Computers and Electronics in Agriculture 145 (2018) 319–325

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Detection of stored-grain insects using deep learning

Yufeng Shen[a], Huiling Zhou[a,*], Jiangtao Li[a], Fuji Jian[b], Digvir S. Jayas[b]

[a] Beijing University of Posts and Telecommunications, PO Box 137, Road Xitucheng 10, Haidian District, Beijing 100876, China
[b] Department of Biosystems Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada

ARTICLE INFO

Keywords:
Stored-grain insect
Object detection
Insect classification
Convolutional neural network
Trap images
Faster R-CNN

ABSTRACT

A detection and identification method for stored-grain insects was developed by applying deep neural network. Adults of following six species of common stored-grain insects mixed with grain and dockage were artificially added into the developed insect-trapping device: *Cryptoleste Pusillus*(S.), *Sitophilus Oryzae*(L.), *Oryzaephilus Surinamensis*(L.), *Tribolium Confusum*(Jaquelin Du Val), *Rhizopertha Dominica*(F.). Database of Red Green and Blue (RGB) images of these live insects was established. We used Faster R-CNN to extract areas which might contain the insects in these images and classify the insects in these areas. An improved inception network was developed to extract feature maps. Excellent results for the detection and classification of these insects were achieved. The test results showed that the developed method could detect and identify insects under stored grain condition, and its mean Average Precision (mAP) reached 88.

1. Introduction

Image recognition to detect and identify insects in a stored product is the critical component of a stored-grain insect monitoring system. The main challenges in the image recognition of these insects are to identify areas of the image containing insects in grain mixed with other materials (mostly the fine materials and broken grain kernels) and to classify the small body size insects conglutinated with other insect species and/or the same species and/or the other materials in the target area.

Object detection system such as pedestrian detection and vehicle detection applies the region proposal algorithms to infer locations of objects (Girshick et al., 2013). The early developed region proposal algorithms include Selective Search, Sliding Window, Rigor, Superpixels and Gaussian (Hosang et al., 2015). The Region Proposal Network (RPN) Ren et al., 2017 was proposed in 2016, which applied convolutional neural network method to get the areas of interest more quickly and accurately through the acceleration of Graphics Processing Unit (GPU).

In the field of the insects classification based on computer vision, most of researches focused on the extracting of insects' features including texture, shape, and local characteristics (Qiu et al., 2003; Zhang et al., 2009, 2005; Wu et al., 2015; Jayas, 2017). Procedure of this feature extraction was complex under in-situ situations, and those extracted features might not accurately represent the image characteristics of insects. In the practical application, the variation of image background, impurities, illumination and insect's gestures will also increase the difficulty of feature extraction.

Krizhevsky et al. (2012) used deep convolutional neural networks got first on ImageNet Large Scale Visual Recognition Challenge in 2012. Ding and Taylor (2016) used the Sliding Window method to obtain regions of interest and applied a 5-layer convolutional neural network to determine whether the regions contained a moth. They got a higher recall rate using convolutional neural network than that using LogReg algorithm. Liu et al. (2016) applied the GrabCut for the segmentation of paddy field pests and classified the pests using a 8-layer convolutional neural network. The accuracy of the convolutional neural network was higher than that of the Histogram of Oriented Gradient (HOG), Speeded Up Robust Features (SURF). However, the network used by Liu et al. was too shallow, and could not extract more effective features when the targets were similar in appearance. In this study, we used an Online Insect Trapping Device (OITD) (Wang et al., 2016) to capture the images of live insects with or without fines, foreign materials, dockages and broken grains (referred to as FFDB) under laboratorial conditions. The insects imaged were: *Cryptoleste Pusillus*(S.), *Sitophilus Oryzae*(L.), *Oryzaephilus Surinamensis*(L.), *Tribolium Confusum*(Jaquelin Du Val), *Rhizopertha Dominica*(F.) and *Lasioderma Serricorne*(F.). A dataset with 739 images of the insects with or without FFDB was established, and artificially marked. To increase the accuracy of the convolutional neural network, we applied a 27-layer convolutional neural network to extract the features from the images of stored-product insects, and adopted the Softmax as classifier to identify the insects. The Faster R-CNN method was used to locate and classify the insects. First, the inception network (Szegedy et al., 2015) extracted the feature maps of

[*] Corresponding author.
E-mail address: huiling@bupt.edu.cn (H. Zhou).

https://doi.org/10.1016/j.compag.2017.11.039
Received 18 June 2017; Received in revised form 16 October 2017; Accepted 26 November 2017

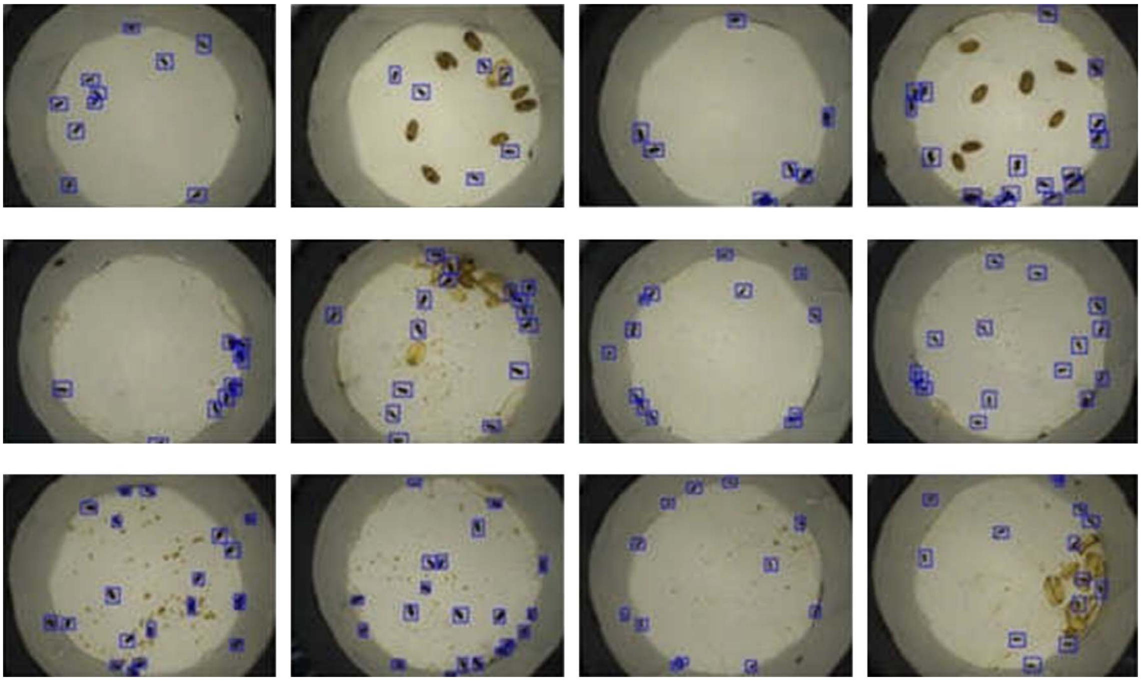

Fig. 1. Ground truth.

each image, then the RPN returned the coordinates of the areas which might have insects in the feature maps. These coordinates were merged by Non Maximum Suppression (NMS) Neubeck and Van Gool, 2006 and mapped to the feature maps. These target areas were classified by the improved inception network. The coordinates of the target areas were also corrected at the same time (Girshick, 2015). Finally, NMS was used to merge the overlapping target boxes.

In this study, we developed a method based on the Faster R-CNN, which can be used to identify the insects mixed with FFDB under different illumination conditions. This method improved the accuracy of the insect detection. The rest of this article is organized to introduce this image processing procedure.

2. Image acquisition and preprocessing

The resolution of the images taken by the OITD system was 1944 × 2592 pixels. These images were collected under the conditions of multi live insects mixed with artificially added FFDB in wheat in order to increase the difficulty of detection and simulate the real situation in grain warehouses. The FFDB included fine materials and broken grains (Fig. 1). Multiple live insects of single species were added into OITD system every time and multiple pictures were snapshotted. Some pictures were randomly chosen as the testing set, and the rest were used as the training set. This procedure was repeated and new insects were added for the next replicate. Table 1 shows the number of the images and insects in the training and testing sets.

To enrich the training set, extract image features accurately, and generalize model to prevent overfitting, the image dataset was augmented by flipping and color jittering. To account for the change of illumination level and insects' posture, the color jittering was conducted by randomly adjusting the saturation, contrast, brightness, and sharpness of the image. After augmentation, the size of training set was increased by 12 folds of the original training set. The original images had a high resolution. To improve the training and testing speed and to reduce the GPU consumption, the image resolution was lowered to 600 × 800 pixels. Each insect in the images was artificially marked by a blue bounding box (Fig. 1) as ground truth. The marked blue bounding box was used for training.

Table 1
Number of images and insects used for training and testing.

		SO[a]	LS[b]	TC[c]	RD[d]	OS[e]	CP[f]
Training	Images	77	54	98	114	63	117
	Insects	1206	1088	1423	1908	830	2957
Testing	Images	33	28	42	21	23	69
	Insects	465	514	446	335	358	978
Total	Images	110	82	140	135	86	186
	Insects	1671	1602	1869	2243	1188	3935

[a] SO = *Sitophilus Oryzae*.
[b] LS = *Lasioderma Serricorne*.
[c] TC = *Tribolium Confusum*.
[d] RD = *Rhizopertha Dominica*.
[e] OS = *Oryzaephilus Surinamensis*.
[f] CP = *Cryptolestes Pusillus*.

3. Object detection network

The detection steps for the target object were: acquire the region proposals of the image by RPN, merge these proposals as candidate boxes by NMS, map these candidates boxes to the feature maps, classify these regions of candidate boxes by classification network, use the NMS to merge these overlapping candidate boxes. The detection process is shown schematically in Fig. 2 and details are provided in the following sections.

3.1. Feature extraction network

In order to obtain high-quality model, the width (different sizes of kernels were used to extract the same feature maps) and depth of the neural network model should be increased. The inception structure (Szegedy et al., 2015) was adopted to extend the convolutional layers of the model. The main purpose of the inception structure was to find a simple dense component to replace an optimal local sparsity structure and repeated this structure spatially. This procedure would cluster together the units with high relative correlation to form the next layer and this next layer would connect to its top layer. The adopted inception structure is shown in Fig. 3. To reduce the number of parameters

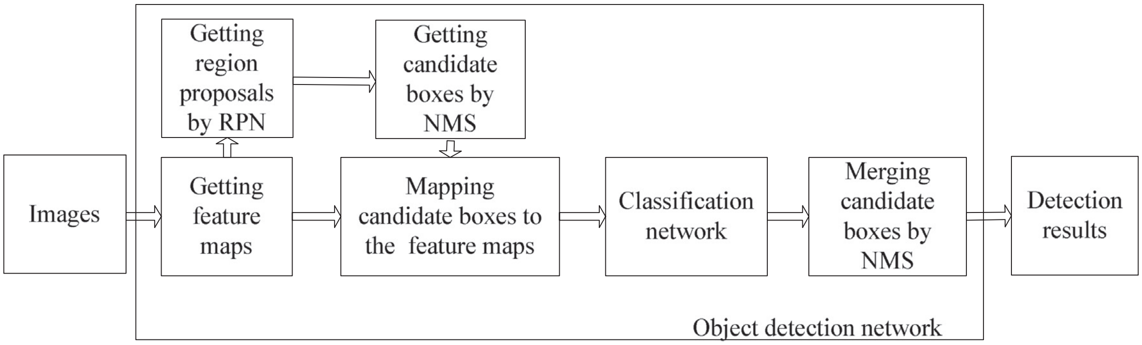
Getting region proposals by RPN
Getting candidate boxes by NMS
Images
Getting feature maps
Mapping candidate boxes to the feature maps
Classification network
Merging candidate boxes by NMS
Detection results
Object detection network

Fig. 2. Flow chart of object detection.

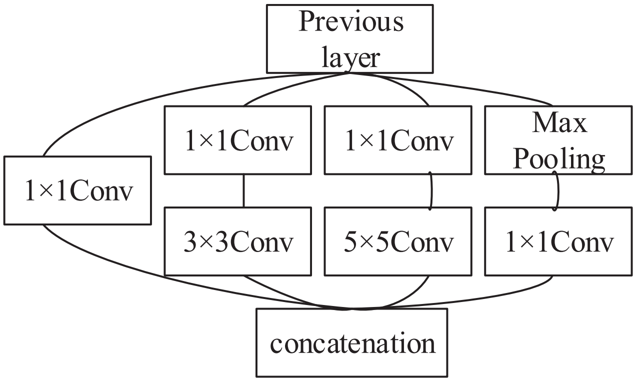

Fig. 3. Inception structure.

and to improve the speed of operation, the 1 × 1 convolution kernels were used before 3 × 3 and 5 × 5 convolution kernels.

3.2. Region proposal network

In the literature (Ren et al., 2017), VGG16 (Simonyan and Zisserman, 2015) was usually used as a base network, but the VGG16 network contained a large number of parameters, because it adopted the fully connection layer of 4096 × 4096. We used the inception network to replace VGG16 network, and assigned the output of the seventh inception structure as a feature map. To further extract features, two inception structures were connected with the feature maps, then the convolutional layer with kernels of 3 × 3 was applied to reduce the thickness of convolution features to 256 dimensions. This output was assigned as the features of region proposal network. We coded the features as nine scales of bounding boxes' coordinates and the bounding boxes' scores.

Training followed multitask loss. The loss function was defined as (Ren et al., 2017):

$$L(\{p_i\},\{p_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \tag{1}$$

where N is the total amount of anchors (Ren et al., 2017), i is the anchor’s index in the mini-batch, p_i is the prediction probability of the ith anchor, p_i^* is the label of the ith anchor, t_i is the coordinate of the predicted bounding box, and t_i^* is the ground truth coordinate. Classification loss (L_{cls}) is the logarithmic loss of two categories: foreground and background (Ren et al., 2017):

$$L_{cls}(p_i,p_i^*) = -(\log(p_i^* p_i) + \log((1-p_i^*)(1-p_i))) \tag{2}$$

Regression loss L_{reg} (Ren et al., 2017):

$$L_{reg}(t_i,t_i^*) = \text{smooth}_{L_1}(t_i-t_i^*) \tag{3}$$

smooth_{L_1} is the robust loss function:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if}|x| < 1 \\ |x|-0.5 & \text{otherwise} \end{cases} \tag{4}$$

When the resolution of input images was 600 × 800 pixels, the output was 50 × 37 × 36 and 50 × 37 × 18. This output was reshaped to 16,650 × 4 and 16,650 × 2, respectively. Then 16,650 regression boundaries and foreground/background scores were obtained. During training, the 2000 rectangular boxes with the highest score were casted to the classification network as proposal regions. Then the proposal regions were merged by NMS as the candidate boxes.

3.3. Classification network

To classify the region in a candidate box, the RoI (region of interest) Pooling Layer (Ren et al., 2017) was used to map the candidate box to the feature map which was the output of the seventh inception structure. The RoI Pooling Layer was the simplified form of Spatial Pyramid Pooling layer (He et al., 2015) with only one scale. In this study, the RoI Pooling Layer sampled the feature map to scale 7 × 7. After that, two fully connection layers were used to unify the feature map as a 1024 dimension feature vector.

The classification network and the RPN shared the parameters of convolutional layers. Two fully connection layers were used to code the features to a 44 dimension vector and a 11 dimension vector, respectively, in the classification network. After adjusting 500 candidate boxes input by region proposal network, the coordinates and the corresponding category score (defined by the probability of the true category) were obtained. During the training, the loss value of the classification network was consisted of the logarithmic loss and the regression loss. Logarithmic loss (L_{cls}) was calculated by the corresponding probability (P_u) of the true category (u) Girshick, 2015:

$$L_{cls} = -\log P_u \tag{5}$$

Regression loss (L_{reg}) (Girshick, 2015) was calculated as:

$$L_{reg} = \sum_{i=1}^{4} \text{smooth}_{L_1}(t_i^u-v_i) \tag{6}$$

where v_i is the corresponding prediction parameter of the true class, t_i^u is the real translation and scaling parameters. The total loss (L) Girshick, 2015 is shown in Eq. (7):

$$L= \begin{cases} L_{cls} + \lambda L_{reg} & \text{(when u is foreground)} \\ L_{cls} & \text{(when u is background)} \end{cases} \tag{7}$$

In the insect classification, to balance between the precision and the recall of the classification network, we defined a rule: when the P_u of a candidate box was higher than 0.5, the box was considered to contain an insect which belonged to the category of u. In order to prevent overfitting and increase the sparsity of the network, the dropout layer (Srivastava et al., 2014) was added before the fully connection layer.

3.4. Non maximum suppression

After obtaining the coordinates of the 500 candidate boxes and their corresponding category scores, each candidate box was saved as a vector (r, c, h, w, n). The r, c, h and w were the coordinates of the candidate box, and n was the score of the candidate box. The procedure of this non maximum suppression was: calculate the areas of all windows and the overlapping area of the candidate box, find the candidate box with the highest score and the candidate box with the second-highest score, calculate the intersection over union (IoU) of the two candidate boxes. If the IoU is higher than the threshold, suppress the target box with the second-highest score. During our testing, we found the threshold value of 0.7 got the highest mAP.

3.5. Model optimization

Faster R-CNN with deep networks had two issues. The first issue is that the optimization ability of the model was significantly reduced because the flow of the deep inception network's information was blocked (He et al., 2016). In the literature, Szegedy et al. (2015) added two additional Softmax layers in the inception network to calculate new loss value, and the gradient of the network was calculated using the new loss value. He et al. (2016) introduced the method of the shortcut connection to solve the problem of flow blocking. The shortcut connections were leveraged to reduce the influence of gradient disappearance and improve the degradation phenomenon of information flow. The second issue is that the body size of stored-grain insects is relatively small. The feature map size of the output of the seventh inception was only 1/16 of the original image. The receptive field of each neuron was too large, so it was not sensitive to a small target. To solve these two issues an improved inception network was developed. We combined the output of the second inception with the output of the seventh inception through the fully connection layers. We also added a Rol Pooling layer after the second inception layer (Fig. 4). The proposal regions were mapped to the output of the second inception. This developed method could directly back-propagate the gradient from the deep layers to the shallow layers of the inception network. By adding the shortcut connection, the network could extract the feature maps of the shallow layers which were 1/8 of the original image. This method was more sensitive to small targets than that used by He et al. (2016).

If the classification and positioning function were achieved by the fully connection layer, a large amount of memory was required. The Singular Value Decomposition (SVD) was operated on the fully connection layer and the number of parameters was reduced, so a small amount of memory was required. The SVD was used to decompose the parameter matrix (W), and W was approximated by the former t eigenvalues:

$$W= U\sum V^{T} \approx U(:,1:t)\cdot \sum (1:t,1:t)\cdot V(:,1:t)^{T} \tag{8}$$

The forward propagation was divided into two steps (Girshick, 2015). This was equivalent to splitting the fully connection layer into two layers and these two layers were connected by the fully connection layer with lower dimension in the middle. In this study, the two fully connection layers were divided by SVD to reduce the number of parameters. The improved object detection network is shown in Fig. 4.

4. Experiment and analysis

4.1. Experiment

The training set was augmented by flipping and color jittering, the images in Fig. 5 are examples of the "data augmentation". The images of augmented training set were normalized to 600 × 800 pixels. Two images as a batch were send into the neural network for training. Neural network adopted the training method of end to end (training RPN and classification network at the same time). A dropout layer after the fully connection layer was added and only 50% of the fully connection layer's neurons were activated in each training iteration. The network weights were initialized by the pre-trained model trained by the ImageNet, and the method of Stochastic Gradient Descent (SGD) with momentum was adopted to update parameters (Wilson and Martinez, 2003). For each region proposal generated by RPN, nine candidate boxes were generated with three sizes (128 pixels, 256 pixels and 512 pixels) and three aspect ratios (0.5, 1 and 2). During training, 2000 candidate boxes obtained by the non-maximum suppression were divided into foreground and background as a training set to train the classification network. Four loss values were analyzed: regression and logarithmic loss of RPN, regression and logarithmic loss of classification network. The training iteration was 140,000 times. The sum of four loss values is shown in Fig. 6.

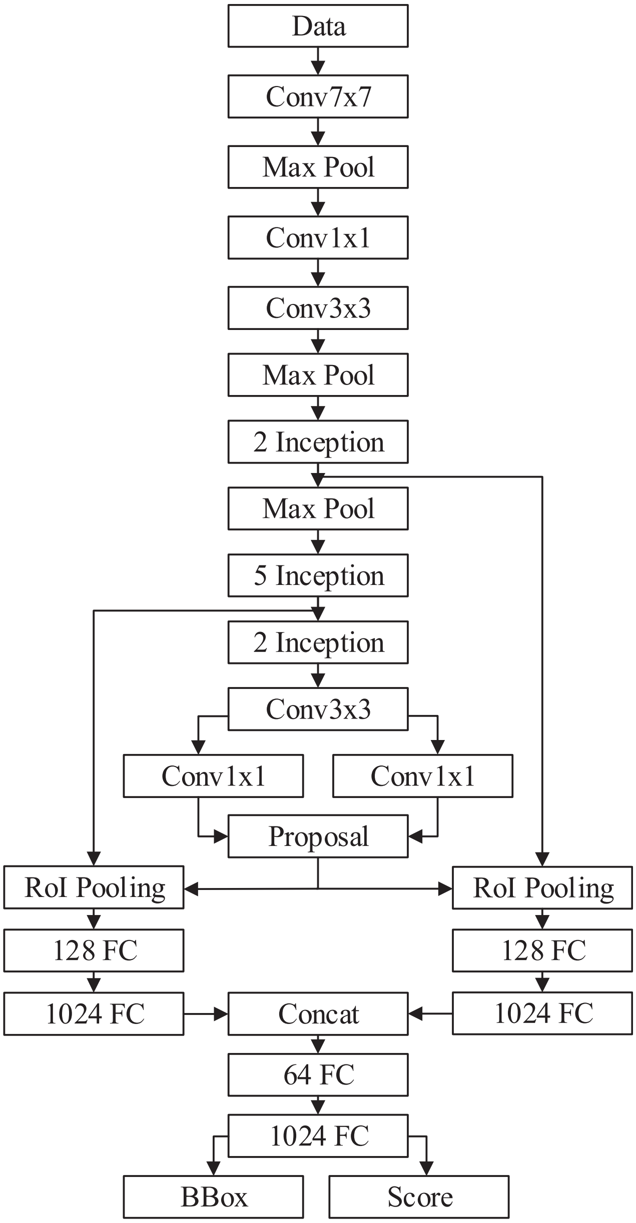

Fig. 4. The improved detection network.

During testing, the candidate boxes generated by RPN were ordered by their scores. Five hundred candidate boxes with the highest score were selected as the candidate boxes. Classification and regional position adjustment were conducted for these candidate boxes by the fully

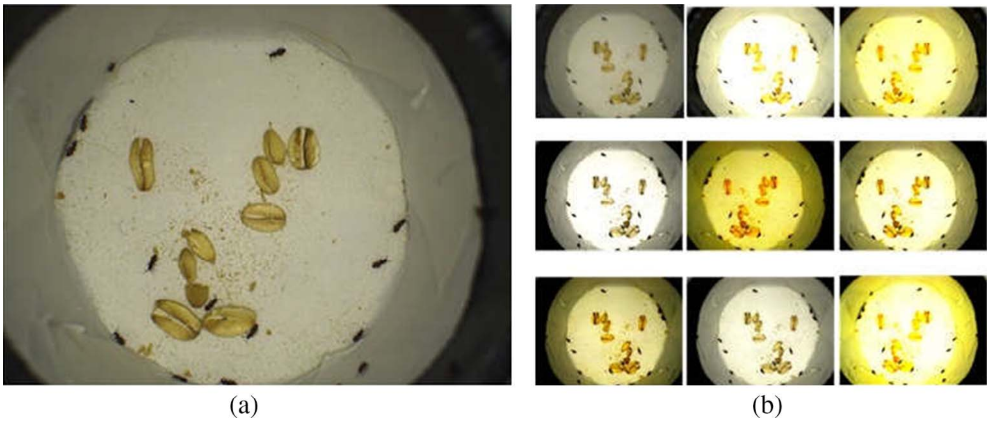

Fig. 5. Data augmentation. (a) Original image; (b) images generated by data augmentation.

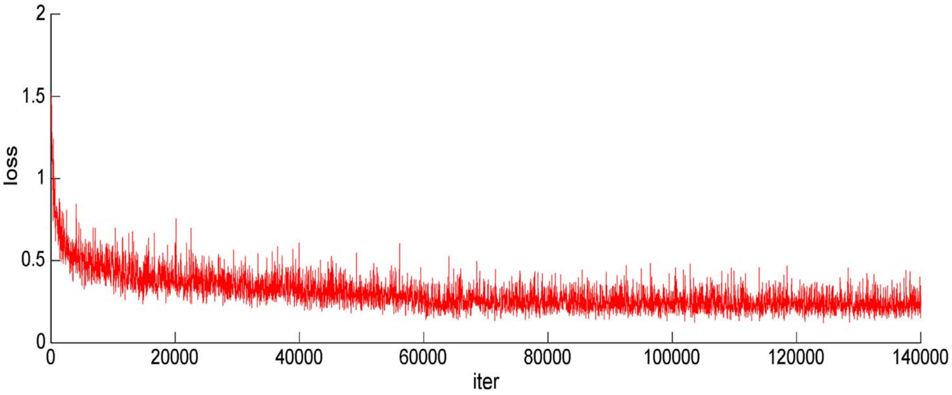

Fig. 6. Loss graph for 14,000 iterations.

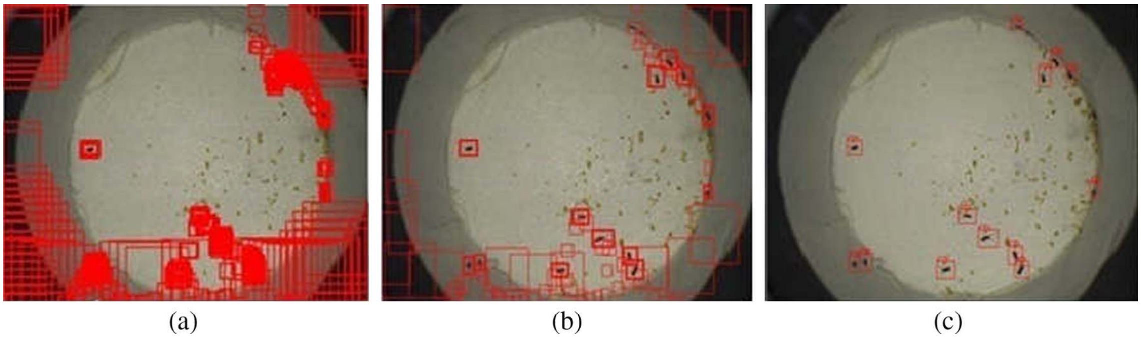

Fig. 7. Detection results. (a) 500 candidate boxes extracted by RPN; (b) Boxes through classification network merged by NMS; (c) Boxes whose category scores ≥ 0.5.

Table 2
The mAP, run time and model size of different models.

Model	SO[a]	LS[b]	TC[c]	RD[d]	OS[e]	CP[f]	mAP	Run time (s)	Model size
VGG16	90.19	71.55	93.17	89.66	71.54	79.82	82.66	0.226	547M
Inception	89.76	70.91	91.20	84.49	74.97	76.98	81.39	**0.168**	160M
Improved inception	**95.48**	**77.82**	95.72	92.44	79.54	86.95	**87.99**	0.182	261M
R-FCN + ResNet101	88.54	74.81	95.26	93.63	**80.47**	**88.56**	86.88	0.246	200M
Improved inception + SVD	95.31	72.92	**96.86**	**93.79**	79.86	86.17	87.49	0.183	**62M**

The bold values mean the best performance in these models.

connection layer of the classification network. In Fig. 7, (a) shows the 500 candidate boxes extracted by RPN and merged by NMS, (b) shows the boxes through classification network merged by NMS, and (c) shows the boxes whose category scores were over 0.5. We used the mAP as a model of performance evaluation indicators (Wojek et al., 2012), the mAP was the mean value of each category's average precision (AP). The computation formulas of recall (R), precision (P) and average precision (AP) are as shown below:

$$R=\frac{\text{number of correct detection}}{\text{total number of object}} \tag{9}$$

$$P=\frac{\text{number of correct detection}}{\text{total number of detection}} \tag{10}$$

Table 3
The mAP of the improved inception model trained on different datasets.

Dataset	SO[a]	LS[b]	TC[c]	RD[d]	OS[e]	CP[f]	mAP	Run time (s)
600 × 800	92.52	71.09	91.95	92.00	74.97	85.35	84.65	**0.182**
600 × 800 + augmentation	95.48	**77.82**	**95.72**	92.44	**79.54**	86.95	87.99	**0.182**
1000 × 1300 + augmentation	**96.04**	73.89	95.63	**93.67**	79.21	**89.65**	**88.02**	0.227

The bold values mean the best performance in these models.

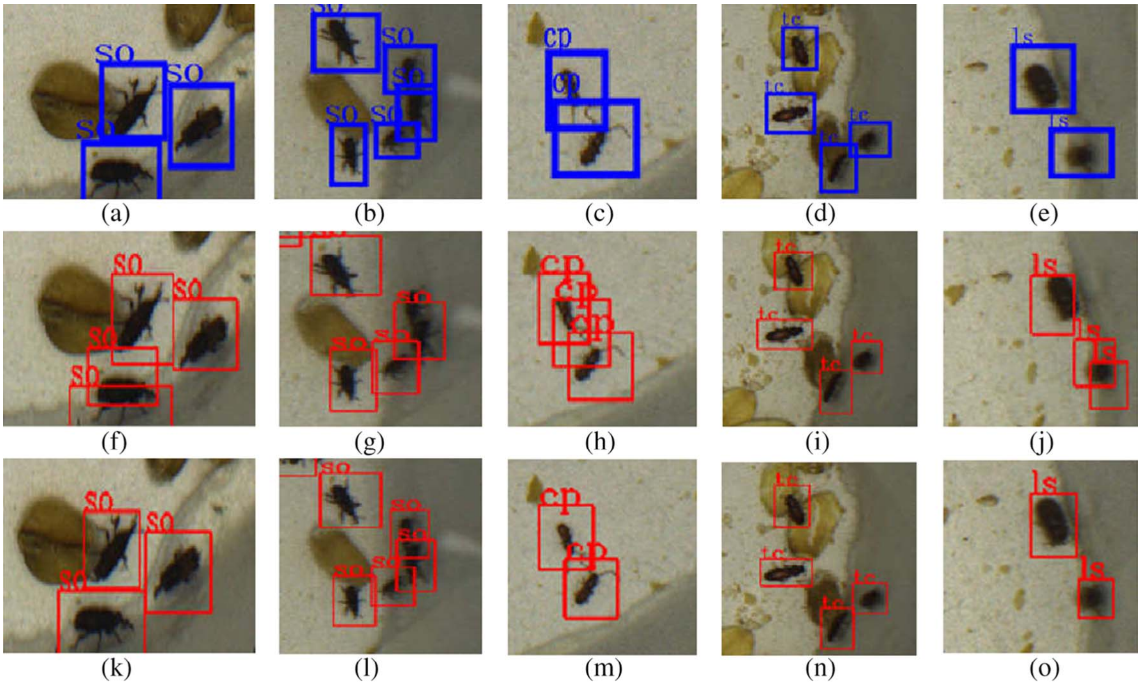

Fig. 8. Ground truth and the performance of different models. (a–e) Ground truth; (f–j) VGG16; (k–o) Improved inception model with SVD operation.

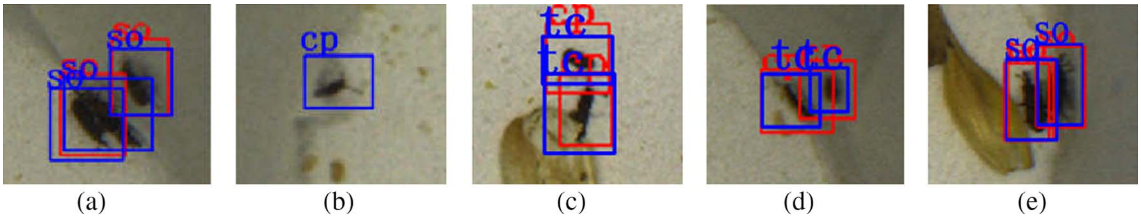

Fig. 9. Examples of error detection. (a) Insects with close adhesion; (b) Missing detection of occluded insects; (c) Insects with distortion; (d) Unstable focus of insects; (e) One species was classified as two species.

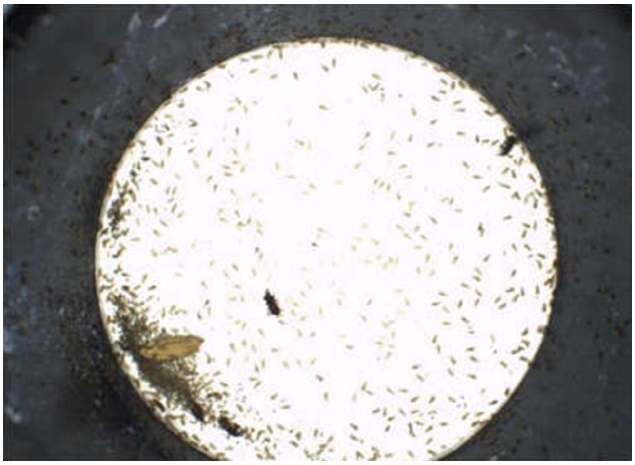

Fig. 10. An example image uploaded by OITD in a grain warehouse.

$$AP = \int_0^1 PdR \tag{11}$$

4.2. Analysis

4.2.1. Model evaluation

The inception network adopted the structure of inception, had better width, better diversity of extracted features, small number of parameters, and faster running speed than that of VGG16. In order to evaluate the insect detection capability of the Faster R-CNN with improved inception network, Faster R-CNN with VGG16, Faster R-CNN with inception, and R-FCN (Wojek et al., 2012) with ResNet101 (Dai et al., 2016) were also trained. In addition, the SVD operation was conducted on the improved inception network to reduce the model parameters, and the impact of SVD operation on the model detection was observed. The comparison of test results of different models trained on the augmented dataset with 600 × 800 pixels is shown in Table 2. These results indicated that the improved inception network had the best performance on the testing set and inception network had the fastest speed. The improved inception model with SVD operation was 199M smaller than the original model size, while the mAP value just had a small drop.

In order to find the influence of the data augmentation and the image resolution on the model performance, images with 600 × 800 pixels, images with 600 × 800 pixels and augmentation, and images with 1000 × 1300 pixels and augmentation were used to train the improved inception model. The mAP of the improved inception model trained on different datasets was shown in Table 3. The performance of this model was significantly improved by the augmentation, while the high resolution of training images slightly improved the performance and had a lower running speed.

4.2.2. Discussion

The improved inception model with SVD operation trained on images of 600 × 800 pixels resolution with augmentation was selected as the final model. The comparison of detection results between the VGG16 and the developed inception network is shown in Fig. 8. The developed detection method could effectively detect insects with different gestures and slight adhesion. However, this method could make errors when there were severe adhesion, occlusion, distortion, and unstable focus of the insects (Fig. 9). The reason for one species classified as two species was that both category scores of *Sitophilus Oryzae* and *Tribolium Confusum* were higher than 0.5.

The inception network published in the literature (Szegedy et al., 2015) had deeper structures than VGG16, so the gradient diffusion problem was more serious and the detection performance was poorer than VGG16. The improved inception network had improved the detection performance compared with inception network and VGG16. Although the R-FCN with ResNet101 could achieve a good mAP, one image took 0.246 s on this model was slower than other models. SVD operation had a significant effect on the compression of the fully connection layer, which reduced the size of the model to about 60M.

The data augmentation could significantly improve the performance of the model. The high resolution of images had no contribution to the higher mAP, and images with high resolution increased the unnecessary details. In addition, increasing resolution would increase the memory consumption and reduce the detection efficiency of images.

However, the dataset adopted in this paper just contained six types of stored-grain insects, and the pictures taken in the laboratory differ from the pictures taken by OITD in the grain warehouses. Fig. 10 shows an image uploaded by OITD in a grain warehouse, the image contained lots of booklice. By May 2017, we had arranged OITD for 8 different warehouses of China. In the future research, we will enrich our image dataset with these images from these grain warehouses, improve our algorithm, and apply the automatically detection system inside stored grain bins.

5. Conclusions

This paper applied the object detection algorithm, which was based on Faster R-CNN, to detect stored-grain insects under field condition with impurities. The method could detect the insects with slight adhesion. An improved inception network was also developed to enhance the accuracy of small insect detection through the deep convolutional neural networks. The improved inception network had achieved a higher mAP (87.99) than the inception network proposed in literature (Liu et al., 2016) (81.39) and VGG16 (82.66). Besides, we enriched the datasets and got a 3.34 improvement of mAP and compressed the model from 261M to 62M by SVD operation with 0.5 reduction on mAP. We also got the images from grain warehouses to improve the identification accuracy in the future.

Acknowledgements

This work is partially supported by the Program of Introducing Talents of Discipline to Universities of China (B08004), and the 2015 China Special Fund for Grain-scientific Research in Public Interest (201513002).

References

Dai, J., Li, Y., He, K., et al., 2016. R-FCN: object detection via region-based fully convolutional networks. Comput. Sci.

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

Girshick, R., 2015. Fast R-CNN. Comput. Sci.

Girshick, R., Donahue, J., Darrell, T., et al., 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. Computer Vision and Pattern Recognition. IEEE, pp. 580–587.

He, K., Zhang, X., Ren, S., et al., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37 (9), 1904–1916.

He, K., Zhang, X., Ren, S., et al., 2016. Deep residual learning for image recognition. Conference on Computer Vision and Pattern Recognition. IEEE 770–778.

Hosang, J., Benenson, R., Dollár, P., et al., 2015. What makes for effective detection proposals? IEEE Trans. Pattern Anal. Mach. Intell. 38 (4), 814–830.

Jayas, D.S., 2017. The role of sensors and bio-imaging in monitoring food quality. Resour. Mag. 24(2), 12–13.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems. Curran Associates Inc, pp. 1097–1105.

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

Neubeck, A., Van Gool, L., 2006. Efficient non-maximum suppression. In: International Conference on Pattern Recognition. IEEE, pp. 850–855.

Qiu, D., Zhang, H., Chen, T., et al., 2003. Software design of an intelligent detection system for stored-grain pests based on machine vision. Trans. Chinese Soc. Agric. Mach. 34 (2), 83–85.

Ren, S., He, K., Girshick, R., et al., 2017. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. Comput. Sci.

Srivastava, N., Hinton, G., Krizhevsky, A., et al., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15(1), 1929–1958.

Szegedy, C., Liu, W., Jia, Y., et al., 2015. Going deeper with convolutions. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–9.

Wang, D., Zhou, H., et al., 2016. Research on image acquisition and recognition for stored grain pests. In: International Conference on Artificial Intelligence & Industrial Engineering.

Wilson, D.R., Martinez, T.R., 2003. The general inefficiency of batch training for gradient descent learning. Neural Netw. 16 (10), 1429–1451.

Wojek, C., Dollar, P., Schiele, B., et al., 2012. Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. 34 (4), 743.

Wu, Y., Wang, K., Tao, F., 2015. Classification of stored-grain insects based on the Extend Shearlet Transform, Krawtchouk Moment and SVM. J. Chinese Cereals Oils Assoc. 30(11), 103–109.

Zhang, H., Fan, Y., Tian, G., 2005. Research of the stored-grain insects classification based on image processing techniques. J. Zhengzhou Inst. Technol. 26(1), 19–22.

Zhang, H., Mao, H., Qiu, D., 2009. Feature extraction of image classification on stored-grain insects. Trans. Chinese Soc. Agric. Eng. 25 (2), 126–130.