IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 2, 2020, accepted August 25, 2020, date of publication September 4, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021830

Detection and Identification of Stored-Grain Insects Using Deep Learning: A More Effective Neural Network

ZHICHAO SHI, HAO DANG, ZHICAI LIU, AND XIAOGUANG ZHOU
School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China
Engineering Research Center of Information Network, Ministry of Education, Beijing 100876, China

Corresponding authors: Hao Dang (danglee@bupt.edu.cn) and Xiaoguang Zhou (zxg_bupt@126.com)

ABSTRACT The detection and identification of stored grain insects is important to ensure the safety of grain during grain storage. At present, insect identification methods primarily rely on manual classification; therefore, the automatic, rapid and accurate detection of stored grain insects remains a challenge. This paper proposes an improved detection neural network architecture based on R-FCN to solve the problem of detection and classification of eight common stored grain insects. In this network, we use the multiscale training strategy with a fully convolutional network to extract more features of the insects and automatically provide the location of potentially stored grain insects through an RPN from the feature map. By using the position-sensitive score map to replace some fully-connected layers, our network is more adaptive to detect insects in complicated backgrounds, and our detection speed is improved. In addition, we also used soft-NMS to solve the superposition interference between insects and to further improve the detection accuracy. Sufficient comparative experiments are performed using our two stored grain insect detection datasets, which are carefully annotated by entomologists. Quantitative comparisons against several prior state-of-the-art methods demonstrate the superiority of our approach. Experimental results show that the proposed method achieves a higher accuracy and is faster than the state-of-the-art insect image classification algorithms.

INDEX TERMS Stored-grain insect, object detection, insect classification, improved R-FCN, soft-NMS.

I. INTRODUCTION

Insects are one of the direct causes of loss during postharvest operations [1]; therefore, it is crucial to detect and identify stored grain insects by using a stored grain insect monitoring system.Due to the disadvantages of traditional methods [2] (e.g., near -infrared, acoustical methods, electrical conductivity), such as difficulty in sampling, slow speed, and manual work, rapid and accurate pest detection has long been a difficult problem to solve. However, the image recognition method based on deep learning can acquire many images of grain insects with low cost and a high recognition rate through seduction and other methods [3].

Deep learning has made many improvements in agriculture through the progress of science and research, such as leaf diseases identification [4] and insect recognition [5], [6]. In addition, deep learning is used to classify the feature vectors from insect image features based on the generalized learning ability of big data to quickly identify the categories of different insects. Meanwhile, the target detection technology based on deep learning can automatically learn and generalize the characteristics of large picture data. Image detection also has notably high research value in the field of stored grain pest detection.

In the field of insect classification based on computer vision, the extraction of insect texture, shape, and local characteristics have long been the focus of research [7]– [10]. Xie *et al.* [11] combined a sparse-coding technique for encoding insect images with multiple-kernel learning (MKL) techniques to construct an insect recognition system, which achieved an mAP (mean average precision) of 85.5% on 24 common insects in crop fields. Lim *et al.* [12] adopted Alexnet and Softmax to build an insect classification system, which was optimized by adjusting the network architecture. Yalcin [13] proposed an image-based insect classification method by using four feature extraction

The associate editor coordinating the review of this manuscript and approving it for publication was Shuping He.

methods: Hu moments (Hu), Elliptic Fourier Descriptors (EFD), Radial Distance Functions (RDF) and Local Binary Patterns (LBP), but these images need preprocessing manually, which is undoubtedly very time consuming. Pjd *et al.* [14] proposed a prototype automated identification system that distinguishes five parasitic wasps by identifying the differences of the wing structure. Mayo and Watson [15] developed an automatic identification system using SVM to recognize the images of 774 live moths, without manually specifying the region of interest (ROI). Because of the different characteristics of different species and because adults of the same species may exhibit the same characteristics, the classification of stored grain pests has reached a high level of accuracy. However, in the practical application process, there are many problems in the image detection of grain insects, such as the confusion between insects and grain particles, the shielding of grain grains from grain insects, and the shielding of grain insects from grain grains.

To further solve the problem of multiple pests in real pictures, the researchers also did a lot of research. Ding and Taylor [16] proposed a neural network model based on deep learning to classify and count the number of moths and achieved successful results under relatively ideal experimental conditions. Shen *et al.* [17] proposed a faster-RCNN framework based on a convolutional neural network to detect 6 stored grain pests with an accuracy rate of 88.02%. Liu *et al.* [18] proposed the PestNet network structure and used RPN, position score sensitivity and other technologies to detect pests in the field, which obtained good results. Xia *et al.* [19] proposed the Region Proposal Network based on a convolutional neural network to detect insects in crop yield, which achieved relatively high accuracy. The rapid identification of insects by existing models in the field of stored grain insects has revealed that there are still some problems to be solved, especially in the detection and identification of multiple insects in a single picture. The problems that need to be solved primarily occur when features are not clear (the insects were too small relative to the picture), features are similar (such as the features of Sitophilus oryzae and Sitophilus zeamais), and the target insects are superposed.

In this context, we proposed improved neural network architecture, the improved R-FCN, for rapid and accurate pest detection. In this architecture, we used a convolutional neural network for feature extraction, a region proposal network, and a position-sensitive score map for target detection. The strategy of a position-sensitive score map and position-sensitive ROI pooling are proposed in the network structure of the R-FCN [20]. As the network structure of R-FCN eliminated the time consumption of the two-layer full connection layer in the RCNN series network [21]– [23], it adopted the global average pooling strategy to reduce the computation of consumption in the two-stage network, and the speed was notably improved. Therefore, we adopted the network structure of R-FCN as the basic structure and further improved it.

In this study, we developed a method based on R-FCN, which can be used to detect the insects rapidly and accurately. Firstly, we introduced the backbone DenseNet [24] as a feature extraction convolutional neural network and used the depth separable convolution technique [25] to reduce the number of parameters and computations. Through the improved DenseNet, the features on the picture could be better extracted to achieve accurate identification of insects and similar species. Then, the multi-scale training strategy [26] was used to extract more generalized features for accurate identification. In addition, the soft-NMS algorithm [27] was used to optimize the attenuation of the adjacent detection box scores of the overlapped parts, which solved the superposition problem between insects and other insects, insects and grains, and insects and grain pores. Thus, the detection performance of the algorithm was further improved. Finally, we not only carried out corresponding experiments on the images in a laboratory environment, but also verified the images in the actual grain storage environment, and the results were in line with our expectations.

The rest of this article is organized as follows. Section 2 describes the datasets we have created and introduces our proposed network architecture in detail. Section 3 describes the experiments and analyses. Section 4 presents the discussion and limitations of our work. Section 5 presents the conclusion of our work.

II. MATERIALS AND METHODS

A. IMAGE DESCRIPTION AND PREPROCESSING

For stored grain insect detection and identification, almost no standard open-source databases exist. Therefore, with the help of ASAG China (Academy of State Administration of Grain), eight kinds of stored grain insects were obtained in Fig. 1. Dataset 1 was created in a laboratory environment. We obtained 1716 original images from pest culture dishes with a resolution of 2592 × 1944. The total images were randomly sampled and 70% as the training set, and the rest were used as the testing set. The average insect density was approximately 14 per picture. Dataset 2 was created to simulate the actual situation in the green warehouses. We obtained 784 original images with a resolution of 2592 × 1944. The average insect density was approximately 8 per picture. In this paper, we used dataset 2 to verify the model in complex cases. Table 1 below shows the number and species of each original insect image in the training set and test set, as well as the number of stored grain insects in all images.

For the preprocessing of the original image in Fig. 2, we used Labeling (an open-source target detection image annotation software) to carry out the data annotation and store the annotation information in an XML file. In accordance with the format of the PASCAL VOC standard data set, the file directory was arranged to facilitate the data reading of the model.

The biological forms of adult insects of a single species were basically the same. The main differences of insects of different species were in size, shape, and color gloss, among which the differences of shape were mainly in the

TABLE 1. The number of images and insects used for training and testing.

			dataset 1								dataset 2	
			TC[a]	CF[b]	CT[c]	SO[d]	SZ[e]	RD[f]	OS[g]	LS[h]	SO[d]	SZ[e]
original	Training	Images	152	146	165	155	148	135	164	156	272	278
		Insects	2285	2044	2392	2139	2368	1936	2230	2187	2203	2196
	Testing	Images	62	64	70	54	63	58	66	58	116	118
		Insects	936	896	1036	788	806	796	838	788	916	961
	Total	Images	214	210	235	209	211	193	230	214	388	396
		Insects	3221	2940	3428	2927	3174	2732	3068	2975	3119	3157
augmentation	Training	Images	912	876	990	930	888	810	984	936	1632	1668
		Insects	13710	12264	14352	12834	14208	11616	13380	13122	13218	13176
	Testing	Images	372	384	420	324	378	348	396	348	696	708
		Insects	5616	5376	6216	4728	4836	4776	5028	4728	5496	5766
	Total	Images	1284	1260	1410	1254	1266	1158	1380	1284	2328	2376
		Insects	19326	17640	20568	17562	19044	16392	18408	17850	18714	18942

[a] TC = Tribolium Confusum.
[b] CF = Cryptolestes Ferrugineus.
[c] CT = Cryptolestes Turcicus.
[d] SO = Sitophilus Oryzae.
[e] SZ = Sitophilus Zeamais
[f] RD = Rhizopertha Dominica.
[g] OS = Oryzaephilus Surinamensis.
[h] LS = Lasioderma Serricorne.

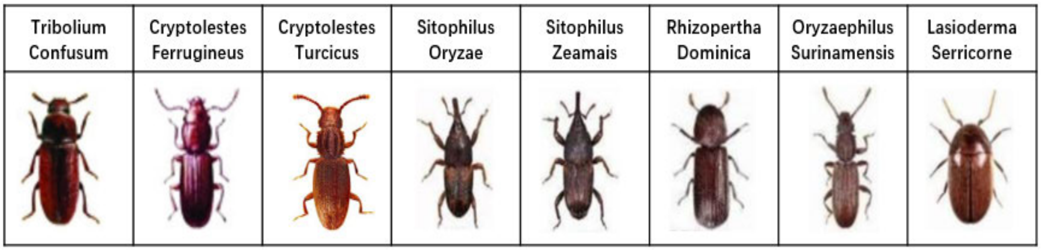


FIGURE 1. Sample images of 8 grain insects in our Work. Note that these sample images are from the specimens, and there is a big gap with the actual situation of the images.

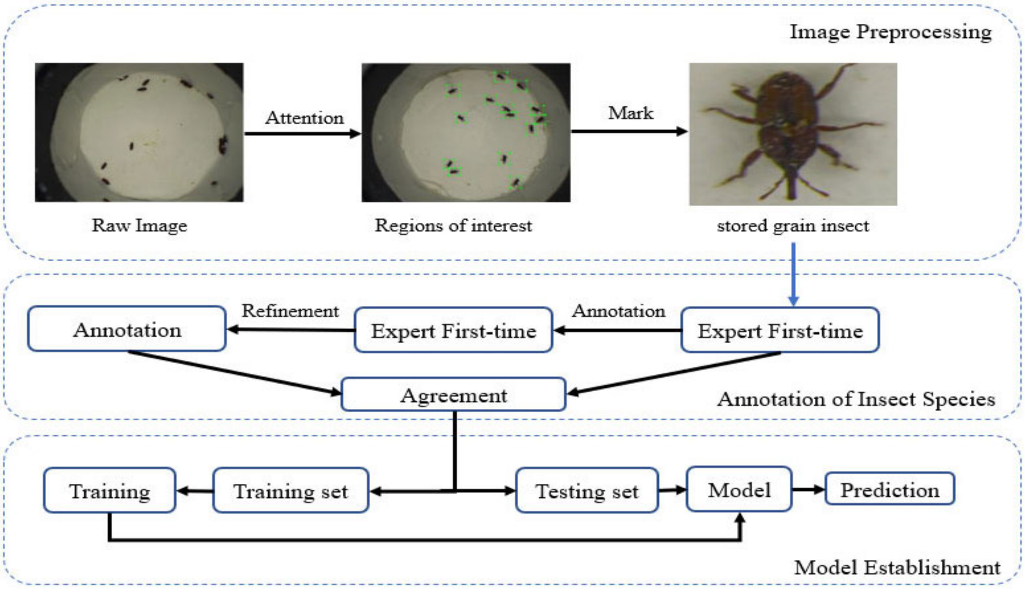


FIGURE 2. The data flow in our research. First, the stored grain insects were marked in raw images. Second, the stored grain insects were finely annotated through two steps by entomologists in grain stored. Finally, the dataset was separated into two subsets and fed into the network for training and prediction.

antennae, head, and backplate. Therefore, in the process of classification, fewer feature values of shape needed to be extracted, and generally satisfactory results could be obtained through a deep feature extraction network. But for the identification process, the location, the direction, and superposition of insects made the identification a difficult problem.

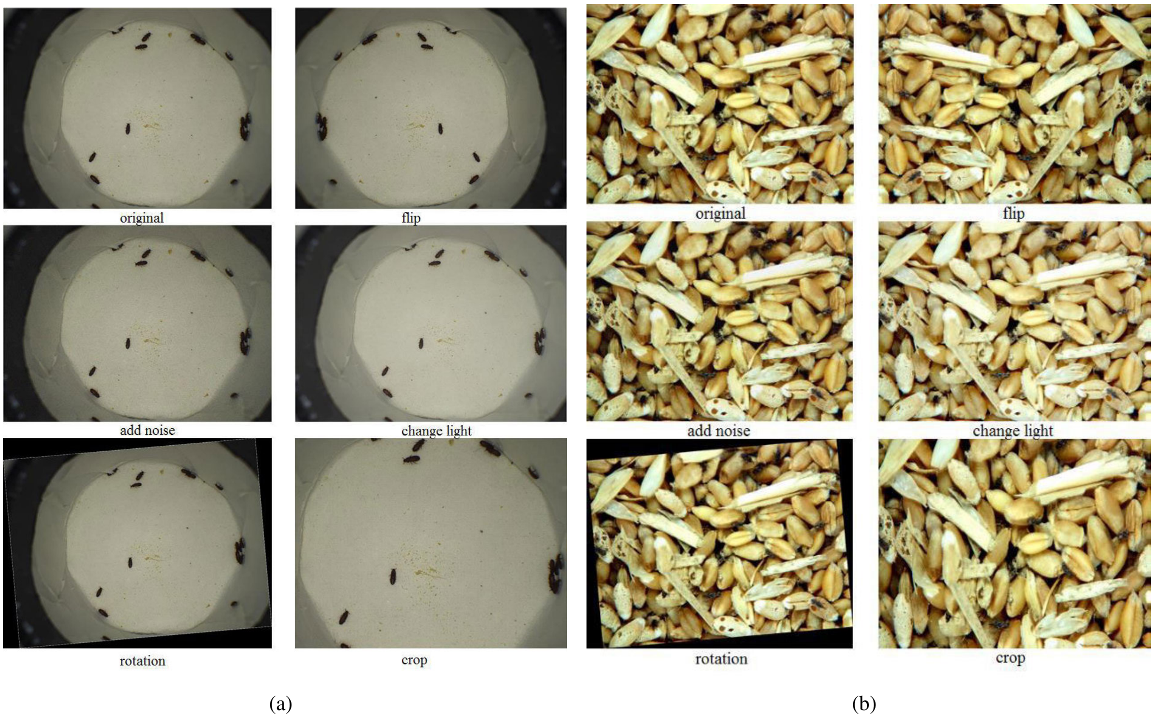


FIGURE 3. Data augmentation. (a) Dataset 1 was created in a laboratory environment; (b) Dataset 2 was simulated in actual situation.

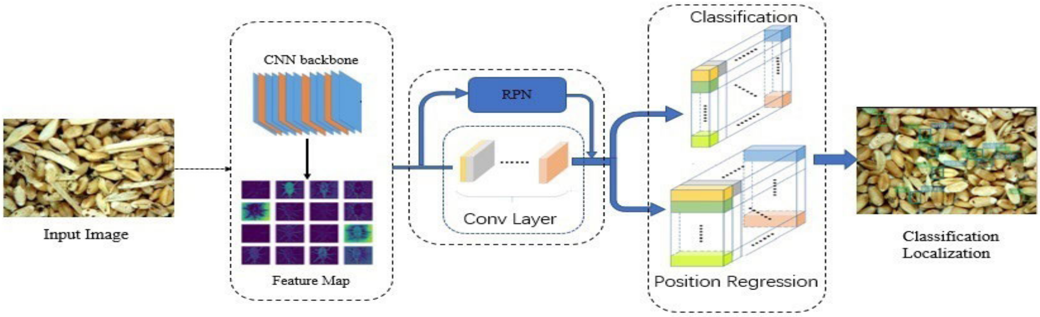


FIGURE 4. Schematic structure of the proposed detection network.

To enrich the training set and improve the generalization ability of the model, five data enhancement methods were adopted to expand the training set [28]. The methods are as follows: 1) enriched the different postures of the insect through horizontal mirror image; 2) added Gaussian noise to the training set to enhance the generalization ability of the model; 3) adjusted the image brightness such that the model could generalize the detection of insects in scenes with different levels of brightness; 4)rotated images to create different orientations and postures of insects; 5) capture different parts of the original image through clipping such that the training set could have more images with different views. Through these methods, our new data volume is 10,296 images, close to 144,000 insect images. Basically, meet the experimental requirements. Table 1 below shows the number and species of each data-enhanced insect image in the training set and test set, as well as the number of stored grain insects in all images. The original image and the data-enhanced image are shown in Fig. 3.

B. OBJECT DETECTION NETWORK

Our neural network in Figure.4 consisted of three stages: insect feature extraction, insect regions search and insect prediction. First, the input image was fed into the CNN backbone to extract feature maps, where DenseNet was used for feature extraction. Then, we fused the RPN with position-sensitive score maps to provide insect regions and insect predictions. During the merging of the overlapping candidate boxes, we used soft-NMS instead of the original NMS to further improve our results. Next, we will discuss each of these stages in more detail.

1) CNN FEATURE EXTRACTION NETWORK

Convolutional Neural Networks (CNNs) are a special type of neural network inspired by the cognitive mechanism of biological vision. The core of the convolutional neural network is the convolution operation. Therefore, a convolutional neural network has excellent performance in image classification, object detection and other computer vision tasks. Without complex image preprocessing, a convolutional neural network can automatically extract the effective features from a large number of original input data, which makes image feature extraction simple and efficient.

In order to obtain better feature results, more complex feature extraction is usually carried out with a deeper network structure [29]– [31]. A variety of novel and advanced CNN network structures greatly improve the classification performance and target positioning performance of the current system [32]. However, as the network deepens,such problems as gradient disappearance/explosion often occur. DenseNet was adopted to optimize the CNN layers of the model. One of the advantages of DenseNet is that each layer in its network structure is directly connected to its front layer, effectively solving the problem of gradient disappearance through realizing the repeated utilization of features. At the same time, the number of channels in each layer of the network is designed to be notably small, strongly reducing the number of parameters and redundancy. The dense block in original DenseNet-121 is shown in Fig.5.

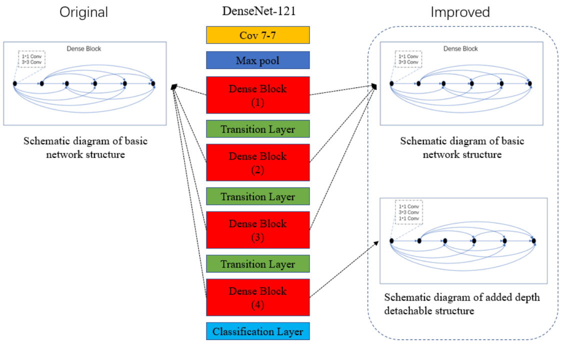


FIGURE 5. Structure of original and improved DenseNet-121. The original part is the initial structure; the improved part is that we use deep separable convolution to optimize the network.

2) RPN

In our network, we adopted an RPN module to obtain object proposals for the potential task of identifying the regions of the objects. As the RPN generates proposals from the points on the feature map obtained from the CNN feature extraction network, it can automatically provide effective regions. In contrast to other relevant methods e.g. selective search [33] and edge boxes [34], RPN adopted the sharing of convolution layer parameters to greatly improved the generation speed of the proposals and introduced various anchor boxes for box regression reference to ensure the quality of the proposed objects.

Fig.6 shows the framework of the RPN module in the training phase. The RPN network mainly completes two tasks: 1) the classification of background and foreground and 2) regression correction of the proposed objects while obtaining their approximate coordinates. First, we map each point on the feature map back to the original image to generate 9 anchors; the Anchor generated by each point has 3 kinds of length-width ratios $(1:1, 1:2, 2:1)$ and scales $(128 \times 128, 256 \times 256, 512 \times 512)$. In this paper, we computed our scales $(8 \times 8, 16 \times 16, 32 \times 32)$ specifically, which ensured an effective receptive field for finding tiny insects on the input images.

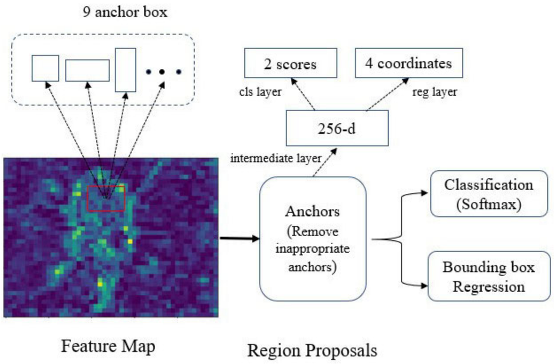


FIGURE 6. Region Proposal Network(RPN).

For predicting categories and creating bounding boxes, we employed softmax regression which is an expansion of logistic regression. Besides, we defined a threshold to filter most of the boxes which held low scores, and non-maximum suppression (NMS) was also applied to retain regions with locally maximal scores, in which Intersection-over-Union (IoU) was adopted as a metric to eliminate most of the overlapping boxes. Training followed Multi-task Loss Function. The loss function included Softmax Classification Loss L_{cls} and Bounding Box Regression Loss L_{reg} [20], that is:

$$\begin{aligned} RPN\_loss &= \frac{1}{N_{cls}} \sum_{i} L_{cls}(p_i, p_i^*) \\ &\quad + \lambda \frac{1}{N_{reg}} \sum_{i} p_i^* L_{reg}(t_i, t_i^*) \end{aligned} \tag{1}$$

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

$$L_{reg}(t_i, t_i^*) = \sum_{i \epsilon x,y,w,h} smooth_{L1}(t_i - t_i^*) \tag{3}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & \text{if |x|<1} \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{4}$$

where N is the numbers of anchors, i is the anchor's index in the training, p_i is the prediction probability of the ith anchor,

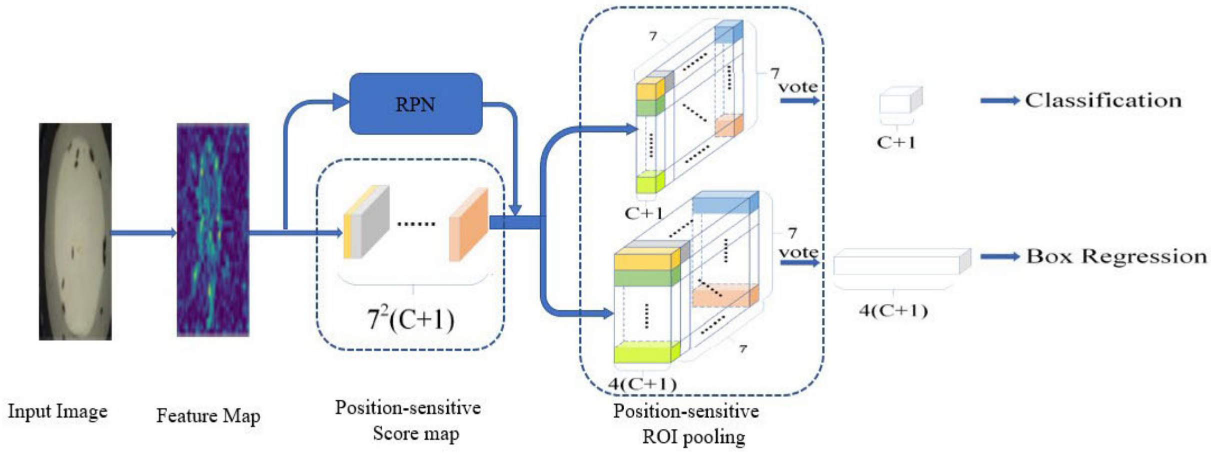


FIGURE 7. The architecture of position-sensitive score map.

p_i^* is the label of the ith anchor, and t_i and t_i^* denote $\{x, y, w, h\}$ of the predicted and true bounding boxes, respectively.

3) POSITION-SENSITIVE SCORE MAP AND POSITION-SENSITIVE ROI CLASSIFICATION

To solve the problem of location insensitivity in classification network and location sensitivity in the detection network, we adopted the position-sensitive score map(PSSM) to encode position information. As shown in Fig.7, a new convolutional layer is extended after feature map to produce a $7^2(C+1)$ channels score map to develop a location sense for each category, in which $(C+1)$ is the number of object categories plus the background. Each region through the RPN is divided into a 7×7 grid and mapped into a score map. Then, the local corresponding score map is processed by the ROI pooling layer to reduce the weight and height by applying an average pooling in each region. Finally, the score map calculated 3*3 confidence scores for $C+1$ categories that represent the possibility of each position, and then the 3×3 scores are averaged and used to vote for the final class score. Similarly, we also fine-tuned the bounding box by augmenting an extra $4 \times 7 \times 7 \times (C+1)$ channels convolutional layer and produced $4 \times (C+1)$ channels in a similar way. Therefore, in the PSSM method, the two score maps are sensitive to the positions of region proposals because various channels indicate different positions.

4) TRAINING

Our loss function defined on each RoI is the summation of the cross-entropy loss and the box regression loss[R-FCN]:

$$R - FCN\_loss = L_{cls}(p_{c^*}) + \lambda[c^* > 0]Lreg(t, t^*) \quad (5)$$

$$L_{cls}(p_{c^*}) = -log(p_{c^*}) \quad (6)$$

$$L_{reg}(t, t^*) = \sum_{i\epsilon x,y,w,h} smooth_{L1}(t_i - t_i^*) \quad (7)$$

In this equation, c^* is the RoI's ground-truth label ($c^* = 0$ represents the background). $[c^* > 0]$ is an indicator that equals 1 if the argument is true and 0 otherwise. We define positive examples as the RoIs that have an intersection-over-union (IoU) overlap with a ground-truth box of at least 0.5.

C. MODEL OPTIMIZATION

1) IMPROVED DenseNet-121

In this paper, we used the improved DenseNet-121 in Fig.5 as the CNN feature extraction network by improving on the original Densenet-121. On the basis of original Densenet-121, we changed the3 $\times$ 3 convolution in the fourth Dense Block with deep separable convolution. In this way, we reduced the parameters and made the network lightweight, and there was an improvement in accuracy.

2) soft-NMS

In this paper, we changed the traditional NMS to soft-NMS [27] to improve the detection accuracy of stored-grain insects with mutual occlusion. The traditional NMS algorithm directly delete objects with lower scores in the two overlapped areas, thereby causing missed detection of occluded objects. In addition, the threshold value of NMS requires relevant experience and many trials to determine a more appropriate value.Soft-NMS made the following improvements:

$$s_i = \begin{cases} s_i, & iou(M, b_i) < N_t \\ s_i(1 - iou(M, b_i)), & iou(M, b_i) >= N_t \end{cases} \quad (8)$$

$$s_i = s_i e^{\frac{iou(M, b_i)^2}{\sigma}}, \quad \forall b_i \notin D \quad (9)$$

In this equation, σ is the attenuation factor,and D was preserved after the maximum inhibition treatment suggestion box set. In this experiment, σ was set to 0.5. Through soft-NMS, we attenuated the scores of adjacent detection boxes with overlapping detection boxes M. The larger the overlap was, the greater the attenuation of the scores was. At the same time, by changing score reset function into a continuous function, the result of the score reset function was prevented from mutating and caused a substantial change in the category score sequence.

3) TRAINING STRATEGY IMPROVEMENT

To further improve the detection performance of the algorithm, multi-scale training techniques were used in the training, and three input image scales were set: 800 × 640, 1024 × 800 and 1280 × 1024. One of the scales was randomly selected for training every 1000 iterations. The experiment proved that more generalized features of the input image at different scales could be learned through multi-scale training, which brought about an improvement inaccuracy. Compared with the previous experiment, it achieved an average improvement of approximately 1%.

III. EXPERIMENTS AND EVALUATION

A. EXPERIMENT SETTING

In this section, we presented some experiments to validate effectively of our neural network. All experiments in this paper were running on two NVIDIA TITAN XP GPUs, and our codes were based on CUDA 9.0 with Pytorch deep learning environment. Before training, we expanded the data set (see section2.1 for details) and compared the difference of the results after the expansion of the data set through Faster R-CNN network. We verified the effectiveness and advantages of several existing exposure models and performed many experiments on our stored grain insect data set. In this paper, we used two-stage detection models (Faster R-CNN, R-FCN), a single-stage detection model (YOLO) [27] and other target detection deep learning frameworks. On this basis, we conducted experiments to change the use of different basic networks in these frameworks and used the open-source model as the benchmark to verify the effectiveness of the improved model in this paper.

In our basic R-FCN network structure, we used a weight decay of 0.0005 and a momentum of 0.9. By default, we use single-scale training where the images were resized such that the scale (shorter side of image) is 600 pixels [27]. Each GPU held 1 image and selected 128 RoIs for the backdrop. We fine-tune the R-FCN using a learning rate of 0.001 for 20k mini-batches and 0.0001 for 10k mini-batches on VOC. We adopted the 4-step alternating training in [21] to make R-FCN share features with RPN (Fig.6), alternating between training RPN and training R-FCN.

B. EVALUATION CRITERION

In target detection, mean average precision (mAP) is often used as the measurement standard of the result. The mAP is the mean value of each category's Average Precision (AP). The calculation formulas are as follows:

$$AP_C(c) = \int_0^1 Precision(c)dRecall(c) \tag{10}$$

$$Precision(c) = \frac{number\ of\ correct\ detection}{total\ number\ of\ detection} \tag{11}$$

$$Recall(c) = \frac{number\ of\ correct\ detection}{total\ number\ of\ object} \tag{12}$$

$$mAP = \frac{1}{N_C}\sum_{c \in C} AP_C(c) \tag{13}$$

Here, c is the category.

C. RESULTS AND DISCUSSION

During the experiment, the trained DenseNet weight parameters on the ImageNet data set were used to initialize the basic network, and the MSRA initialization method was used to initialize the weight parameters of the changed network layer. During the training, the adaptive gradient descent Adam algorithm was adopted to update the parameters backpropagation, and the expanded training set was updated and iterated 60,000 times. The loss curve of the algorithm is shown in Fig.8.

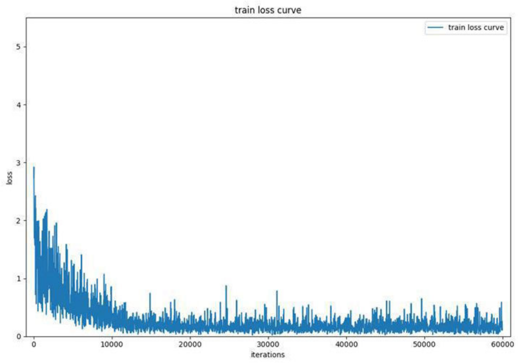


FIGURE 8. Loss graph for 60,000 iterations.

Data enhancement could increase the richness of the data set, so that the convolutional neural network could learn more generalized features, thereby improved the detection performance of the network. Table.2 showed that the average accuracy of the data enhancement Faster R-CNN increases by approximately 4%. Since the effect was obvious and the theory was consistent with our experimental results, we adopted data enhancement in all the networks thereafter. Compared with the two convolution neural networks under Faster R-CNN algorithm, the used of ResNet-101 improved the average accuracy by approximately 3%. We found that a deeper and better basic network could obtain better image features to improve the performance of the detection algorithm. By comparing R-FCN and Faster R-CNN, the network detection speed was greatly improved by maximizing the shared convolution operation and reducing the repetition calculation such that the detection speed of a single image was reduced from 0.217 s to 0.124 s with an average accuracy of 83.44%. The improved image recognition framework substantially improved the recognition effect, and the R-FCN framework had a significant speed advantage.

TABLE 2. Performance of open-source model on stored grain insect data set.

		Faster R-CNN			R-FCN	YOLO
		VGG-16	VGG-16	ResNet-101	ResNet-101	GoogleNet
Dataset		original	augmentation	augmentation	augmentation	augmentation
AP (%)	TC[a]	75.25	80.15	84.23	85.53	71.32
	CF[b]	70.23	75.56	79.56	82.25	65.39
	CT[c]	73.87	73.16	76.17	79.85	68.26
	SO[d]	69.49	74.87	78.25	80.59	65.78
	SZ[e]	68.31	71.89	75.41	77.49	63.52
	RD[f]	76.46	81.54	84.35	87.61	71.81
	OS[g]	80.33	84.76	87.14	87.59	74.85
	LS[h]	79.98	83.69	85.75	86.63	69.24
mAP(%)		74.24	78.20	81.36	83.44	68.77
detection time(s)		0.198	0.198	0.217	0.124	0.022

TABLE 3. The results of the improved model on the dataset by using DenseNet-121.

	AP(%)								mAP	detection
	TC[a]	CF[b]	CT[c]	SO[d]	SZ[e]	RD[f]	OS[g]	LS [h]	(%)	time(s)
R-FCN	87.05	84.37	81.42	82.18	79.94	90.87	87.69	88.73	85.28	0.122
R-FCN+[i]	87.37	86.38	82.69	85.02	80.23	91.47	87.95	89.96	86.38	0.118
R-FCN++[j]	88.35	86.74	83.17	85.23	81.45	92.75	88.53	90.67	87.11	0.118
R-FCN+++[k]	88.94	87.25	83.24	86.36	84.63	93.71	89.32	91.05	88.06	0.118
R-FCN+++(Dataset2)				80.74	78.16				79.45	0.118

[i] R-FCN+ used improved DenseNet-121 which means the 3x3 convolution in the fourth dense block of DenseNet-121 was replaced with a deep separable convolution.

[j] R-FCN++ used the multi-scale training skills in R-FCN+.

[k] R-FCN+++ changed the traditional NMS to soft-NMS in R-FCN++.

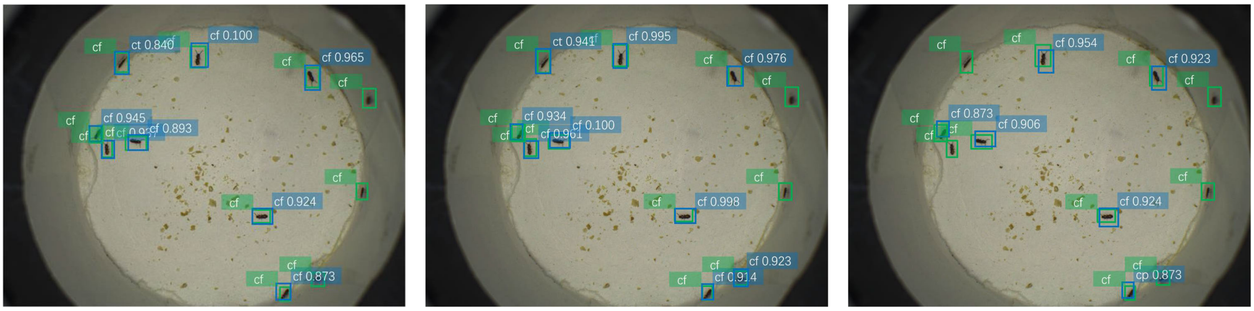
Compared to two-stage detection models (R-FCN and Faster R-CNN) and the single-stage detection model (YOLO), the integration testing framework had the advantage of a single image detection time of 0.022 s. However, the single-stage detection precision was worse than that of the two phase-detection models, and the average accuracy was only 68.77%. Therefore, we used the R-FCN algorithm in the two-stage model and improved it on this basis to obtain a better network structure and better results.

We conducted 4 comparative experiments based on R-FCN algorithm and DenseNet-121 network, and the results are shown in Table 2 and Fig.9. During the basic experiment, we first replaced ResNet-101 with DenseNet-121 which was deeper and had fewer parameters. We scaled the original image to 800*640 resolution as the input of the network and iterated 60,000 times by updating the training set. The detection time of a single image was 0.122s, and the average accuracy mAP on the test set reached 85.28%. The detection performance of our model was an improvement from the detection performance of the model in Table 2 to a certain extent.

For R-FCN+, we used the improved DenseNet-121, which altered the last dense block of the separable convolution in the original DenseNet-121. In this way, we reduced the number of basic network parameters and obtained a faster detection time for individual images (0.118 s) in Table 3. The method's average accuracy on the test set was approximately 1% better than that of the previous network. Compared with the original network structure, our R-FCN+ exhibited an improvement in speed and accuracy.

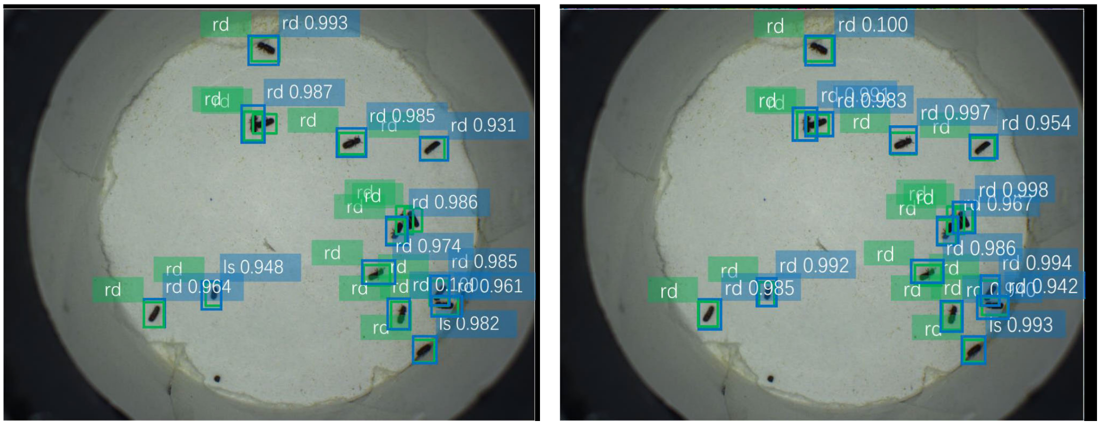
For R - FCN++, we further used the multi-scale training skills, setting $800 \times 640, 1024 \times 800$ *and* 1280×1024 as the three dimensions of input images for training. During the training, one of the three scales was randomly selected as the resolution of the input image in every 1000 iterations, and the training set was updated and iterated 60,000 times. In the test set, the AP of each type of stored grain insect had a slight improvement, and its average detection accuracy mAP reached 87.11%.

For R - FCN+ + +, we improved R - FCN++ and introduced the Soft-NMS ($\sigma = 0.5$) algorithm instead of the traditional NMS algorithm. This method avoided missing the detection of two overlapping insects, as seen in Fig.11. As seen from the experimental results, the final mAP was only improved by less than 1% (88.06%). At the same time, we also compared various values of the parameters σ



(a) (b) (c)

FIGURE 9. Detection results of insect(CF). The green represents our insects tag and the blue represents the result of machine detection. (a) Faster R-CNN(VGG-16); (b)R-FCN(ResNet-101); (c) YOLO.



(a) (b)

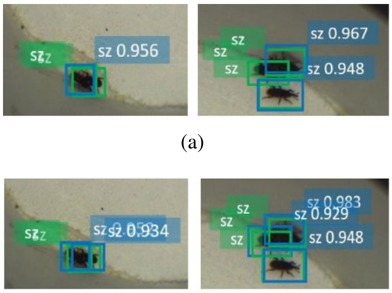
FIGURE 10. Detection results of insect(RD). The green represents our insects tag and the blue represents the result of machine detection. (a) R-FCN(ResNet-101), (b) R-FCN+ + +.

TABLE 4. Sensitivity analysis across the various parameter values of σ for soft-NMS using R-FCN++ on Dataset1.

σ	0.3	0.5	0.7	0.9
$mAP(\%)$	87.95	88.06	87.89	87.72

through experiments and obtained more appropriate parameters in Table 4. According to the results in Table 4, compared with the NMS method, soft-NMS was significantly improved the results, and the effect decreased with the increase in parameters. In addition, the detection accuracy of SZ was strongly improved (approximately 3%) due to more occlusion phenomenon in the image.Test results in Fig.10.

For dataset 2, we only used the network to train images of rice elephants and corn elephants (because they look similar but are relatively accurate in a laboratory setting). We found that the detection accuracy of the RCN+ + + (σ = 0.5) used on real images decreased by 6% on average, but the detection time was not augmented. As seen from the test results in Fig. 12, the actual grain storage environment was more complex, and it was easy to generate shielding between grain grains and grain worms, resulting in missed detection.



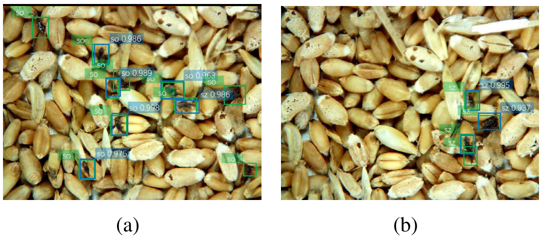
(a)

(b)

FIGURE 11. Detection effect of Soft-NMS on mutual occlusion of insects. (a) the result of R-FCN++; (b) the corresponding result of R-FCN+ + +. Green represents our insect tag, and blue represents the result of machine detection.

IV. DISCUSSION

In this paper, we proposed the R-FCN+ + + as the final model. In this model, we started with the R-FCN+ which used the multi-scale training method to obtain a richer feature convolution to some extent. At the same time, we further put forward the R-FCN++ based on the R-FCN+. We used the improved DenseNet feature extraction network to solve the



(a) (b)

FIGURE 12. **Detection effect of R-FCN+ + + on actual grain storage environment. Green represents our insects tag and blue represents the result of machine detection. (a) the Sitophilus Oryzae(SO); (b) the Sitophilus Zeamais(SZ).**

problem of gradient disappearance of the deep network due to its network structure, and depth-separable convolution to further reduce the calculation amount of the feature extraction network. Finally, we proposed the R-FCN+ + + which used soft-NMS to solve the exciting problem of overlap between stored grain insects.

We compared the detection results of Faster R-CNN and R-FCN frames under different feature extraction networks (VGG-16, ResNet-101, DenseNet-121) in Table 2 and Table 3. Compared to the Faster R-CNN, our speed increased from 0.217s to 0.118s, and the accuracy increased from 81.36% to 88.06%. Also compared with R-FCN network, our speed increased from 0.122s to 0.118s, and accuracy increased from 85.28% to 88.06%. From the results, the accuracy and detection speed of our proposed R-FCN+ + + had been significantly improved compared with the original model. However, our model still needed further optimization.

Table 5 summarizes the various techniques used by researchers to automatically classify and detect insects using images in agriculture. The table includes research literature, databases, feature extraction, accuracy, detection time picture description, and purposes. These studies achieved good results for insect classification. However, there were some limitations to the image data. There was only one insect in a single image, and the pictures of the insects were enlarged. Although good classification results were obtained under complex background conditions, problems were still observed in realizing the automatic monitoring process relative to the small size of the insects.

Insect detection has long been a difficult problem, not only because of the insect's small size but also due to the easily disturbed background and other factors. Although the literature [14]achieved excellent accuracy in the case of codling moths, they are relatively large compared with other insects. Later, Shen *et al.* [17] and Liu *et al.* [18] conducted relevant studies, both of which achieved good results. Among these studies, the improved inception model [17] published in the literature had a wider network structure and a smaller model size. However, our model showed a significant improvement in detection speed and a slight improvement in detection accuracy compared with the improved inception model, ranging from 0.182 s in the literature to our 0.118 s. In addition, the model proposed by Shen *et al.* [17] is based on faster-RCN, while the model proposed by us is based on R-FCN structure. We replaced the last full connection layer with a position-sensitive score to further increase the speed. Although the Pes-Net proposed by Liu *et al.* [18] was also based on the RFCN structure, we made a series of improvements on this basis such that our model had better results in insect detection.

Table 5 shows a variety of feature extraction methods that were applied to insect image detection and classification. Among these methods, the convolutional network, as a better method, has been widely used by scholars. However, considering the relatively few characteristics of stored grain pests, future studies are necessary to determine whether increasing the depth of the CNN feature extraction network will further improve the network effect. To some extent, our model reduced the missing detection and error detection in areas where stored grain insects overlap in fig.7, but the missing detection still existed in the places where the overlap was highly serious, and the error detection still occurred near the edges of the vessels.

The Sitophilus Oryzae(SO) and the Sitophilus Zeamais(SZ) are insect varieties with similar physical features, which are difficult to distinguish by only the naked eye. Therefore, this paper chose these two varieties for an example. In the case of high resolution, we can find that the images in the laboratory environment can still achieve a high detection effect, while in the simulated real environment, the average value decreased by approximately 6%, which is still considerably higher than the detection effect of human eyes. Therefore, pest detection based on computer vision for stored grain is of high significance; computer vision can not only simplify the operation of relevant grain managers but also improve the identification of confusing varieties of insects.

The data augmentation could significantly improve the performance of the model. However, the high resolution of the images might increase the unnecessary details. In addition, the high resolution might increase memory consumption and reduce the detection efficiency of images. In future research, we will enrich more kinds of resolution images.

To a certain extent, the pictures that we obtained did not meet the actual requirements. In this article, there was only one species of insect per image, and therefore, the effects on detection still need further verification. We hope to further supplement the relevant data in future studies and to obtain more real and reliable results.

The dataset 1 adopted in this paper contained eight types of stored-grain insects, and the images taken in the laboratory differ from the pictures taken by equipment in the grain warehouse. Because the pictures in the actual storage environment were more complex, our model could not reach the expected results in the actual application (Table 3), but the results were in line with our expectations. In future research, we will enrich our image dataset with images from grain warehouses, improve our algorithm, and apply the automatic detection system inside stored grain bins.

TABLE 5. Summary of selected studies conducted for the automated detection of insects in agriculture.

Author,Year	Database	feature extraction	One Picture	Purposes	Accuracy(%)	detection time(s)
Xia D *et al.*, 2018 [19]	24 common images of Insects in crop fields from internet	VGG19	one insect	classification	89.22	0.083
X. Cheng *et al.*, 2017 [6]	10 classes of crop pest images	ResNet-101	one insect	classification	98.67	
C. Xie *et al.*, 2018 [5]	40 common images of Insects in crop fields	multi-level deep feature learning	one insect	classification	89.3	
Weiguang Ding *et al.*, 2016 [14]	codling moth	CNN	multiple insects	detection	93.1	
Shen Y *et al.*, 2018 [17]	6 storage insects	Improved inception	multiple insects	classification detection	87.49	0.183
Liu L *et al.*, 2019 [18]	16 classes of crop pest from the data MPD2018	ResNet101	multiple insects	classification detection	75.46	0.441
Proposed model	8 storage insects	Improved DenseNet-121	multiple insects	classification detection	88.06	0.118

V. CONCLUSION

This paper proposed an effective multi-class stored-grain insect object detection network, which was based on R-FCN. This method can detect insects with weak adhesion rapidly and accurately. To improve the detection accuracy and detection speed of deep convolutional neural networks in detecting grain storage insects, an improved DenseNet-121 was proposed. Second, utilizing data enhancement, multi-scale training techniques were used to further improve accuracy. Finally, through the soft-NMS algorithm, the problem of missing detection of two adjacent target objects due to the hard threshold was improved, and the accuracy of detection was further improved. In addition, our model was verified by the actual pictures of two kinds of stored grain insects, and the results were acceptable. We hope to obtain images from actual grain storage in the future to improve the accuracy and speed of identification under a complex background, light, impurities, and other conditions.

REFERENCES

[1] F. Jian and D. S. Jayas, “The ecosystem approach to grain storage,” *Agricult. Res.*, vol. 1, no. 2, pp. 148–156, Jun. 2012.

[2] S. Neethirajan, C. Karunakaran, D. S. Jayas, and N. D. G. White, “Detection techniques for stored-product insects in grain,” *Food Control*, vol. 18, no. 2, pp. 157–162, Feb. 2007.

[3] I. Y. Zayas and P. W. Flinn, “Detection of insects in bulk wheat samples with machine vision,” *Trans. ASAE*, vol. 41, no. 3, pp. 883–888, 1998.

[4] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, “Identification of maize leaf diseases using improved deep convolutional neural networks,” *IEEE Access*, vol. 6, pp. 30370–30377, 2018.

[5] C. Xie, R. Wang, J. Zhang, P. Chen, W. Dong, R. Li, T. Chen, and H. Chen, “Multi-level learning features for automatic classification of field crop pests,” *Comput. Electron. Agricult.*, vol. 152, pp. 233–241, Sep. 2018.

[6] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, “Pest identification via deep residual learning in complex background,” *Comput. Electron. Agricult.*, vol. 141, pp. 351–356, Sep. 2017.

[7] Q. D. Z. Hongtao and Z. Tiejun, “Software design of an intelligent detection system for stored-grain pests based on machine vision,” *Trans.Chinese Soc. Agric. Mach.*, vol. 34, no. 2, pp. 83–85, 2003.

[8] Y. Wu, K. Wang, and F. Tao, “Classification of stored-grain insects based on the extend shearlet transform,” *Krawtchouk Moment SVM. J. Chin. Cereals Oils Assoc*, vol. 30, no. 11, pp. 103–109, 2015.

[9] H. Zhang, H. Mao, and D. Qiu, “Feature extraction of image classification on storedgrain insects,” *Trans. Chin. Soc. Agric. Eng.*, vol. 25, no. 2, pp. 126–130, 2009.

[10] D. S. Jayas, “The role of sensors and bio-imaging in monitoring food quality,” *Resour. Mag.*, vol. 24, no. 2, pp. 12–13, 2017.

[11] C. Xie, J. Zhang, R. Li, J. Li, P. Hong, J. Xia, and P. Chen, “Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning,” *Comput. Electron. Agricult.*, vol. 119, pp. 123–132, Nov. 2015.

[12] S. Lim, S. Kim, and D. Kim, “Performance effect analysis for insect classification using convolutional neural network,” in *Proc. 7th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2017, pp. 210–215.

[13] H. Yalcin, “Vision based automatic inspection of insects in pheromone traps,” in *Proc. 4th Int. Conf. Agro-Geoinform. (Agro-Geoinform.)*, Jul. 2015, pp. 333–338.

[14] P. J. D. Weeks, M. A. O’Neill, K. J. Gaston, and I. D. Gauld, “Automating insect identification: Exploring the limitations of a prototype system,” *J. Appl. Entomology*, vol. 123, no. 1, pp. 1–8, Jan. 1999.

[15] M. Mayo and A. T. Watson, “Automatic species identification of live moths,” *Knowl.-Based Syst.*, vol. 20, no. 2, pp. 195–202, Mar. 2007.

[16] W. Ding and G. Taylor, “Automatic moth detection from trap images for pest management,” *Comput. Electron. Agricult.*, vol. 123, pp. 17–28, Apr. 2016.

[17] Y. Shen, H. Zhou, J. Li, F. Jian, and D. S. Jayas, “Detection of stored-grain insects using deep learning,” *Comput. Electron. Agricult.*, vol. 145, pp. 319–325, Feb. 2018.

[18] L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, S. Sudirman, and W. Liu, “PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification,” *IEEE Access*, vol. 7, pp. 45301–45312, 2019.

[19] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, “Insect detection and classification based on an improved convolutional neural network,” *Sensors*, vol. 18, no. 12, p. 4169, Nov. 2018.

[20] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 7, 2016, pp. 379–387.

[21] Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[22] R. Girshick, "Fast R-CNN," 2015, *arXiv:1504.08083*. [Online]. Available: http://arxiv.org/abs/1504.08083

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[24] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "DenseNet: Implementing efficient ConvNet descriptor pyramids," 2014, *arXiv:1404.1869*. [Online]. Available: http://arxiv.org/abs/1404.1869

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[26] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[27] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5561–5569.

[28] B. Xiao, J.-F. Ma, and J.-T. Cui, "Combined blur, translation, scale and rotation invariant image recognition by radon and pseudo-Fourier–Mellin transforms," *Pattern Recognit.*, vol. 45, no. 1, pp. 314–321, Jan. 2012.

[29] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[33] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[34] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv:1506.02640*. [Online]. Available: http://arxiv.org/abs/1506.02640

ZHICHAO SHI received the B.S. degree in automation from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree in control science and engineering with the School of Automation. His research interests include post-harvest grains storage data analysis, intelligent systems, and machine learning.

HAO DANG received the M.S. degree in pattern recognition and intelligent system from the Henan University of Technology, Zhengzhou, China, in 2016. He is currently pursuing the Ph.D. degree in control science and engineering with the School of Automation, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include the pattern recognition, intelligent systems, machine learning, and so on.

ZHICAI LIU received the master's degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2019. His research interests include grain storage pests, computer vision, and neural networks.

XIAOGUANG ZHOU received the M.S. degree from the Department of Precision Instrument, Tsinghua University, in 1984, and the Ph.D. degree in engineering from the Tokyo University of Agriculture and Technology, Japan. From 2001 to 2002, he was a Visitor Professor with the Tokyo University of Agriculture and Technology. From 2013 to 2014, he was a JSPS Researcher with Tokyo University. He is currently a Professor and a Ph.D. Supervisor with the School of Automation, Beijing University of Posts and Telecommunications. He is also the Director of the Engineering Research Center of Information Networks, Ministry of Education. He is the author of over ten books, over 100 articles, and over 16 inventions. His research interests include control theory and its application in engineering, deep learning, computer vision, the Internet of Things, automated logistics systems, and mechatronics technology. He is a Permanent Member of the Chinese Association of Automation/Manufacturing Technology Committee and the China Institute of Communications/Equipment Manufacturing Technical Committee.