Available online at www.sciencedirect.com

ScienceDirect

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

Research Paper

A coarse-to-fine network for aphid recognition and detection in the field

Rui Li [a,b], Rujing Wang [a,*], Chengjun Xie [a,**], Liu Liu [a,b,***], Jie Zhang [a], Fangyuan Wang [a,b], Wancai Liu [c]

[a] Institute of Intelligent Machines, Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China

[b] University of Science and Technology of China, Hefei 230026, China

[c] National Agro-Tech Extension and Service Center, Beijing 100125, China

ARTICLE INFO

Article history:
Received 9 April 2019
Received in revised form
20 August 2019
Accepted 22 August 2019

Keywords:
Aphid Detection
Aphid Recognition
Convolutional Neural Network
Coarse-to-Fine Network

In agriculture, aphids are one of the most destructive pests, responsible for major reductions in wheat, corn and rape production leading to significant economic losses. However, manual pest recognition approaches are often time-consuming and laborious for Integrated Pest Management (IPM). In addition, the existing pest detection methods based on Convolutional Neural Network (CNN) are not satisfactory for small aphid recognition and detection in the field because aphids are tiny and often in dense distributions. In this work, a two-stage aphid detector named Coarse-to-Fine Network (CFN) is proposed to address these problems. The key idea of our method is to develop a Coarse Convolutional Neural Network (CCNN) for aphid clique searching as well as a Fine Convolutional Neural Network (FCNN) for refining the regions of aphids in the clique. Specifically, The CCNN detects approximately all the object regions from natural aphid images with various aphid distributions including dense aphid cliques and sparse aphid objects, in which an Improved Non-Maximum Suppression (INMS) strategy is proposed to eliminate overlapping regions. Then, the FCNN further refines the detected aphid regions from the CCNN. The final recognition and detection result would be obtained by combining the outputs from CCNN and FCNN together. Experiments on our dataset show that our CFN achieves an aphid detection performance of 76.8% Average Precision (AP), which improves 20.9%, 18%,13.7% and 12.5% compared to four state-of-the-art approaches.

* Corresponding author.
** Corresponding author.
*** Corresponding author. Institute of Intelligent Machines, 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), liuliu66@mail.ustc.edu.cn (L. Liu).

https://doi.org/10.1016/j.biosystemseng.2019.08.013
1537-5110/

Nomenclature

Abbreviations

AP	Average Precision
CCNN	Coarse Convolutional Neural Network
CFN	Coarse-to-Fine Network
CNN	Convolutional Neural Network
DSSD	Deconvolutional Single Shot Detector
Faster R–CNN	Faster Regions with Convolutional Neural Network
FCNN	Fine Convolutional Neural Network
FPN	Feature Pyramid Networks
FPS	Frames Per Second
INMS	Improved Non-Maximum Suppression
IPM	Integrated Pest Management
mAP	mean Average Precision
NMS	Non-Maximum Suppression
PR	Precision-Recall
R–FCN	Region-Fully Convolutional Network
RPN	Region Proposal Network
ReLU	Rectified Linear Unit
SSD	Single Shot Detector
YOLO	You Only Look Once

1. Introduction

Aphids are one of the prime pests in wheat, rape and corn, where they feed on the sap of stem and leaf. In agriculture, aphids can cause major damage in fields and result in significant crop yield losses. Since the aphids are small, manual recognition and counting are very time-consuming, labour intensive and inefficient, which might affect the investigation efficiency of Integrated Pest Management (IPM) in the field. Therefore, automatic aphid recognition and detection in the field can reduce labour requirements and improve work efficiency.

With the rapid development of computer vision technology, a lot of research on pest recognition and detection has been undertaken. Tian, Chen, Dong, and Li (2016) used a fusion method of infrared sensor and machine vision for pest identification and counting in orchard ecosystems that achieved an accuracy of 80%. Deng, Wang, Han, and Yu (2018) proposed a pest image detection and recognition method based on bio-inspired techniques to obtain categories and number of the pests. Wen, Guyer, & Li (2009) used a local feature method for orchard insect identification and classification. Xie, Zhang, Li, Li, Hong, & Xia (2015) proposed an automatic classification method for in-field crop insect identification using multiple-task sparse representation and multiple-kernel learning. Faithpraise and Chatwin (2013) applied the correspondence filter for the segmentation of paddy field pests and classified the pests using K-Means method. Besides, other classifiers have also been employed for insect recognition such as Support Vector Machine (Liu, Chen, Wu, Sun, Guo, & Zhu, 2016b), (Ebrahimi, Khoshtaghaza, Minaei, & Jamshidi, 2017) Random Forest (Yang, Liu, Xing, Qiao, Wang, & Gao, 2010, pp. 545–548), (Yuan & Hu, 2016) and AdaBoost (Yao, Xian, Liu, Yang, Diao, & Tanget, 2014). However, though all the above methods could achieve satisfactory performance, they aim to detect pests under simple background conditions rather than with the complex backgrounds typical of the field environment. In practical application, the dense distribution of the pest, complexity of image background, illumination, scales and different attitudes are the major challenges in pest recognition and detection.

In order to address these issues, many researchers have focused on developing novel methods. Recent advances in deep learning techniques based on Convolutional Neural Network (CNN) (Krizhevsky, Sutskever, & Hinton, 2012; Lecun, Bengio, & Hinton, 2015) have led to significantly promising progress in the field of object recognition and detection under natural conditions. In contrast to conventional machine learning methods, CNN can avoid the use of hand-crafted feature extractors and automatically learn the appropriate features from the training subset. Therefore, most of research has extracted pest features by using CNN and achieved a satisfactory recognition and detection accuracy. Wang and Zhang (2018) applied the deep belief network to predict the large-scale occurrence of cotton pests by using cotton's growth environment information, but the growth environment information was easily disturbed by noise and had uncertainty, which directly affected the prediction results. Ding and Taylor (2016) located pest regions by using the sliding window on feature maps extracted by CNN, and they achieved precision-recall rate of 93%, higher than LogReg algorithm (Hilbe J M, 2009; Liu, Gao and Yang, 2016c) which used GrabCut, a variant of CNN, to compute a saliency map and detected pest. Furthermore, there are many excellent CNN-based object detection methods like Faster-RCNN (Ren, He, Girshick, & Sun, 2015), SSD (Liu, Anguelov, Erhan, Szegedy, Reed, Fu and Berg, 2016a), YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016, pp. 779–788), R–FCN(Dai, Li, He, & Sun, 2016), Feature Pyramid Network (FPN) (Lin, Dollar, Girshick, He, & Hariharan, 2017, pp. 936–944), DSSD (Fu, Liu and Ranga, 2017), and other extended variants of these networks (Cai & Vasconcelos, 2018, pp. 6154–6162; Chen, Li, Sakaridis, & Dai, 2018, pp. 3339–3348). Faster RCNN, SSD and YOLO are effective object detectors towards general object detection in a specific field, but intractable for use in practical detection of tiny objects. Although these methods like FPN could achieve satisfied performance, they cannot accurately detect tiny pests that are densely distributed.

Different from other common pests in the field, aphids are often distributed in dense regions (Fig. 1(a)) and are always tiny in static images as shown in Fig. 1(b). Specifically:

1) The intuitive features of aphids in dense regions are easily confused with complex background in field environments.
2) Features of aphids like scale invariance, rotation invariance, translation invariance are too weak and insensitive to be recognised.
3) Tiny sizes might weaken the features of aphids in feature maps because down-sampling operations can lead to information loss. Thus, the aphid detection in

Fig. 1 – (a) Features of aphids in dense distributions are too weak and insensitive to be recognised. (b) The sizes of aphids would be at most 1.5% of the whole image size, which may be smaller and more insensitive after each pooling layer of CNN.

the field might be significantly affected by their distributions.

In Fig. 2, we illustrate some of the aphid detection visualisation performed by Feature Pyramid Network (FPN) (Lin et al., 2017, pp. 936–944). As shown, it is obvious that densely distributed aphids may be more difficult to detect than well-separated aphids. 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. In this case, a potential way to alleviate this challenge in natural aphid detection is to localise the regions of aphid cliques containing a large number of aphids and then refine them into single objects. Therefore, in this paper, we have developed a novel architecture named Coarse-to-Fine Network (CFN), which can improve the detection accuracy of aphids in dense distribution regions. The key idea of our method is to develop a Coarse Convolutional Neural Network (CCNN) to search for aphid cliques as well as a Fine Convolutional Neural Network (FCNN) for refining the regions of aphids in the clique. The final output is obtained by combining the final detection results from CCNN and FCNN.

Fig. 2 – (a) the result for well-separated aphids (b) the result for dense distribution aphids. The red boxes are the results detected by using Feature Pyramid Network. The blue boxes are the results miss detected. It shows that densely distributed aphids are more difficult to detect than well-separated aphids. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

2. Materials and methods

2.1. Image acquisition and preprocessing

In this paper, we select 2200 aphid images, which are collected under the field environment. The resolution of these images taken by CCD camera with 4 mm focal length and an aperture of f/3.3 is 1440 × 1080 pixels. We randomly split the dataset into training subset and validation subset at ratio of 9:1. The statistics of these two subsets are illustrated in Table 1.

After image acquisition, we label these images with annotations by using LabelImg (https://tzutalin.github.io/labelImg/). LabelImg is a graphical image annotation tool, which is written in Python and uses Qt for its graphical interface. The regions of object are selected to annotate the rectangular boxes and class name by using mouse. The rectangular boxes are described by two pairs of coordinates: top-left and bottom-right point. Annotations are saved as XML files in PASCAL VOC-style. Specifically, aphids are firstly annotated with dense distribution regions and aphids are then labelled by agricultural experts to guarantee the accuracy of these annotations (Fig. 3 (a)). Then the dense distribution regions are cut into different blocks, and the aphids would be annotated with bounding boxes, as shown in Fig. 3(b).

Before training our model, some data augmentation methods are applied to expand the amount of data. Firstly, given the change of the rotation invariability and changes of shooting angle, the original images are rotated with unchanging image resolution. Then we vertically and horizontally flip the training subset to obtain the extra 2-fold images.

2.2. Coarse-to-fine network (CFN) architecture

To address the issue of aphid detection in dense distributions in the field, we describe here our proposed CFN architecture (Fig. 4).

In the first stage, a CCNN architecture is proposed to detect dense regions of aphids (which we call cliques). The input image taken in the field environment is fed into a CNN backbone to generate feature maps, and then we use region proposal network (RPN) (Ren et al., 2015) module to predict the clique regions and well-separated aphids in sparse areas. Finally, a new optimisation method, improved non-maximum suppression (INMS), is proposed to eliminate overlapping bounding boxes of aphids in dense distribution regions.

Secondly, the FCNN is used to refine the regions of aphids in the clique. After CCNN, the aphids are greatly enlarged in the image. Therefore, it is easy to recognise and detect aphids. For improving the speed of recognition and detection, we adopt a one-stage object detection architecture in FCNN.

Finally, the results from CCNN and FCNN are combined as the final aphid detection output.

2.2.1. Convolutional neural network

Our architecture adopts CNN for automatic feature extraction, mainly consisting of four parts: convolutional layers, batch normalisation layer, activation function and pooling layers. In general, images are fed into convolutional layers for obtaining convolution features and then via pooling layers for down-sampling. In this paper, different combinations of these layers are applied for extracting convolutional features, and a feature map is obtained at the end.

Convolutional layer: The convolutional layer is an important part of CNN and the convolutional features are extracted through some slide of the convolutional kernels on the image. Convolution is a linear operation and can be defined as:

$$a_{i,j} = \sum_{m=0}^{S_h} \sum_{n=0}^{S_w} w_{m,n} x_{i+m,j+n}$$

where $w_{m,n}$ is the weight of convolutional kernel at m and n, $x_{i,j}$ is the pixel value of image at position i and j, S_h and S_w are the height and width of convolutional kernel.

The image is a two-dimensional discrete signal, so the process of convolution is to use the convolution kernels to slide on the image, multiply the value of the image pixel by the weight on the convolution kernel, and then add the product result to the pixel value of the corresponding position on the feature map, as shown in Fig. 5.

Batch normalisation layer: During the training phase, the distribution of output from the convolutional layer would vary with large number of learnable parameters, which result in slower convergence. Therefore, it is difficult for the CNN to train a set of parameters with optimal performance. As the depth increases, the training process might become increasingly difficult and slow because of gradient dispersion problem. In addition, larger learning rates cannot be used in CNN. For improving the speed of convergence and alleviating the gradient dispersion of the network, batch normalisation has been introduced (Ioffe & Szegedy, 2015). The batch normalisation function is defined as:

$$y_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} \gamma + \beta$$

where x is the mini-batch, $\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$ represents the mean deviation and $\sigma = \sqrt{\frac{1}{2}\sum_{i=1}^{m}(X_i - \mu)^2}$ is the standard deviation of the mini-batch, of which m is the number of training images in the mini-batch, γ, ε is a small constant which prevents the divisor equalling zero.

Table 1 – Statistics on training and validation subsets of our dataset for each crop.

Crop name	Training		Validation		All	
	#images	#objects	#images	#objects	#images	#objects
Wheat	900	16144	100	2401	1000	18545
Rape	450	11375	50	1049	500	12424
Corn	630	12841	70	1614	700	14455
Total	1980	40360	220	5064	2200	45424

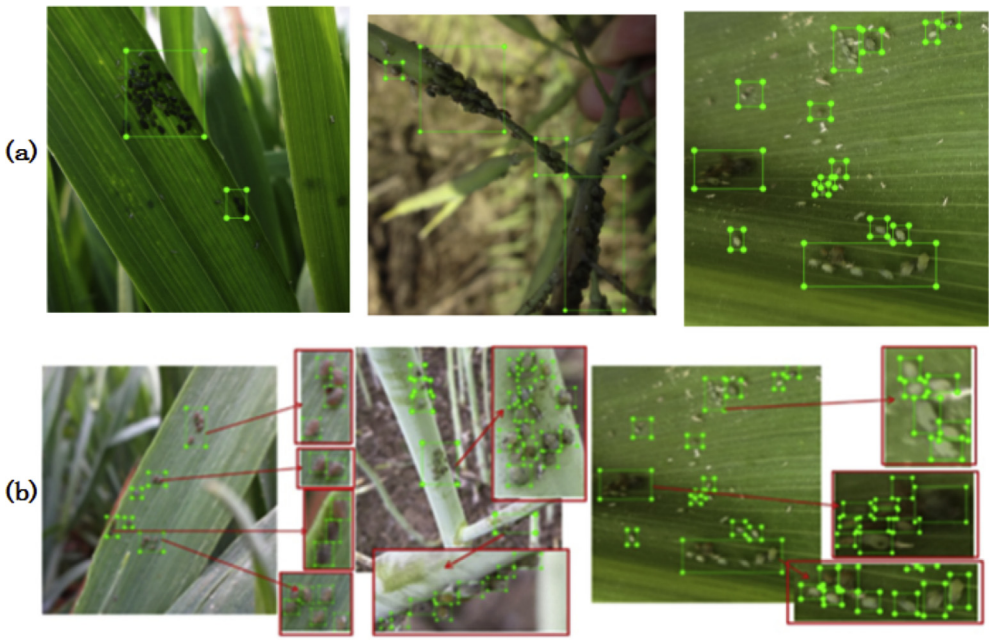

Fig. 3 – (a) Training set is annotated with aphids and dense distribution regions. (b) Aphids are annotated in dense distribution regions.

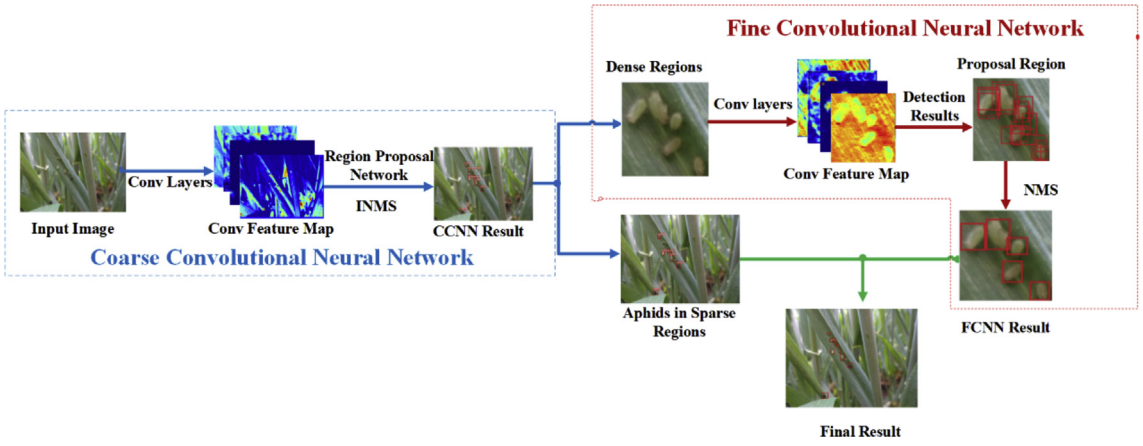

Fig. 4 – Coarse-to-Fine Network (CFN) structure. The CFN structure mainly includes coarse network and fine network, which are responsible for detecting clique regions and aphids in the cliques respectively.

There are some advantages of using a batch normalisation layer:

(1) Larger learning rates can be used in the CNN to learn the best parameters.
(2) It can improve the speed of convergence and alleviating the gradient dispersion of the CNN.

Activation function: Most of the training datasets are non-linearly separable (Huang, Xu, Schuurmans, & Szepesvari, 2016), in order to improve classification accuracy, and the activation functions are introduced to get non-linear factors. The activation function is important for CNN to learn the non-linear complex functional mapping between inputs and response. Their main purpose is to convert an input signal in convolutional neural network to an output signal. That output signal is used as an input in the next layer. To avoid vanishing gradient and sparsity, the rectified linear unit (ReLU) is usually used as activation function in CNN. The function of ReLU is defined as:

$$ReLU(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

When $x > 0$, the gradient is always 1. There is no gradient dissipation that could contribute to the effectiveness of

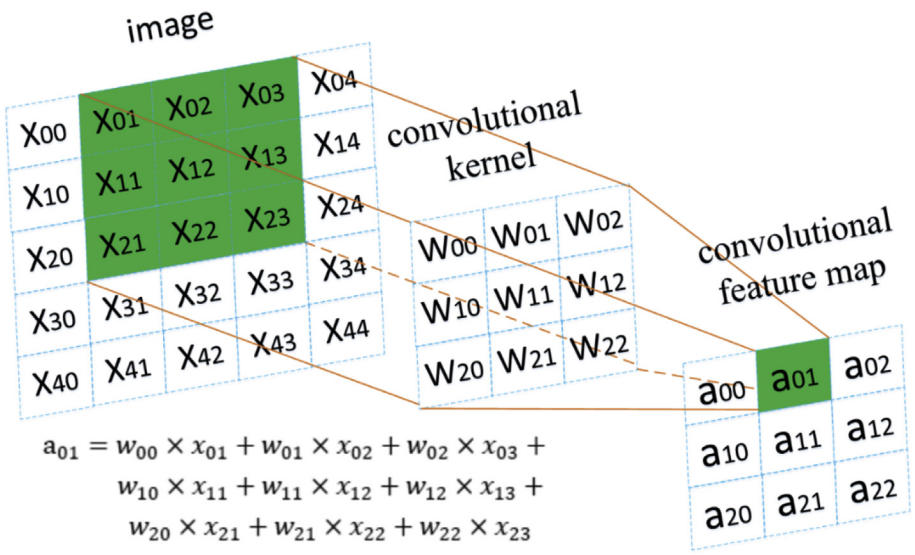

Fig. 5 – Process of convolution. The convolutional feature map is obtained by multiplying the image pixels by the weights on the convolution kernel.

convergence. When x < 0, the output of this layer is 0, which increases the sparsity of the network. Much sparser features would bring up more representative ability and stronger generalisation of the network. Therefore, ReLU can reduce risk of the vanishing gradient and improve the speed of training.

Pooling layer: Pooling layers are commonly used in CNN, as they can reduce the data dimensionality and improve the computational efficiency. Max pooling is used in this paper, which extracts maximum value within a k*k neighbourhood (Fig. 6).

2.2.2. *Coarse Convolutional Neural Network*

The state-of-the-art object detectors can be divided into two-stage approaches and one-stage approaches. The two-stage approaches have been achieving the highest accuracy, especially in tiny and multi-scale object recognition and detection (Zhang, Wen, Bian, Lei, & Li, 2018, pp. 4203–4212). Aphids are tiny pests and their scale is much smaller than the dense distribution regions. Therefore, in order to better detect dense distribution regions and aphids at the same time, two-stage object detection architecture is used in CCNN. In the first stage, the feature maps are extracted by using CNN. In the second stage, RPN is used for producing a large list of candidate regions, which is proposal-free.

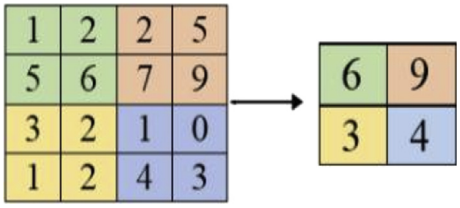

Fig. 6 – Max pooling with 2 × 2 filters and stride 2.

Region Proposal Network: RPN takes feature maps from CNN as an input and outputs probability of dense regions and well-separated aphids, as shown in Fig. 7. There is a sliding window on the feature map, and the dense regions and aphids are all obtained vis the sliding window, whose size is 6 × 6 in this paper. A low dimensional vector is generated by using the sliding window, which is then fed to classification layer and box-regression layer. Several dense distribution regions and aphids are predicted by location of sliding window. Generally, the number of dense distribution regions and aphids for each sliding window is donated as *l*. Therefore, 2*l* scores including foreground and background of each region and 4*l* coordinates (coordinates of points in upper left and lower right) of l boxes are output by box-classification layer and box-regression layer.

Improved Non-Maximum Suppression (INMS): The target regions and their corresponding category scores are obtained after RPN. However, the sliding windows of RPN can cause many target regions that are largely intersected with other regions. In order to overcome this obstacle, non-maximum suppression (NMS) (Neubeck & Gool, 2006, pp. 850–855) is chosen to select the highest scores in those regions, but this still has some limitations as shown in Fig. 8. The problem is that double boxes lie in the same region after using NMS, which may cause some aphids to be repeatedly detected, seriously affecting the counting accuracy. To solve the issue of overlapping boxes, we propose a variant of NMS named Improved Non-Maximum Suppression (INMS), whose process is shown in Algorithm 1.

Algorithm 1.

Input: $B = \{b_1, \cdots, b_N\}$,$S = \{s_1, \cdots, s_N\}$,N_t
B is the list of initial detection boxes of objects
S contains corresponding detection scores of objects
N_t is the threshold of NMS
N_b is the threshold of overlapping bounding box

begin
$D \leftarrow \{\}$
while $B \neq \phi$ do
$m \leftarrow argmax\, S$
$M \leftarrow b_m$
$D \leftarrow D \cup M; B \leftarrow B - M$
for b_i **in** B **do**
if $iou(M, b_i) \geq N_t$ **then**
$B \leftarrow B - b_i; S \leftarrow S - s_i$
end
end
end
while $B \neq \phi$ do
$M \leftarrow b_m$
for b_i **in** B **do**
if $iou(M, b_i) \geq N_b$ **then**
$M \leftarrow M \cup b_i; B \leftarrow B - b_i$
End

2.2.3. *Fine Convolutional Neural Network*

After aphid clique searching, we could re-scale the dense regions into a higher resolution. Then the sizes of aphids are enlarged so they could be detected better in our FCNN. Inspired by the one-stage detection approach such as YOLO (Redmon et al., 2016, pp. 779–788), DSSD (Fu, Liu, Ranga, & Tyagi, 2017), our FCNN structure is a highly efficient one-stage detection framework, it adopts regression method to recognise objects and locate their position. The FCNN incorporates the idea of multi-scale detection and the tiny object detection accuracy has been significantly improved. Inspired by the residual network (He, Zhang, Ren, Sun, 2016), some residual blocks are used in CNN, which is called Darknet-53 (Redmon, Farhadi, 2018). FCNN is a structure based on YOLO. Darknet-53 has 5 pooling layers and the size of the image is reduced by one-half after passing one pooling layer, so the image size is better set to a multiple of 32. Through our experiment, the image is resized to 608 × 608, which not only has a high computational efficiency but also has a better detection accuracy (Table 2). The experimental results show that the bigger size cannot provide better detection accuracy. This is because, when the image scale reaches a certain size, the sharpness of image becomes blurred, making it difficult to distinguish the aphids from the background. As shown in Fig. 9, the detecting process of FCNN is as follows:

1) The original image is firstly resized to 608 × 608, and then the image is divided into S × S cells, in this paper we set S to 14.
2) The feature maps are extracted from Darknet-53.
3) Logistic Regression is used to predict the class, bounding box and the probability value of the object.
4) Some bounding boxes with low confidence are filtered out by a threshold.
5) The final bounding boxes are obtained by using NMS.

2.3. *Evaluation metrics*

For validating the performance of our model in detecting aphids, we select Precision-Recall (PR) curve and Average Precision (AP) (Zhang & Zhang, 2016) as the evaluation metrics.

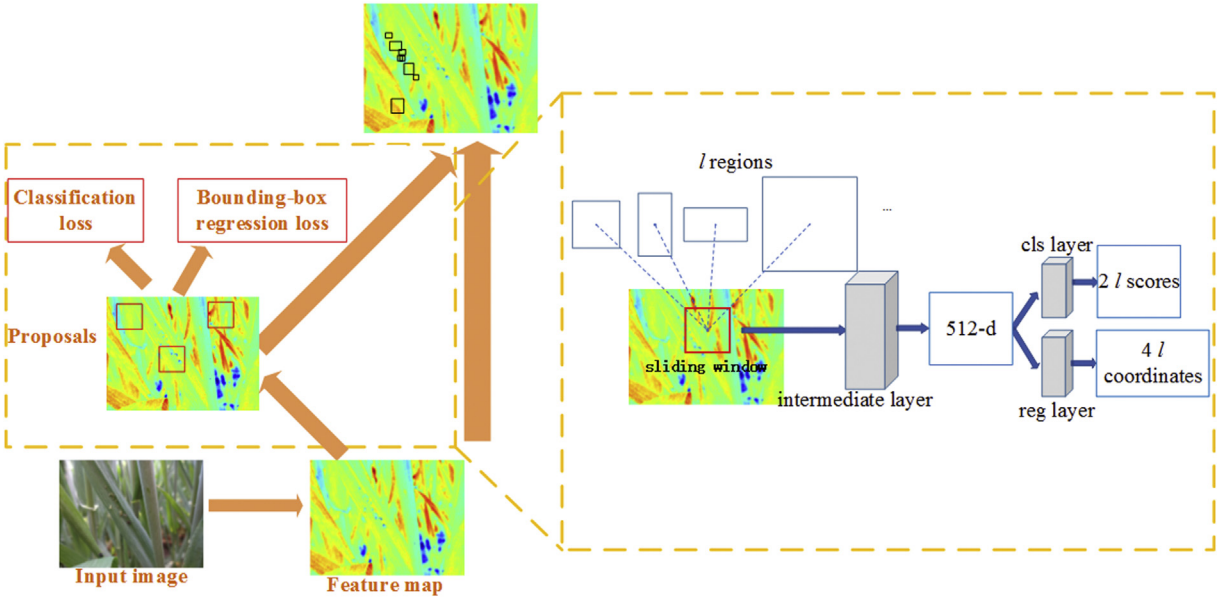

Fig. 7 – Region Proposal Network (RPN) Framework. The training stage is shown in the left part before selective search of these regions. The implementation process is indicated as right dotted box.

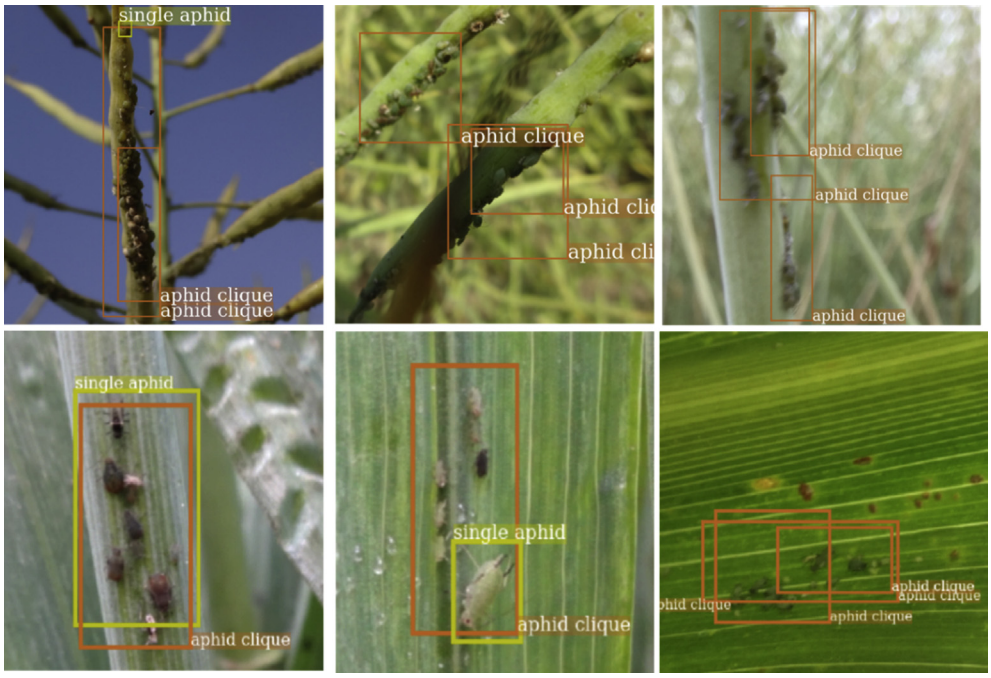

Fig. 8 – The problem of NMS: there are some overlapping bounding box after NMS. The chocolate box is the dense region and the yellow box is a well-separated aphid. It shows that the dense distribution regions boxes contain the other dense regions and some well-separated aphids. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

The Precision-Recall (PR) curve represents the balance between reducing false positives and misdetections. We used area under precision-recall curve to precisely measure the performance of the method, The Precision-Recall (PR) is calculated by:

$$Precision(c) = \frac{TP(c)}{TP(c) + FP(c)}, \quad Recall(c) = \frac{TP(c)}{TP(c) + FN(c)}$$

where c denotes the class, in which TP, FP and FN represent True Positive, False Positive and False Negative samples respectively, so the Precision measures the samples that are incorrectly detected while Recall measures the misdetection samples.

AP is updated for combining localisation and classification tasks together. Given an IoU threshold, the AP is defined as the area under Precision-Recall:

$$AP = \int \text{Precision } d \text{ Recall}$$

in which the Precision measures the samples that are incorrectly detected and Recall measures the misdetection samples.

Table 2 – Detection results training time and AP value (%) of FCNN. The number of dense region images is 7120, During the training phase, the image mini-batch size is set to 2, learning rate is initialised to 0.001 and the number of training iterations is 30000.

Image Size	Training time	AP
576 × 576	1 h 45 min	88.7%
608 × 608	**1 h 52 min**	**90.9%**
640 × 640	2 h 0 min	90.8%
672 × 672	2 h 10 min	90.6%

Mean Average Precision (mAP) is the mean of Average Precision (AP) value among classes and is obtained by taking mean:

$$mAP = \frac{1}{|C|}\sum_{c \in C} AP(c)$$

where c denotes the class.

In order to accurately evaluate the model of FCNN, Frames Per Second (FPS) is used as metric evaluation for computational efficiency, defined as:

$$FPS = \frac{Fn}{T}$$

where Fn is the total number of frames during the time of T.

3. Experiment and discussion

3.1. Experimental settings

In this paper, the proposed method and contrast method are all performed under PyCharm platform. Caffe2 (Jia, Shelhamer, Donahue, Karayev.et al., 2014, pp. 675–678) with Python API 2.7 is used in our experiment and run on 12 GB T P40 GPU and two Intel Xeon E5-2600V3/V4s running 64-bit Ubuntu 16.04 LTS. All models are trained by Stochastic Gradient Descent (SGD) over 2 GPUs with a total of 2 images per mini-batch. We initialise the learning rate to 0.001 and the learning rate will be divided by 10 per 1000 iterations. We are going to consider two different CNN architectures as our prior CNN backbones for feature extraction which are ResNet-50 (He et al., 2016) and Darknet-53 (Redmon, Farhadi, 2018). The

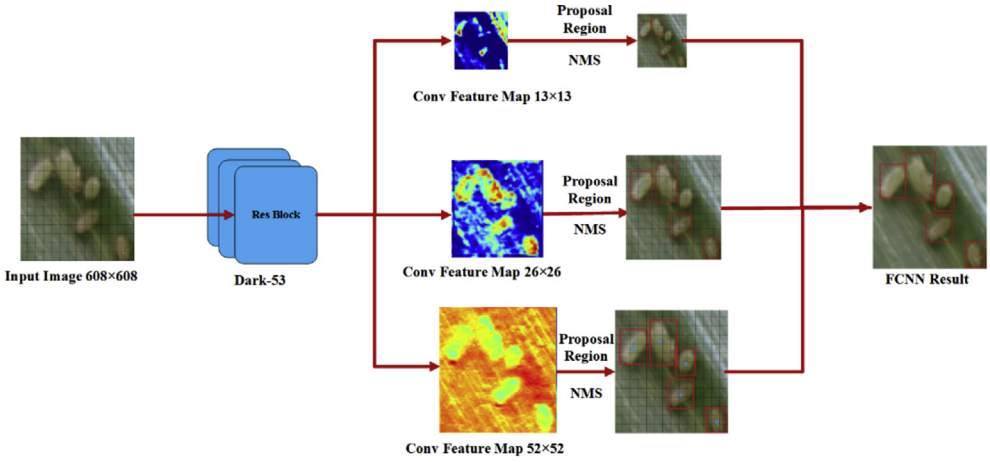

Fig. 9 – Fine Convolutional Neural Network (FCNN) structure. Multi-scale detection is used in FCNN structure. The high-level feature map is used to detect bigger aphids and the low-level feature map is used to detect smaller aphids.

input image resolution is 2160 × 1620. In addition, we adopt transfer learning strategy to initialise the parameters of backbone CNN from that pre-trained on ImageNet dataset and fine-tune them during training in our task.

3.2. *CCNN results*

Three state-of-the-art two-stage methods are used in CCNN architecture. Table 3 presents the detection results of dense distribution regions and aphids not in these regions using these methods. The Faster-RCNN-INMS, R–FCN–INMS and FPN-INMS are the methods using INMS, the others are approaches without INMS. The best result is marked with the bold font. Table 3 shows that compared with the methods without INMS, the methods using INMS have higher detection accuracy. The experiment shows that the INMS method can improve the detection rate. Among all of the approaches, the best detection performance occurs in FPN-INMS using ResNet-50 as backbone, which achieves mAP with 71%. Compared with Faster-RCNN(Ren et al., 2015) and R–FCN(Dai et al., 2016), FPN (Lin et al., 2017, pp. 936–944) is a feature pyramid network structure, so it improves the detection rate of aphids in sparse distribution areas while ensuring the accuracy of dense region detection. It is noteworthy that CCNN is used to detect "dense regions" not the aphids. Table 3 shows that the dense regions have higher detection accuracy than separated aphids, because aphids are much smaller than dense regions. Very small size will weaken the features of aphids in feature maps because max pooling can lead to information loss. As Fig. 10 shows, compared with the detection results when not using INMS, INMS could effectively eliminate overlapping regions.

3.3. *FCNN results*

The dense distribution regions are cropped and rescaled after CCNN, the proportion of aphids in the new distribution region image are significantly enlarged compared with the original image. Therefore, the problem of tiny pest detection can be solved.

Five state-of-the-art methods were studied and the results are shown in Table 4. Compared with other four methods, we firstly observe that FCNN has the highest computational efficiency. Though FCNN gives a lower AP value than FPN, it could achieve a higher speed (four times faster than FPN). Thus, we select FCNN as our Fine Network. The results for FCNN are illustrated in Fig. 11.

3.4. *CFN results*

By combining the results from CCNN and FCNN, we could obtain the detection result of CFN. Table 5 presents the final detection results for CFN with approaches of Faster-RCNN (Ren et al., 2015), DSSD (Fu, Liu and Ranga, 2017), R–FCN(Dai et al., 2016), FPN(Lin et al., 2017, pp. 936–944) using ResNet-50 as backbone and an input resolution of image of 2160 × 1620. The best results are marked in bold font. Among all of the approaches, the best detection performance occurs in CFN using FPN as Coarse Network and FCNN as Fine Network, which achieves AP with 76.8%. The CFN using Faster-RCNN as Coarse Network and DSSD as Fine Network achieves AP with 66.8%, which outperforms FPN by 2.5% AP.

FPN and FCNN are all multi-scale detection structures and, therefore, they have more accurate recognition and detection

Table 3 – Detection Results AP value (%) over IoU 0.5 of CCNN.

Method	Well-separated aphids (AP)	dense distribution regions (AP)	mAP
Faster-RCNN	59.2%	64.1%	61.7%
Faster-RCNN-INMS	**59.7%**	**64.9%**	**62.3%**
R–FCN	62.1%	68.9%	65.5%
R-FCN-INMS	**62.4%**	**69.5%**	**66.0%**
FPN	66.5%	74.6%	70.6%
FPN-INMS	**66.8%**	**75.1%**	**71.0%**

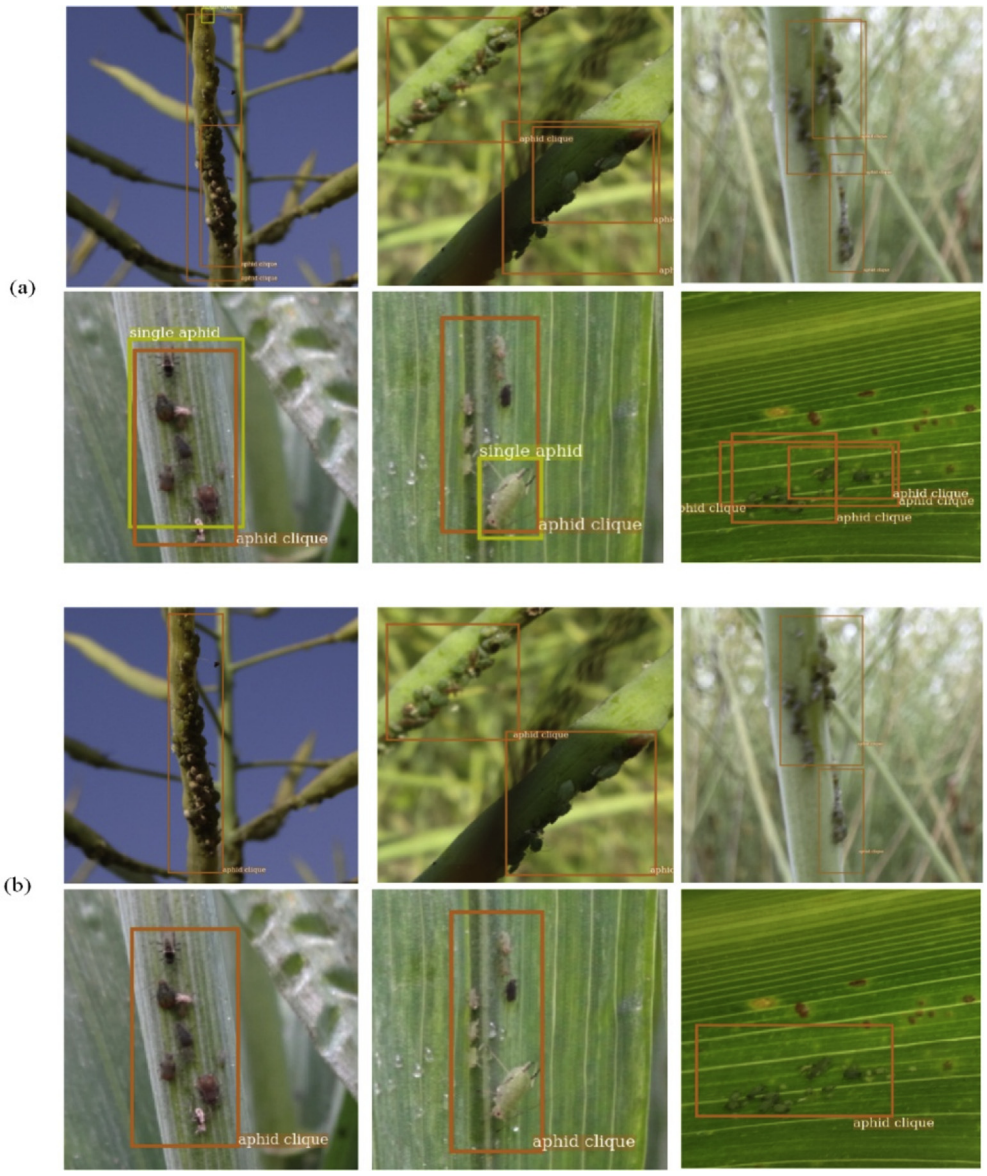

Fig. 10 – (a) The results when not using INMS and (b) when using INMS. There are many overlapping boxes after NMS, which may cause some aphids to be repeatedly detected. The number of overlap boxes could be effectively reduced by using INMS.

Table 4 – Detection results AP over IoU 0.5 and FPS of fine network.

Method	AP	mean FPS
FCNN	**90.9%**	**19**
DSSD	87.6%	8
Faster-RCNN	85.4%	9
R–FCN	89.7%	7
FPN	93.2%	5

results in multi-scale detection. As Fig. 12 shows, compared with the detection results of FPN, R–FCN, DSSD and Faster-RCNN, our proposed method could effectively detect the aphids in dense distribution regions. Thus, it indicates that our approach could improve the detection accuracy of aphids.

3.5. *Results for different crops*

The aphids in our dataset come from three crops: wheat, rape, and corn. Aphids have different distribution densities in different crops, and the aphid Precision-Recall (PR) curves for each crop are shown in Fig. 13. As the PR curve shows, there are some differences in performance between the different crops. Specifically, in our dataset, the number of aphid images for wheat are greater than for other crops (Table 1), and furthermore wheat aphids are more dispersed. Therefore, the wheat has the best result. The distribution of rape aphids is

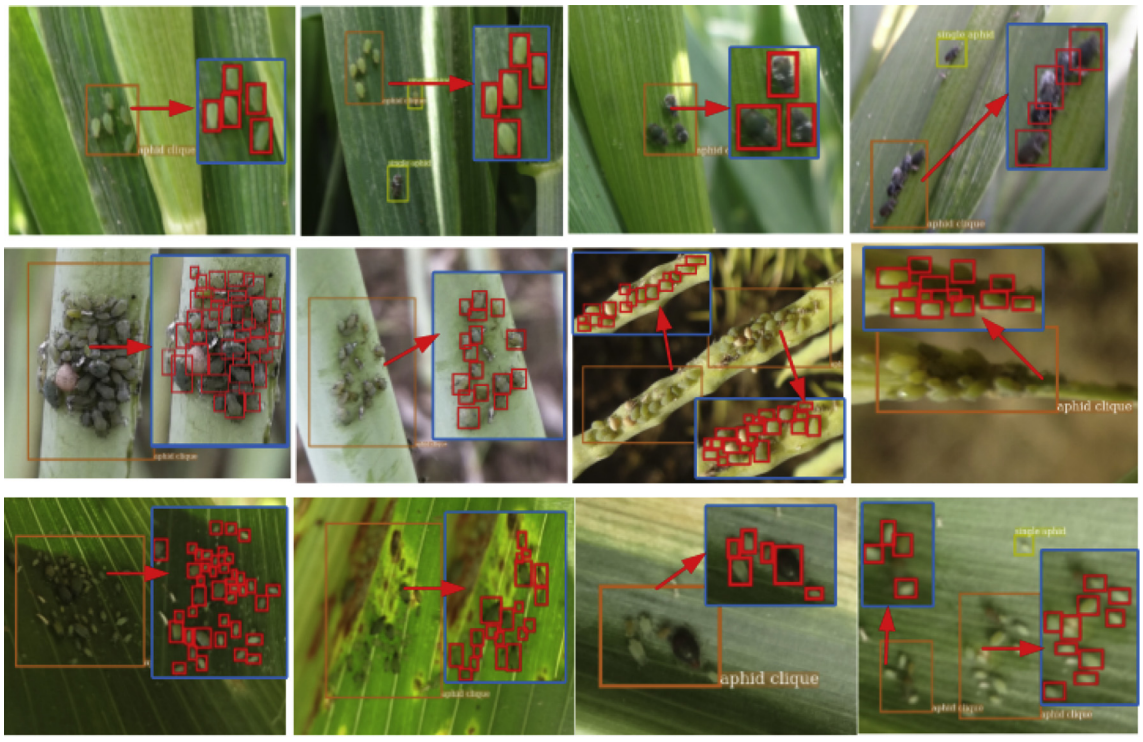

Fig. 11 – Detection results of Fine Convolutional Neural Network (FCNN). The chocolate boxes are the dense regions detected by using CCNN and the red boxes are the aphids detected by using FCNN. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 5 – Detection results AP value over IoU 0.5 of CFN. To improve detection accuracy and computational efficiency, we adopt two-stage approach as Coarse Network and one-stage approach as Fine Network method.

Method	Method of Coarse Network	Method of Fine Network	AP
CFN	Faster-RCNN-INMS	DSSD	66.8%
		FCNN	68.6%
	R–FCN–INMS	DSSD	71.3%
		FCNN	73.5%
	FPN-INMS	DSSD	75.1%
		FCNN	**76.8%**
FPN	–	–	64.3%
R–FCN			63.1%
DSSD			58.8%
Faster-RCNN			55.9%

the densest, and it has the lowest detection accuracy. This indicates that it may be easier to detect aphids in wheat and the aphids in rape may be more difficult to detect.

3.6. *Result for resolution*

Since aphids are tiny pests, the sharpness of the image may affect the detection results. The original size in our datasets is 1440 × 1080. To investigate the effect of image size for performance, the input image sizes were set to 960 × 720 (0.5 multiple), 1140 × 1080 (1 multiple), 2160 × 1620 (1.5 multiple) and 2880 × 2160 (2 multiple) respectively. As Fig. 14 shown, the image size set to 2160 × 1620 has the best result, because the tiny aphid is easily detected by amplification. However, it is not the biggest size that has the best result. When the image scale reaches a certain size, the original image becomes blurred, making it difficult to distinguish the aphids from the background. From Fig. 14, we can see that the resolution of 2880 × 2160 gives the poorest result compared with the other three resolutions. Thus, the sharpness of image can affect the detection result and the biggest image size is not the best.

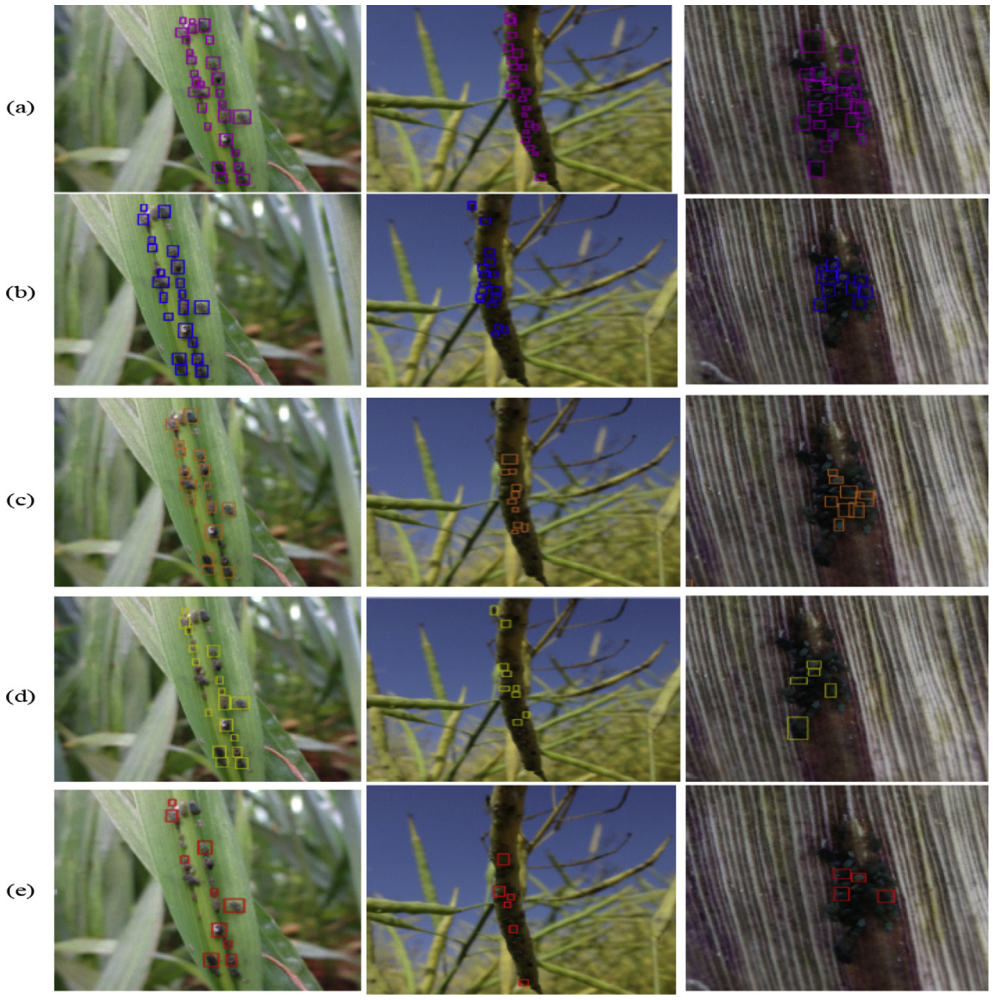

Fig. 12 – The final result with CFN compared with four state-of-the-art methods: (a) The result for CFN (b) The result for FPN (c) The result for R–FCN (d) The result for DSSD (e) The result for Faster-RCNN.

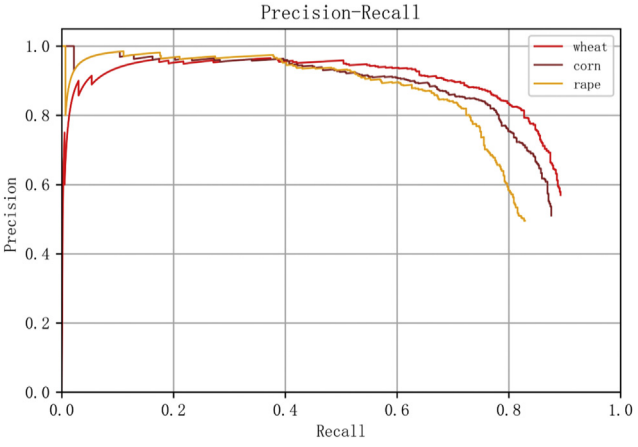

Fig. 13 – Precision-Recall Curve for wheat, corn and rape.

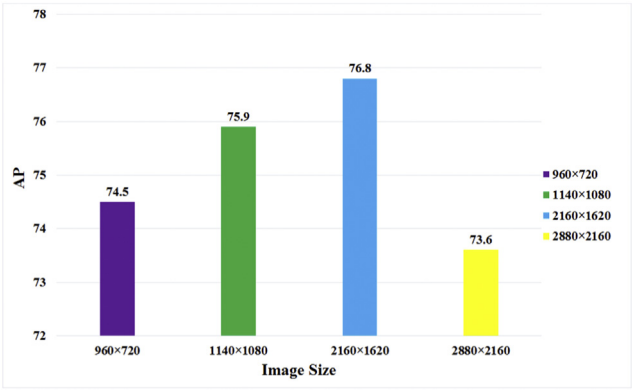

Fig. 14 – Average Precision (AP) for different image sizes.

4. Conclusion

In this work, we present a CNN-based Coarse-To-Fine Network (CFN) for tiny and densely distributed aphid detection in images taken in natural field conditions. In our method, CCNN is responsible for detecting the aphid cliques, in which we develop an improved NMS (INMS) to eliminate overlapping regions. In the next phase, FCNN is proposed to refine the clique regions and obtain the single aphid objects. Final results are obtained by combining the outputs from CCNN and FCNN. Under our enriched wild aphid dataset, CFN could deliver 76.8% AP on aphid detection task, which outperforms four state-of-the-art methods.

The major contributions of CFN are:

(1) A domain specific dataset for aphid recognition and detection in the field containing more than 2000 images and 45424 annotated aphids is published in this paper. Specifically, this dataset is evaluated for high application value in practical aphid monitoring.
(2) A coarse-to-fine network (CFN) towards aphid recognition and detection in dense distribution regions is proposed, which is feasible to apply for practical aphid prevention. Experimental results demonstrate the advantages of the proposed algorithm over other four state-of-the-art approaches.
(3) An improved non-maximum suppression (INMS) method is proposed in this paper. Our method can eliminate the overlapping bounding boxes, which further improve the performance of our method.

In the future, we will target improvements in the generalisation of our CFN and transferring it into generic dense object detection tasks.

Acknowledgements

This work was supported by The National Key Technology R&D Program of China (grant number 2018YFD0200300), the National Natural Science Foundation of China (grant numbers 31401293, 31671586, 61773360), Chinese Academy of Science and Technology Service Network Planning (No.KFJ-STS-ZDTP-048-02) and Fundamental Research Funds for the Central Universities in China (grant numbers ACAIM190101).

REFERENCES

Cai, Z., & Vasconcelos, N. (2018). *Cascade R-CNN: Delving into high quality object detection, 2018 IEEE conference on computer vision and pattern recognition (CVPR)*. https://arxiv.org/abs/1712.00726.

Chen, Y., Li, W., Sakaridis, C., Dai, D., & Van Gool, L. (2018). *Domain adaptive faster R-CNN for object detection in the wild. 2018 IEEE conference on computer vision and pattern recognition (CVPR)*. https://doi.org/10.1109/CVPR.2018.00352.

Dai, J., Li, Y., He, K., & Sun, J. (2016). *R-FCN: Object detection via region-based fully convolutional networks*. https://arxiv.org/abs/1605.06409.

Deng, L., Wang, Y., Han, Z., & Yu, R. (2018). Research on insect pest image detection and recognition based on bio-inspired methods. *Biosystems Engineering, 169*, 139–148. https://doi.org/10.1016/j.biosystemseng.2018.02.008.

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.

Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture, 137*, 52–58. https://doi.org/10.1016/j.compag.2017.03.016.

Faithpraise, B., & Chatwin, C. (2013). Automatic plant pest detection & recognition using k- means clustering algorithm &correspondence filters. *International Journal of Advanced Biotechnology and Research, 4*(2), 189–199.

Fu, C. Y., Liu, W., Ranga, A., & Tyagi, A. (2017). *DSSD : Deconvolutional single Shot detector*. https://arxiv.org/abs/1701.06659v1.

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

Hilbe, J. M. (2009). *Logistic regression models*. CRC Press. https://doi.org/10.1002/9780470012505.tal017.

Huang, R., Xu, B., Schuurmans, D., & Szepesvari, C. (2016). *Learning with a strong adversary*. Computer Science. https://arxiv.org/abs/1511.03034.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. https://arxiv.org/abs/1502.03167v2.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). *Caffe: Convolutional architecture for fast feature embedding, proceedings of the 2014 ACM conference on multimedia*. https://doi.org/10.1145/2647868.2654889.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems*, 25(2). https://doi.org/10.1145/3065386.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436.

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

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016a). SSD: Single Shot multibox detector. In *European conference on computer vision* (pp. 21–37). ECCV). https://doi.org/10.1007/978-3-319-46448-0_2.

Liu, T., Chen, W., Wu, W., Sun, C., Guo, W., & Zhu, X. (2016b). 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, Z., Gao, J., Yang, G., Zhang, H., & He, Y. (2016c). Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. *Scientific Reports, 6*, 20410. https://doi.org/10.1038/srep20410.

Neubeck, A., & Gool, L. J. V. (2006). *Efficient non-maximum suppression, 18th international conference on pattern recognition (ICPR 2006)*. https://doi.org/10.1109/ICPR.2006.479.

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

Redmon, J., & Farhadi, A. (2018). *Yolov3: An incremental improvement*. https://arxiv.org/abs/1804.027678.

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.

Tian, R., Chen, M., Dong, D., & Li, W. (2016). Identification and counting method of orchard pests based on fusion method of infrared sensor and machine vision. *Transactions of the Chinese Society of Agricultural Engineering, 32*(20), 195–201. https://doi.org/10.11975/j.issn.1002-6819.2016.20.025.

Wang, X., Zhang, C., Zhang, S., & Zhu, Y. (2018). Forecasting of cotton diseases and pests based on adaptive discriminant deep belief network. *Transactions of the Chinese Society of Agricultural Engineering, 34*(14), 157–164. https://doi.org/10.11975/j.issn.1002-6819.2018.14.020.

Wen, C., Guyer, D. E., & Li, W. (2019). Local feature-based identification and classification for orchard insects. *Biosystems Engineering, 104*(3), 299–307. https://doi.org/10.1016/j.biosystemseng.2009.07.002.

Xie, C., Zhang, J., Li, R., Li, J., Hong, P., Xia, J., et al. (2015). Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Computers and Electronics in Agriculture, 119*, 123–132. https://doi.org/10.1016/j.compag.2015.10.015.

Yang, H., Liu, W., Xing, K., Qiao, J., Wang, X., Gao, L., et al. (2010). *Research on insect identification based on pattern recognition technology*. 2010 Sixth International Conference on Natural Computation. https://doi.org/10.1109/ICNC.2010.5583156.

Yao, Q., Xian, D., Liu, Q., Yang, B., Diao, G., & 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.

Yuan, Y., & Hu, X. (2016). Random forest and objected-based classification for forest pest extraction from UAV aerial imagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1093–1098. https://doi.org/10.5194/isprsarchives-XLI-B1-1093-2016. XLI-B1.

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). *Single-shot refinement neural network for object detection. 2018 IEEE conference on computer vision and pattern recognition (CVPR)*. https://doi.org/10.1109/CVPR.2018.00442.

Zhang, E., & Zhang, Y. (2016). *Average precision. Encyclopedia of database systems*. https://doi.org/10.1007/978-1-4899-7993-3_482-2.