

Research Article

Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests

Yong He ^{1,2,3} Hong Zeng,^{1,2,3} Yangyang Fan,^{1,2,3} Shuasheng Ji,^{1,2,3} and Jianjian Wu^{1,2,3}

¹College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China

²State Key Laboratory of Modern Optical Instrumentation, Zhejiang University, Hangzhou 310058, China

³Key Laboratory of Spectroscopy Sensing, Ministry of Agriculture and Rural Affairs, Hangzhou 310058, China

Correspondence should be addressed to Yong He; yhe@zju.edu.cn

Received 22 February 2019; Revised 10 June 2019; Accepted 23 June 2019; Published 10 July 2019

Academic Editor: Nicola Bicocchi

Copyright © 2019 Yong He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we proposed an approach to detect oilseed rape pests based on deep learning, which improves the mean average precision (mAP) to 77.14%; the result increased by 9.7% with the original model. We adopt this model to mobile platform to let every farmer able to use this program, which will diagnose pests in real time and provide suggestions on pest controlling. We designed an oilseed rape pest imaging database with 12 typical oilseed rape pests and compared the performance of five models, SSD w/Inception is chosen as the optimal model. Moreover, for the purpose of the high mAP, we have used data augmentation (DA) and added a dropout layer. The experiments are performed on the Android application we developed, and the result shows that our approach surpasses the original model obviously and is helpful for integrated pest management. This application has improved environmental adaptability, response speed, and accuracy by contrast with the past works and has the advantage of low cost and simple operation, which are suitable for the pest monitoring mission of drones and Internet of Things (IoT).

1. Introduction

Oilseed rape is one of China's four major oilseed crops (soybean, oilseed rape, peanut, and sunflower), and its oil production efficiency is extremely high, which accounts for 18.9% of the total annual oil production of oil crops in the world and 55% of the total oil production of oil crops in China [1]. It plays a significant role in the market as edible vegetable oil. People have tried many pest control measures to protect oilseed rape from pests. However, the pesticides are used excessively or misused, which will cause huge economic losses and serious environmental contamination [2, 3]. The Food and Agriculture Organization of the United Nations (FAO) estimates that, despite the application of about 2 million tons of pesticides, annual global crop production losses due to diseases and pests are about 20–40% [4]. Therefore, accurate and timely diagnosis of oilseed rape pests is essential for the oilseed rape

production. For a long time, pest identification mainly depends on crop technicians and experienced farmers by the morphology. However, the artificial identification is subjective, inefficient, and delayed. Therefore, it is necessary to find an objective, efficient and rapid method for detecting pests.

Because pest images can be expanded for further processing after acquisition, images are often used as data, and object recognition is one of the most challenging tasks in the area of computer vision (CV) [5]. CV refers to the automatic extraction, analysis, and understanding of useful information from a single image or a sequence of images [6]. The tasks of CV include classification, localization, object detection, segmentation [7], etc. Object detection is the process of finding real-world instance objects in images or videos [8]; it produces bounding boxes and class labels to locate and identify multiple targets. Generally, there are three steps in the object detection, which include extracting

features from the input image, selecting detection window, and designing a classifier [9]. The traditional detection algorithm is built on handcraft features and shallow trainable architectures, which lacks robustness and sensitivity to diversity changes. In addition, the region selection strategy based on sliding window is computationally expensive and time-consuming and usually produces redundant windows [10]. For most CV tasks, the detection algorithm based on deep learning performs better than traditional detection techniques [11]. Deep learning can understand high-dimensional data very efficiently and has achieved certain results in the fields of classification, speech recognition, natural language processing, recommendation system, and automatic driving [12, 13]. Deep neural networks do not require designing features manually, which makes them more reliable and adaptable for different types of variations in the image data [14].

In recent years, with the rapid development of deep learning in object detection, a great deal of deep detection models has been proposed. Typical detectors include Faster R-CNN [15], R-FCN [16], YOLO (you only look once) [17], SSD (single-shot multibox detector) [18], etc. The common evaluation of these two-stage detectors are that they are in the detection precision but need a big computational cost. One-stage detectors, such as YOLO and SSD, have lower accuracy but faster speed, which means they are fit to be used on mobile devices. After comparing the mAP values, detection time, and model size, this paper chose SSD as the detection framework.

Recently, the growing interest in sustainable agriculture has extended its application to diagnose the crop diseases and pests [19]. Some research studies have been performed to develop pest detection models based on deep learning, whose results showed that they are promising in pest detection tasks compared to traditional detection algorithms and can better adapt to the complex field environments. But for this study, there are three main challenges: first, there is lack of available public pest dataset, and most of current researches use insect images collected in ideal laboratory environments [20]. Second, the field environment is complex, so the existing pest models could not be directly used for pest detection which needs further training and adjustment to achieve the best performance. Third, the equipment of the pest detection system used first is expensive and bulky and of poor practicability.

In this paper, a variety of oilseed rape detection models based on deep convolutional neural networks have been evaluated and tested on oilseed rape pest dataset to construct a detection system for pests of oilseed rape in complex environments. Furthermore, we have developed the first Android application of pest detection based on deep learning in China. This paper is organized as follows. Related works are discussed in Section 2. In Section 3, we present the oilseed rape pest dataset and methodologies. In Section 4, we compare the performance of the six models and propose two methods to improve the mAP. In Section 5, we designed and evaluated a pest detection system of oilseed rape based on Android. We offered the conclusion in Section 6.

2. Related Works

In this section, we will introduce some major object detection methods briefly. In general, advanced object detection methods can be categorized into two categories, including one-stage and two-stage detectors. All of these proposed improvements of detection performance are based on standard datasets such as MS COCO [21], PASCAL VOC [22], and ILSVRC [23]. The two-stage detectors have a box proposal network and a classification network, such as R-CNN, R-FCN, and FPN [24]. The method extracts features and generates region of interests (RoIs), which could be used for the subsequent object classification and bounding box regression. The first breakthrough in the object detection is the proposal of R-CNN. After then, massive improved models have been suggested with multiple convolution layers, which requires huge calculation capacity. Then, Faster R-CNN solves the speed bottleneck of fast R-CNN and R-CNN, as it proposed a region proposal network (RPN) to generate region proposal. Hereafter, R-FCN is proposed to further improve the performance.

Nowadays, the challenges in the object detection are the requirements for the faster speed and high efficiency, which is obvious that in the one-stage detectors like YOLO and SSD, these algorithms are usually faster but reach a lower mAP than the two-stage detectors. The one-stage detectors combine region proposals and classification into one network. For the input image, the bounding boxes and classification are simultaneously predicted at multiple positions of the image without the region proposals. YOLO runs real-time by processing the image on CNN once. SSD can detect multiple scales by a multiscale feature map. Although these models can be used for object detection, it is difficult to find a model that achieves a balance between accuracy and speed.

Object detection has been dominated by a sliding window. For instance, Ding and Taylor [25] have collected the images of the codling moth in the field and designed an automatic moth detection system based on sliding window to realize pest detection and counting. The experimental results showed that the optimal model accuracy is 93.4%, and the log-average miss rate is 9.16%. However, the approach is not only computationally expensive but also only suitable for specific task. When it is in a nonlaboratory environment, experiment cannot reach good detection performance. Recently, more object detection methods based on deep learning have been used in agriculture, mainly for the disease and pest detection. Great changes have taken place in the field of object detection, and the original sliding window approaches were replaced by the region proposals [26]. In order to achieve efficient detection of main organs of tomato, Zhou et al. [27] built a classification network model based on VGGNet, designed TD-Net with Fast R-CNN, and completed a tomato organ detector. The average precision (AP) of the detector for fruit, flower, and stem were 81.64%, 84.48%, and 53.94%, respectively, whose performance and speed have been improved compared to R-CNN and fast R-CNN. Fu et al. [28] have reported on a kiwifruit

recognition system based on the LeNet network and studied the overlap and occlusion of multicluster kiwifruit. The overall recognition rate achieved 89.29%, and the average time of recognizing one fruit is 0.27 s, which is a strong support for the harvesting robot. In the classification task, there are many applications for developing a prediction model based on spectral characteristics and image data and have achieved good results [29–32].

3. Materials and Methods

3.1. Dataset

3.1.1. Data Collection. Due to the lack of publicly available oilseed rape pest dataset, we established a new oilseed rape pest dataset by two ways. The first means is downloading pictures through the Internet. But the resolution and brightness of the picture from the Internet are different, and the target pests in the image are moderately sized and in various postures. The second means is capturing in both laboratory and field to enrich the dataset. In the field condition, mobile phone (iPhone 6S, Apple Inc.) and digital camera (Canon 5D Mark III, Canon) are used to collect live pests images in multiple places and different poses. The acquired images contain much background clutter making them in poor quality [33], so the target pests are small and fuzzy. At the same time, the pests were captured in the field and photographed in the laboratory using a digital camera with good lighting condition to ensure high quality. An image acquisition device was constructed with reference to an image acquisition system designed by Yao et al. [34], as shown in Figure 1. This system includes a digital camera, 2 light sources, a glass plate placed on a table, tripods used to hold the camera and light sources; a desktop computer to control the camera; and a mobile phone with Wi-Fi to get images from the camera directly. The oilseed rape pests must be spread on the glass plate evenly, and it will make the following image segmentation and identification easier. Two light sources light all images from bilateral symmetry above to eliminate shadows.

The size of the pests is too small, and the shooting equipment is limited, so the images obtained in the field and lab are small and need to be further cropped. A total of 3,022 oilseed rape pest images were obtained under multiangle shooting by three ways. We unified the images format into JPEG and named them in consecutive numbers [35]. According to the species of pests, they were divided them into 12 categories. Some images are shown in Figures 2(a)–2(c).

3.1.2. Dataset Arrangement. Before the annotation, the dataset needs to be refined manually to avoid the image type annotation errors and the images duplication caused by pest interspecies similarity. Then, the image annotation tool LabelImg (v1.3.0) was used to mark the categories and rectangular bounding boxes of the pest images. Table 1 shows the number of images and the number of targets

for each sample. It contains 12 typical oilseed rape pests, listed by the first letter of the name.

From Table 1, it is clear that there is an uneven class balance in the dataset and the number of various data varies greatly. There are 610 images of *Pieris rapae* but only 30 images of *Entomoscelis suturalis*. *Lipaphis erysimi* has 1,474 annotated objects while *Entomoscelis suturalis* only has 36 objects. When the number of images in the dataset is insufficient, it is necessary to increase the data so that the maximum number of images is kept at a certain ratio with the minimum number of images.

3.1.3. Data Augmentation (DA). DA refers to keeping the labels unchanged and transforming the training data to increase the amount of the training data, which is a common strategy for increasing the quantity of training data and has been applied to specific tasks [36, 37]. By making data more diverse and increasing the interference artificially, the problem of inadequate data and imbalanced label distribution could be overcome [38], and the generalization of the model could be improved. DA can be divided into two categories, one is common DA and the other is complex DA. Complex DA is a scheme that artificially expands the dataset by using domain specific synthesis to generate more training data but is computationally expensive and time-consuming to implement. Common DA calculation is low cost and easy to implement [39]. Thus, for the small datasets and experimental environments given in this paper, Common DA is a more appropriate method.

Generally, the process of object detection will be affected by the interference such as viewpoint, pose, illumination variation, and occlusion. Common DA used in the deep learning include cropping, rotation, color jittering, and adding noise to create more images, and each of them can simulate a type in real-world. The specific examples are presented in Figures 3(a)–3(h). We performed some DA processing on the original image, such as brightness, contrast, saturation, image flipping, rotation, cropping, translation, etc.

3.2. Methodologies. In this paper, we focused on the three major meta-architectures for object detection: Faster R-CNN, R-FCN, and SSD (single-shot detector) and used the dropout method to avoid overfitting.

3.2.1. Dropout. In neural networks, overfitting is the major weakness which makes training large neural networks extremely hard. It is impossible to perform model combination in a short time and avoid overfitting. Thus, we adopt the “dropout” method, which will drop out units to overcome this problem in neural networks. By removing units, we got new thinned networks, and it is easy to train smaller sub-models and achieved better performance. The dropout is random at each unit. For example, each unit that has the fixed probability p will not work independently.

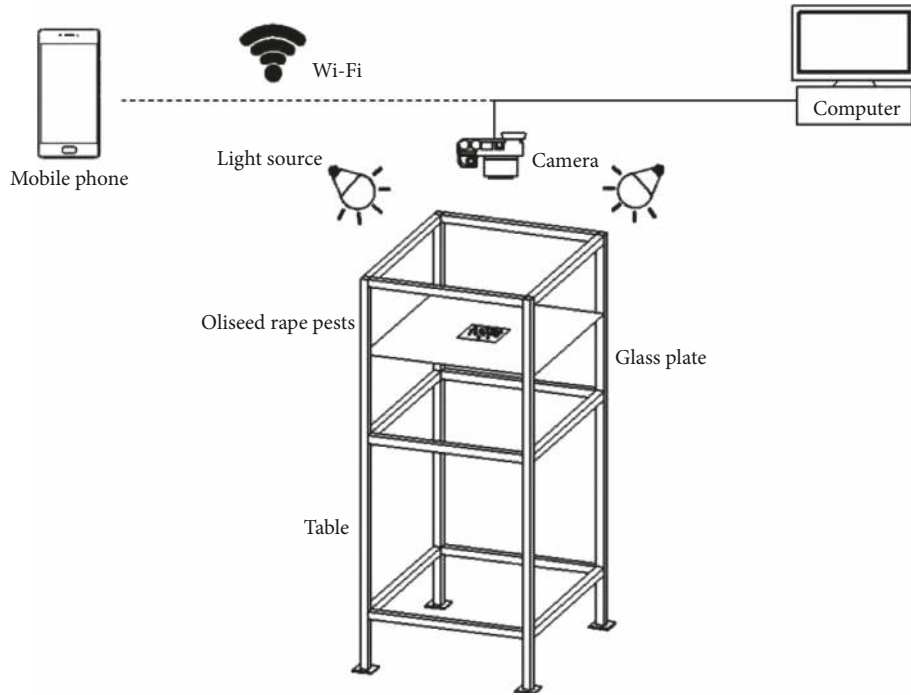


FIGURE 1: An oilseed rape pests imaging system.

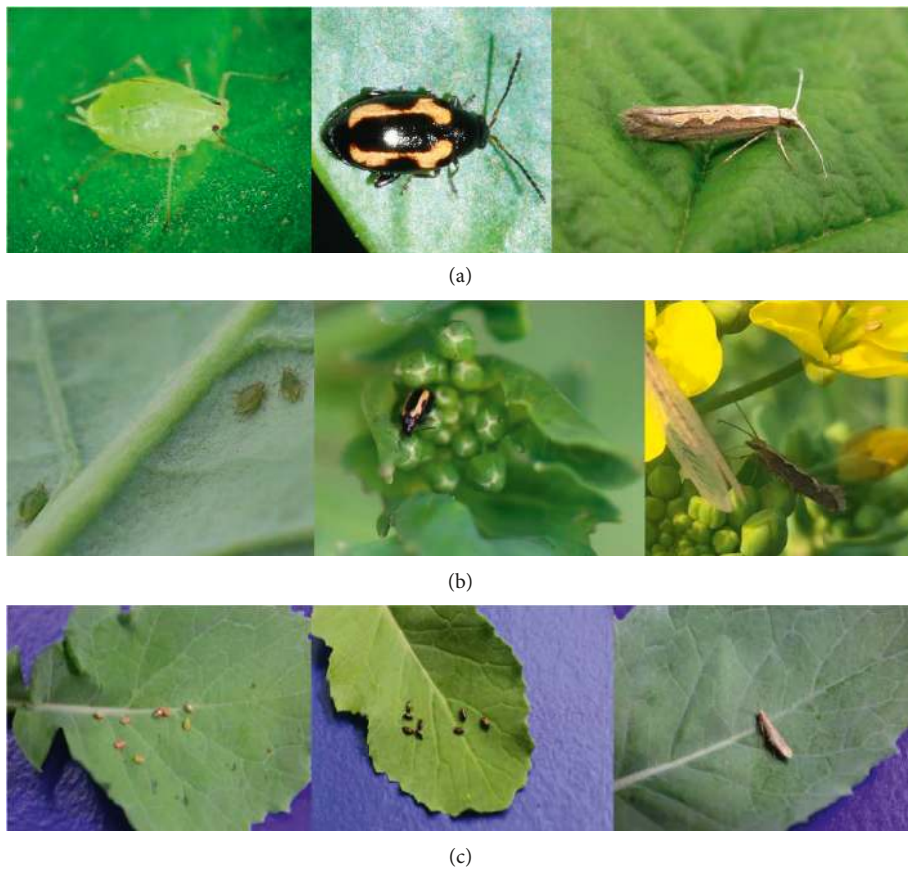


FIGURE 2: A representation of insect pests of oilseed rape. Images (a) downloaded from the Internet; (b) taken in the field; (c) taken in a laboratory.

TABLE 1: Details of oilseed rape pest dataset.

Species	Number of images	Number of objects
<i>Athalia rosae japonensis</i>	352	375
<i>Cretonotus transiens</i>	175	182
<i>Entomoscelis adonidis</i>	87	134
<i>Entomoscelis suturalis</i>	30	36
<i>Hellula undalis</i>	320	322
<i>Lipaphis erysimi</i>	150	1474
<i>Mamestra brassicae</i>	334	343
<i>Meligethes aeneus</i>	164	566
<i>Phyllotreta striolata</i>	220	272
<i>Pieris rapae</i>	610	690
<i>Plutella xylostella</i>	475	484
<i>Psylliodes punctifrons</i>	105	138
Total	3022	5016

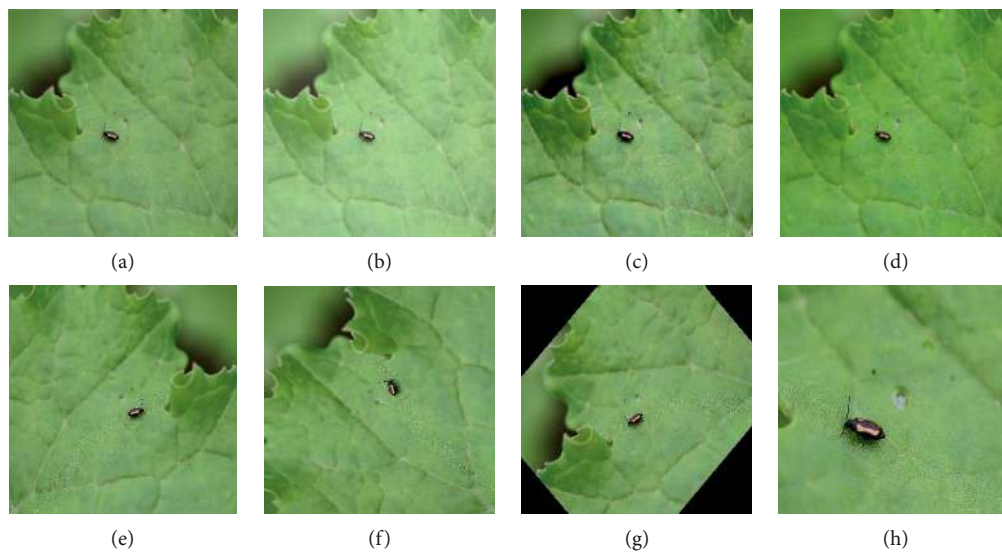


FIGURE 3: Examples of data augmentation. (a) Origin image. (b) Brightness adjusted. (c) Contrast adjusted. (d) Saturation adjusted. (e) Flipped horizontally. (f) Flipped diagonally. (g) Rotated 45 degrees. (h) Random cropping.

3.2.2. *Faster R-CNN*. Specifically, the innovation of Faster R-CNN lies in replacing the previous slow selective search algorithm with region proposal network (RPN), which runs much faster than before. Faster R-CNN methods include four steps: region proposal generation, feature extraction, classification, and location optimization. For inputting images of any size, the RPN uses CNN to directly generate a batch of region proposals and then fine-tunes candidate frames and classifications. Faster R-CNN output will predict every anchor being object or background. It resulted in a 10x increase in inference speed and better accuracy.

3.2.3. *R-FCN*. R-FCN is a region-based fully convolutional network which proposes position-sensitive score maps to avoid too much computation. This model increases the speed by sharing calculation on the entire image. And the convolution layer is added to generate the position-sensitive score maps. R-FCN can solve the contradiction between the image classification translation-invariance and the object detection translation-variance. Some studies have shown that

R-FCN can achieve comparable accuracy to Faster R-CNN with shorter running times [16, 40, 41], which means it achieves a good balance between speed and accuracy.

3.2.4. *SSD*. As R-FCN and SSD run faster than Faster R-CNN, but the working mode is significantly different from R-FCN. For the input image, the bounding boxes of the different aspect ratio and classification are simultaneously predicted at multiple positions of the image. The features are extracted in different convolutional layers which will decrease the size and support the multiple scales. Prediction is running at each location using the suitable kernel and produces accuracy of every box. It is worth noting that SSD works well in low-resolution image which means the model will achieve a better score because the pest shape is often in small size.

We chose SSD (single-shot detector) as the meta-architecture of optimal detection model because of the multiscale and the fastest high-accuracy feature. The specific implementation process of the network is shown in Figure 4.

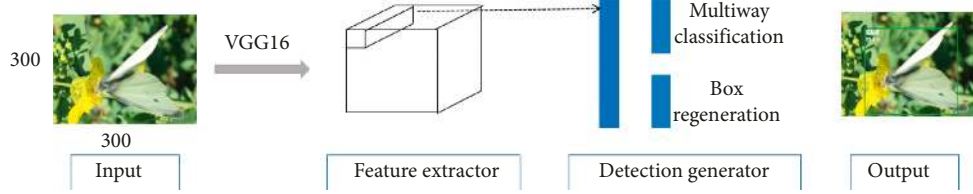


FIGURE 4: A diagram of the SSD-based detection network.

During training, we input the image first. The feature maps will be generated by the shared convolution layer of Inception. And, the location and label information of each object can be obtained finally.

3.3. Evaluation Indicators. A significant nonuniform distribution of dataset results in a simple accuracy-based metric cannot properly quantify the model performance. Thus, to evaluate the generalization ability of the model, the standard metric for this paper is based on PASCAL VOC2007 [19]. The notation used for evaluation is shown in Table 2 [42].

Precision and recall are commonly used in the literature to measure how well the detected objects correspond to the reference objects. Recall is defined as the proportion of items that are correctly detected among all the items that should have been detected. Precision is the proportion of all examples above that rank which are from the positive class. Precision is the proportion of detected items that are correct [43].

The precision and recall are computed as

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}. \end{aligned} \quad (1)$$

Average precision (AP) is a measure of the performance of a given class. It is a common performance indicator for object detection and is used to evaluate the performance of the dataset on this model. AP summarizes the shape of the precision/recall curve and defines its score as the mean precision of a set of equally spaced recalls values (0, 0.1, 0.2, ..., 1):

$$\text{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} \text{Pinterp}(r). \quad (2)$$

The $\text{Pinterp}(\bar{r})$ is defined as

$$\text{Pinterp}(r) = \max_{\bar{r}: \bar{r} \geq r} P(\bar{r}), \quad (3)$$

where $p(\bar{r})$ is the measured precision at recall (\bar{r}).

In multiclass detection, the mean of all AP values is mean average precision (mAP). To evaluate the performance of the dataset on each model, AP and mAP are selected as the metrics of the models.

Intersection over union (IoU) is an evaluation metric for position accuracy. It measures how good are the real object boundary on an image and the pest boundary generated by

TABLE 2: Confusion matrix.

	Actual positive	Actual negative
Predicted positive	True positives (TP)	False positives (FP)
Predicted negative	False negatives (FN)	True negatives (TN)

TP: examples correctly labeled as positives. FP: negative examples incorrectly labeled as positive. FN: positive examples incorrectly labeled as negative. TN: correspond to negatives correctly labeled as negative.

the algorithm. When IoU is 1, it means that the two are completely coincident. When IoU is 0, it means that the predicted bounding box and the true bounding box do not overlap at all. In this paper, IoU was 0.6. The formula for calculating IoU is as follows:

$$\begin{aligned} \text{IoU} &= \frac{\text{area}(\text{true bounding box} \cap \text{predicted bounding box})}{\text{area}(\text{true bounding box} \cup \text{predicted bounding box})} \\ &> 0.5. \end{aligned} \quad (4)$$

3.4. Experiment. The training and test were run on a computer equipped with an Intel Core i7 7800X, 32 GB of RAM and two NVIDIA GTX 1080Ti GPU for parallel computing (Table 3).

In our experiments, in order to solve the problem of insufficient dataset, we trained the detection model based on a pretrained model which has been trained in the COCO dataset. Faster R-CNN and R-FCN models were trained using stochastic gradient descent (SGD) with momentum of 0.9, initial learning rate of 0.0001, and batch size of 1. The initialization of the weight affects the convergence rate of the network. The normal distribution with the mean of 0 and the standard deviation of 0.01 is used to randomly initialize the weights of all layers of the network. The hyperparameters in SSD are the same as above except that the batch size was 24 and standard deviation of the normal distribution was 0.03. The initial learning rate was 0.001, and decaying every epoch was an exponential rate of 0.95.

4. Results

This section can be divided by subheadings. It provides a concise and precise description of the experimental results.

4.1. Object Detection Models. For a given set of data, the proportion of data division has a significant effect on the model performance. Firstly, data were divided into three subsets: training set, validation set, and test set in a ratio of 0.7:0.2:0.1. The data were trained on the training set and

TABLE 3: Experiment equipment.

Equipment	Specification
Memory	32 GB
Processor	Intel core i7 7800X CPU
Graphics	GeForce GTX 1080Ti
Operating system	Linux Ubuntu16.04

evaluated in a validation set to select model parameters, and finally, the test set was used to evaluate model performance. Considering the cost of training time and calculations, the experiment used a public model that has been pretrained on the COCO dataset. Transfer learning was employed in this study which used the same oilseed rape pest dataset to train different models. The meta-architectures (Faster R-CNN, R-FCN, and SSD) were combined with feature extractors (ResNet, Inception, and MobileNet) to find the optimal model by considering the mAP, running speed, memory usage, and loss. Table 4 and Figure 5 are the statistical results and training loss curves on different models, respectively, which can be used as the basis for model selection.

According to Table 4, the mAP of Faster R-CNN w/ ResNet101, SSD w/Inception, and R-FCN w/ResNet101 are all higher than 0.65, which have achieved good detection results. The running times of SSD are nearly three times faster than other architectures. Observing the training loss of each model, all curves converged rapidly and no longer fluctuated strongly after 80,000 iterations. The loss values are stable within a smaller range finally. As the model will be applied to portable and mobile devices, the memory footprint should not be too large. In summary, by analyzing the data, it is apparent that SSD w/Inception has better performance. We can develop mobile application based on this model to realize real-time and high-precision detection of oilseed rape pests.

4.2. Data Augmentation Processing. As mentioned before, the insect oilseed rape pest dataset in this experiment is small and may lead to overfitting of the model but collecting many images is an expensive process. Hence, the DA technology has been used to enlarge the training data. In order to simulate various shooting conditions in the field as closely as possible, this paper finally chose the DA methods as given below:

Color jittering: we adjusted the brightness, contrast, and saturation of the images randomly to detect objects in different lighting conditions and cameras

Random flipping/translation: it could prevent the objects from appearing only in certain position of the image

Random rotate: it could make the model learn the objects across various viewpoints

Random cropping: it could reduce the influence of the background and enhance the stability of the model

During training, one or more combined DA methods were used to augment the dataset randomly, which are shown in Figures 6(a) and 6(b).

As shown in Figure 6(a), random cropping provides the greatest improvement, followed by random rotation and random flipping. Figure 6 shows that the top three methods all used cropping and flipping. These two methods are suitable for improving the performance of the dataset in this paper. Then, we compared the impact with DA (flip + crop) and without DA on each pest, and the results are presented as Table 5.

The experiments use SSD with Inception. After DA, we have a total of approximately 10575 images for training. Table 5 shows the effect of DA on the models. It could be learned from Table 5 that the mAP value has been improved after DA. Among them, the AP values of *Athalia rosae japonensis*, *Mamestra brassicae*, *Pieris rapae*, and *Plutella xylostella* decreased, and the AP value of *Pieris rapae* decreased by 0.1383, while the AP values of other pest samples increased significantly. By analyzing the dataset, *Athalia rosae japonensis*, *Mamestra brassicae*, *Pieris rapae*, and *Plutella xylostella* have the largest amount images in the dataset, but there might also be overfitting. After increasing the images, the results are closer to the real state, resulting in a decrease in the AP value of this part of the pests. The remaining small sample is insufficient. After DA, more features are learned, and the AP values are increased greatly, thereby improving the mAP value of the model. Therefore, we observed an appropriate increase in the number of images is conducive to improve the model performance.

4.3. Impact of Dropout on Performance. In this work, there are fewer training data, and the model may have overfitting. Dropout is an effective way to prevent overfitting in deep neural network. It combines exponentially different neural network architectures by randomly dropping out neurons in the network. Each time the data are applied to different neural networks, the final comprehensive averaging strategy can avoid the overfitting problem. The neurons were dropped out (aka removed) by the layer according to the probability q ($q = 1 - p$) during training, and the effect of change of dropout rates on pest dataset is shown in Figure 7.

Srivastava et al. [41] gave the typical values of p ; generally p takes 0.5 to 0.8 for hidden layer. The choice of p is combined with the number of hidden units n , and the smaller p requires a larger n . Figure 7 shows that the model obtains the best mAP when p is 0.8. This is because when p is between 0.5 and 0.7, the excessive feature information is hidden and the generated model has weakened ability to fit the data. The value of mAP increases as p increases, reaching a peak at $p = 0.8$. When the p further increased, it can be found that the detection performance of the model begins to decrease, because the dataset is too small to limit the size of the model parameter [44].

5. Design and Implementation of Oilseed Rape Pest Detection System

In the previous works, a pest detection model is trained and works well. So, we transplant this oilseed rape pest detection model to Android mobile platform. The system utilizes the

TABLE 4: The results for five different architectures.

Meta-architecture Feature extractor	Faster R-CNN		R-FCN		SSD	
	ResNet101	Inception	ResNet101	Inception	Inception	MobileNet
<i>Athalia rosae japonensis</i>	0.8321	0.8393	0.7703	0.8218	0.8218	0.8302
<i>Cretonotus transiens</i>	0.5743	0.5527	0.6083	0.586	0.586	0.5951
<i>Entomoscelis adonidis</i>	0.8765	0.8246	0.9161	0.6176	0.6176	0.6030
<i>Entomoscelis suturalis</i>	0.6562	0.4162	0.4344	0.7555	0.7555	0.4980
<i>Hellula undalis</i>	0.7563	0.7029	0.7106	0.7738	0.7738	0.6247
<i>Lipaphis erysimi</i>	0.2149	0.3003	0.3606	0.3772	0.3772	0.3236
<i>Mamestra brassicae</i>	0.9288	0.8445	0.9280	0.967	0.967	0.8431
<i>Meligethes aeneus</i>	0.5375	0.2637	0.3556	0.2247	0.2247	0.3476
<i>Phyllotreta striolata</i>	0.7604	0.5197	0.6124	0.6275	0.6275	0.6760
<i>Pieris rapae</i>	0.8143	0.8040	0.8432	0.9111	0.9111	0.6609
<i>Plutella xylostella</i>	0.8629	0.7890	0.7880	0.8767	0.8767	0.7846
<i>Psylliodes punctifrons</i>	0.4537	0.4202	0.5368	0.554	0.554	0.7593
mAP@0.6	0.6890	0.6064	0.6554	0.6744	0.6744	0.6288
Time (s)	0.158	0.13	0.148	0.052	0.052	0.045
Memory (MB)	191.3	52.9	201.7	60.6	60.6	23.6

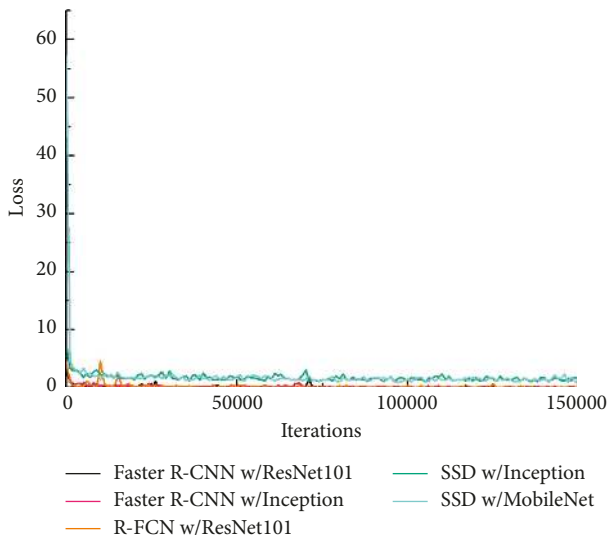


FIGURE 5: Plot of model loss on dataset.

calculating ability of popular mobile device and provides real-time detection in agricultural production. Compared with most of the previous work, the advantages of this system are follows:

- (i) Capture and detection are processed in the same device at the same time and it shows the detection results instantly without waiting
- (ii) The system can process images captured by various camera devices in different resolutions
- (iii) The system has high adaptability to different lighting conditions and complex background (laboratory and field environment); the model works well with different pest poses.

The system is easy to use and robust to detect.

5.1. Software Environment. The system development environment is based on the MacOS Mojave version 10.14.1

operating system and is configured by Android Studio 3.2 JDK 1.8.

5.2. Overall Structural Design. The overall structure of the oilseed rape pest detection system is shown in Figure 8.

The system adopts modular design idea, which is divided into image acquisition module, image preprocessing module, image detection module, and result display module. The image acquisition module captures an image through a mobile device camera, or mobile album; the image preprocessing module cuts the input image into a fixed size, which can reduce the interference of the background to the result, so that the model can detect the object more accurately; the image detection module can detect 12 common oilseed rape pests; the results display module displays the species and location of pests in the image. The users can view more test results and detailed information of the corresponding results, learn the skills of artificially identifying of pest species, and understand pest control programs. The detailed functions of the application are presented in Figures 9(a)–9(c).

This application is called Dr. Pest. When the user launches the application, the startup interface appears, including the name of the app, the icon, and the photo button. Click the Identify Pest button; user can directly take or select a photo from the mobile album, determine the image to be detected, and wait a few seconds before entering the detection result display interface. In the image, different color boxes are used to represent different pest categories. At the bottom of the page, the example images, species, recognition accuracy, and related expressions of pests are displayed. If you want to know the detailed introduction of the pest and control measures or access the relevant web page, click on the corresponding pest to enter the next interface.

5.3. Software Testing. The program is packaged into the Android installation package file, which is installed into the HUAWEI Honor V10 mobile phone with the Android

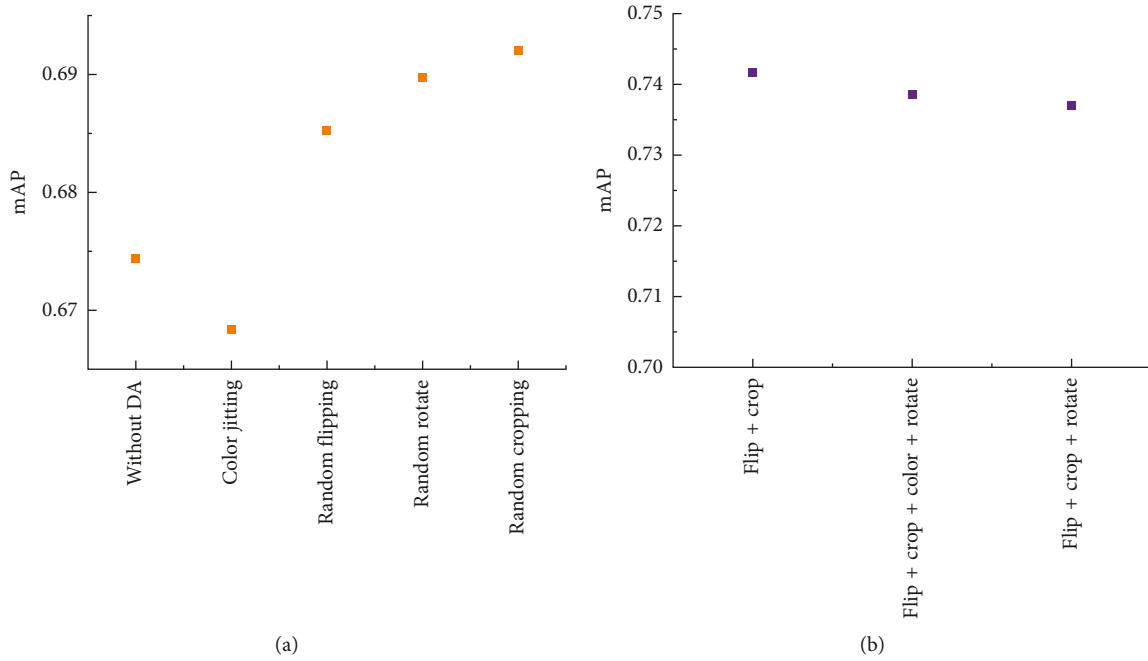


FIGURE 6: Experimental result for the SSD w/Inception with data augmentation. (a) Each experiment uses only one data augmentation method. (b) Test of all the combinations of DA methods, showing the top 3 methods.

TABLE 5: Validation set results for the model with and without data augmentation.

Method	<i>Athalia rosae japonensis</i>	<i>Cretonotus transiens</i>	<i>Entomoscelis adonidis</i>	<i>Entomoscelis suturalis</i>	
Without data augmentation	0.8218	0.586	0.6176	0.7555	
Data augmentation	0.7858	0.6106	0.8593	0.9167	
Method	<i>Hellula undalis</i>	<i>Lipaphis erysimi</i>	<i>Mamestra brassicae</i>	<i>Meligethes aeneus</i>	
Without data augmentation	0.7738	0.3772	0.967	0.2247	
Data augmentation	0.842	0.5914	0.9577	0.301	
Method	<i>Phyllotreta striolata</i>	<i>Pieris rapae</i>	<i>Plutella xylostella</i>	<i>Psylliodes punctifrons</i>	mAP
Without data augmentation	0.6275	0.9111	0.8767	0.554	0.6744
Data augmentation	0.7759	0.7728	0.8062	0.6808	0.7417

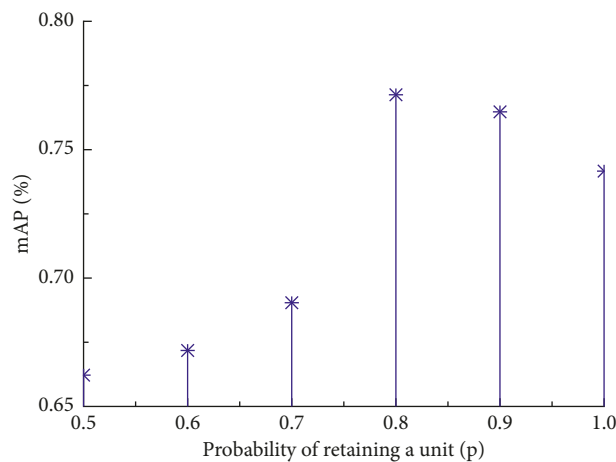


FIGURE 7: Effect of change dropout rates on mAP. We set $p = 0.5, 0.6, 0.7, 0.8,$ and $0.9,$ and the mAP changes accordingly. When $p = 1.0,$ it means the model without dropout.

version 9.0. The mobile terminal configuration is shown in Table 6.

We tested the accuracy and detection time of the software. According to the observation, it takes about 1 second

to detect an image on the mobile phone. The experiment prepared a total of 303 pest test images, 540 objects, including 12 categories in Table 1, covering as many different lighting conditions and poses as possible. Test mainly

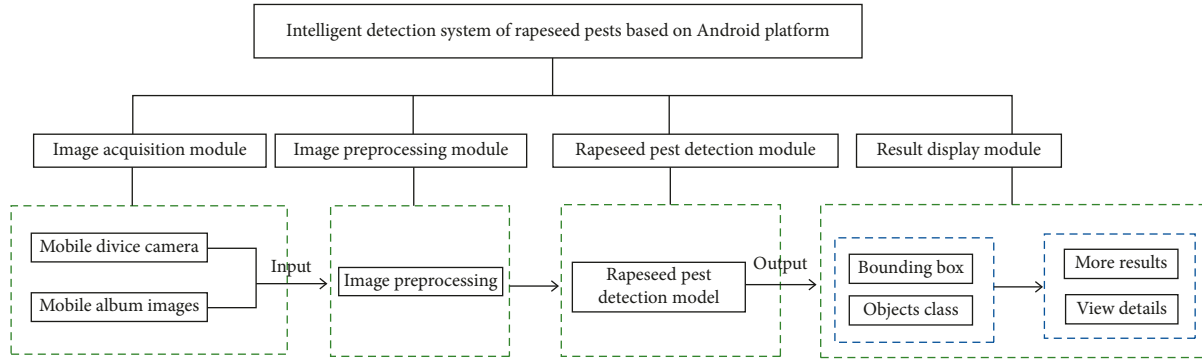


FIGURE 8: Overall structure of the oilseed rape pest detection system.

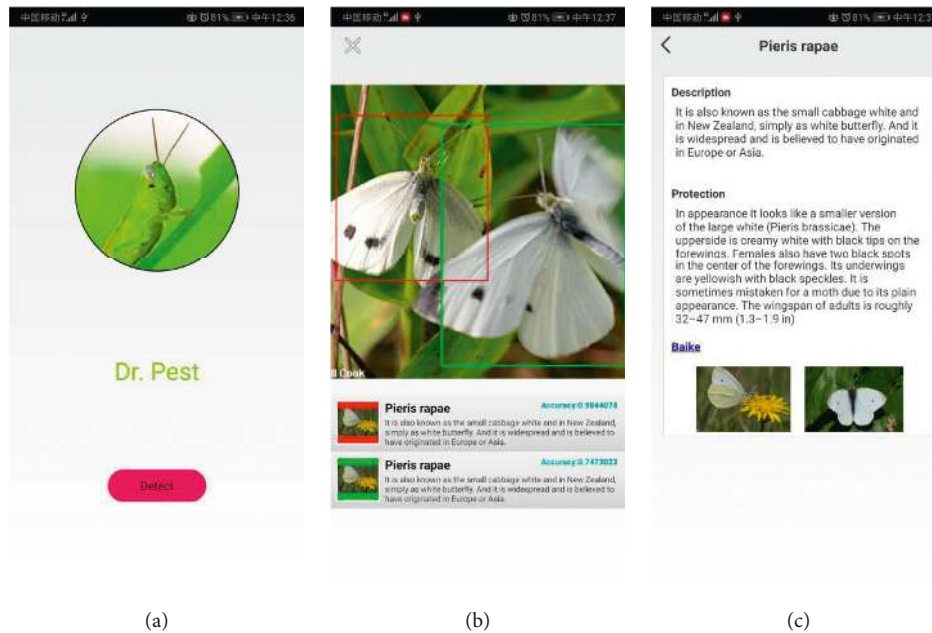


FIGURE 9: Android oilseed rape pest detection application interface. (a) Application startup interface. (b) Detection result display interface. (c) Detailed view interface.

calculates the accuracy, missing rate, and false reject rate of the objects, as shown in Table 7.

5.4. Discussion. Figures 10(a)–10(f) show examples of test images that are correctly detected by the model.

Figures 10(a) and 10(b) are taken from the side of the pests, while the rest are mostly frontal, reflecting the change of perspectives. Figures 10(b) and 10(f) detect ambiguous objects correctly, which indicate that when the pests in the image are blurred, the model can also accurately detect the object. Figure 10(c) shows the postures of pest is diverse, and the color of each sample in Figure 10(d) is different, which indicates that the model effectively handles the shape and color difference within the class. Feature extraction and occlusion processing are two key elements in multiobject detection [45]. The two pests overlap each other in Figure 10(c), and half of the

TABLE 6: The configuration of the mobile terminal.

Items	Parameter values
CPU frequency	HiSilicon Kirin 970
ROM	64 GB
Size of screen	2160 * 1080
Number of CPU cores	2
RAM	4 GB
Pixels of camera	5120 * 3840

The experiments use SSD with Inception.

TABLE 7: The test results of the mobile application.

Method	Accuracy	Missing rate	False reject rate
Oilseed rape pest detection system	0.6982	0.2074	0.0944

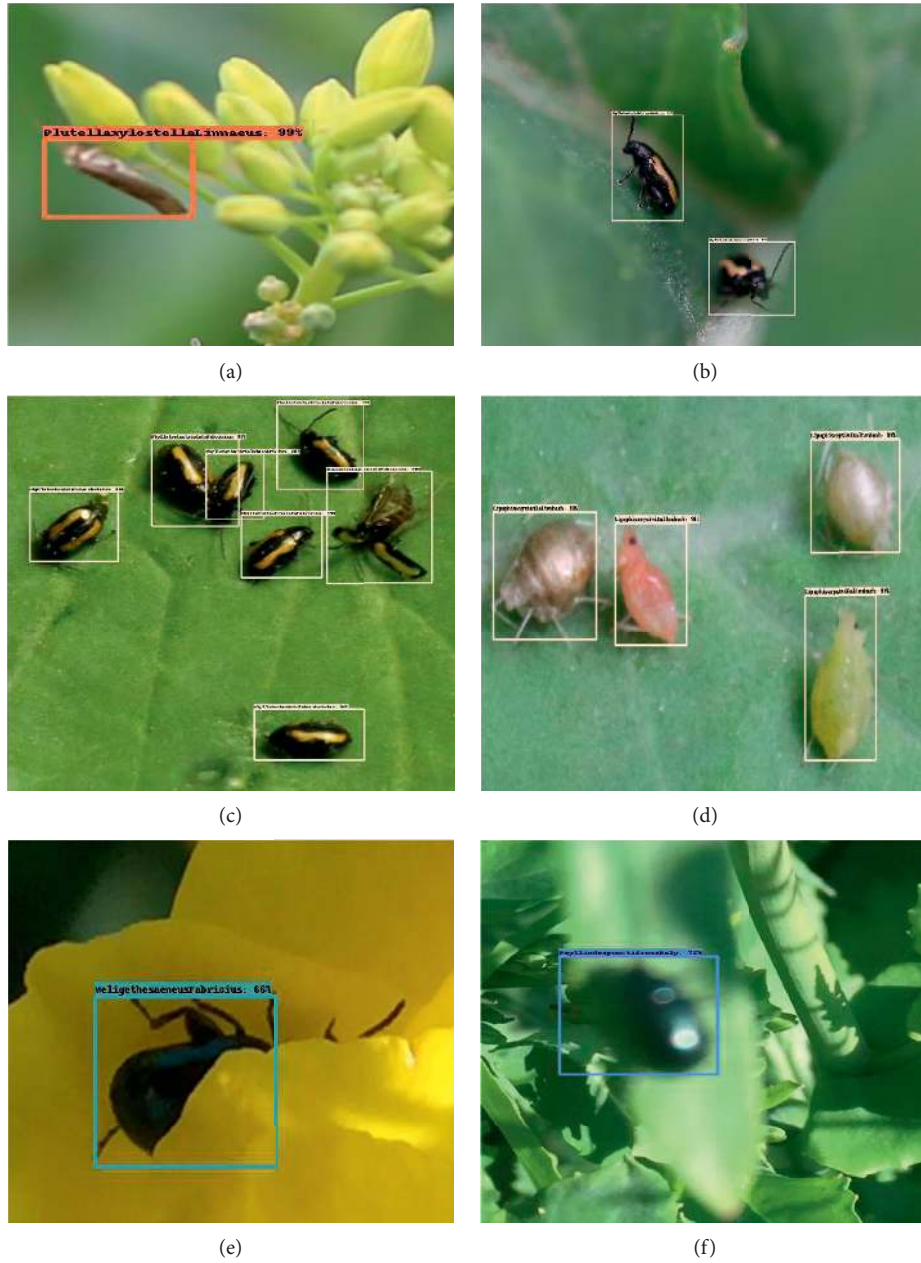


FIGURE 10: Examples of test images for correct detection. The eight images reflect the model that can achieve good performance in some cases. (a) *Plutella xylostella*. (b) *Phyllotreta striolata*. (c) *Phyllotreta striolata*. (d) *Lipaphis erysimi*. (e) *Meligethes aeneus*. (f) *Psylliodes punctifrons*.

features of the pests are occluded in Figure 10(e); neither of them captures the complete pests, which shows that the model has a certain generalization ability for missing features. These results confirmed the feasibility of the detection system. Especially when the image contains only one class of objects, the model can accurately detect each object in different backgrounds.

There still needs improvements in the detection system, and the typical examples that were wrongly detected are shown as Figures 11(a)–11(d). It is concluded the detection ability of the model is insufficient in the following situations,

which can be used as the next research focus in the future. In Figure 11(a), two pests overlap and the system only regards them as one. This indicates that when the overlapping area of pests adjacent is large, the results of multiobject detection are not good and it is impossible to distinguish whether the overlapping objects are independent individuals or not. The system shows that there are two pests in Figure 11(b), one of which is detected accurately and the other is detected flower buds as *Lipaphis erysimi*. This is because the shape and color of the oilseed rape flower buds are similar to those of the *Lipaphis erysimi*. The flower buds are repeatedly mistaken

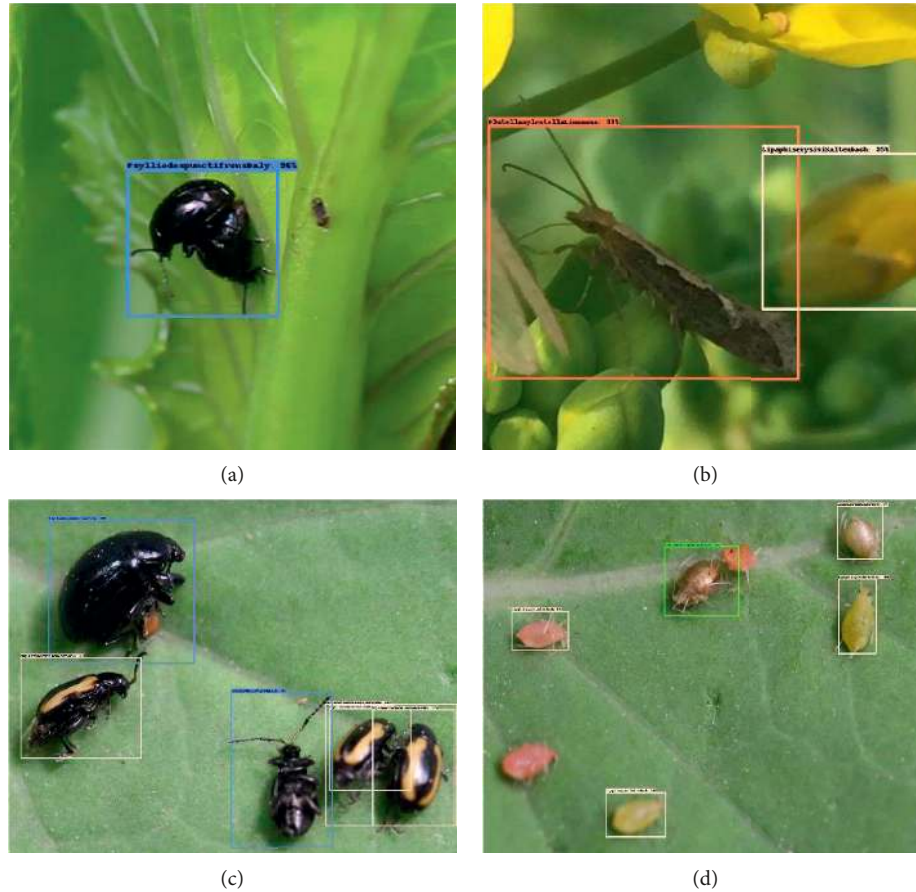


FIGURE 11: Examples of test images for failure detection. The model cannot perform well when detecting these types of objects, reflecting the shortcomings of the model. (a) *Psylliodes punctifrons*. (b) *Plutella xylostella*. (c) *Psylliodes punctifrons* and *Plutella xylostella*. (d) *Lipaphis erysimi*.

for *Lipaphis erysimi* in the detection, and the number of pests will greatly disturbed with the result. *Plutella xylostella* has three postures in Figure 11(c); the object can be detected correctly when the side and back facing up, but the abdomen face up is mistaken for *Psylliodes punctifrons*. Because the data collected by the experiment are mostly taken from the side and back when a new pest pose appears, it is impossible to accurately identify the object. Seven *Lipaphis erysimi* pests were found in Figure 11(d), but only five were detected. The dataset of this experiment is mostly from the Internet. The size of pests in the Internet images is large, and the dataset collected from the field and lab contain only a few small objects, so the detection model has poor ability to detect small objects.

6. Conclusion

This paper created an oilseed rape pest dataset that covers the 12 typical oilseed rape pests and filled in the blank of the open dataset of oilseed rape pests. The application of common object detection algorithms in oilseed rape pest detection is compared and analyzed. Among them, SSD w/ Inception balances the performance indicators such as detection accuracy, time, and memory usage and can be applied to mobile devices. On the basis of the model, the DA

method is selected to solve the uneven class balance problem in the experimental dataset. The experimental results showed that the method can improve the detection accuracy by 0.7 percentage points. In order to avoid data overfitting, the dropout layer is added. When $p = 0.8$, the mAP value of the model was as high as 77.14%. It was tested on images collected under different illuminations and backgrounds. The model has good ability to detect oilseed rape pests under complex noise. Eventually, the trained model was applied to the Android platform to develop an oilseed rape pest detection system. It is also the first application of oilseed rape pest detection based on deep learning in China. Users can accomplish detecting the pests of oilseed rape in the field real-time by operating the intuitive and simple interface based on an Android mobile phone, which effectively improves the efficiency of agricultural production.

The detection capability of the model is still insufficient, encountering multiobject overlap, feature occlusion, and small object detection tasks, which will be the next research direction. In the future work, we will try to solve the above problems by expanding the oilseed rape pest dataset and adjusting the model network structure. In addition, the oilseed rape pest detection system has only completed preliminary development work, and more functional improvement works are needed in the next step.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

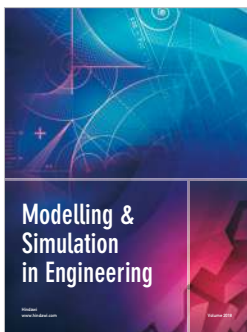
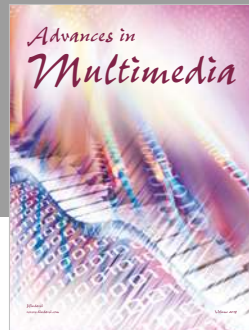
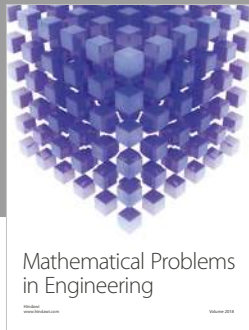
Acknowledgments

This work was funded by the Major Science and Technology Projects in Zhejiang with grant no. 2015C02007.

References

- [1] Y. Yin, H. Z. Wang, and X. Liao, "Analysis and strategy for 2009 rapeseed industry development in China," *Chinese Journal of Oil Crop Sciences*, vol. 31, pp. 259–262, 2009.
- [2] D. J. Ecobichon, "Pesticide use in developing countries," *Toxicology*, vol. 160, no. 1–3, pp. 27–33, 2001.
- [3] D. Pimentel, H. Acquay, M. Biltonen et al., "Environmental and economic costs of pesticide use," *Bioscience*, vol. 42, no. 10, pp. 750–760, 1992.
- [4] A. King, "Technology: the future of agriculture," *Nature*, vol. 544, no. 7651, pp. S21–S23, 2017.
- [5] A. Fuentes, D. H. Im, S. Yoon, and D. S. Park, "Spectral analysis of CNN for tomato disease identification," in *Proceedings of the International Conference on Artificial Intelligence and Soft Computing*, pp. 40–51, Cham, Switzerland, June 2017.
- [6] Wikipedia, Computer vision, 2019, https://en.wikipedia.org/w/index.php?title=Computer_vision&oldid=874749249.
- [7] S. Hwang and H.-E. Kim, "Self-transfer learning for fully weakly supervised object localization," 2016, <https://arxiv.org/abs/1602.01625>.
- [8] S. Gossain and J. S. Gill, "A novel approach to enhance object detection using integrated detection algorithms," *International Journal of Computer Science and Mobile Computing*, vol. 3, pp. 1018–1023, 2014.
- [9] S. Ding and K. Zhao, "Research on daily objects detection based on deep neural network," in *Proceedings of the International Symposium on Application of Materials Science and Energy Materials*, pp. 28–29, Shanghai, China, December 2017.
- [10] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. D. Wu, "Object detection with deep learning: a review," 2018, <https://arxiv.org/abs/1807.05511>.
- [11] A. Vouloimos, N. Doulamis, A. Doulamis et al., "Deep learning for computer vision: a brief review," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 7068349, 13 pages, 2018.
- [12] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [13] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.
- [14] J. Ubbens, M. Cieslak, P. Prusinkiewicz, and I. Stavness, "The use of plant models in deep learning: an application to leaf counting in rosette plants," *Plant Methods*, vol. 14, no. 1, p. 6, 2018.
- [15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
- [16] J. Dai, Y. Li, K. He, and J. Sun, "Object detection via region-based fully convolutional networks," 2016, <https://arxiv.org/abs/1605.06409>.
- [17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Seattle, WA, USA, June 2016.
- [18] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Amsterdam, The Netherlands, October 2016.
- [19] I. Bechar and S. Moisan, "On-line counting of pests in a greenhouse using computer vision," in *Proceedings of the VAIB 2010-Visual Observation and Analysis of Animal and Insect Behavior*, Istanbul, Turkey, August 2010.
- [20] Y. Zhong, J. Gao, Q. Lei, and Y. Zhou, "A vision-based counting and recognition system for flying insects in intelligent agriculture," *Sensors*, vol. 18, no. 5, p. 1489, 2018.
- [21] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *Proceedings of the European Conference on Computer Vision*, pp. 740–755, Zurich, Switzerland, September 2014.
- [22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [23] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [24] X. Chen, Z. Wu, and J. Yu, "TSSD: temporal single-shot detector based on attention and LSTM for robotic intelligent perception," 2018, <https://www.arxiv-vanity.com/papers/1803.00197/>.
- [25] W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management," *Computers and Electronics in Agriculture*, vol. 123, pp. 17–28, 2016.
- [26] A. Shrivastava and A. Gupta, "Contextual priming and feedback for faster R-CNN," in *Proceedings of the European Conference on Computer Vision*, pp. 330–348, Amsterdam, The Netherlands, October 2016.
- [27] Y. C. Zhou, T. Y. Xu, W. Zheng, and H. B. Deng, "Classification and recognition approaches of tomato main organs based on DCNN," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 33, pp. 219–226, 2017.
- [28] L. S. Fu, Y. L. Feng, E. Tola, Z. H. Liu, R. Li, and Y. J. Cui, "Image recognition method of multi-cluster kiwifruit in field based on convolutional neural networks," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 34, pp. 205–211, 2018.
- [29] X. Wei, F. Liu, Z. Qiu, Y. Shao, and Y. He, "Ripeness classification of astringent persimmon using hyperspectral imaging technique," *Food and Bioprocess Technology*, vol. 7, no. 5, pp. 1371–1380, 2014.
- [30] F. Liu and Y. He, "Classification of brands of instant noodles using Vis/NIR spectroscopy and chemometrics," *Food Research International*, vol. 41, no. 5, pp. 562–567, 2008.
- [31] H. Wang, J. Peng, C. Xie, Y. Bao, and Y. He, "Fruit quality evaluation using spectroscopy technology: a review," *Sensors*, vol. 15, no. 5, pp. 11889–11927, 2015.

- [32] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deep fruits: a fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, 2016.
- [33] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, and G. Venturini, "A survey on image-based insect classification," *Pattern Recognition*, vol. 65, pp. 273–284, 2017.
- [34] Q. Yao, J. Lv, Q.-J. Liu et al., "An insect imaging system to automate rice light-trap pest identification," *Journal of Integrative Agriculture*, vol. 11, no. 6, pp. 978–985, 2012.
- [35] Y. Wang and X. Zhang, "Autonomous garbage detection for intelligent urban management," in *Proceedings of the International Conference on Electronic Information Technology and Computer Engineering*, pp. 12–14, Shanghai, China, October 2018.
- [36] I. Rebai, Y. BenAyed, W. Mahdi, and J. P. Lorre, "Improving speech recognition using data augmentation and acoustic model fusion," in *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 316–322, Marseille, France, September 2017.
- [37] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.
- [38] X. Y. Zhu, Y. F. Liu, J. H. Li, T. Wan, and Z. C. Qin, "Emotion classification with data augmentation using generative adversarial networks," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 349–360, Melbourne, Australia, May 2018.
- [39] L. Taylor and G. Nitschke, "Improving deep learning using generic data augmentation," 2017, <https://arxiv.org/abs/1708.06020>.
- [40] J. Huang, V. Rathod, C. Sun et al., "Speed/accuracy trade-offs for modern convolutional object detectors," 2016, <https://www.arxiv-vanity.com/papers/1611.10012/>.
- [41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [42] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240, Pittsburgh, PA, USA, June 2006.
- [43] A. Godil, R. Bostelman, W. Shackelford, T. Hong, and M. Shneier, "Performance metrics for evaluating object and human detection and tracking systems," NISTIR 7972, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2014.
- [44] Z. Y. Liu, J. F. Gao, G. G. Yang, H. Zhang, and Y. He, "Localization and classification of paddy field pests using a saliency map and deep convolutional neural network," *Scientific Reports*, vol. 6, no. 1, article 20410, 2016.
- [45] J. Li, H.-C. Wong, S.-L. Lo, and Y. Xin, "Multiple object detection by a deformable part-based model and an R-CNN," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 288–292, 2018.



Hindawi

Submit your manuscripts at
www.hindawi.com

