



## A deep learning based system for real-time detection and sorting of earthworm cocoons

Ali ÇELİK<sup>1</sup> , Sinan UĞUZ<sup>2,\*</sup> 

<sup>1</sup>Department of Mechatronics Engineering, Isparta University of Applied Sciences, Isparta, Turkey

<sup>2</sup>Department of Computer Engineering, Isparta University of Applied Sciences, Isparta, Turkey

Received: 08.03.2022

Accepted/Published Online: 02.06.2022

Final Version: 22.07.2022

**Abstract:** Vermicompost, created by earthworms after eating and digesting organic waste, plays an important role as an organic fertiliser in sustainable agriculture. In this study, a deep learning-based smart system was developed to separate earthworm cocoons used in the production of vermicompost from the compost and return it to production. In the first stage of the study, a dataset containing 1000 images of cocoons was created. The cocoons in each image were labeled and training was performed using a deep learning architecture, one-stage and two-stage models. The models were trained over 2000 epochs with a learning rate of 0.01. From the experimental results, faster R-CNN with ResNet50-FPN model detected the earthworm cocoons better compared to other models. The best performance was obtained by this model with an average precision (AP) of 0.89. In the other stage of the study, the cocoons detected by the software were separated from the compost using a specially designed conveyor belt system. In this process, the detected cocoons are separated from the compost using 10 pneumatic valves that spray air at the separation point. The study is the first of its kind that enables earthworm cocoons to be returned to production with the use of a real-time intelligent system. It also contributes to the literature on small object detection using deep learning.

**Key words:** Object detection, vermicompost, earthworm cocoon, agriculture

### 1. Introduction

In recent years, interest in small object detection has increased thanks to the success of deep learning models. Small object detection problems are more difficult than other object detection problems in terms of distinguishing from background and other classes. Another difficulty is that there are more possibilities for the localization of small objects [1].

Small object detection applications in agriculture differ for living and nonliving objects. In studies on living objects, it is seen that the deep learning approach is widely used on the visual detection and classification of agricultural pests. In a study, Yang [2] developed a convolutional neural networks (CNN) model based insect recognition system. They achieved an accuracy of 91.5% for 7706 images obtained from the tea garden. In another study [3], a dataset called Pest24 with 25378 labeled pest images was created. The dataset with 24 different pest categories was trained using faster R-CNN, single shot multibox detector (SSD), YOLOv3 and Cascade R-CNN architectures. In the study, the highest mean average precision (mAP) value was obtained with the YOLOv3 model at 63.5%. In another study on pest detection, Karar et al. [4] developed a prediction model for five different pest groups using SSD, backpropagation neural networks, faster R-CNN architectures. The

\*Correspondence: [sinanuguz@isparta.edu.tr](mailto:sinanuguz@isparta.edu.tr)

result showed that the faster R-CNN was the most successful model with 55%. Moreover, Jiao et al. [5] achieved an mAP score of 56.4% using the proposed model (AF-RCNN) based on R-CNN architecture for classification and recognition of 24 different pests.

There are also studies for buds on trees in agricultural applications for small object detection. Xia et al. [6] obtained a 56.4% mAP score with the proposed the new CNN model based ResNet architecture for apple bud categorization. In another study, Xu et. Al. [7] achieved an accuracy of 95.7% with the proposed two-level fusion network approach, using a dataset with tea buds from different angles. The study carried out by Kokehi et al. [8] on the detection of Pacific oyster larval differs from previous studies in terms of subject. The dataset created by labeling 1515 larval images was trained using the ResNeXt architecture. The study presented a model for larval shell heights using the least squares method.

In recent years, it is an important requirement to have fertilizers containing macro- and micronutrients in the soil for the realization of sustainable agriculture, which aims to develop new agricultural practices that are safe and harmless to the environment [9, 10]. Chemical fertilizers, widely used in modern agriculture, have contributed significantly to increasing yields and controlling agricultural diseases and pests. However, in addition to these benefits, their use has also led to significant problems in terms of food security, human and environmental health, protection of the ecological balance, and soil biodiversity [11]. Therefore, the demand for sustainable agriculture has increased day by day [12].

Vermicompost, an important application of sustainable agriculture, is a natural fertilizer rich in nutrients and ionic substances, with a high water-holding capacity. It is produced after the worms have eaten and digested organic waste [13, 14]. If there are not enough worms in vermicompost, fertilizer production will not reach the desired level. Vermicompost producers want a minimum number of cocoons in the solid fertilizer they sell. This is because it is important to use the cocoons for the production of new worms and not for the fertilizer that is applied to the field. Separating the cocoons from the vermicompost is very difficult and costly because of their small size. However, it is important that these cocoons remain in the production of new vermicompost and that they are returned to worm production. Solving this problem will provide an advantage to earthworm production facilities in terms of labor, cost and time.

The aim of this study is to develop a real-time system to separate earthworm cocoons used in vermicompost production to be separated from compost and return them to production using deep learning techniques.

The main contributions of this paper are as follows: (1) This is the first study to describe the return of earthworm cocoons to production through the use of an intelligent real-time system. (2) When examining the literature on the separation of agricultural products using computer vision techniques, it can be said that the identified studies mostly refer to objects larger than the cocoons used in our study. For this reason, the contribution of such studies to small object detection by deep learning is limited. (3) By using different state-of-the-art object detection models on a conveyor belt system cocoons were detected and the best AP value of 89% in prediction success was obtained.

The workflow of the paper is as follows. Section 2 presents the materials and methods for earthworm cocoons detection using one-stage and two-stage CNN models. In this section, the preparation of the dataset, the training models and the developed hardware system was examined. The experimental results are discussed in Section 3. Finally, conclusions are drawn in Section 4.

## 2. Materials and methods

The processes of the Cocoon Detection and Sorting System (CoDeSS) which enable earthworm cocoons to be returned to production using a real-time intelligent system are as shown in Figure 1. As can be seen in the first stage of the process, 1000 images of compost were obtained, and the cocoons in each image were labeled as cocoon class. After obtaining the dataset, the different CNN models were used for training in terms of cocoon identification and classification. In this study, cocoon images were trained using one-stage and two-stage object detection CNN models. Average precision (AP) and average recall (AR) metrics were used to evaluate model performances. At the end of the training process, the model with the highest performance was selected for the sorting process. Nvidia Jetson TX2 embedded AI computing device is preferred to run the selected model.

The software was developed for the selected model, and the cocoons detected by this software could be monitored in real-time. One of the tasks of the GUI software was to calculate the time required for the detected cocoons to reach the separation point from the region where they were detected and to transmit this information to an Nvidia Jetson TX2. In the final stage of the study, a real-time system was developed on the conveyor belt to separate the detected cocoons from the compost. The light box on the conveyor contains an industrial camera, lens and lighting unit. The Nvidia Jetson activates the pneumatic valves for 200 ms at the separation point to separate the detected cocoons from the compost. These steps of the general scheme are explained in detail in the following subsections.

### 2.1. Image dataset

The dataset created in this study consists of compost prepared under controlled conditions and earthworm cocoon images. To the best of our knowledge, no cocoon dataset was found during the literature search. The images for the dataset were obtained from an earthworm production facility in Turkey. These images were taken in high resolution ( $4000 \times 3000$  pixels) using a Panasonic DMC-FZ100 digital camera and converted to RGB format and  $800 \times 600$  pixels. The dataset included a total of 1000 compost images. Each image contained an average of four cocoons and a total of 3809 cocoons.

The coordinates of the cocoons in the images were individually labeled using LabelImage software<sup>1</sup> and after all the images were labeled, they were divided into 80% training and 20% testing sets.

In this work we used several methods of data augmentation. The original training images were augmented by  $\pm 15^\circ$  rotation,  $0^\circ$ – $20^\circ$  cropping,  $\pm 15^\circ$  horizontal and vertical shear, 3 boxes of 10% size cutout for each image. Figure 2 shows randomly selected original and augmented images.

### 2.2. Implementation details of CNN models

The experiments on two-stage models in this study were conducted using the Detectron2<sup>2</sup>, a state-of-the-art PyTorch-based modular object detection library released by Facebook AI Research. Detectron2 Model Zoo offers a variety of pretrained base models. Three proficient backbone architectures, i.e. ResNet50-FPN, ResNet101-FPN, and ResNeXt101-FPN, were selected and used with faster R-CNN. Another base model used is RPN and fast R-CNN. This model was used with the RPN-R50-C4 and RPN-R50-FPN backbone architectures.

Three one-stage models YOLOv5s, YOLOv5m and SSD, were used in the study. While CSPDarknet is used as the backbone for the YOLO models, VGG16 is used as the backbone for SSD.

<sup>1</sup>LabelImg (2017), Website <https://github.com/tzutalin/labelImg> [accessed 07 March 2022]

<sup>2</sup>Detectron2 (2019), Website <https://github.com/facebookresearch/detectron2> [accessed 07 March 2022]

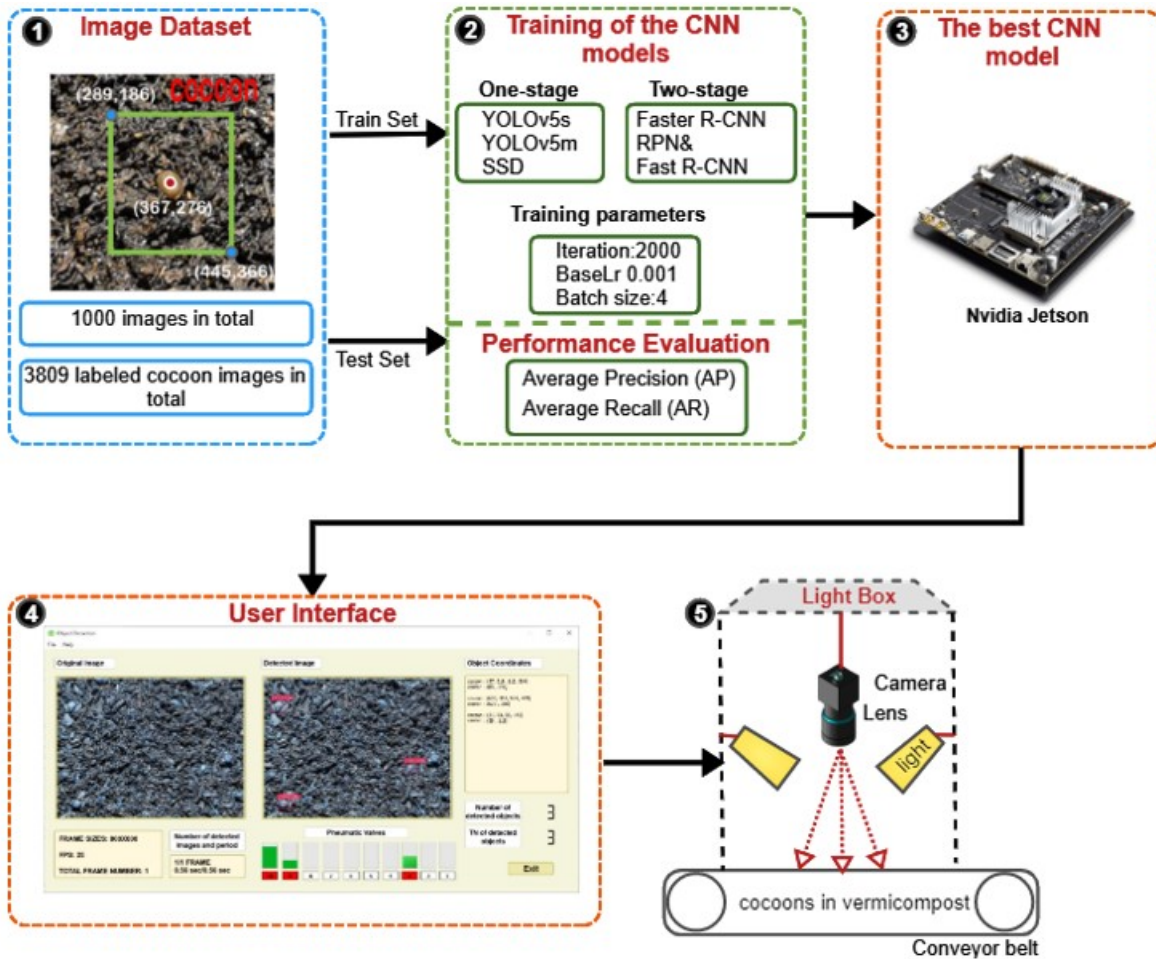
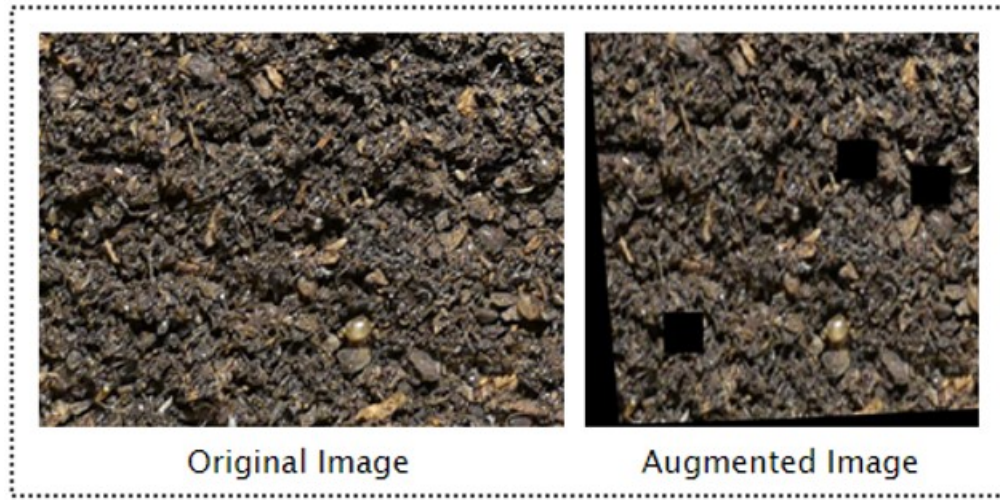


Figure 1. General operation scheme for CoDeSS.

All models were configured to use the base models, and the training hyperparameters (Table 1) were predefined and remained static for all experiments. The PyTorch framework was used for the developed software. For the GUI software of the CoDeSS, the Qt Designer and various Python libraries were used. The training and testing processes for the detection of cocoons in this work is performed on a workstation with the configurations of Intel Xeon CPU, 16GB Nvidia Quadro RTX5000 GPU with 16GB RAM.

Table 1. Main hyperparameters used the train the object detection models.

Hyperparameters	Value
Learning rate (LR)	BaseLR=0,001; 100 warmup epochs then 0,12x /baseLR, 0,5x BaseLR, 1x BaseLR at 100, 500, 1000–2000 epochs respectively
Batch size	4
Epochs	2000



**Figure 2.** Comparison of original image and augmented image.

### 2.3. Performance evaluation

In ImageNet large scale visual recognition challenge (ILSVRC) [15] known as ImageNet challenge, objects with various class labels were classified by training millions of images. ImageNet, Microsoft Common Objects in Context (MS-COCO) [16], Pascal VOC [17], etc., datasets were used to identify successful models. The performance evaluations of the successful models in the competitions were conducted using different metrics. In this study, the performance evaluation metrics AP and AR were used for faster R-CNN, RPN & fast R-CNN, YOLOv5s, YOLOv5m, and SSD architectures. For the performance evaluation of CoDeSS, AP and AR were preferred, which are commonly used in the literature [18–20] for object detection problems in agriculture. The evaluation metrics chosen in this work are similar to those of MS-COCO dataset [18], namely mean average precision (mAP) and mean average recall (mAR). mAP is the mean value of APs under 10 intersection over union (IoU) threshold values and AP represents mAP in the evaluation system of MS-COCO, so  $AP$  is defined as Equation 1

$$AP = \frac{1}{10}(AP_{50} + AP_{55} + AP_{60} + \dots + AP_{95}). \quad (1)$$

In the evaluation system of MS-COCO, we refer to AR as mAR. There are different subindices depending on the number of detections. Three numbers, 1, 10 and 100, are selected in the evaluation system of MS-COCO, and the corresponding subindices are AR1, AR10 and AR100 [21].

### 2.4. User interface

The GUI software developed for CoDeSS can be seen in Figure 3.

The cocoons detected in the image can be seen on the screen of the software in real-time. The pneumatic valve used to separate the cocoons can be seen using ten progress bars in the software. The distances of the cocoons to the separation point can be monitored in real-time from these progress bars. In addition, results such as the number of processed images, the processing time, the number of cocoons detected, and the total number of cocoons detected can be viewed in the software in real-time.

The software uses the coordinates on the image to calculate how quickly the cocoon will reach the



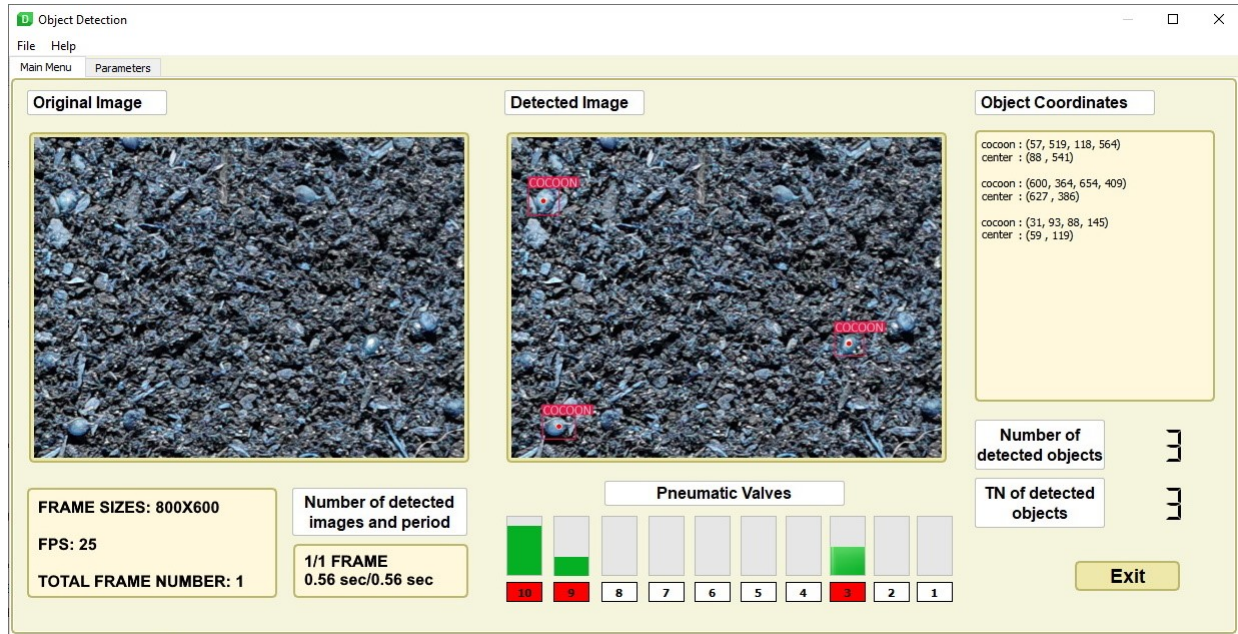


Figure 3. GUI developed for CoDeSS.

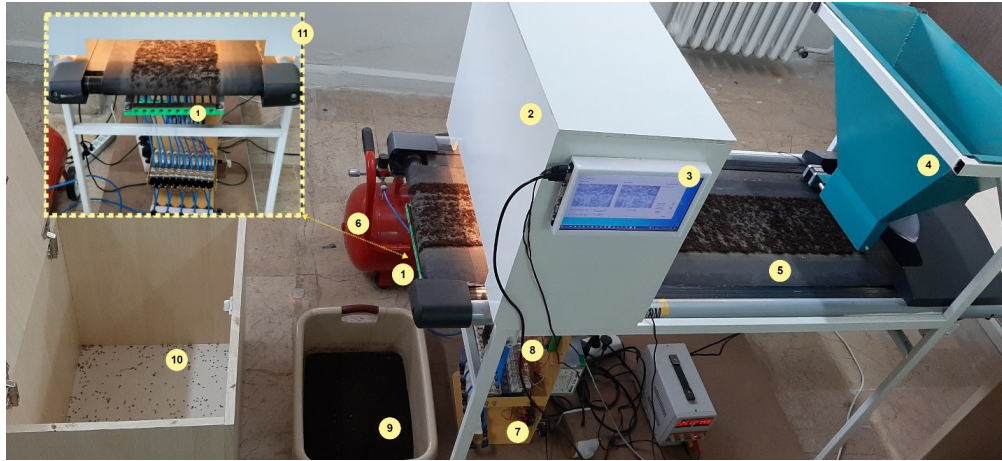
separation point. The air spraying time of each pneumatic valve controlled by the Nvidia Jetson TX2 was determined as 200ms. If this time is extended, more compost will fall into the separation chamber along with the cocoons because more air is blown through the valves.

### 2.5. The CoDeSS hardware system

The hardware units of CoDeSS that enable earthworm cocoons to be returned to production using an intelligent real-time system are shown in Figure 4. The embedded AI computing device in CoDeSS and the industrial camera in the light box are the basic units used for cocoon detection. The industrial camera used in this study has area scan, rolling shutter, USB 3.1, and 143 fps. A 6mm lens is preferred. With this lens, a field of view of 1360 mm × 858 mm can be captured. This value is quite sufficient for this study. The interior of the light box used is illuminated with 2 light strips.

Another unit is the feeder unit where compost is added. There is a mechanism inside the feeder unit to spread the compost on the conveyor belt. The mechanism is connected to a motor with adjustable speed. Thus, homogeneous spread of compost on the conveyor belt is ensured. Through the LCD screen connected to Nvidia Jetson, the GUI can be both controlled and monitored in real-time. The compost that passes through the light box enters the separation unit. Pneumatic valves located just below the conveyor separate the cocoons in the falling compost by spraying air. The compressed air required is provided by an air compressor. Another important step regarding CoDeSS is that the cocoons arrive at the separation point, where the ten pneumatic valves are located, and are separated from the compost. At this stage, calculating the time between the location where the cocoons are detected and the separation point is a problem that was solved in this study.

The CoDeSS images of the solution developed for this problem can be seen in Figures 5a and 5b. The distance traveled by a detected cocoon up to the separation point is expressed by the  $x$  shown in Figure 5b. This distance consists of two parts. The first is the distance  $x_1$  from the point where a cocoon is detected to



**Figure 4.** CoDeSS hardware system. 1. Separation unit, 2. Light box, 3. Software GUI, 4. Compost feeder unit, 5. Conveyor belt and compost, 6. Air compressor, 7. NVIDIA Jetson TX2, 8. Pneumatic valves, 9. Compost, 10. Separated cocoons, 11. Front view of separation unit.

the end of the conveyor belt while the second is the distance  $x_2$  between the point where the conveyor belt ends and the separation point.

The  $x_1$  distance (Equation 2) is obtained by adding the distance between the point where the image ends and the point where the belt ends  $rcl$  to the difference between the height of the detected image  $ih$  and the position of the cocoon on the image  $cp$ .

$$x_1 = \frac{(ih - cp)}{res} + rcl \quad (2)$$

The time spent by cocoons on the belt along the  $x_1$  distance is calculated as in Equation 3. Here, the difference between  $ih$  and  $cp$  is given in pixels. This value is divided by the resolution of the image and converted to centimeters.

$$t_1 = \frac{x_1}{cs} = \frac{\frac{(ih - cp)}{res} + rcl}{cs}, \quad (3)$$

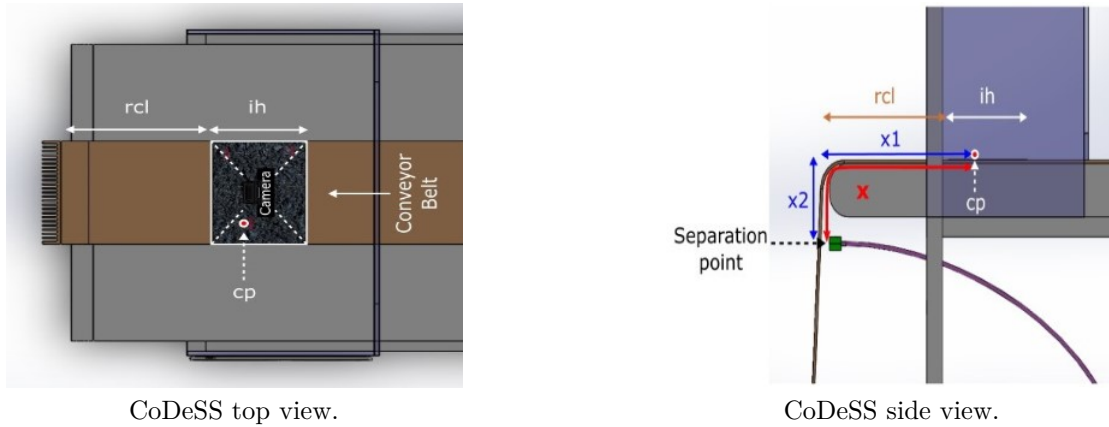
where  $ih$  (pixel) is the height of the detected image,  $cp$  (pixel) the center point of the detected cocoon, and  $res$  the resolution of the detected image (pixels/cm),  $rcl$  (cm) is the distance between the point where the image ends and the point where the conveyor belt ends, and  $cs$  (cm/sec) represents the speed of the conveyor belt.

The time spent by the cocoons on the  $x_2$  distance is calculated as in Equation 4. In this study, the  $x_2$  distance is  $8.5cm$ . When this value is written in Equations 4 and 5, a time of  $0.132sec$  is obtained. Therefore,  $t_2 = 0.132sec$  was determined for the study.

$$x_2 = 8.5cm = \frac{1}{2}gt_2^2 \quad (4)$$

$$t_2 = \sqrt{\frac{2x_2}{g}}, \quad (5)$$

where  $x_2$  refers to the distance between the point where the cocoons leave the conveyor belt and the separation point, and  $g$  is the gravity acceleration  $9.8m/sn^2$ .



**Figure 5.** View of the CoDeSS from different angles

The time it takes a cocoon to travel this distance can be calculated as the sum of the times from Equations 3 and 5 and given as  $t$  in Equation 6.

$$t = t_1 + t_2 \Rightarrow \frac{x_1}{cs} + \sqrt{\frac{2x_2}{g}} \quad (6)$$

### 3. Results and discussion

Deep learning-based object detection architectures are divided into two groups: one- and two-stage detection. The faster R-CNN [22] is a typical two-stage detection network that is executed depending on the region proposal. The high computational cost associated with the region proposal process leads to unsatisfactory results in terms of time, especially for object detection with real-time applications. YOLO [23] series are typical one-stage detection networks and produces more successful results in terms of time compared to faster R-CNN and similar architectures, especially in real-time applications.

To verify the effectiveness of the proposed method for earthworm cocoon detection, five object detection algorithms were compared in this study, including faster R-CNN, RPN & Fast R-CNN, YOLO v5s, YOLO v5m and SSD [24].

#### 3.1. Comparison of state-of-the-art-models

Rapid and accurate detection of earthworm cocoons is not only helpful in sorting out more cocoons from compost, but also ensures more compost processing. Three backbones, ResNet50-FPN, ResNet101-FPN and ResNeXt101-FPN, were used for faster R-CNN. RPN-R50-C4 and RPN-R50-FPN were preferred as backbones for the RPN and fast R-CNN model. While CSPDarknet is used for one-stage models such as YOLOv5s and YOLOv5m, VGG16 is preferred for SSD model.

In this section, we compare the state-of-the-art models with different backbone networks as shown in Table 2. The AP50 value is 89.0 when ResNet50-FPN is used as the backbone for the faster R-CNN model. This AP50 value is the best result obtained from the experiments using the three backbones. The lowest inference time for the faster R-CNN model was obtained in the experiments with the ResNet50-FPN backbone at 0.26.

Apart from AP, which directly measures the performance of a detector, AR is also an important metric. Just like AP, a high AR value is expected for a good model. The AR values obtained in the experiments for



RPN and fast R-CNN, another state-of-the-art model, are shown in Table 2. It can be seen that both backbone models are quite close. The lowest inference time for the RPN and fast R-CNN model was found to be 0.13 in the experiments with the RPN-R50-C4 backbone. It can be said that one-stage models have better inference time performance than two-stage models. From the experimental results, faster R-CNN with ResNet50-FPN model detected the earthworm cocoons appropriately compared with RPN and fast R-CNN, SSD and YOLOv5 models.

The speed of object detection is a very important factor in real-time studies. Since our study has an industrial application area, the speed parameter must be considered to separate more cocoons. Although the faster R-CNN model is slightly better in terms of AP, the YOLO models are more effective in terms of speed.

**Table 2.** Performance comparison of state-of-the-art models.

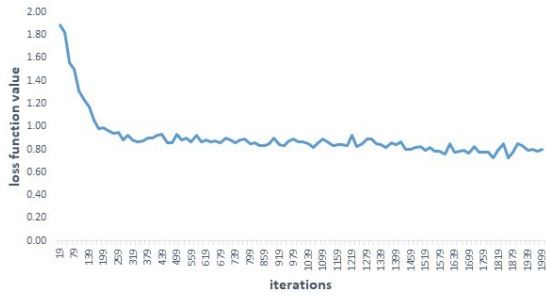
State-of-the-art models	Backbone	AP(50..95) (%)	AP50 (%)	AP75 (%)	AP100 (%)	Inference time (s/im)
Faster R-CNN (two-stage)	ResNet50-FPN	36.0	89.0	14.7	-	0.26
	ResNet101-FPN	33.5	87.4	11.4	-	0.36
	ResNeXt101-FPN	38.5	88.2	16.7	-	0.58
RPN and fast R-CNN (two-stage)	RPN-R50-C4	-	-	-	54.0	0.13
	RPN-R50-FPN	-	-	-	54.6	0.21
YOLOv5s (one-stage)	CSPDarknet	45.5	87.0	17.1	85.6	0.09
YOLOv5m (one-stage)	CSPDarknet	45.0	88.7	18.2	86.0	0.08
SSD (one-stage)	VGG16	30.2	84.9	10.6	-	0.11

The training loss curves of one-stage models converged, as shown in Figure 6, where the coordinates represent the iterations and the values of the loss functions. It is seen that loss values have dropped down to 0.80 and 0.90 respectively in Figures 6a and 6b. The loss curve of the faster R-CNN model (with ResNeXt101-FPN as backbone) is shown in Figure 6c. The loss value for this model has decreased to 0.70. The loss curves of the RPN Fast R-CNN models are shown in Figures 6d and 6e. The loss curve of the YOLOv5s is shown in Figure 6f and the loss value for this model has decreased to 0.75. In Figure 6g, it can be seen that the loss value for the YOLOv5m model has decreased to 0.80. The loss curve of the SSD model is shown in Figure 6h. It can be said that the loss values of these models are the lowest among all models.

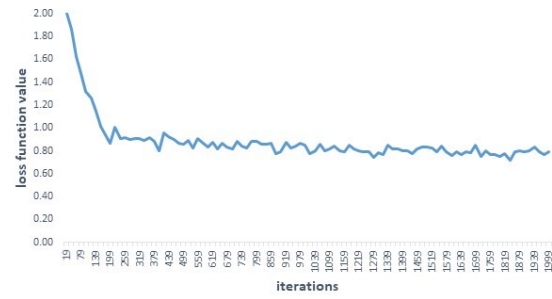
As the number of training iterations continually increased, the loss values of all models decreased suddenly at first and then decreased slowly. The loss curve of faster R-CNN model (almost all backbone models) gradually converged toward 0.8 after about 300 iterations, while that of RPN and fast R-CNN model varied depending on the backbone type. Accordingly, the loss curve of the RPN and fast R-CNN model (with RPN-R50-C4 as backbone) with gradually converged toward 0.2 after about 300 iterations, while that of RPN-R50-FPN converged toward 0.35 after about 300 iterations. The convergence of the loss curves indicated that the predicted results were credible and that the trained models had learned the features of the earthworm cocoons for their detection.

For two-stage models, just like in one-stage models, it can be said that the loss curve starts to become stagnant after about 300 iterations. For the YOLO models, the values of the loss function decreased to the value of 0.80.

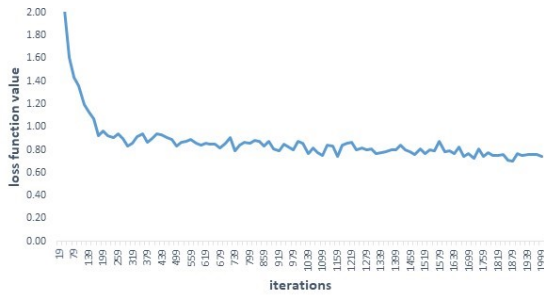
Example results for each model when CoDeSS is run are shown in Figure 7. Cocoon detection is a difficult object detection problem. This is because the detected cocoons are small and close to the color of the compost.



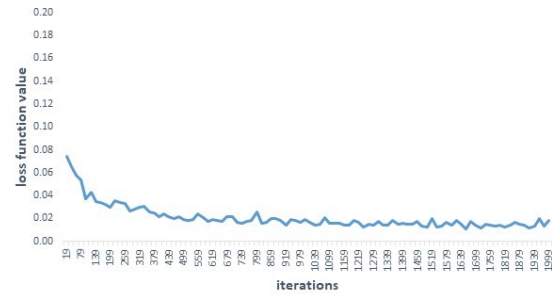
The loss curve of faster R-CNN model (with ResNet50-FPN as backbone).



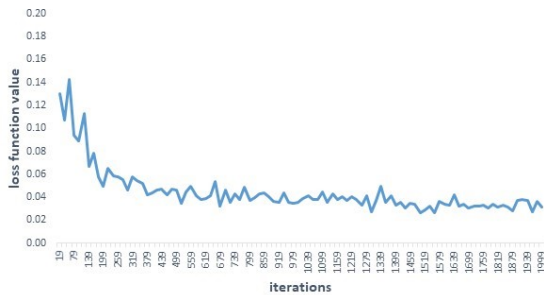
The loss curve of faster R-CNN model (with ResNet101-FPN as backbone).



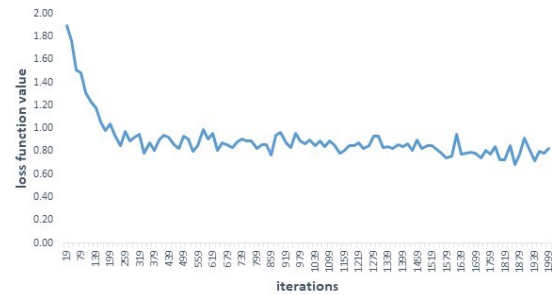
The loss curve of faster R-CNN model (with ResNeXt101-FPN as backbone).



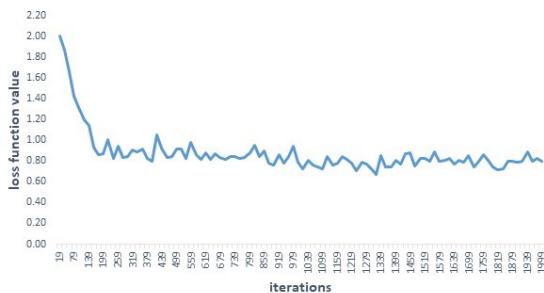
The loss curve of RPN and fast R-CNN model (with RPN-R50-C4 as backbone).



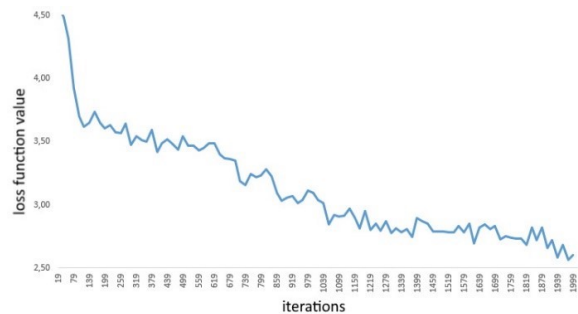
The loss curve of RPN and fast R-CNN model (with RPN-R50-FPN as backbone).



The loss curve model of YOLOv5s.



The loss curve model of YOLOv5m.



The loss curve model of SSD.

Figure 6. The loss curves for all models.

Some models detected all cocoons in the compost, while other models such as that faster R-CNN model (with ResNeXt101-FPN as backbone) and YOLOv5s were insufficient.

When the studies related to the field of agriculture using object detection models are examined, performance comparisons can be made in terms of the AP or mAP values obtained. Since our study was concerned with small object detection, a performance comparison can be made with studies on topics such as plant disease detection. Bhatt et al. [25] achieved an mAP of 86% with the YOLO-based models they developed for disease and pest detection on tea plants. In another study [26], the VGG16-based SSD system, developed to prevent damage to paddy seedlings during mechanical weed removal in paddy fields, achieved a mAP value of 89.6%, while the MobileNet-based SSD system achieved a mAP value of 92.8%. Jiang et al. [27] obtained a mAP score of 78.8% using the INAR-SSD (SSD with Inception module and Rainbow concatenation) developed to detect five different apple leaf diseases. Fuentes et al. [28] achieved a mAP score between 76% and 88% in their studies on disease and pest detection in tomato plants using SSD architecture. In another study [29], a dataset with 47696 labeled pest images was created. The dataset with 140 different categories of weed seeds was trained with the five CNN architectures. In the study, the highest accuracy value was obtained with the GoogLeNet model with 93.11%. Another small object detection application in agriculture, AgriPest21, created a pest dataset collected by an automatic pest image acquisition device. The dataset includes 24,412 images with 21 categories. The proposed method achieved an average recall of 89.0% and a mAP of 78.7%, outperforming other state-of-the-art methods, including SSD, RetinaNet, Free-Anchor, PISA, Grid RCNN and Cascade RCNN detectors [30]. Li et al. [31] set traps to catch the pest in the greenhouses. The pests attached to the traps are classified into 3 classes. The obtained dataset is trained with faster R-CNN and the recommended model. Their proposed model was more successful with a value of 95.2% mAP.

Table 3 shows the comparison between our study and some other studies. The only study on earthworm cocoon detection in the literature is our study. Therefore, the studies on insect and pest detection with physical features close to the cocoon were considered. These studies only performed object detection. No sorting system was developed in any study. Only our work proposed both an object detection and a sorting system. Our study was conducted with different CNN models and the data were augmented to better train the models. The proposed CNN-based approach was validated by experimental evaluation. The fact that our study was conducted in real time was an important contribution.

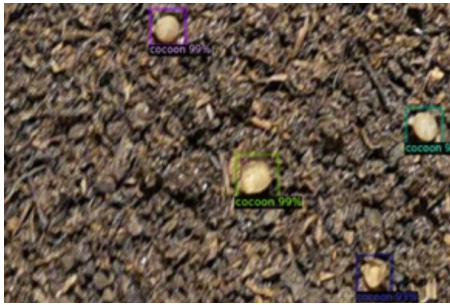
#### 4. Conclusion

In this study, the Cocoon Detect and Sorting System (CoDeSS) was developed to detect earthworm cocoons using state-of-the-art deep learning architectures and to separate the detected cocoons from the compost and return them to production. The training, performed with a total of 1000 images of cocoons, was completed after 2000 epochs, and a AP result of 89% was obtained for the faster R-CNN with the ResNet-50-FPN model with the lowest loss value. Cocoons detected by the cocoon separation mechanism developed in the study were separated from the compost using an air spray system. The study is the first example in the literature of returning earthworm cocoons to production using an intelligent real-time system.

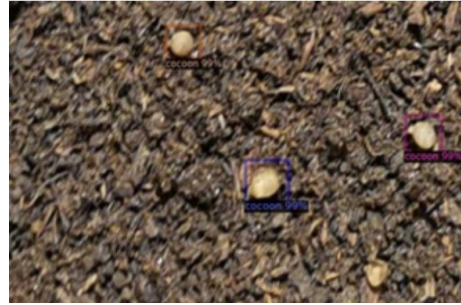
The small size of the detected cocoons and their similarity to the color of the compost is one of the factors that could reduce the success rate of object detection. On the other hand, a binary classification problem offers an advantage over multiclass problems because there is only one-class the cocoon class.

In similar studies, different CNN models can be tried and the effects of the different models on the AP value can be compared. Different background images can be included in images containing objects detected as





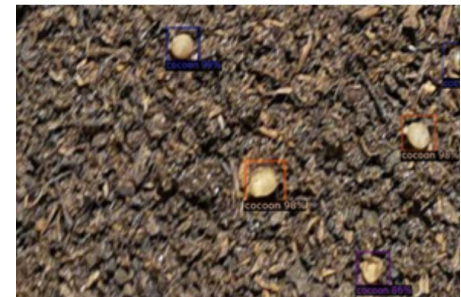
Example of faster R-CNN model (with ResNet50-FPN as backbone).



Example of faster R-CNN model (with ResNet101-FPN as backbone).



Example of faster R-CNN model (with ResNeXt101-FPN as backbone).



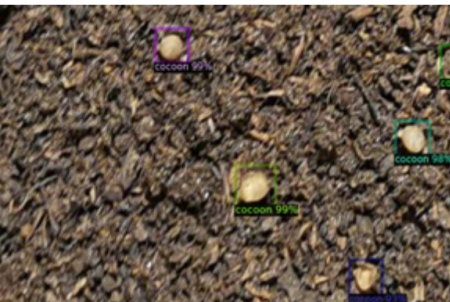
Example of RPN and fast R-CNN model (with RPN-R50-C4 as backbone).



Example of RPN and fast R-CNN model (with RPN-R50-FPN as backbone).



Example of YOLOv5s model.



Example of YOLOv5m model.



Example of SSD model.

**Figure 7.** Sample results of earthworm cocoons images detected by models.



**Table 3.** Results from previous studies on small object detection in agriculture.

Ref	Image number	Task type	CNN networks	Backbone	Sorting system	Performance
[2]	7706	Insect recognition	Proposed CNN Model	-	No	91.5% accuracy
[3]	25378	Pest detection	Faster R-CNN, SSD, YOLOv3, Cascade R-CNN	-	No	63.5% mAP
[4]	500	Pest detection	SSD, faster R-CNN	MobileNet	No	55% mAP
[25]	250	Pest detection	YOLOv2, YOLOv3	DarkNet-19, MobileNet, Inception v2, ResNet-101, DarkNet-53	No	86% mAP
[26]	-	Weed detection	Faster R-CNN, SSD	-	No	89.6% mAP
[30]	24412	Pest detection	SSD, RetinaNet, Cascade R-CNN	ResNet-50	No	78.7% mAP
[31]	17579	Pest detection	Faster R-CNN, Proposed Model	VGG-16	No	95.2% mAP
Our method	2000	Cocoon detection	Faster R-CNN, RPN and fast R-CNN, SSD, YOLOv5s, YOLOv5m	ResNet50-FPN, ResNet101-FPN, ResNeXt101-FPN, RPN-R50-C4, VGG16	Yes	89% AP

part of an object detection exercises. Since only compost is included in the background in this study, 1000 images were considered sufficient for training purposes. In similar studies, a change in performance can be observed when the data are augmented with the use of different agricultural products used in compost production. In such a case, the compost images can be displayed in different colors and textures.

In this study, images obtained from a single earthworm facility were used. Images of composts created with different nutrient contents could be used for similar studies. Another factor that can affect performance is training with images taken with higher image quality cameras. Since cocoons are alive, experiments could be conducted with images taken with thermal imaging cameras. Another important issue when using cameras is the speed of the conveyor belt. In this study, a fixed belt speed was used. If CoDeSS is to run at higher belt speeds, cameras with higher fps and transmission rates should be used. In this study, we considered not only object detection, but also cocoon separation with the developed CoDeSS. This provided a new and different contribution to deep learning studies in agriculture.

### References

- [1] Tong K, Wu Y, Zhou F. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*. 2020 ;40 (8):103910. doi: 10.1016/j.imavis.2020.103910
- [2] Yang G, Bao Y, Liu Z. Localization and recognition of pests in tea plantation based on image saliency analysis and convolutional neural network. *Transactions of the Chinese Society of Agricultural Engineering*. 2017:156-162

- [3] Wang QJ, Zhang SY, Dong SF, Zhang GC, Yang J et al. Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. *Computers and Electronics in Agriculture*. 2020; 175:105585. doi: 10.1016/j.compag.2020.105585
- [4] Karar ME, Alsunaydi F, Albusaymi S, Alotaibi S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alexandria Engineering Journal*. 2021;60 (5):4423–4432. doi: 10.1016/j.aej.2021.03.009
- [5] Jiao L, Dong S, Zhang S, Xie C, Wang H. AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection. *Computers and Electronics in Agriculture*. 2020;174:105522. doi: 10.1016/j.compag.2020.105522
- [6] Xia X, Chai X, Zhang N, Sun T. Visual classification of apple bud-types via attention-guided data enrichment network. *Computers and Electronics in Agriculture*. 2021;191:106504. doi: 10.1016/j.compag.2021.106504
- [7] Xu W, Zhao L, Li J, Shang S, Ding X et al. Detection and classification of tea buds based on deep learning. *Computers and Electronics in Agriculture*. 2022;192:106547. doi: 10.1016/j.compag.2021.106547
- [8] Kakehi S, Sekiuchi T, Ito H, Ueno S, Takeuchi Y et al. Identification and counting of Pacific oyster *Crasostrea gigas* larvae by object detection using deep learning. *Aquacultural Engineering*. 2021;95:102197. doi: 10.1016/j.aquaeng.2021.102197
- [9] Lichtfouse E, Navarrete M, Debaeke P, Souchère V, Alberola C et al. Agronomy for sustainable agriculture: a review. *Sustainable agriculture*. 2009: 1-7.
- [10] Aziz MZ, Naveed M, Abbas T, Siddique S, Yaseen M. Alternative fertilizers and sustainable agriculture. *Innovations in Sustainable Agriculture*. 2019: 213-245.
- [11] Prasad MNV. *Agrochemicals detection, treatment and remediation*. Butterworth-Heinemann. 2020
- [12] Gafsi M, Legagneux B, Nguyen G, Robin P. Towards sustainable farming systems: Effectiveness and deficiency of the French procedure of sustainable agriculture. *Agricultural systems*. 2006;90 (1-3):226–242. doi: 10.1016/j.agsy.2006.01.002
- [13] Samal K, Mohan AR, Chaudhary N, Moulick S. Application of vermitechnology in waste management: a review on mechanism and performance. *Journal of Environmental Chemical Engineering*. 2019;7 (5):103392. doi: 10.1016/j.jece.2019.103392
- [14] Sharma K, Garg VK. Comparative analysis of vermicompost quality produced from rice straw and paper waste employing earthworm *Eisenia fetida* (Sav.). *Bioresource technology*. 2018; 250:708-715. doi: 10.1016/j.biortech.2017.11.101
- [15] Russakovsky O, Deng J, Su H, Krause J, Satheesh S et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*. 2015;115 (3):211-252. doi: 10.1007/s11263-015-0816-y
- [16] Lin TY, Maire M, Belongie S, Hays J, Perona P et al. Microsoft coco: Common objects in context. *European conference on computer vision*. 2014:740-755.
- [17] Everingham M, Van GL, Williams CKI, Winn J, Zisserman A. The pascal visual object classes (voc) challenge. *International journal of computer vision*. 2010;88 (2):303-338. doi: 10.1007/s11263-009-0275-4
- [18] de Aguiar ASP, dos Santos FBN, dos Santos LCF, de Jesus FVM, de Sousa AJM. Vineyard trunk detection using deep learning—An experimental device benchmark. *Computers and Electronics in Agriculture*. 2020; 175:105535. doi: 10.1016/j.compag.2020.105535
- [19] Li Y, Li M, Qi J, Zhou D, Zou Z et al. Detection of typical obstacles in orchards based on deep convolutional neural network. *Computers and Electronics in Agriculture*. 2021; 181:105932. doi: 10.1016/j.compag.2020.105932
- [20] Xudong Z, Xi K, Ningning F, Gang L. Automatic recognition of dairy cow mastitis from thermal images by a deep learning detector. *Computers and Electronics in Agriculture*. 2020; 178:105754. doi: 10.1016/j.compag.2020.105754

- [21] He Z, Jiang S, Zhang J, Wu G. Automatic damage detection using anchor-free method and unmanned surface vessel. *Automation in Construction*. 2022;133:104017. doi: 10.1016/j.autcon.2021.104017
- [22] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*. 2015;28
- [23] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016: 779-788
- [24] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S et al. Ssd: Single shot multibox detector. *European conference on computer vision*. 2016: 21-37. doi: 10.1007/978-3-319-46448-0\_2
- [25] Bhatt PV, Sarangi S, Pappula S. Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations. *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV*. 2019;11008:1100808. doi: 10.1117/12.2518868
- [26] Liu F, Yang Y, Zeng Y, Liu Z. Bending diagnosis of rice seedling lines and guidance line extraction of automatic weeding equipment in paddy field. *Mechanical Systems and Signal Processing*. 2020;142:106791. doi: 10.1016/j.ymsp.2020.106791
- [27] Jiang P, Chen Y, Liu B, He D, Liang C. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access*. 2019;7:59069-59080
- [28] Fuentes A, Yoon S, Kim SC, Park DS. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*. 2017;17 (9):2022. doi: 10.3390/s17092022
- [29] Luo T, Zhao J, Gu Y, Zhang S, Qiao X et al. Classification of weed seeds based on visual images and deep learning. *Information Processing in Agriculture*. 2021. doi:10.1016/j.inpa.2021.10.002
- [30] Wang R, Jiao L, Xie C, Chen P, Du J et al. S-RPN: Sampling-balanced region proposal network for small crop pest detection. *Computers and Electronics in Agriculture*. 2021;187:106290. doi:10.1016/j.compag.2021.106290
- [31] Li W, Wang D, Li M, Gao Y, Wu J et al. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. *Computers and Electronics in Agriculture*. 2021;183:106048. doi:10.1016/j.compag.2021.106048