Computers and Electronics in Agriculture 187 (2021) 106268

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Convolutional neural network based automatic pest monitoring system using hand-held mobile image analysis towards non-site-specific wild environment

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

[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

ARTICLE INFO

Keywords:
Convolutional Neural Network
Rice Planthopper Search Network
Sensitive Score Matrix

ABSTRACT

Due to cost-effectiveness and efficient automation, image analytic based automatic pest monitoring techniques are widely utilized in specialized control of pests in the agricultural crops industry. They could achieve good recognition performance in certain species of pests in site-specific environment, but are sensitive to content and characteristics of image like appearance variances and clustered background. For pests with small size and indistinct features like rice planthopper, it is hard to manually and timely select suitable features. In this paper, we propose an effective CNN based automatic hand-held mobile pest monitoring system to detect and count rice planthoppers. A rice planthopper search network (RPSN) approach is proposed for automatically extracting multiple high-quality proposal regions from large-scale pest images with tiny objects. Additionally, sensitive score matrix (SSM) is employed to further enhance the performance of classification and bounding box regression. The experimental results under the proposed approaches evaluating three types of density pest images show that our system performs well on detecting rice planthoppers in non-specific wild environment with recognition recall up to 91% in industrial circumstance, which outweighs the state-of-the art approaches.

1. Introduction

Specialized pest control and effective disease prevention for crops industry is always a highly-priority agricultural issue in all over the world, especially in many developing countries (Santangelo, 2018). In order to effectively inspect and prevent the occurrence of pests, many advanced technological solutions have been developed and applied in nowadays agricultural crop industry, like novel chemical pesticides (Bures et al., 2006), image analytic systems (Liu et al., 2017), automatic adjustable spraying device (Berenstein and Edan, 2018), remote sensing (Luo et al., 2013), etc. Due to great cost-effectiveness and efficient automation, image analytic based pest monitoring approaches are widely utilized in practical crops monitoring systems. Typically, their applications need to employ either hand-held or stationary pest monitoring devices in the fields for collecting massive pest image datasets, and then apply advanced image processing algorithms (Ding and Taylor, 2016; Zayas and Flinn, 1998; Cho et al., 2007; Wang et al., 2013; Aragón et al., 2014; Wu et al., 2014; Yan et al., 2017; Enes et al., 2020) to automatically detect and recognize pest associated image data for supporting decision-making and prediction in various agriculture applications. But these two types of pest monitoring facilities are usually utilized into different working environment towards different species of pests.

Stationary based pest monitoring approaches are based on the use of insect trap conveniently spread over the specified control area and applying hand-crafted high-quality features and advanced image processing algorithms in processing these trap images. They perform well in detection and classification of certain type of pest in site-specific environments. But the quality of features is driven by many factors like illumination, movement of the trap, decay or damage to the insect, etc. The manual selection of suitable features among these trap images is a time-consuming and laborious task. Also, considering the cost of hardware deployment, stationary based automatic pest monitoring systems are mostly utilized into small or middle scale site-specific environments.

For non-site-specific wild environment, hand-held device based pest monitoring approaches are more cost-effective and flexible than stationary ones. But their image quality is usually worse and complex than the trap images, which contains some pest with small size and indistinct

[*] Corresponding author.
E-mail address: cjxie@iim.ac.cn (C. Xie).

https://doi.org/10.1016/j.compag.2021.106268
Received 22 October 2020; Received in revised form 3 April 2021; Accepted 5 June 2021
Available online 23 June 2021

features. Many researchers have done some work in this field. Wang et al. (2013) presents a mobile smart device based vegetable disease and insect pest recognition method with region-labelling based pest extraction and morphology based object separation algorithms. They get good recognition performance with high efficiency, but only work in simple vegetable disease and insect. Similarly, Aragón et al. (2014) presents a method that uses Mahalanobis distance in the detection of pests. It exaggerates the role of small variables and is easily influenced by the environment. Aforementioned researches have shown that combining multiple complementary features such as texture, color, shape and spatial distance is more valuable than single features in detecting pests. While above advanced image processing techniques in mobile pest monitoring applications enable great success in recognition and classification of certain type of insect, one key limitation of these approaches appears that most researchers focus on increasing the recognition performance of certain type of insect by manually introducing new features or employing new machine learning algorithms, yet paying more attentions on developing practical useful and robust automatic pest monitoring systems. In general, due to the numerous similarities between pest feature and environment texture in real application scenarios like non-specific wild environment, it is challenging to manually and timely select suitable hand-crafted features among various pest images and address this issue through traditional methods, so that there is a trade-off between recognition performance and efficiency on many hand-held or mobile pest monitoring systems.

Recent developments in deep learning technology help us handle these problems (Deutsch and He, 2018; Pourbabaee et al., 2017; Yaseen et al., 2018). Instead of using hand-crafted features to recognize objects, Convolutional Neural Network (CNN) extracts feature from the highly semantic information from the raw image without using manual feature, which address the issues that conventional methods are difficult to obtain complex hand-crafted features in the unstructured environment (Girshick et al., 2014; Xiao et al., 2012). Inspired by this, this paper attempts to investigate the possibilities of solving in-field pest detection problem by CNN methods and improve performance and efficiency of automatic smartphone or mobile pest monitoring system in non-site-specific environments with innovative CNN techniques. Existing CNN method always attempts to search objects by sliding windows through each pixel, but this method is disturbed by some noises in the field. Some traditional methods point out that the local features are helpful to solve this issue. Inspired by this, we consider to bind the local information of the object to the specific CNN channel, therefore, the different CNN channel can learn the local feature of the object. Particularly, we target at large-scale insect pest with small size in wild environment and use rice planthopper as a case.

Thus, in this paper, we propose an effective CNN based automatic mobile pest monitoring system to accurately detect and count rice planthoppers. This system is built on 15 years term of industrial-level multiple insects' data with over 5 million images. This system first proposes a rice planthopper search network (RPSN) under CNN architecture for automatic extraction of rice planthopper region and features. Then we design a sensitive score matrix (SSM) to effectively integrate the features of different local position in a rice planthopper and the regions obtained in the first stage are fine-tuned. Besides, the number of rice planthoppers is counted. The experimental results under the proposed approaches evaluating three types of density pest images show that our system performs well on detecting rice planthoppers in non-specific wild environment and improves greatly compared with the traditional computer vision method and human annotations. The major contributions of this paper are as follows:

(1) A novel convolutional neural network based automatic hand-held mobile pest monitoring system for non-specific-site environment is designed and developed. This system enables accurately and effectively detect and count rice planthopper in the wild environment.
(2) A rice planthopper search network (RPSN) approach integrating with a new sensitive score matrix (SSM) is proposed for automatically extracting multiple high-quality local features from large-scale pest images with tiny objects, and fusing them with better recognition performance.
(3) A comprehensive and in-depth experimental evaluation on practical industry level large-scale practical real wild environment dataset (over 15 years over 5 million) is provided for verifying the usefulness and robustness of proposed system and approaches.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 give an overview to our pest monitoring system and the technical details of our system are introduced in Section 4. Then Section 5 describe the system implementation and discuss the experimental results. Finally, we conclude this paper in Section 6.

2. Related work

Typical approaches for pest recognition focus on agricultural object identification using either computer vision or image processing for insects and plants, including two key stages (Dyrmann et al., 2016): (1) feature extraction that extracts information as feature vectors from images. (2) pattern recognition that trains a model to classify categories of input images. Some researchers used branch length similarity (BLS) entropy calculated from the shape of butterfly as the basis to identify the butterfly by neural network (Kang et al., 2012). To address the limitations of single feature, they improve the effect of algorithm by combining butterfly shape viewed from different angles (Kang et al., 2014). Xia et al. (2014) identified and classified the three common pest species collected in the greenhouse with color features extracted by Mahalanobis distance. Bearup et al. (2015) proposed a mean-field mathematical model of pest trapping based on diffusion equation to detect and count the pests in the trap. In addition, there are many literatures (Wen et al., 2015; Liu et al., 2016a; Liu et al., 2016b; Espinoza et al., 2016; Ding and Taylor, 2016; Sun et al., 2017; Ebrahimi et al., 2017) focusing on constructing the insect appearance models through some local features. However, these detection methods would likely take up large computing resources resulting in poor performance on highly similar pests because these local features are sensitive to changes in translation, rotation and scale.

Although the aforementioned pest detection systems had great progress, most of them adopted image derived from trap with high-intensity lighting and fixed position. Another disadvantage is that most features (such as color, texture, HOG, Gabor and Scale Invariant Feature Transform (SIFT)) used in these models are hand-crafted, which leads to the difficulty in detecting pest images from certain angles and views.

Fortunately, CNN can tackle these problems by extracting semantic information from images, which means we don't need to design pest features in complicated environments. Compared to the majority of previous image processing algorithms, convolutional neural network is able to extract global features and local features from the original image to recognize pests. In this case, convolutional neural network has been applied successfully in face recognition (Taigman et al., 2014; Schroff et al., 2015), plant recognition (Grinblat et al., 2016; Dyrmann et al., 2016; Lu et al., 2017), general object detection (Do et al., 2017; Hou et al., 2017) as well as pest detection under simple and stationary devices (Liu et al., 2019; Loris et al., 2020) and achieved state-of-the-art performance. These convolutional neural network methods originate from a pioneering work on object detection (Redmon et al., 2016), which proposed capability of selective search as well as feature extraction in convolutional neural network and achieved the good recognition performance. Besides, deep CNN architecture could achieve automatic feature extraction.

However, in terms of conventional CNN methods, they try to employ the fixed scale cell to match all targets, which is not appropriate for pest detection caused by the size of pests varies greatly at growth phase (Redmon et al., 2016). In addition, the fixed scale cell in CNN could extract false positive proposal regions. Yu et al. (2016) employs iou loss to remove these false positive samples by calculating the difference between the sliding window and the target, but it is difficult to ignore the impact of noise in the unstructured environment. Thus, we attempt to investigate the possibilities of solving in-field pest detection problem by CNN methods and propose an automatic hand-held mobile pest detection system based on deep CNN architecture targeting at the tiny size pest in wild environment.

3. System overview

The proposed system work was inspired on using 'sliding window' in convolutional neural network (CNN) for insect classification and proposes a region-based CNN detection technique for Rice Planthopper detection shown in Fig. 1, in which the components surrounded by red bold dotted box will go through the prior training phase on training images before test phase.

In addition, we design an image capture equipment to build our task-specific dataset Rice Planthopper Dataset in Anhui 2018 (RPDA2018) involving 10,267 rice planthoppers with 11 different challenging environments, and some of insects' images are shown in Fig. 2. Note that these sample images are partially taken from images of our dataset. Images are firstly input into a CNN backbone and the output is so-called 'feature maps'. Then our system employs Rice Planthopper Search Network (RPSN) to compute probability rice planthopper regions for each feature map. These regions could distinguish roughly between insects and non-insects so they indicate potential insects' positions.

At the second stage, the sensitive score matrix is created to detect and fine-tune the rice planthopper. In many other pest detection systems, feature extraction for complete pest lead to poor performance which is caused by the similarity of texture, shape and color of pest. Inspired by the research of image segmentation algorithm, firstly, regions obtained from the first stage are divided into several parts. Meanwhile, data dimensionality of each part is reduced to avoid affection of noise, thus improving the robustness of the system. Then, the sensitive score matrix is calculated which represents the confidence score of each part of region corresponding to each position of pest. Finally, detection results are voted by scores of each position.

4. Materials and methods

4.1. Rice planthopper dataset in Anhui 2018 (RPDA 2018)

For agriculture insect identification, there exist a few open databases released such as Butterfly Dataset. However, to our best knowledge, few suitable datasets that cover in-field insects are released while our purpose is to detect rice planthopper in different kinds of environments. As a result, we establish a database for our rice planthopper detection task. This system is built on 15 years term of industrial-level multiple insects' data with over 5 million images datasets. The equipment for capturing images of multiple insects is designed in our task shown in Fig. 3.

The in-field rice planthopper images analyzed in this paper were collected in the Anhui province, a typical rice-producing region in China. All the images were captured by independent research and development device called rice planthopper intelligent collection equipment, whose structure and usage are shown in Fig. 3 as well as Fig. 4. As for image acquisition, CCD camera whose parameters were set to 4 mm focal length with an aperture of f/3.3. It should be noted that only one RGB color image (1440*1080) from each time series is labelled and used in this paper, therefore the labelled rice planthopper are unique.

Through the use of rice planthopper intelligent collection equipment, we collect numerous rice planthopper images with 10,267 rice planthoppers, and record the temperature, humidity and geographic information of rice planthopper images. The rice planthopper images collected in Anhui Province is named as rice planthopper dataset in Anhui 2018(RPDA2018). Note the fact that RPDA2018 has been randomly divided into 10 folds, where nine-tenths are used as training set while the residual part is used as test set.

4.2. Rice planthopper search network (RPSN)

Our detection architecture is a region-based CNN method, in which the propose RPSN module is employed to search potential locations of pest that follows base CNN module. In terms of conventional CNN methods e.g. selective search and edge boxes, they try to employ the fixed scale cell to match all targets, which is not appropriate for pest detection caused by the size of pests varies greatly at growth phase. Indeed, the proposed RPSN establishes sliding windows of different scales on each pixel in the image instead of simple grid, which is more likely to search each object. We first calculate the category to which the sliding window belongs, if sliding window belongs to the object, we need to further compute the regression parameters of bounding box. The object in the image collected by hand-held equipment can be retrieved completely, and the misdetection of pests in unstructured environment

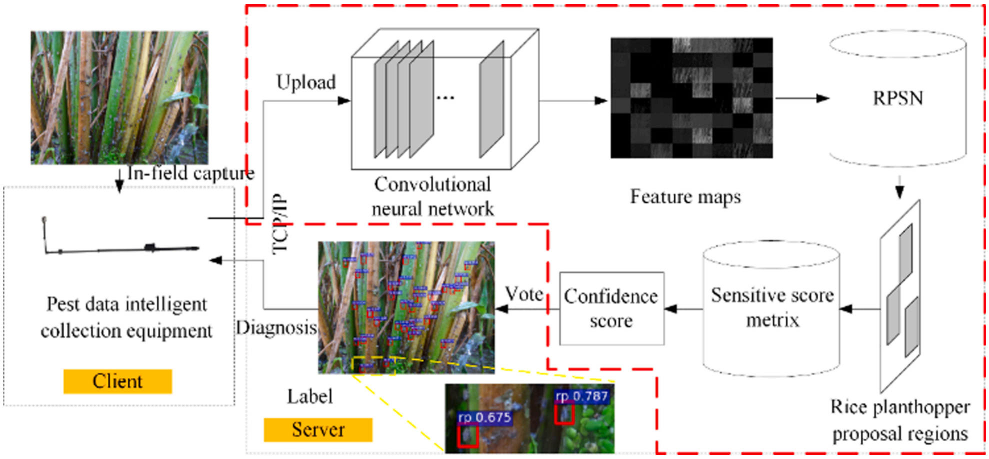


Fig. 1. Technical pipeline of our system architecture.

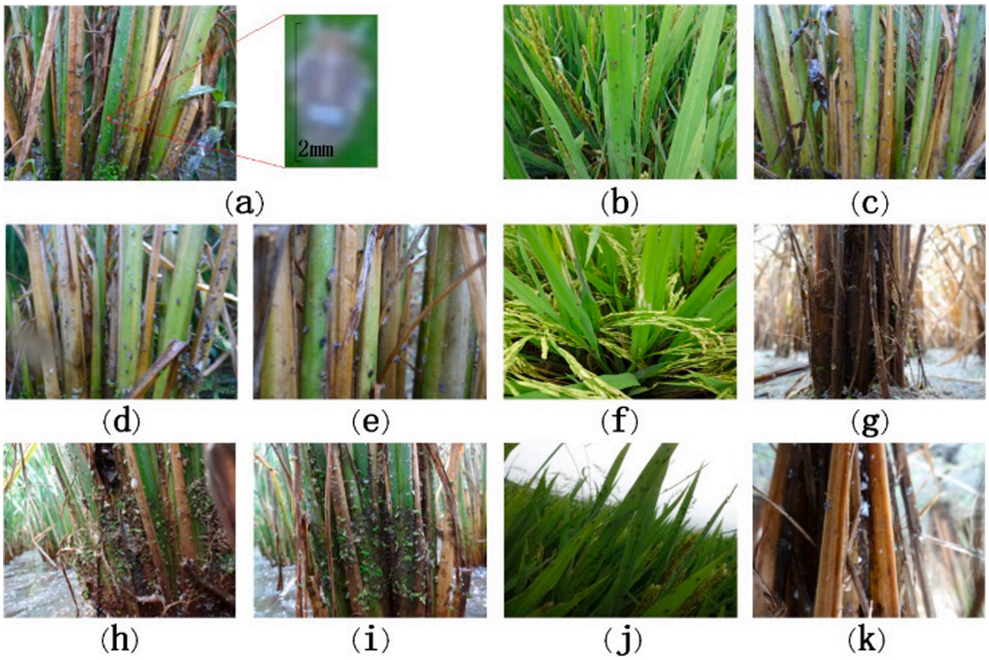


Fig. 2. Some typical challenges in the rice planthopper detection.

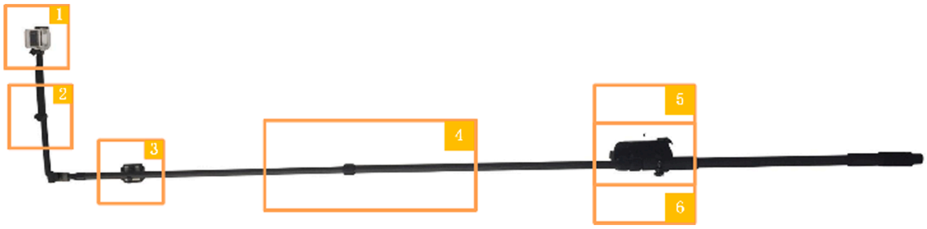


Fig. 3. Rice planthopper intelligent collection equipment. (1) CCD camera; (2) soil moisture sensor; (3) ambient temperature and humidity sensor; (4) carbon fiber telescopic rod; (5) mobile client; (6) global positioning system.

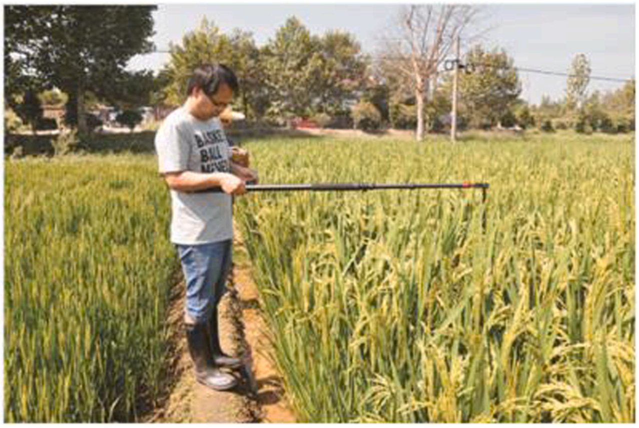


Fig. 4. The usage of rice planthopper intelligent collection equipment in the field.

can be reduced.

As shown in Fig. 5, for each pixel in the feature map which fed into RPSN, numerous sliding window is placed to search rice planthopper. Generally, we use 3 different aspect ratios of {1 × 1, 1 × 2 and 2 × 1} and scales of {32px, 64 px and 128 px} by experience which also mean 3 × 3 rice planthopper regions are predicted for a single pixel. For a feature map of a size w*h in convolutional layer, there are totally 3 × 3 × w × h rice planthopper regions. The shape of these sliding windows is between (32px, 32px) and (256px, 128px).

In the training stage, the overlap ratios (IoU) between predicted sliding windows and ground truth in the training set are calculated. Once the IoU greater than a given threshold thr, the sliding window is considered as a correct proposal region, whose classification score should be supervised to 1. The rest of the samples fail to detect pests and their classification scores should be set to 0. Additionally, the pest location regression for those correct proposal regions could be obtained as shown below:

$$\Delta X = (X_{real} - X_{predict})/X_{predict}$$
$$\Delta Y = (Y_{real} - Y_{predict})/Y_{predict}$$

$$\Delta W = \log\left(W_{real}/W_{predict}\right)$$
$$\Delta H = \log\left(H_{real}/H_{predict}\right)$$

where $X_{predict}$, $Y_{predict}$, $W_{predict}$ and $H_{predict}$ indicate the centerness and scale of predicted rice planthopper region. X_{real}, Y_{real}, W_{real} and H_{real} denote the location of ground truth.

The classification and regression for rice planthopper are accomplished in the box-classification layer as well as box- regression layer in Fig. 5. Generally, the number of sliding windows in feature map is donated as k. Therefore, 4 k coordinates ($X_{predict}$, $Y_{predict}$, $W_{predict}$ and $H_{predict}$) of predicted rice planthopper regions are produced by box-regression layer and k scores (possibility of rice planthopper) are outputted by box-classification layer.

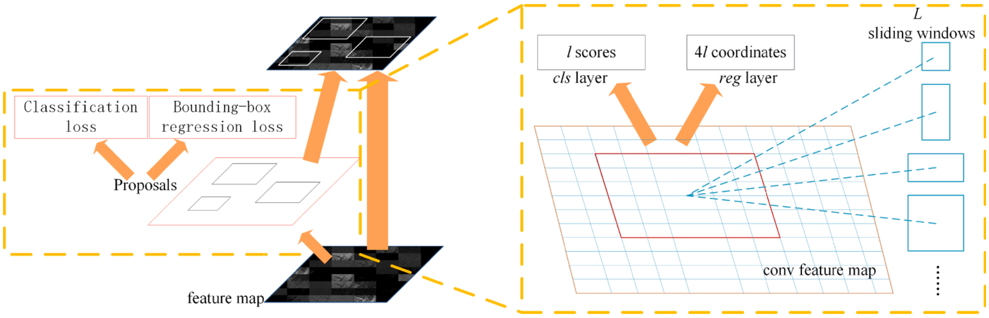


Fig. 5. Pipeline of Rice Planthopper Search Network.

4.3. Sensitive score matrix (SSM)

Generally, in RPSN module, we iterate through each pixel and establish the sliding window in order to reduce the false negatives in unstructured environment. However, it leads to a large number of false positives in proposal regions to some extent and reduces the classification effect of proposal regions. Thus, we need to further confirm the classification results and refine the bounding box of these proposal regions.

Specifically, after getting the proposal regions of rice planthopper from RPSN, position-sensitive feature maps are constructed by $r^2(c+1)$-channel convolutional layer as shown in Fig. 6, in which c + 1 and r denotes the number of classes (c classes object and background) and the shape of kernel, respectively. Each channel in the position-sensitive feature map is sensitive to the corresponding position of the object after learning to identify rice planthopper. For instance, the last c + 1 channels (deep blue block) in Fig. 6 is sensitive to the top-left region, while the first c + 1 channels (red block) at the front is sensitive to the bottom-right region. After that, our sensitive score matrix is obtain in which each block responses from one position-sensitive feature map by RoI pooling (Girshick, 2015).

Due to the fact that the network has one-to-one mapping of local sensitive score in SSM to local position-sensitive feature map, SSM is a matrix with $r^2 r \times r \times (c+1)$ bins. For sensitive score matrix of a size $w^{'} \times h^{'}$, in which each block is of a size $w^{'}/r \times h^{'}/r$ Here sensitive score matrix represent the confidence score of local patches in feature map corresponds to different positions of rice planthopper as described earlier.

In order to obtain the final classification result, we need to aggregate features in sensitive score matrix. SSM vote on each rice planthopper proposal regions by global average pooling on the r^2 position-sensitive scores, which produce c + 1 classification confidence for each rice planthopper proposal regions.

We also address the issue about proposal region regression of rice planthopper in a similar approach. The only difference between bounding-box regression process and classification process in rice planthopper images is that bounding-box has four dimensionality which are position of center point (X, Y) and scale of bounding-box (W, H). For one specific sliding window generated by RPSN, therefore, we obtain $4r^2(c+1)$-channel feature as the parameters of bounding box. By RoI pooling on position sensitive score map and global average pooling SSM, 4(c + 1) regression predictions are produced. Rice planthopper is selected as the object in our case, so that c = 1 and 8 regression predictions are obtained. Note that half of them are location which supervised to ground truth, and rest of labels do not need to calculate because they belong to negative samples (background). Similar to the regression process in RPSN, 4 boundary prediction for each bin is defined as $V = (V_X, V_Y, V_W, V_H)$, in which (V_X, V_Y) denotes the center point difference between the prediction and ground truth and (V_W, V_H) demonstrates the scaling of width as well as height.

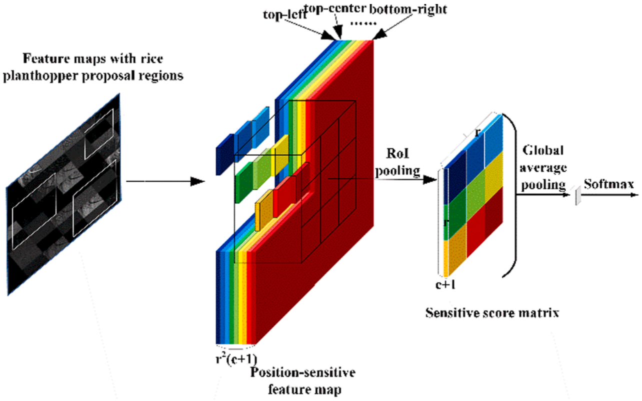


Fig. 6. Processing pipeline of constructing sensitive score matrix (SSM).

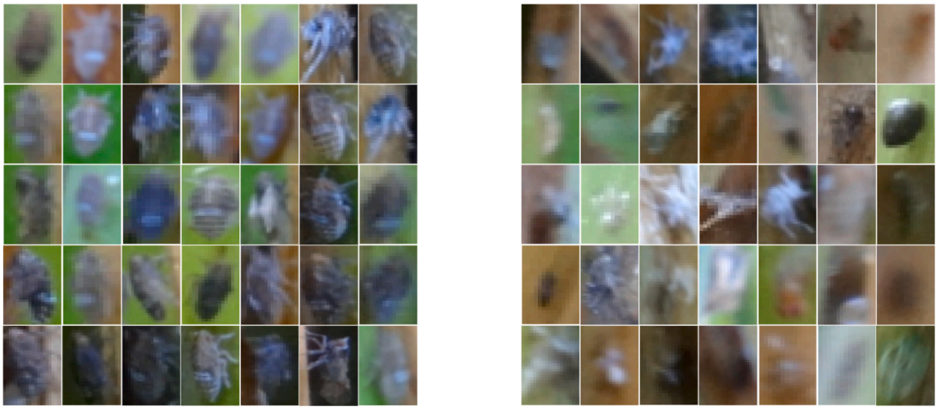
5. Experimental results and discussion

5.1. Experimental settings

In order to verify that our method could be used in our dataset for detecting insects, we build some experiments to evaluate the performance of our model. Our codes are based on Caffe with Python API and run on a GeForce GTX TITAN X GPU. Some experiment details are given in this section. The RMSprop is chosen as our optimizer with momentum 0.9, which updates parameters based on one mini-batch at each iteration. This optimizer could partly keep the update gradient at previous iteration and fine-tune the final gradient considering the current mini-batch. In order to avoid over-fitting problem, we utilize dropout method with 0.5 dropout rate as well as early-stopping strategy to select the best training iteration. 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 50,000 iterations. In addition, mini-batch size is set to 1 and the number of region proposals of every training example is at least 128. During training phase, we adopt approximate joint training strategy to speed up the training process, which trains the RPSN module jointly with the SSM module.

Here, positive patches refer to patches which are labeled by insect experts manually, while negative patches are considered as the area without rice planthoppers. It is obvious that the area without rice planthoppers is much larger than that containing rice planthoppers. Therefore, we set the appropriate threshold to obtain negative samples, whose amount roughly matches that of positive samples. Fig. 7 shows positive patches and negative patches extracted from the training set.

Before training our model, we utilize some data augmentation strategies to expand the data diversity. Firstly, considering the variability of the shooting angle and the rotation invariability of the plant images, we rotate and flip the original images with unchanging image resolution. We horizontally and vertically flip the raw images to obtain the other 2-fold images as well as the use of image rotation results in 7-fold increase at data diversity (image is rotated by 45 degree at each time). Therefore,



(a) Positive patches

(b) Negative patches

Fig. 7. Positive patches and negative patches used in our experiments.

a total of 24-fold plant images can be obtained. Additionally, the captured plant images are cropped into different sizes which could further enlarge the data diversity.

We consider the modified ZF, VGG-16, ResNet-50 and ResNet-101 (Robinson et al., 2007; Zeiler and Fergus, 2014; Simonyan and Zisserman, 2014; He et al., 2016) as our backbone models. Note that we remove the fully connection layer and softmax regression of these networks. Moreover, we adopt RPSN and SSM to improve the effectiveness of these four conventional CNN models. The pre-trained models of four baseline architectures are trained on the ImageNet ILSVRC2012.

For a given image of rice with rice planthoppers, the performance of the experiment is evaluated by precision and recall from results which obtained from experiment. For a given threshold thr in RPSN, the description of evaluation metrics is defined as:

$$\begin{array}{ll} TruePositiveAmount(TP)_{thr} = N_{pred} & \text{s.t.IoU(pred, gt)} \geq thr \\ FalsePositiveAmount(FP)_{thr} = N_{pred} & \text{s.t.IoU(pred, gt)} < thr \\ FalseNegativeAmount(FN)_{thr} = N_{gt} & \text{s.t.IoU(pred, gt)} < thr \end{array}$$

$$precision_{thr} = \frac{TP_{thr}}{TP_{thr} + FP_{thr}} recall_{thr} = \frac{TP_{thr}}{TP_{thr} + FN_{thr}}$$

$$mAP = \int_0^1 precision_{thr} d(recall_{thr})$$

where N_{pred} and N_{gt} denote the amount of prediction and object bounding box. Generally, as a hyperparameters in CNN, domain definition of threshold thr can be selected between 0 and 1. By recording precision and recall under each threshold in RPSN, the precision-recall curve is plotted which represents the balance between precision and recall. We used mean average precision (mAP) to precisely measure the performance of the method, the higher the precision for a certain recall, the better the model performance. mAP is calculated by the formula:

5.2. Result analysis

Fig. 8 corresponds to tendency of training loss as the iterations of model increases (Tabor and Spurek, 2014). One can observe that the loss of our system tends to the minimum after 40 K iterations and remains in the subsequent iterations. In order to obtain the mAP as high as possible without over-fitting the network, the training stage was stopped at 40 K iterations and each model is recorded every 10 K iterations to reduce the huge memory footprint of too many models. Fig. 9 shows the precision-recall curve and performance comparison for different convolutional

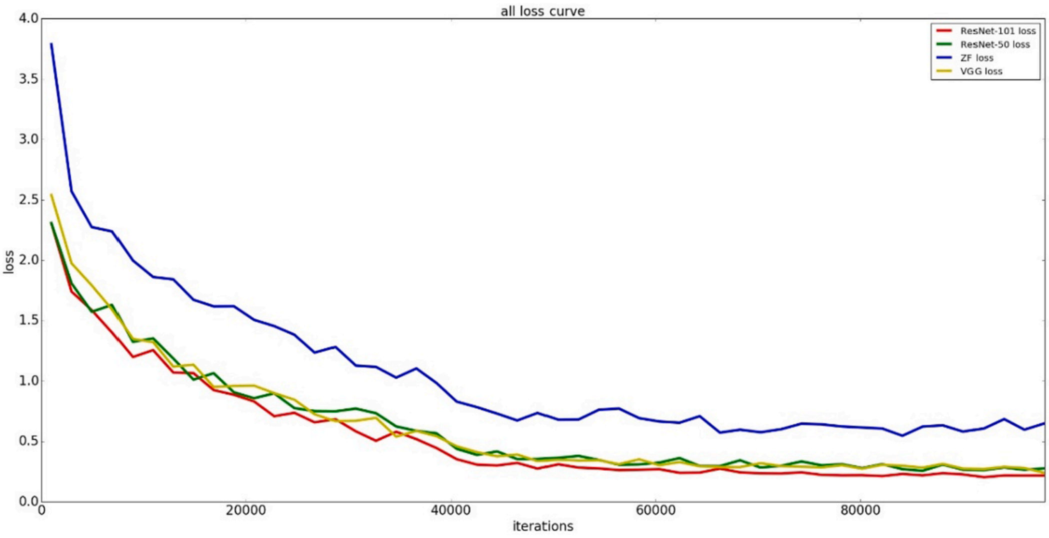


Fig. 8. Training loss of four modules used in our experiments.

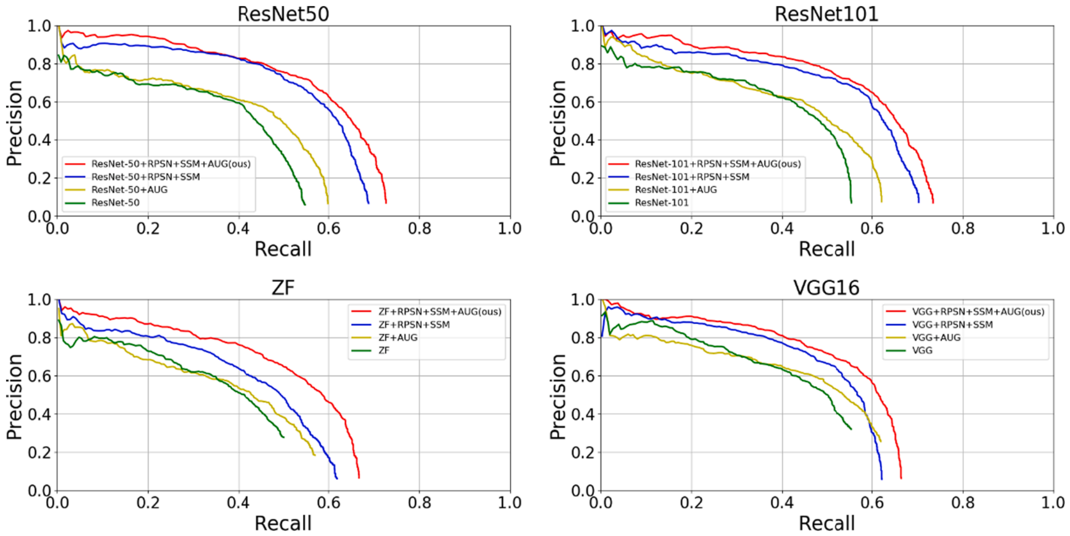


Fig. 9. Precision-recall curve for different models.

neural network models. Different blocks in Fig. 9 represent the different architectures. The ‘ResNet-50’, ‘ResNet-101’, ‘ZF’ and ‘VGG’ are four baseline CNN models. ‘RPSN’ as well as ‘SSM’ are our proposed method mentioned in Section IV and ‘AUG’ indicates the mirrored rotational augmentation used in data augmentation. Precision-recall curve demonstrates the performance of the system when a trade-off between precision and recall. Obviously, the closer this curve is to the upper right corner the better. As can be seen from Fig. 9, CNN models with RPSN and SSM (our method) have the optimal effect and greatly improvement compared with other algorithms.

In order to illustrate the performance of each algorithm more clearly, the mAP and operation time of each model are shown in Table 1. We firstly observe the results of four baseline CNN models with RPSN and SSM (our method) and conventional CNN models. Our method based on ResNet-101, ResNet-50, ZF and VGG could improve the mAP by 14.5%,18.6%, 13.4% and 8.8% respectively compared to those CNN backbones without our method. Then, in terms of different baseline architecture with our method, ResNet-101 and ResNet-50 improve the mAP by 7.6% and 4.7% compared to ZF and VGG, respectively.

Moreover, from the depth of the architecture perspective, the mAP of ResNet-50 + RPSN + SSM is 0.1% higher than that of ResNet-101 + RPSN + SSM, which indicate the weak influence caused by the depth of the architecture. In addition, we also explore the effectiveness of data augmentation on detection performance by using (1) mirrored rotational augmentation and (2) no augmentation. The results in Table 1 show that the augmentation obviously improves the performance compared to no augmentation at all.

We also evaluate the operating speed of the proposed approach, as shown in Table 1. Fortunately, we can observe that the proposed method maintains operating speed even when RPSN and SSM are associated for improving the effectiveness of CNN models. All architectures used in our method can detect a rice planthopper image less than 0.2 s, including ResNet-50 and ZF which spend less than 0.1 s detecting rice planthopper.

Furthermore, we compare the performance of the algorithm in this paper with other state-of-the-art deep learning methods in Table 2. One can observe that our mAP is raised to 57.4%, whereas the mAP with Faster RCNN is only 56.3% even with ResNet-101. Besides, proposed method performs better than other state-of-the-art CNN models including YOLOv3 and SSD. In-field rice planthopper detection is difficult, the reason that our algorithm can exceed other state-of-the-art algorithms is that our method employs the local features in the field, which makes our RPSN with SSM can search the rice planthopper target where large numbers of local features are present even if RPSN does not provide the high confidence.

Table 1
mAP and operating speed for different models.

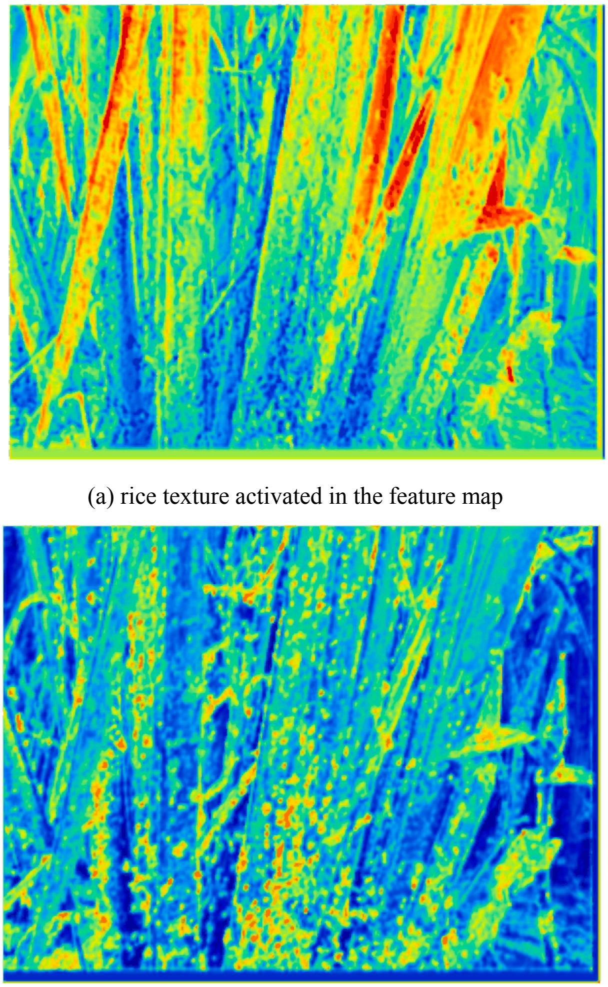
Backbone	Data Augmentation	RPSN + SSM	mAP	Operating speed (second/image)
ResNet-101			0.394	0.126
		√	0.529	0.129
	√		0.428	0.138
	√	√	0.573	0.141
ResNet-50			0.366	0.096
		√	0.525	0.097
	√		0.388	0.096
	√	√	0.574	0.095
ZF			0.359	0.054
		√	0.428	0.043
	√		0.364	0.051
	√	√	0.498	0.048
VGG			0.412	0.133
		√	0.488	0.118
	√		0.439	0.14
	√	√	0.527	0.116

5.3. *Feature visualization*

In order to better describe what is learned by our models, Fig. 10 shows the feature maps after feature extraction by ResNet-50 + RPSN + SSM. Note that a pixel in each feature map is a rectified activation, the brighter the feature map is, the higher the activation value of the pixel

Table 2
Performance comparison with other prevalent object detection methods.

Backbone	Methods	mAP
ResNet-101	RPSN + SSM	0.573
ResNet-50	RPSN + SSM	0.574
ZF	RPSN + SSM	0.498
VGG	RPSN + SSM	0.527
VGG	SSD512	0.472
VGG	SSD300	0.433
ResNet-50	SSD512	0.479
ResNet-50	SSD300	0.445
DarkNet-53	YOLOv3	0.495
ResNet-101	Faster RCNN	0.563
ResNet-50	Faster RCNN	0.552



(a) rice texture activated in the feature map

(b) rice planthopper activated in the feature map

Fig. 10. Visualization of feature maps in the process of training ResNet-50 + RPSN + SSM model by RPDA2018.

turns into in the feature map. As we can see in the Fig. 10(a), many features of the original image, such as space feature, texture feature and color feature are activated in the feature maps. Furthermore, Fig. 10(b) illustrates the pixels in rice planthopper are activated in some feature maps, which is an important reason that our model can detect the rice planthopper. This feature visualization can demonstrate what is learned by our architecture.

Fig. 11 shows all channels in position-sensitive maps. White color indicates that the pixels in this area are activated. The part in red dotted box shows the activation for background pixels and the image with activated pixels of rice planthopper are embedded in the bottom yellow dotted box. In this paper, feature maps are divided into grid at r2 = 72 = 49 grids, the number of channels in each color block is c + 1 = 2, therefore, r2(c + 1) = 98 channels are captured in position-sensitive maps. We can observe that half the channels represent the extraction of the background content of the image, and a large number of image background pixels are activated, while the pixels belonging to the rice planthopper region are not activated. Meanwhile, the remaining 49 images represent the extraction of rice planthoppers in the image. It is obvious that almost all the pixels involved in the rice planthopper are activated, whose areas are filled with white color.

5.4. Results demonstrations

In order to show the performance of counting results better, RPDA2018 is divided into low-density images, medium-density images and high-density images on the basis of the number of rice planthopper in the image. An image of less than 5 rice planthoppers is considered as low-density image, while high-density images contain more than 30 rice planthoppers, and the left images are medium-density images. The parameters for the distribution of RPDA2018 are strictly calculated to ensure that the number of images in different densities is roughly the same. Some demos of rice planthopper detection results by ResNet-50 + RPSN + SSM are showed in Fig. 12. ‘rp’ is the abbreviation of rice planthopper in detection results, and the number after ‘rp’ is the confidence level which architecture considers the object is the rice planthopper. The white boxes show the rice planthoppers in the images while the red boxes represent the detection results of the method proposed in this paper.

Table 3 illustrates the counting results of different densities images with threshold = 0.5/0.8 in RPSN. The rice planthopper images with high-density achieved the best performance at precision and recall when threshold equals to 0.5. Another two types of rice planthopper images also have great counting results based on ResNet-50 + RPSN + SSM model. After re-observing the experimental results of rice planthopper images with different densities, we found that part of bounding box and ground truth did not match well due to the low threshold in RPSN, which

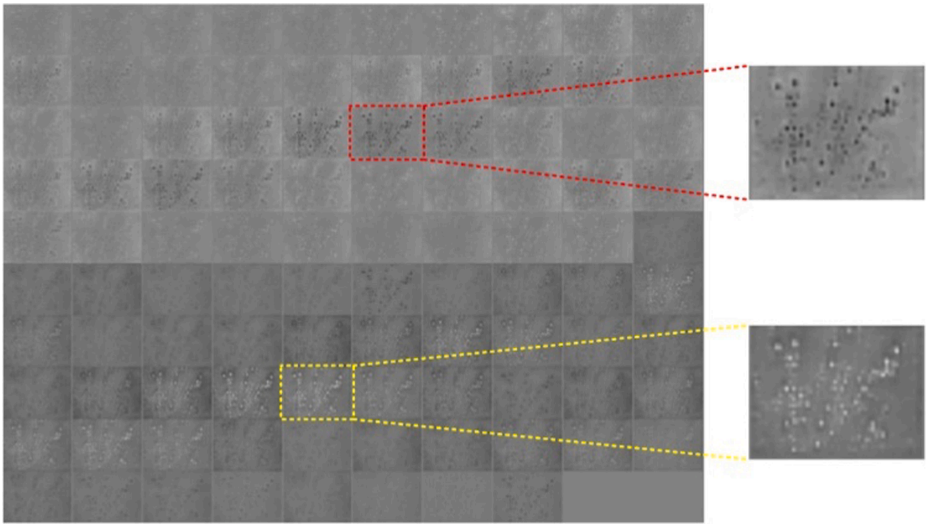
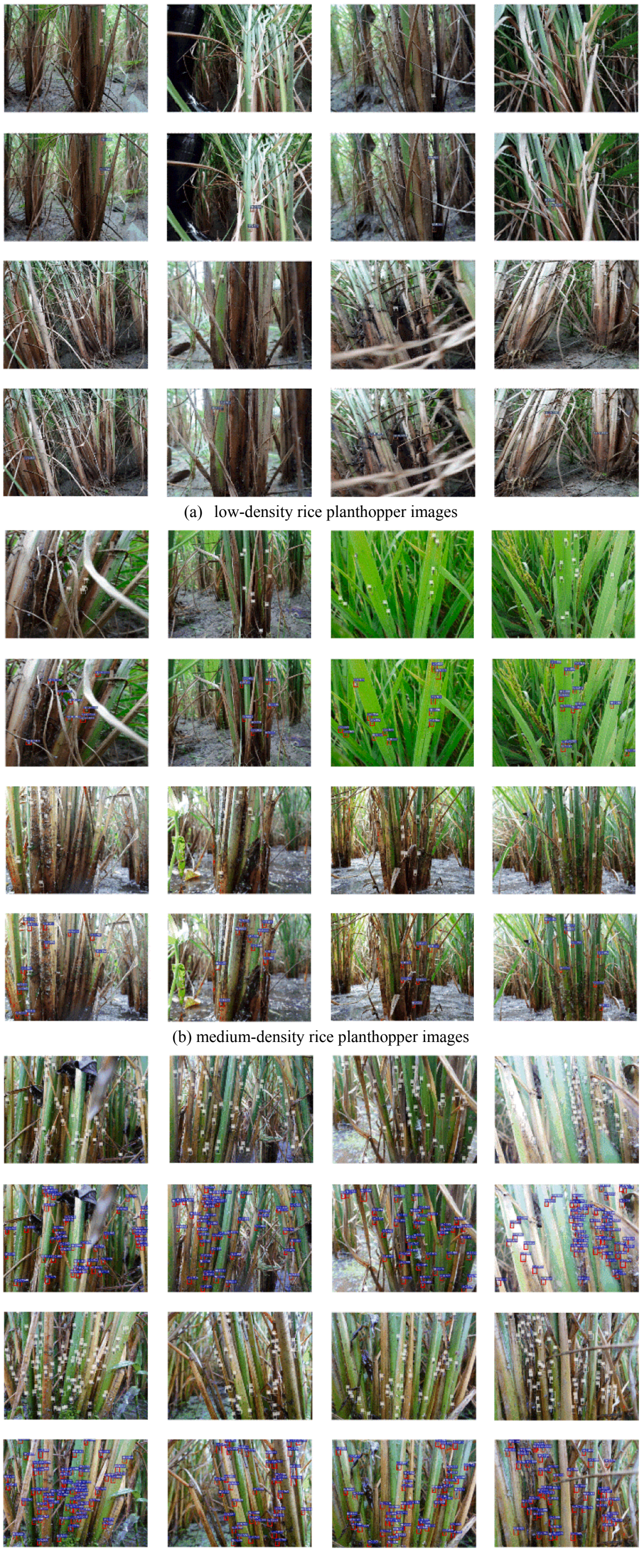


Fig. 11. Position-sensitive maps in trained RPDA2018 based on ResNet-50 + RPSN + SSM model.


(a) low-density rice planthopper images

(b) medium-density rice planthopper images

(c) high-density rice planthopper images

Fig. 12. Some typical examples in the rice planthopper detection.

means that there are slight deviations. Some prediction boxes on high-density images can be matched to another rice planthopper due to the high-density, while on low-density data, such offset prediction boxes will be considered as False positive samples, which means it is not rice planthopper, but predicted as rice planthopper. Therefore, if we set a higher threshold (0.8) in RPSN, the counting performance gap between those three different density images will decrease.

Table 3
Comparison of counting results of different densities images based on threshold = 0.5/0.8 in RPSN.

	Numbers	Precision	Recall
High-density images	28,608	82.79% (0.5) 76.45% (0.8)	92.48% (0.5) 85.77% (0.8)
Medium-density images	27,856	81.24% (0.5) 76.23% (0.8)	91.51% (0.5) 85.79% (0.8)
Low-density images	25,672	80.23% (0.5) 76.43% (0.8)	90.14% (0.5) 85.40% (0.8)
Average	27378.67	81.42% (0.5) 76.37% (0.8)	91.38% (0.5) 85.65% (0.8)

5.5. *Real time analysis*

Real-time performance investigation also shows a dramatic improvement on both training and test phase as our method could detect and count the rice planthopper faster comparing to human annotation (Table 4). In terms of real time analysis, the detection time of our system is the mean value calculated from a large number of experimental results. Similarly, the time cost of manual detection and counting is measured several times when our agricultural experts detect and count rice planthopper in the field. Therefore, the credibility of our real time analysis is guaranteed. The detection speed of our models (see Table 1), no matter shallow or deep networks, could far exceed human detection, which is a laborious work. Therefore, our system could be replaced with manual detection in practical agriculture circumstance.

6. Conclusion and future work

In this work we propose a hand-held mobile pest monitoring system for rice planthopper based on convolutional network, which aims to automatically identify the rice planthopper in the non-site-specific wild environment. The system adopting the end-to-end strategy can be used to process the rice planthopper image without any pre-processing before the training on enough rice planthopper samples. Our system has successfully realized the automatic extraction and fusion of high-quality features, which performs well on small size pest detection. Besides, four different convolutional neural network architectures are exploited to perform the rice planthopper detection on RPDA2018 which formed by images photographed in the field. Experimental results show that our proposed system could deliver an average recall up to 91% over three types of image intensity in industrial circumstance, which outweighs the state-of-the art approaches. Experiments on RPDA2018 show that the proposed method achieves remarkable effect under the different convolutional neural network architecture. Even based on typical shallow convolutional neural network architectures such as ZF, our approach can make better performance than using deep convolutional neural network models only (like ResNet-101). Moreover, the convolutional neural network architecture applied in this paper achieve the requirements of the automation and real-time, in which ResNet-50 + RPSN + SSM can take less than 0.1 s to detect and count rice planthopper in one image. In addition, liveness detection which distinguish live and dead rice planthopper module can be developed to further improve the accuracy of our system.

Table 4
Detection time spent on different models.

Methods	Detection Time (s/image)
Deep learning models proposed in this paper	0.05–0.15
Human annotation	More than 120

CRediT authorship contribution statement

Fangyuan Wang: Conceptualization, Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Rujing Wang:** Project administration, Funding acquisition, Supervision. **Chengjun Xie:** Supervision, Writing - review & editing. **Jie Zhang:** Funding acquisition, Supervision. **Rui Li:** Funding acquisition, Supervision. **Liu Liu:** Writing - review & editing.

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.

Acknowledgement

This work was supported by The National Key Technology R&D Program of China under Grant 2018YFD0200300, National Natural Science Foundation of China under grant 31401293, 31671586, 61773360, Major Special Science and Technology Project of Anhui Province under grant 201903a06020006.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compag.2021.106268.

References

Aragón, P., Baselga, A., Lobo, J., 2014. Global estimation of invasion risk zones for the western corn rootworm diabrotica virgifera virgifera: integrating distribution models and physiological thresholds to assess climatic favourability. J. Appl. Ecol. 47, 1026–1035.

Bearup, D., Petrovskaya, N., Petrovskii, S., 2015. Some analytical and numerical approaches to understanding trap counts resulting from pest insect immigration. Math. Biosci. 263, 143–160.

Berenstein, R., Edan, Y., 2018. Automatic adjustable spraying device for site-specific agriculture application. IEEE Trans. Autom. Sci. Eng. 15, 641–650.

Bures, B., Donohue, K., Roe, R., Bourham, M., 2006. Nonchemical dielectric barrier discharge treatment as a method of insect control. IEEE Trans. Plasma Sci. 34, 55–62.

Cho, J., Choi, J., Qiao, M., Ji, C., Kim, H., Uhm, K., Chon, T., 2007. Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis. Red 346, 244.

Deutsch, J., He, D., 2018. Using Deep Learning based approach to Predict Remaining Useful Life of Rotating Components. IEEE Trans. Syst., Man Cybernet.: Syst. 48, 1.

Ding, W., Taylor, G., 2016. Automatic moth detection from trap images for pest management. Comput. Electron. Agric. 123, 17–28.

Do, T., Nguyen, A., Reid, I., Caldwell, D., Tsagarakis, N., 2017. Affordancenet: an end-to-end deep learning approach for object affordance detection. arXiv:1709.07326.

Dyrmann, M., Karstoft, H., Midtiby, H., 2016. Plant species classification using deep convolutional neural network. Biosyst. Eng. 151, 72–80.

Ebrahimi, M., Khoshtaghaza, M., Minaei, S., Jamshidi, B., 2017. Vision-based pest detection based on svm classification method. Comput. Electron. Agric. 137, 52–58.

Enes, A., Hasan, E., Fatih, V., 2020. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. Comput. Electron. Agric. 179, 105809.

Espinoza, K., Valera, D., Molina-Aiz, F., 2016. Combination of image processing and artificial neural networks as a novel approach for the identification of bemisia tabaci and frankliniella occidentalis on sticky traps in greenhouse agriculture. Comput. Electron. Agric. 127, 495–505.

Girshick, R., 2015. Fast R-CNN. IEEE International Conference on Computer Vision, arXiv:1504.08083.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 580–587.

Grinblat, G., Uzal, L., Larese, M., Granitto, P., 2016. Deep learning for plant identification using vein morphological patterns. Comput. Electron. Agric. 127, 418–424.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.

Hou, S., Feng, Y., Wang, Z., 2017. VegFru: A Domain-Specific Dataset for Fine-Grained Visual Categorization. In: IEEE International Conference on Computer Vision, 541–549.

Kang, S., Cho, H., Lee, S., 2014. Identification of butterfly based on their shapes when viewed from different angles using an artificial neural network. J. Asia-Pac. Entomol. 17, 143–149.

Kang, S., Song, S., Lee, S., 2012. Identification of butterfly species with a single neural network system. J. Asia-Pac. Entomol. 15, 431–435.

Liu, H., Lee, S., Chahl, J., 2017. A multispectral 3D vision system for invertebrate detection on crops. IEEE Sens. J. 17, 7502–7515.

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.

Liu, T., Chen, W., Wu, W., Sun, C., Guo, W., Zhu, X., 2016a. Detection of aphids in wheat fields using a computer vision technique. Biosyst. Eng. 141, 82–93.

Liu, Z., Gao, J., Yang, G., Zhang, H., He, Y., 2016b. Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. Sci. Rep. 6, 20410.

Loris, N., Gianluca, M., Fabio, P., 2020. Insect pest image detection and recognition based on bio-inspired methods. Comput. Electron. Agric. 57, 101089.

Lu, J., Hu, J., Zhao, G., Mei, F., Zhang, C., 2017. An in-field automatic wheat disease diagnosis system. Comput. Electron. Agric. 142, 369–379.

Luo, J., Huang, W., Zhao, J., Zhang, J., Zhao, C., Ma, R., 2013. Detecting aphid density of winter wheat leaf using hyperspectral measurements. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 6, 690–698.

Pourbabaee, B., Roshtkahari, J., Khorasani, K., 2017. Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients. IEEE Trans. Syst., Man Cybernet.: Syst. 99, 1–10.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. In: IEEE Conference on Computer Vision and Pattern Recognition.

Robinson, A., Hammon, P., Sa, V., 2007. Explaining brightness illusions using spatial filtering and local response normalization. Vision Res. 47, 1631–1644.

Santangelo, G., 2018. The impact of FDI in land in agriculture in developing countries on host country food security. J. World Bus. 53, 75–84.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. FaceNet: A unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, 815–823.

Simonyan, Karen, Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Sun, Y., Cheng, H., Cheng, Q., Zhou, H., Li, M., Fan, Y., 2017. A smart-vision algorithm for counting whiteflies and thrips on sticky traps using two-dimensional Fourier transform spectrum. Comput. Digital Eng. 153, 82–88.

Tabor, J., Spurek, P., 2014. Cross-entropy clustering. Pattern Recogn. 47, 3046–3059.

Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In: IEEE Conference on Computer Vision and Pattern Recognition, 1701–1708.

Wang, K., Zhang, S., Wang, Z., Liu, Z., Yang, F., 2013. Mobile smart device-based vegetable disease and insect pest recognition method. Intell. Automation Soft Comput. 19, 263–273.

Wen, C., Wu, D., Hu, H., Pan, W., 2015. Pose estimation-dependent identification method for field moth images using deep learning architecture. Biosyst. Eng. 136, 117–128.

Wu, C., Chan, Y., Fu, L., Hsiao, P., Huang, S., Chen, H., 2014. Combining multiple complementary features for pedestrian and motorbike detection. In: IEEE Conference on Intelligent Transportation Systems, 1358–1363.

Xia, C., Chon, T., Ren, Z., Lee, J., 2014. Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. Ecol. Inf. 29, 139–146.

Xiao, B., Ma, J., Cui, J., 2012. Combined blur, translation, scale and rotation invariant image recognition by Radon and pseudo-Fourier–Mellin transforms. Pattern Recogn. 45, 314–321.

Yan, C., Luo, M., Liu, H., 2017. Top-k Multi-class SVM using Multiple Features. Inf. Sci.

Yaseen, M., Anjum, A., Rana, O., Antonopoulos, N., 2018. Deep learning hyper-parameter optimization for video analytics in clouds. In: IEEE Transactions on Systems, Man and Cybernetics: systems, 66.

Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T., 2016. UnitBox: an advanced object detection network. arXiv:1608.01471.

Zayas, I., Flinn, P., 1998. Detection of Insects in Bulkwheat Samples with Machine Vision. Trans. ASAE 40, 883.

Zeiler, M., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, 818–833.