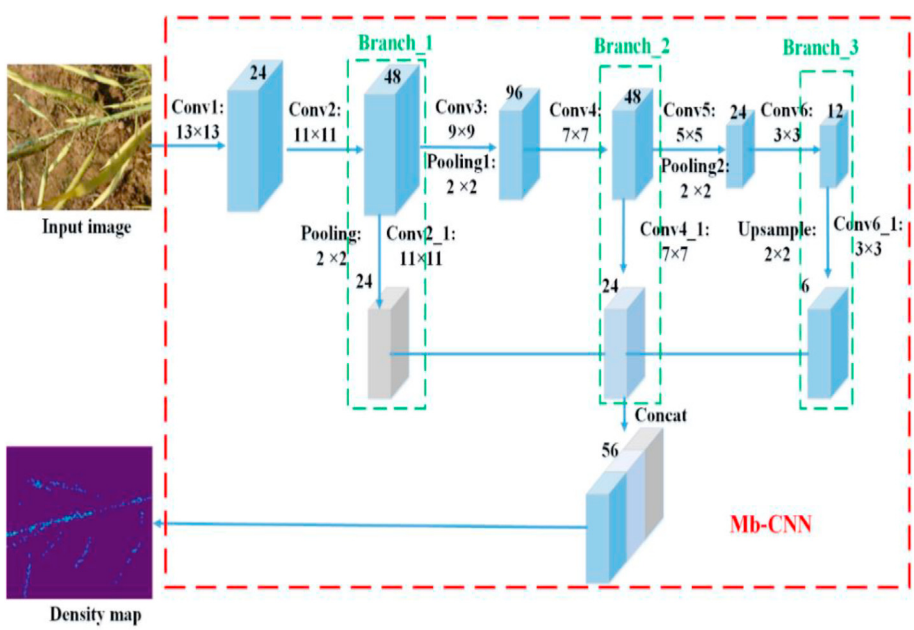

Fig. 7 – The structure of the proposed multi-branch convolutional neural network for aphid counting.

Table 2 – Performance comparison between the state-of-the-art methods.

Based method	Method	MAE	MSE
Detection-based methods	FPN	46.92	48.74
	FCOS	52.87	54.56
	Cascade R-CNN	44.81	46.89
	RPSN	45.50	47.55
Density-based methods	MCNN	11.59	13.84
	CRNet	15.17	17.23
	D2C	10.82	12.63
	Ours	**10.22**	**12.24**

3. Experiment

3.1. *Experimental settings*

In order to verify that Mb-CNN could be applied to aphid counting, some experiments are built to evaluate the accuracy and robustness of model. At present, most of pest automatic counting is all based on the method of object detection. Therefore, three state-of-the-art detection-based methods are experimented in this paper. The methods based on object detection are FPN, fully convolutional one-stage object detection (FCOS) (Tian et al., 2019), Cascade R–CNN and RPSN (Wang, Wang, et al., 2021, Wang, Jiao, et al., 2021). FPN is a pyramidal hierarchy of deep convolutional networks and belongs to two-stage detector. FCOS is a full-convolutional one-stage object detection method, which has high computing efficiency. Cascade-RCNN extended Faster R-CNN to a multi-stage detector through the classic yet powerful cascade architecture, and has good performance on object detection. RPSN belonged to the detection-based methods. RPSN was used for counting planthoppers or other tiny pests via extracting multiple high-quality proposal regions from large-scale pest images with tiny objects (Azulay & Weiss, 2019).

All above methods use ResNet-101 (He et al., 2016), as backbone for feature extraction. The deep learning framework of "mmdetection" (https://github.com/open-mmlab/mmdetection) with Python API 3.7 is applied in this paper. The learning rate is initialised to 0.001 and the min-batch is set to 2.

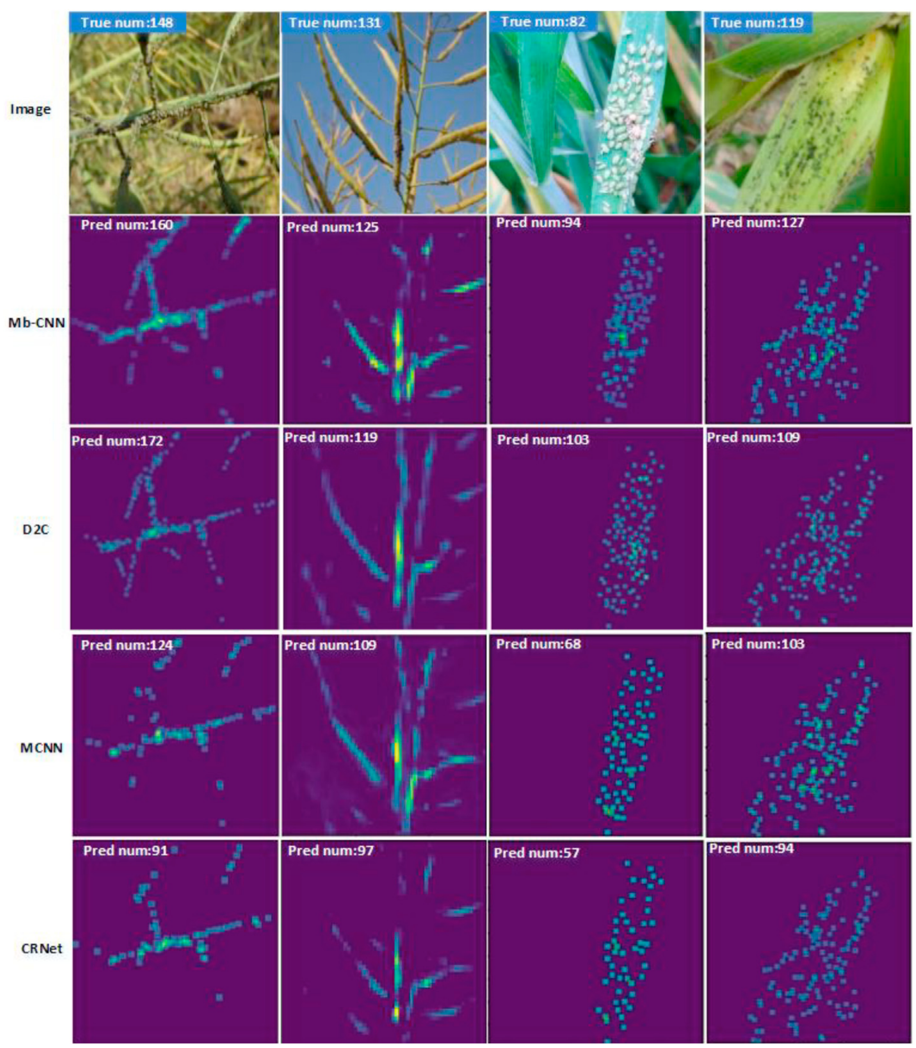

Fig. 8 – The results of density-based methods. The methods of density map do not need locate the position of aphids, which obtain the number of aphids via density map regression. Thus, compared with detection-based methods, they have better performance in complex background and tiny object detection.

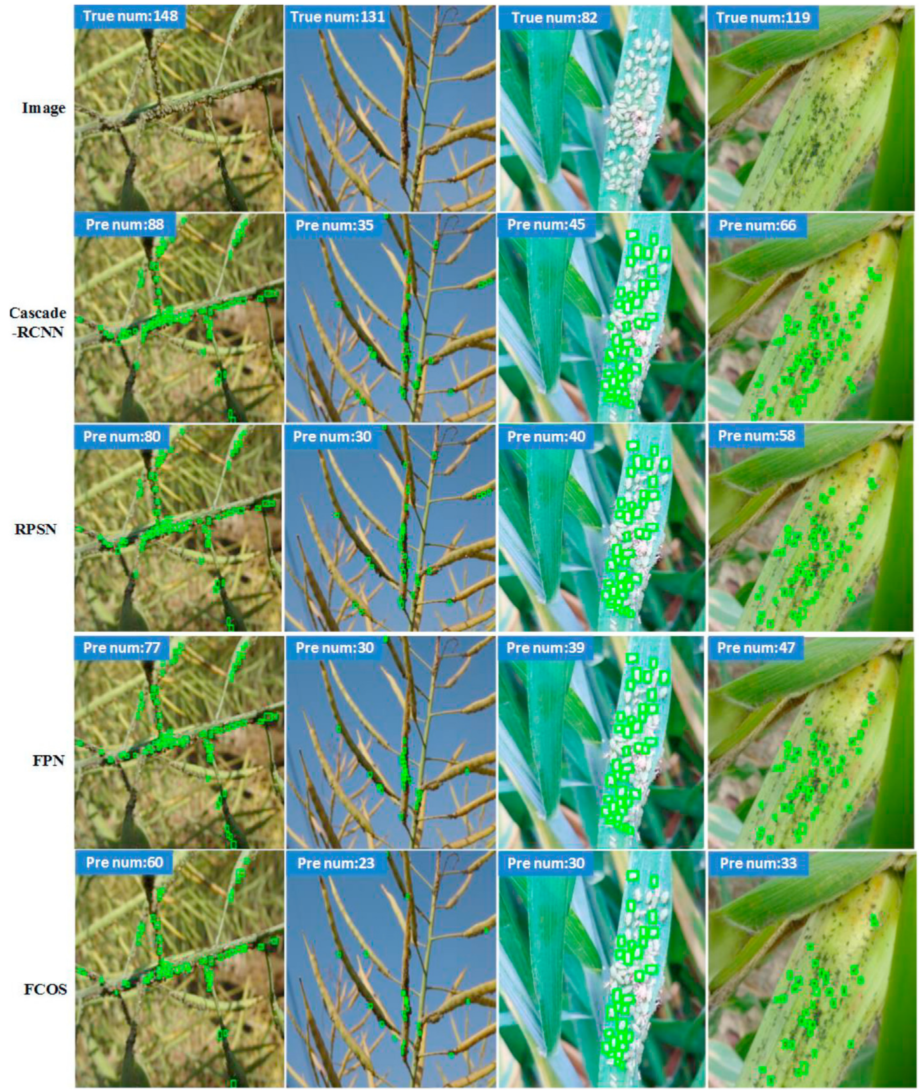

Fig. 9 – The results of detection-based methods. Aphids always gather in a clique; the high density and tiny size of aphids could raise the difficulty to detect them by confusing their features with those of the adjacent aphids. Thus, the detection-based methods have high rate of missed detections.

Besides, three methods based on the density map are also experimented to verify the performance of our method. These methods are MCNN, CRNet and D2C. MCNN is a multi-column convolutional neural network architecture to extract multi-scale features and obtain its density map (Zhang et al., 2016). CRNet could refine predicted density maps progressively based on hierarchical multi-level density priors (Liu et al., 2020). D2C (Cheng et al., 2021) had a good performance in crowd counting, and a probabilistic intermediate representation termed the probability map was introduced in this method. D2C decouple counting into probability map regression and count map regression. All above methods adopt multi-scale architecture to generate density map. The deep learning framework of "pytorch" (https://pytorch.org/) with Python API 3.7 is applied in this paper and run on 12GB GTX 1080Ti GPU. In the training process, the learning rate is set to 0.0001, the number of iterations is 2000. In this paper, the training set, validation set and test set are randomly allocated according to 8:1:1, and the final model would be selected according to the performance of the validation set.

3.2. *Evaluation metrics*

Compared with other detection-based methods, we use Mean Absolute Error (MAE) and Mean Square Error instead of using Average Precision (AP) to evaluate the metric, because the AP is updated for combining tasks of both classification and localization (bounding box). While the point annotation method is used in this paper, which does not contain the information of bounding box, MAE and MSE are more suitable for the model of aphid counting estimation. MAE evaluates the accuracy of the model, and MSE evaluates the robustness of the model, which are defined as:

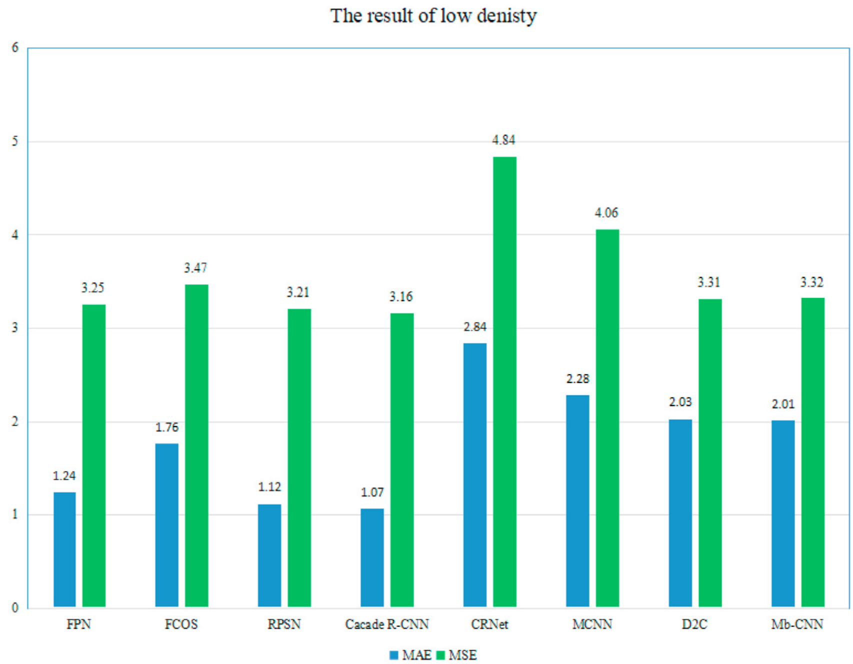

Fig. 10 – The results of low density. The methods of detection-based have lower MAE and MSE than density-based methods when aphids are in low density distribution. Thus, the detection-based methods could be used for counting the number of aphids when aphids are not densely distributed.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|z_i - \hat{z}_i| \tag{6}$$

$$\mathrm{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2} \tag{7}$$

where N is the number images of test set, z_i is the number of labelled aphids in the ith image, and $\hat{z}_i$ is the number of estimated aphids in the ith image.

3.3. *Experimental results*

Table 2 presents the results of detection-based methods and density-based methods. The best result is marked with the

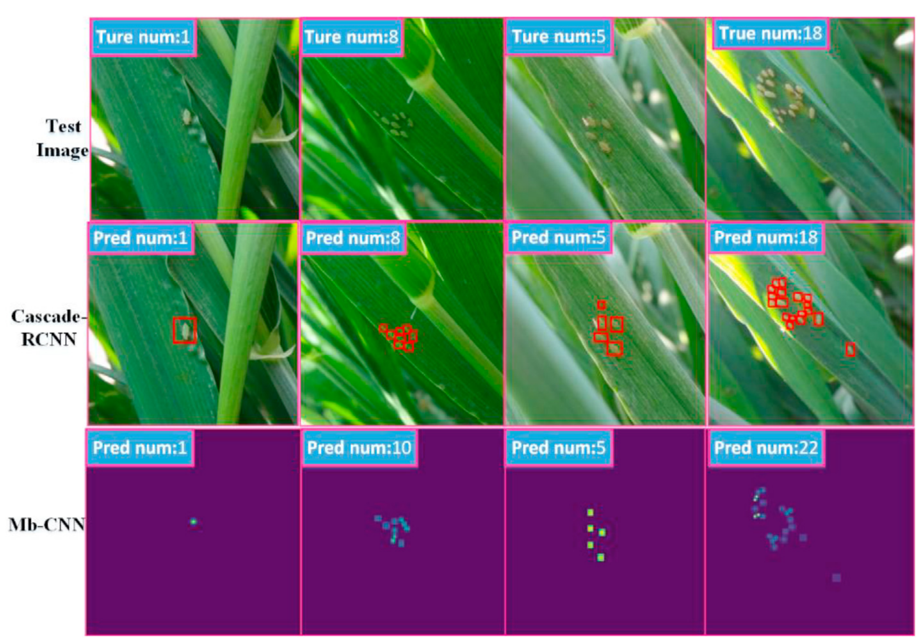

Fig. 11 – The results of low density. There have lower missing and false detection rate by using detection-based method in the case of low density distribution.

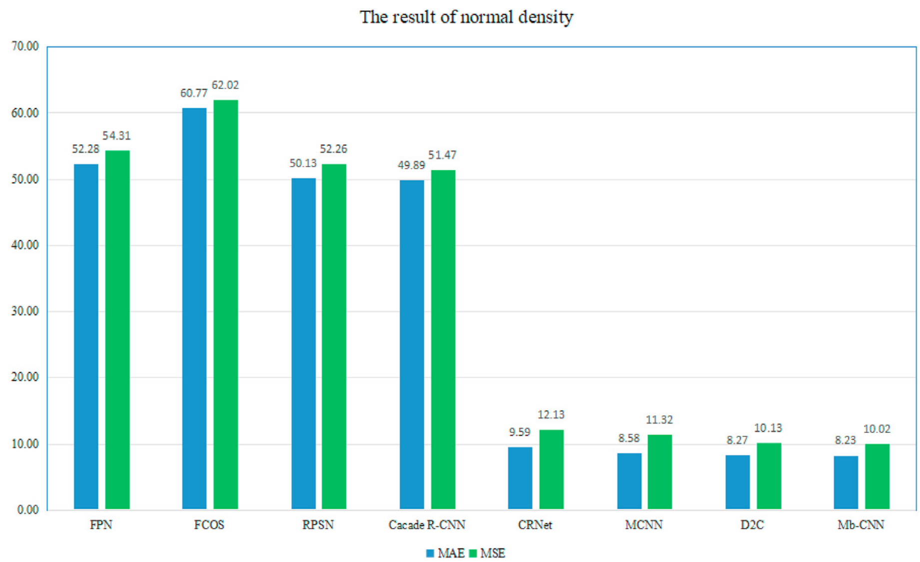

Fig. 12 – The results of normal density. Different from low density, the methods of detection-based have much higher MAE and MSE than density-based methods when aphids are in normal-density distribution. The density-based methods have better performance in the case of normal-density distribution.

bold font. Table 1 shows that compared with detection-based methods, density-based methods have better performance. The results of density-based methods are shown as Fig. 8, and the results of detection-based methods are shown as Fig. 9. The results show that the density-based methods are better than detection-based methods. Because when tiny aphids gather in a clique, the high density of aphids could raise the difficulty to detect them by confusing their features with those of the adjacent aphids (Li et al., 2019). This may lead to an excessively high rate of missed detections as shown Fig. 1(b). Different from detection-based methods, the methods based on the density map estimate the number of aphids by obtaining the density map of the image without the requirement of the aphid location information. Thus, the density-based methods have higher accuracy.

Table 2 shows that Cascade R-CNN has the best performance among the detection-based methods, the reason is that Cascade R-CNN adopted a cascade framework, which

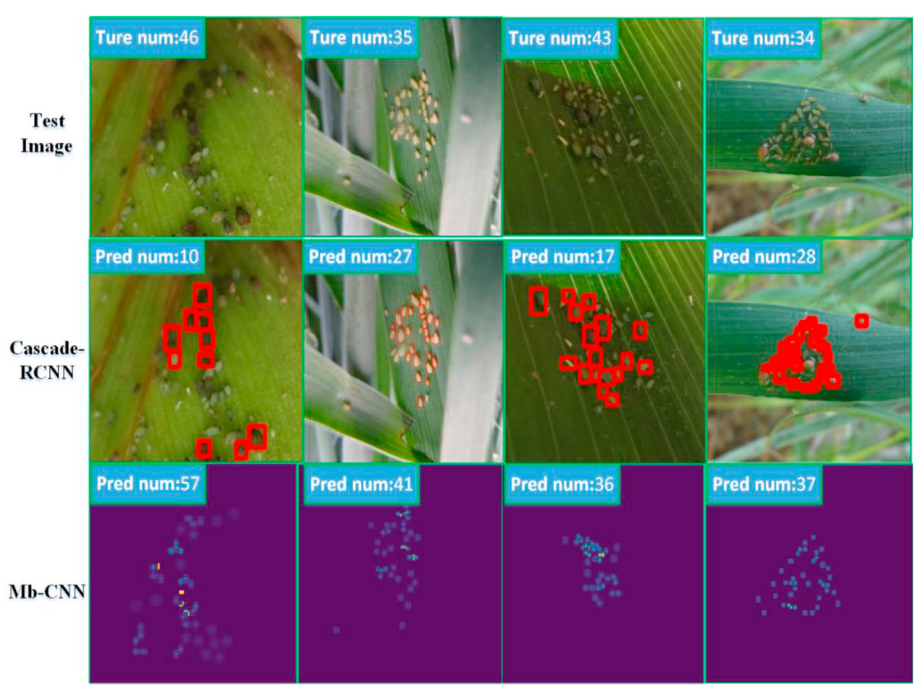

Fig. 13 – The results of normal density. There are higher missing and false detection rates by using detection-based method when aphids are in normal density distribution.

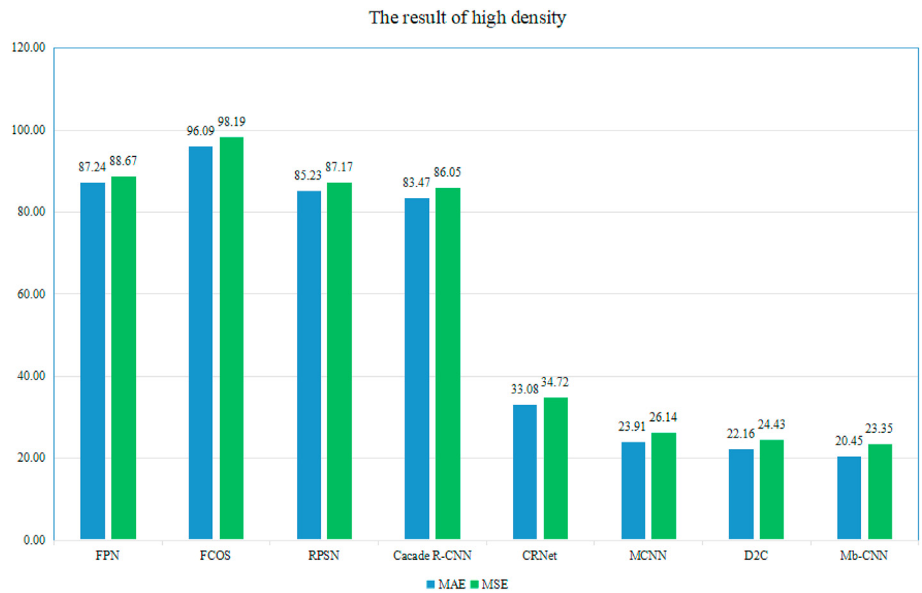

Fig. 14 – The results of high density. When aphids are in high-density distribution, the methods of detection-based have much higher MAE and MSE than density-based methods. As the density of aphids increases, the performance of detection-based methods are getting worse.

could improve the accuracy and robustness of aphid detection. RPSN was specially used to detect tiny pests, it could automatically extract multiple high-quality proposal regions from pest images with tiny objects. Furthermore, sensitive score matrix was used to further enhance the performance of classification and bounding box regression. Thus, RPSN has better results than FCOS and FPN.

The experimental results show that among the state-of-the-art density-based methods, Mb-CNN has the best performances. The pooling layers may lead to information loss, which might weaken the features of objects in feature maps especially when the object is very tiny. Compared with other density-based methods, Mb-CNN reduces the number of pooling layers. Furthermore, Mb-CNN is a pyramid structure, which could integrate the low-level object features and high-level semantic features. Among the density-based methods, MCNN, CRNet and D2C are all developed for estimating the number of crowds. The scale of human head in the crowds is

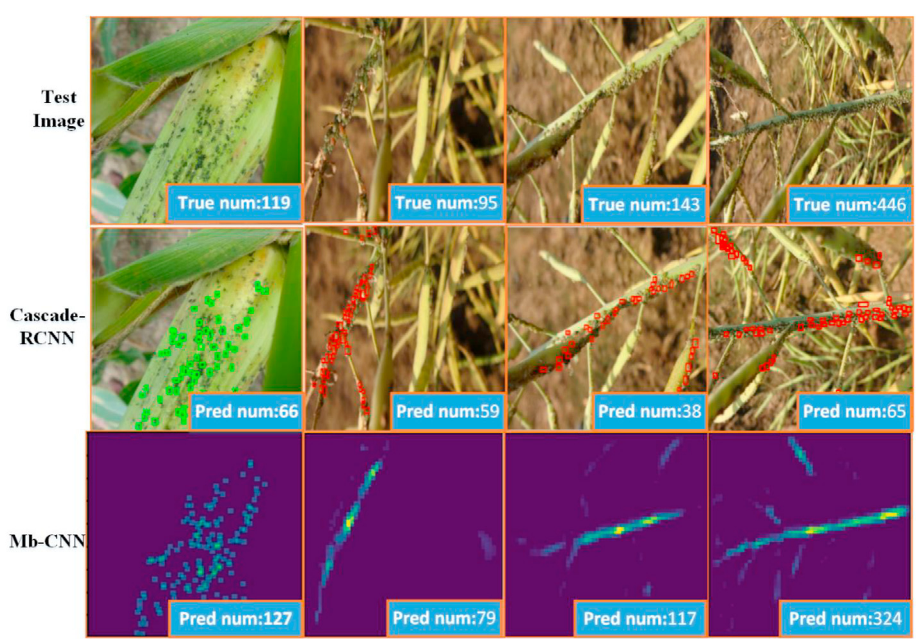

Fig. 15 – The results of high density. There are many aphids cannot be detected by using detection-based method when aphids are in high density distribution. Thus, the density-based methods are more suitable for aphid counting when aphids are in high density distribution.

much larger than that of aphids, and these methods may be not suitable for the estimation of aphids. Thus, among the density-based methods, Mb-CNN has the best results. Table 2 shows that D2C has better results than MCNN and CRNet, because D2C applied probability map regression and count map regression to generate density map, the density map is closer to the number of aphids. MCNN has less number of pooling layers, thus, MCNN has better results than CRNet.

3.4. Experimental analysis

3.4.1. The analysis of dense distribution

Aphids are always densely distributed, different dense distribution may affect the performance of aphid counting. Thus, three kinds of dense distribution (low density, normal density and high density) are experimented in this paper. The results of detection-based methods and density-based methods in different density distributions as shown Figs. 10–15. The visualization of detection-based method (Cascade R-CNN) and density-based method (Mb-CNN) are shown as Figs. 11, 13 and 15. The results of detection-based methods are better than those density-based methods when aphids are in a low-density distribution as shown Figs. 10 and 11. However, with the increase in the density of aphids, the counting accuracy of density-based methods are much better than that of detection-based methods as shown Figs. 12–15. Thus, the experimental results show that the distribution of aphids could affect the accuracy of counting. Specifically:

1) The density-based method have better performance than detection-based method when aphids are in normal density and high density distribution.
2) The detection-based methods have better performance than density-based methods when aphids are sparsely distributed.

3.4.2. The difference between density map and semantic segmentation

Because the output results of Mb-CNN is very similar to semantic segmentation network, thus, it is necessary to discuss the differences between the two methods. Specifically, there are the following differences:

(1) The method of image annotation is different. Our proposed method uses point for image annotation, while semantic segmentation need the pixel-level annotations as shown Fig. 16(b).
(2) The network of the two methods are different. The Mb-CNN is used for extracting the texture features of aphids and the final output is a density map of aphids. Semantic segmentation network is used for extracting the pixel-level features and the final output is a segmented image as shown Fig. 16(a).
(3) The ground truth of the two methods are different. The ground truth of Mb-CNN are density maps of aphid. The ground truth of semantic segmentation network is a pixel-wise segmentation annotation file which gives the class of the object at each pixel.
(4) The two methods have different application fields. Mb-CNN is mainly used for counting the number of dense objects, thus, our proposed method has higher accuracy in the dense aphids counting. The semantic segmentation is mainly used for scene recognition and understanding not for object counting.

Overall, the semantic segmentation network is rarely used for object counting, because when aphids are densely distributed, it is easy to count the adhesive aphids as an aphid using semantic segmentation method as shown Fig. 17. Thus, it is not satisfactory due to the tiny size and the dense

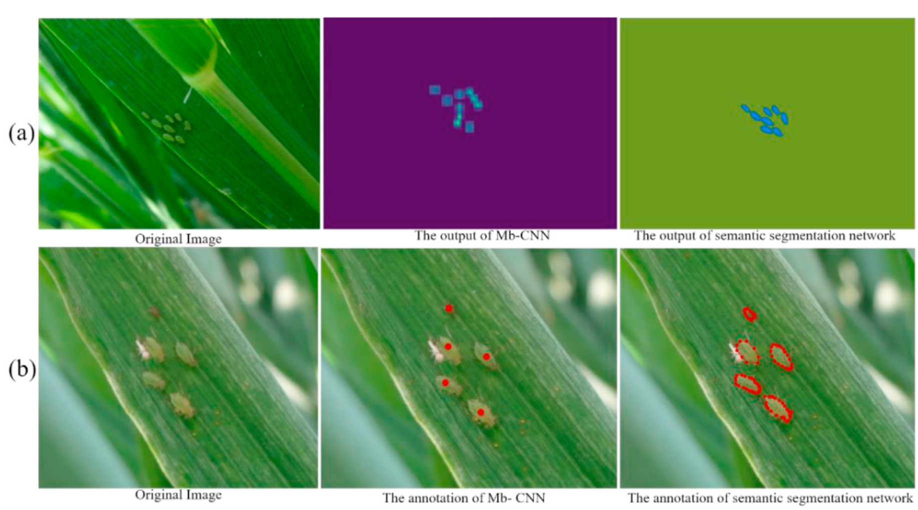

Fig. 16 – The difference between Mb-CNN and semantic segmentation.

Fig. 17 – It is easy to count the adhesive aphids as an aphid using semantic segmentation.

distribution. Because the down-sampling layers in semantic segmentation network are much more than Mb-CNN, and tiny sizes might weaken the features of aphids in feature maps.

4. Conclusion

In order to solve the problem of aphid counting due to serious dense distribution, density map is firstly applied for aphid counting. A multi-branch convolutional neural network with density map for aphid counting is proposed, which is feasible to apply to practical aphid prevention. Experimental results demonstrate that Mb-CNN achieves 10.22 MAE and 12.24 MSE on aphid counting task, which has better performance than the other state-of-the-art approaches. Furthermore, we publish a domain specific dataset for aphid detection and counting in the filed containing more than 1100 images and 28,944 annotated labels in this paper. Specifically, this dataset has a high application value on aphid counting. In the future, we will target at improving the generalization of our Mb-CNN and transferring it into generic dense object counting task.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by The National Key Technology R&D Program of China (grant number ACAIM190101), the National Natural Science Foundation of China (grant number 32171888, 61773360) and Dean's Fund of Hefei Institute of Physical Science, Chinese Academy of Sciences (grant number YZJJ2020QN21).

REFERENCES

Azulay, A., & Weiss, Y. (2019). Why do deep convolutional networks generalize so poorly to small image transformations. *Journal of Machine Learning Research, 20*(2019), 1–25. https://arxiv.org/abs/1805.12177v1.

Chen, J., Su, W., & Wang, Z. (2020). Crowd counting with crowd attention convolutional neural network. *Neurocomputing, 382*, 210–220. https://doi-org-443.webvpn.las.ac.cn/10.1016/j.neucom.2019.11.064.

Cheng, J., Xiong, H., Cao, Z., & Liu, H. (2021). Decoupled two-stage crowd counting and beyond. *IEEE Transactions on Image Processing, 30*, 2862–2875. https://doi.org/10.1109/TIP.2021.3055631

Ding, W., & Taylor, G. (2016). Automatic moth detection from trap images for pest management. *Computers and Electronics in Agriculture, 123*, 17–28. https://doi.org/10.1016/j.compag.2016.02.003

Elison, A., João, P. M., Sandy, I., Juliana, P., Marília, B. L., José, M. C., Paulo, R. V. S. P., & DouglasL, R. R. (2020). A method for counting and classifying aphids using computer vision. *Computers and Electronics in Agriculture, 169*. https://doi.org/10.1016/j.compag.2019.105200

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90

Karar, M., Alsunaydi, F., Albusaymi, S., & Alotaibi, S. (2021). A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alexandria Engineering Journal, 60*(5), 4423–4432. https://doi.org/10.1016/j.aej.2021.03.009

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553). https://doi.org/10.1038/nature14539

Li, R., Wang, R., Zhang, J., Xie, C., Liu, L., Wang, F., Chen, H., Chen, T., Hu, H., Jia, X., Hu, M., Zhou, M., & Li, D. (2019). An effective data augmentation strategy for CNN-based pest localization and recognition in the field. *IEEE Access, 7*(2019), 160274–160283. https://doi.org/10.1109/ACCESS.2019.2949852

Li, W., Wang, D., Li, M., Gao, Y., Wu, J., & Yang, X. (2021). Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. *Computers and Electronics in Agriculture, 183*(2021). https://doi.org/10.1016/j.compag.2021.106048

Li, Y., Zhang, X., & Chen, D. (2018). CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1802.10062.

Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 936–944). https://doi.org/10.1109/CVPR.2017.106

Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., & Sudirman, S. (2019). PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access, 7*, 45301–45312. https://ieeexplore.ieee.org/document/8682057.

Liu, T., Chen, W., Wu, W., Sun, C., Guo, W., & Zhu, X. (2016). Detection of aphids in wheat fields using a computer vision technique. *Biosystems Engineering, 141*, 82–93. https://doi.org/10.1016/j.biosystemseng.2015.11.005

Liu, Y., Wen, Q., Chen, H., Liu, W., Qin, J., Han, G., & He, S. (2020). Crowd counting via cross-stage refinement networks. In *IEEE transactions on image processing*. https://doi.org/10.1109/TIP.2020.2994410

Lu, E., Xie, W., & Zisserman, A. (2018). Class-agnostic counting. In *2018 Asian conference on computer vision (ACCV)*. https://arxiv.org/abs/1811.00472.

Maharlooei, M., Sivarajan, S., Bajwa, S. G., Harmon, J. P., & Nowatzki, J. (2017). Detection of soybean aphids in a greenhouse using an image processing technique. *Computers and Electronics in Agriculture, 132*, 63–70. https://doi.org/10.1016/j.compag.2016.11.019

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *2017 advances in neural information processing systems (NIPS)* (pp. 3856–3866). http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.

Shen, Y., Zhou, H., & Li, J. (2018). Detection of stored-grain insects using deep learning. *Computers and Electronics in Agriculture,* 145, 319–325. https://doi.org/10.1016/j.compag.2017.11.039

Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In *2019 IEEE conference on computer vision and pattern recognition (CVPR)*. https://arxiv.org/abs/1904.01355.

Wang, F., Wang, R., Xie, C., Yang, P., & Liu, L. (2020). Fusing multi-scale context-aware information representation for automatic in field pest detection and recognition. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2020.105222

Wang, F., Wang, R., Xie, C., Zhang, J., Li, R., & Liu, L. (2021). Convolutional neural network based automatic pest monitoring system using hand-held mobile image analysis towards non-site-specific wild environment. *Computers and Electronics in Agriculture, 187*(2021). https://doi.org/10.1016/j.compag.2021.106268

Wang, R., Jiao, L., Xie, C., Chen, P., Du, J., & Li, R. (2021). S-RPN: Sampling-balanced region proposal network for small crop pest detection. *Computers and Electronics in Agriculture, 187*(2021). https://doi.org/10.1016/j.compag.2021.106290

Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., & Shen, C. (2019). From open set to closed set: Counting objects by spatial divide-and-conquer. In *2019 IEEE International Conference on Computer Vision (ICCV)* https://arxiv.org/abs/1908.06473.

Yao, Q., Xian, D. X., Liu, Q. J., Yang, B. J., Diao, G. Q., & Tang, J. (2014). Automated counting of rice planthoppers in paddy fields based on image processing. *Journal of Integrative Agriculture, 13*(8), 1736–1745. https://doi.org/10.1016/S2095-3119(14)60799-1

Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)*. https://doi.org/10.1109/CVPR.2016.70