Nature Environment and Pollution Technology
An International Quarterly Scientific Journal

p-ISSN: 0972-6268
(Print copies up to 2016)
e-ISSN: 2395-3454

Vol. 21 | No. 2 | pp. 899-908 | 2022

Original Research Paper

https://doi.org/10.46488/NEPT.2022.v21i02.053

Open Access Journal

An Improved Convolutional Neural Network for Plant Disease Detection Using Unmanned Aerial Vehicle Images

Dashuang Liang*, Wenping Liu*†, Lei Zhao*, Shixiang Zong and Youqing Luo****

*School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China

**Beijing Key Laboratory for Forest Pest Control, Beijing Forestry University, Beijing 100083, China

†Correspondence: Wenping Liu; wendyl@vip.163.com

Nat. Env. & Poll. Tech.
Website: www.neptjournal.com

Received: 15-06-2021
Revised: 02-08-2021
Accepted: 27-08-2021

Key Words:
Anchor-free detector
Diseased plant detection
Convolutional neural network
Unmanned aerial vehicles
CenterNet

ABSTRACT

Accurate and fast locating of diseased plants is critical for the sustainability of forest management. Recent developments in computer vision made by deep learning provide a new way for diseased plant detection from images captured by unmanned aerial vehicles (UAV). In this paper, we developed an anchor-free detector, an enhanced CenterNet named as Enhanced CenterNet (ECenterNet) model, which significantly improved the overall accuracy over the original CenterNet model without any increase in the running speed or number of parameters. Compared with the original model, in the newly proposed model improvements had been made in the training stage to increase the accuracy of the detector, while procedures in the test stage remained unchanged. Under the hold-out dataset, the proposed model is trained on 5,281 tiles and tested on 3,842 images, the results showed that the overall detection accuracy of ECenterNet reached 54.7% by COCO Challenge metrics (mean average precision (mAP) @[0.5, 0.95]), while mAP accuracy of the original CenterNet was 49.8%. This research indicates that the proposed deep learning detection model provides a better solution for detecting diseased plants from UAV images with high accuracy and real-time speed.

INTRODUCTION

Forests play a vital role in a country's economic, social and environmental benefits (Dash et al. 2017). Plant diseases and pests pose a serious threat to the growth of forests. Traditionally, the range and severity of plant diseases and pests are manually identified and scored by field investigations with expensive cost and low efficiency (Chiu 1993). Therefore, more accurate and faster detection of diseased and pest plants could help in developing an early treatment technology, while substantially reducing economic losses (Fuentes et al. 2017).

In the past, spectral detection technology using satellite remote sensing data or traditional computer vision methods, coupled with global positioning systems and geographic information systems, were widely used in detecting pest distribution and proved to be effective (Cao 2015). However, the efficiency of these methods was low and sometimes failed to accurately locate infected plants. Luckily, with the rapid technological developments of unmanned aerial vehicles (UAV), an inexpensive and fast way of getting high-resolution images of forest distribution becomes available. Based on these images, a variety of image processing methods have been developed to detect the distribution of diseased plants.

Advances in hardware technology have allowed for the evolution of deep convolutional neural network (CNN), which has achieved greater success in many fields, including image classification (He et al. 2016, Krizhevsky et al. 2012, Szegedy et al. 2015), facial recognition (Kshirsagar et al. 2011), segmentation (Chen et al. 2018, Long et al. 2015) and object detection (Dai et al. 2016, Liu et al. 2016, Redmon et al. 2016, Ren et al. 2017). Recently, the application of deep CNN for plant disease severity detection has been proposed and has shown a good performance.

This paper aims at detecting forest plant diseases and insect pests using the object detection method. In the past, many object detection methods had been proposed (e.g., Faster RCNN (Ren et al. 2017), YOLO (Redmon et al. 2016), SSD (Liu et al. 2016), and RFCN (Dai et al. 2016)). All these methods relied on a set of pre-defined anchor boxes and showed good results on the PASCAL VOC (Everingham & Williams 2010), COCO (Lin et al. 2014), and ILSVRC (Russakovsky et al. 2015) datasets. However, anchor boxes result in excessively many hyper-parameters, which need to be carefully tuned to achieve high performance. Meanwhile, anchor-based detection methods also bring complex IoU computation and matching between anchor boxes and ground-truth boxes during training. To avoid these drawbacks, some anchor free detectors are proposed, such as CornerNet (Law & Deng 2020), CenterNet (Zhou et al. 2019a), FCOS (Tian et al. 2020), and ExtremNet (Zhou et al.

2019b), all of which take the object detection as a standard key-point estimation problem. Among various anchor-free object detection algorithms, CenterNet uses the key-point estimation method to regress the center point, the width, and the height of an object. It is a simple, fast, and accurate detector without any Non-Maximum Suppression (NMS) postprocessing. Yet it still has some drawbacks in the application of UAV forestry image, especially in the image where the plants are close to each other. The objective of this study is to propose an enhanced CenterNet (ECenterNet) model for plant diseases and pest detection based on UAV images.

MATERIALS AND METHODS

Study Area

The study area is located in Lingyuan City, Liaoning Province in northeast China. The dominant vegetation is Chinese red pine (Pinus tabuliformis) with a few poplar trees (Populus spp.) occasionally occurring. The pest Dendroctonus (Scolytidae) has caused serious damage and tree mortality within the area. To quickly detect the damage level of the forest, six different sample sites distributed in the county were selected as the study area, namely Site 1, Site 2, Site 3, Site 4, Site 5, and Site 6. Fig. 1 shows the location and distribution of these sample plots on a map.

Data Collection

The UAV-based data acquisition was carried out in the study area on August 11th and August 12th, which was the best time window for catching leaf symptoms. The UAV model was a four-rotor DJI Inspire2, carrying a DJI x5 professional camera with an effective resolution of 2×10^7 pixels, providing an image size of 5280 × 3956 pixels. On each sample plot, a certain number of pictures were toke from 40 to 240 meters above sea level. With the increase in height, the forest land covered by the photos becomes larger and larger, and the size of individual trees in the photos becomes smaller and smaller. To capture images without overexposure or shadows, capture times when the sunlight was too strong or too weak were avoided.

Implementation Details

CenterNet Principles

One aim of CenterNet was to produce a key point heatmap, where C was the number of categories. Another aim of the CenterNet was to output the object size $s_k = \left(x_2^{(c)} - x_1^{(c)}, y_2^{(c)} - y_1^{(c)}\right)$ for each object k. To limit the computational burden, a single-size prediction was used for all object categories. The third aim of CenterNet was to output the offset $\left(x/s_x - \lfloor x/s_x \rfloor, y/s_y - \lfloor y/s_y \rfloor\right)$ between the real center computed and center pixels on the feature map.

For the center of an object (x, y) with class C on the input image, it was mapped to the feature map $\left(\lfloor x/s_x \rfloor, \lfloor y/s_y \rfloor\right)$, which was considered as a positive sample, where S_x and S_y were the horizon and vertical scale parameters, respectively. Points other than the center point were regarded as negative samples. The training objective function was a penalty-reduced pixel-wise logistic regression with focal loss (Lin et al. 2017):

$$L_k = \frac{-1}{N}\sum_{xyc}\begin{cases}\left(1-\hat{Y}_{xyc}\right)^{\alpha}\log\left(\hat{Y}_{xyc}\right) & \text{if } Y_{xyc}=1\\ \left(1-Y_{xyc}\right)^{\beta}\left(\hat{Y}_{xyc}\right)^{\alpha}\log\left(1-\hat{Y}_{xyc}\right) & \text{otherwise}\end{cases} \quad \ldots(1)$$

where α and β were hyper-parameters, respectively, and fixed to $\alpha = 2$ and $\beta = 4$ during training. N was the number of objects in the image. $\hat{Y}_{xyc}$ was the prediction probability of an object with center coordinate (x, y) with the label of c, Y_{xyc} was a heatmap that was created by a Gaussian kernel.

$$Y_{xyc} = \exp\left(-\frac{\left(x-\tilde{p}_x\right)^2 + \left(y-\tilde{p}_y\right)^2}{2\sigma_p^2}\right) \quad \ldots(2)$$

where σ_p was an object size-adaptive standard deviation (Law & Deng 2020). The Gaussian heatmap served as the weight map to reduce the penalty near a positive location in the logistic regression case.

For the offset $\left(x/s_x - \lfloor x/s_x \rfloor, y/s_y - \lfloor y/s_y \rfloor\right)$ between real center computed and center pixels on the feature map, the CenterNet used an L_1 loss to regression it, and all classes shared the same offset.

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\tilde{p}} - \left(x/s_x - \lfloor x/s_x \rfloor\right)\right| \quad \ldots(3)$$

For the size (w, h) of an object, CenterNet also used the L_1 loss to regress it, and all categories shared the same object size.

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_{pk} - s_k\right| \quad \ldots(4)$$

Thus, the total objective function of CenterNet was as follows:

$$L_{det} = L_k + \lambda_{size}L_{size} + \lambda_{off}L_{off} \quad \ldots(5)$$

where λ_{size} and λ_{off} where the weight of size and offset

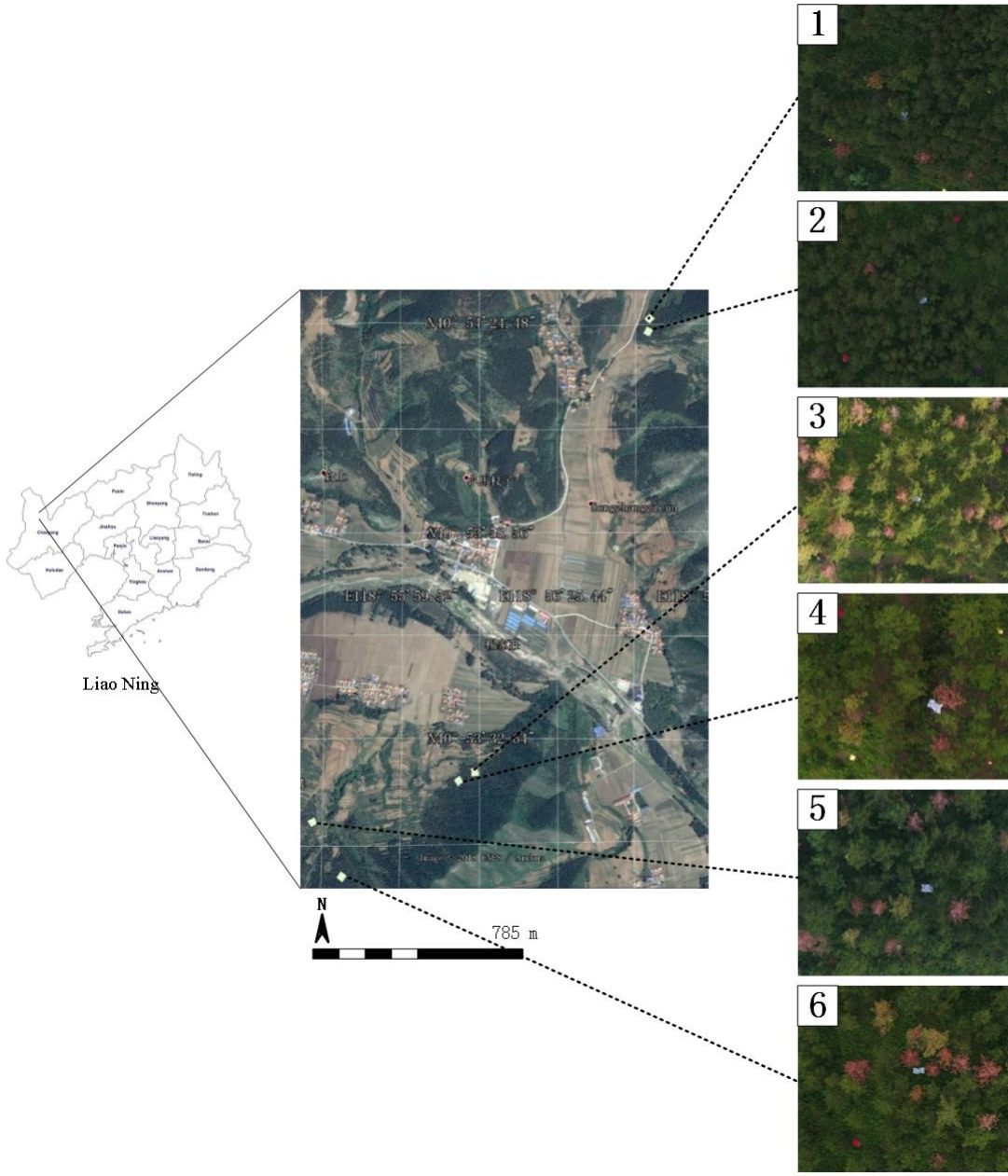


Fig. 1: The study area.

regression, respectively. L_{det} was the total loss of the CenterNet detector, L_k was the heatmap value of the center of an object, more details about CenterNet please see Zhou et al. (2019a).

Separate Overlapping Center Points of Two Objects

In the original CenterNet, the positive sample of an object was the center of the bounding box obtained by mapping the object from the original image to a 128 × 128 size feature map, and the pixels around the center point were considered negative samples. If two objects of the same class were close to each other on a larger original image, the center points of the two objects were mapped to the same point on the 128 × 128 size feature map, as it was shown in Fig. 2(a). This situation would bring confusion to the training of the network. To solve this problem, in this paper, one of the center points was forced to offset by one pixel if two center points on the feature map coincided with each other. The improved

operation which was Separate Overlapping Center Points of two objects (SOCP) aimed to separate two overlapping center points of two objects on the feature map [Fig. 2(b)]. The detailed rules of SOCP were as follows:

Step 1. Between the two objects' center points, the one with a larger bounding box area was selected to make the offset.

Step 2. Offset the selected object's center point by one pixel along the long side of the object's bounding box. Ordinary, either of the two directions (right or left) was selected to make the offset.

Step 3. Choose one of the two directions randomly. If one of them encountered the boundary of the image or the center of the other object's bounding box, this direction was abandoned and another direction was chosen.

Step 4. If two of the above directions were not selected due to the reasons in step 3, the long side direction was given up and the short side direction of the object would be chosen.

Step5. Randomly choose one of the two directions along the short side of the object. If one of them encountered the boundary of the image or the center of another object's bounding box, this direction would be abandoned and another direction would be selected.

Step 6. If neither of the directions was selected to move due to the reason in step 3, the offset operation was given up and the original strategy in original CenterNet was kept.

Controlled Sampling Strategy for Objects on the Boundary

In the original CenterNet, before the data were sent to the network, an affine transformation of random center shift and scaling was performed on the image. At this time, some objects were moved out of the image's boundary, some stayed in the image's boundary, and some fell on the boundary. For the objects that fell on the boundary, the original CenterNet method used clip operation to recalculate the bounding box coordinates of the remaining part. However, the new coordinates of the bounding box calculated by clipping were often inaccurate, as shown in Fig. 3(b), while the accurate coordinates of the left object's bounding box are shown in Fig. 3(c).

To avoid the above-mentioned problems as far as possible, there was a limitation that the bounding box area of the remaining object on the image's boundary should be more than 90% of that of the original one. In detail, before performing the affine transformation, the randomly generated offset

Fig. 2: (a) Two coincide center points on features from two close objects, (b) SOCP operation to two coincide center points on the feature map.

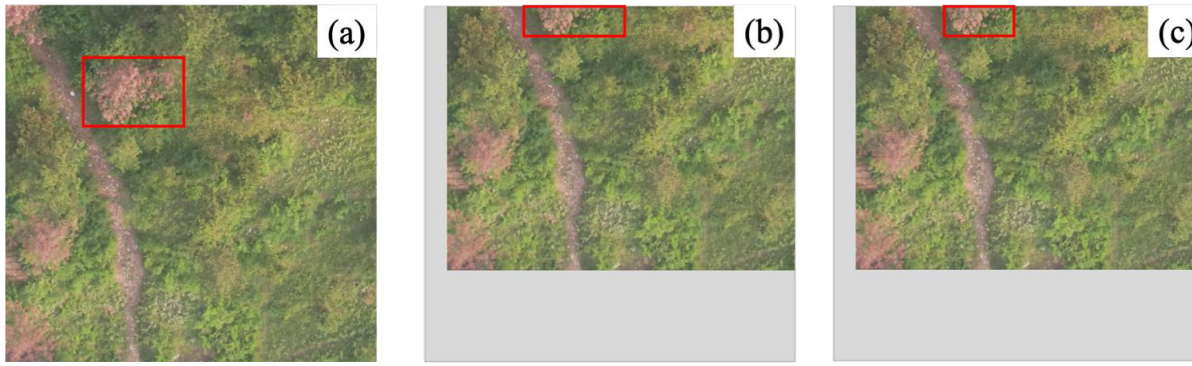


Fig. 3: (a) The original labeled picture, (b) the bounding box create in CenterNet after the affine transform, and (c) the real bounding box of the left part after the affine transform to picture.

parameter O and scaling factor S for affine transformation should be satisfied the limitation mentioned above. If it was not satisfied, a new set of parameters O and S were randomly created and then tried again until it was satisfied. This strategy was called a controlled sampling strategy (CSS). With the help of CSS, the bounding box recalculated by clip operation was very close to the accurate bounding box, especially for the trees whose shapes were approximately circular in this study. In addition, in the experiment, when replacing the affine transformation with the resize operation and no offset operation, the accuracy declined. The reason might be that the resize operation reduced the richness of data compared with the random affine transformation operation.

Positive Pixel Choosing Mechanism

In the original CenterNet, the center of the object's bounding box in the original image was considered to be a positive sample to object' category classification. However, for some objects, as shown in Fig. 4, the center point did not seem to be a good representation of the object. In addition, during the training of the original CenterNet, it was found that some pixels nearby the center pixel had larger IoU formed by the predicted bounding box and ground truth than that of the center pixel (Fig. 5). This phenomenon indicated that some pixels in the neighborhood of the center point were more suitable for an object's bounding box regression.

In view of the above phenomenon, a positive pixel choosing mechanism (PPCM) was designed in this paper to select a better positive sample. In the training stage, not only the center point of the object's bounding box but also the eight neighboring pixels around the center point were considered as candidate positive samples. During the training process, the center point together with the eight neighborhood points competed with each other, and the suitable one was selected as the positive sample of the object. Specifically, after a certain number (experiments show 30 was better) of epochs in the training stage, if one of the eight neighboring pixels' IoU formed by the bounding box and ground truth was 0.2 higher than that of the center pixel, the pixel was selected as a positive sample to calculate the loss of classification and regression in the following training stage.

Experiments

Data Preparation

In the fieldwork, plants with green leaves were classified as healthy plants, while those with yellow leaves were regarded as infected stage plants, and those with red leaves were classified as dead plants. This study only focused on diseased and infected plant detection, therefore, only plants with yellow and red leaves were annotated. As the original size of the images was 5,280 × 3,956 size pixels, which was too big to train a network, a set of image tiles was created by cropping each original aerial image by using a sliding window with random sizes between 1,000 and 2,000 pixels and stride of 1,000 pixels. In this way, one big aerial image was split into several small images. Before training, the images of Site 1, Site 3, Site 4, and Site 6 were split into training and validation datasets, while the images of Site 2 and Site 5 were

Fig. 4: Samples of the center point of the bounding box that cannot represent well of the related object.

chosen as testing datasets. At last, the training and validation datasets contained 5,281 tiles and 1,319 tiles respectively, and the test dataset contained 3,842 images. All the images were manually labeled with ground truth bounding boxes and assigned with class labels "infected" or "dead" (only one per bounding box).

To get as many samples as possible, some more data were created through the method of data augmentation. Several strategies were adopted to do data augmentation, such as flip, random color, random rotation, random crop, and so on.

Training

In all of these experiments, the input sizes of all of the networks were fixed to a size of 512 × 512, while the class number was two, including "dead", and "infected" classes. No matter how large the input image was, it would be scaled to the same size through affine transformation, and then pass through the network of CenterNet structure. ResNet-101 was selected as the backbone part of the network. After passing through the backbone network, the size of the feature layer became 128 × 128 because of the down sampling of the convolution and pooling layer.

As PyTorch is one of the most famous and fastest deep learning frameworks for CNN, it is used to train models in this experiment. The network was then trained on a single NVIDIA Titan 12 GB GPU. The training was stopped after 140 epochs, which took roughly 4 days. The Adam learning method was used as the gradient descent algorithm. The detailed training hyper-parameters were listed in Table 1.

Test and Comparison

To make a full comparison with other models, original CenterNet (Zhou et al. 2019a), SSD (Liu et al. 2016), and Faster RCNN (Ren et al. 2017) were trained with the same data and settings, and then were tested and compared with the proposed ECenterNet model.

To evaluate the final detections, the official COCO API (Lin & Dollar 2016), measured mAP over IoU thresholds from 0.5 to 0.95 with steps of 0.05, simply denoted as mAP@[.5, .95], was used as the performance indicator.

Table 1: Parameters of network training.

Argument	Value
Mini-batch size	8
Num_epochs	140
Lr_policy	Multistep
Stepvalue	90, 120
Initial learning rate	1.25e-4
Gamma	0.1

The Average Precision (AP) was the area under the Precision-Recall curve for the detection task. As in the COCO Challenge, the AP was computed by averaging the precision over a set of spaced recall levels from 0 to 1 with steps of 0.01.

$$AP=\frac{1}{11}\sum_{r\in\{0,0.1,\ldots,1\}} P_{\text{int}erp}(r) \qquad \ldots(6)$$

$$P_{\text{int}erp}(r)=\max_{\tilde{r}:\tilde{r}\geq r} p(\tilde{r})$$

where $p(\tilde{r})$ was the measure precision at recall $\tilde{r}$. AP was a concept of integrating precision as the recall was varied from 0 to 1, and mAP was defined as the average of AP for all of the object classes.

RESULTS

Results of Different Models

The results of different models are shown in Table 2. The proposed ECenterNet performed very well in both classes. Among the anchor-based methods, RetinaNet (Lin et al. 2017) performed best with 0.029 higher accuracies of mAP@[.5, .95] compared with the SSD method (Liu et al. 2016). In anchor free method, CenterNet (Zhou et al. 2019a) had a 0.07 higher accuracy of mAP@[.5, .95] than CornerNet (Law & Deng 2020), while it was 0.19 lower than that of CornerNet (Zhou et al. 2019a) in mAP@[.5, .95]. Compared with the method proposed in this paper, the accuracy of CornerNet (Law & Deng 2020) and CenterNet (Zhou et al. 2019a) were relatively lower. In detail, the proposed method outperformed CornerNet (Law & Deng 2020) with 0.056 in mAP@[.5, .95], 0.011 in mAP@[.5] and 0.053 in mAP@[.75]. When compared with CenterNet (Zhou et al. 2019a), the proposed method was 0.049, 0.030, and 0.055 higher in mAP@[.5, .95], mAP@[.5] and mAP@[.75], respectively.

Detection Samples

In the first two columns of Fig. 6, plants were detected by the original CenterNet (Zhou et al. 2019a) model and the proposed ECenterNet model. The first row showed the images that contained "dead" plants on the boundary of the image. As can be seen from the first row in Fig. 6, the boundary box regressed by the original CenterNet model was larger than the ground truth boundary box, and its category score was only 0.82. In contrast, the ECenterNet's regression of the bounding box was more accurate as shown in the first row and second column in Fig. 6. Moreover, the category score of the boundary box was 0.95, which was also higher than that

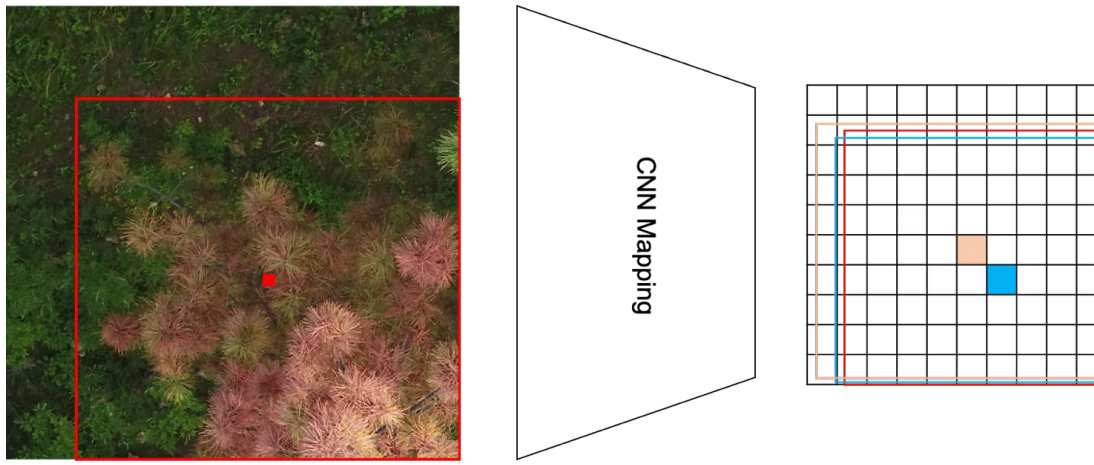


Fig. 5: Red bounding box is the ground truth box; The yellow pixel in the feature map is the positive sample produced by the rule of the original CenterNet, and the yellow bounding box is the related box predicted by this positive sample; the Blue pixel is the neighbor pixel of the yellow one, and the blue bounding box is the related bounding box of the blue pixel; IoU between red and blue bounding box is larger than IoU between red and yellow bounding box.

Table 2: Detection results comparison using different frameworks and network architectures.

Method	backbone	mAP@[.5]	mAP@[.75]	mAP@[.5, .95]
Faster RCNN	ResNet-101	0.693	0.535	0.472
SSD	ResNet-101	0.573	0.489	0.451
RetinaNet	ResNet-101-FPN	0.724	0.546	0.480
CornerNet	Hourglass-104	0.722	0.559	0.491
CenterNet	ResNet-101	0.703	0.557	0.498
ECenterNet (Ours)	ResNet-101	0.733	0.612	0.547

mAP stands for mean average precision.

of the original CenterNet. It could be seen from the second row in Fig. 6, that the original CenterNet might not be able to detect objects, which were very small in the image, while ECenterNet could successfully detect them from the whole image. It could be seen from the third row in Fig. 6 that some plant objects could not be detected by the original CenterNet, while ECenterNet detected them successfully. In addition, compared with ground truth, ECenterNet's boundary box regression was more accurate for most objects.

DISCUSSION

Detection Accuracy with Different IoU

Fig. 7(a-c) shows the precision-recall curves when the IoU thresholds are 0.5, 0.7, and 0.9, respectively. Compared with the original CenterNet, the improvement (area surrounded by the red line and the green line) of mAP (area under the curve) of the proposed method increased with the increase of IoU, suggesting that the proposed method had higher predictive power than original CenterNet (larger IoU indicated the more accurate location of an object). On the one hand, strategy CSS ensured that the newly calculated bounding box was more accurate for objects on the boundary of the image after affine transformation, on the other hand, the training strategy in PPCM enabled to choose of a more suitable pixel with a higher predicted IoU formed by predicted bounding box and ground truth to represent the object. Both of these two strategies improved the accuracy of position prediction to a certain extent.

Architecture Ablation and Diagnosis

To demonstrate the effectiveness of the method proposed in this paper, different supplementary experiments were carried out, and the results were shown in Table 3. Adding SOCP strategy resulted in 0.009 improvements in mAP@[.5, .95], while strategy CSS and strategy PPCM led to 0.16 and 0.24 improvement in mAP@[.5, .95], respectively. The improvement in accuracy brought by SOCP was relatively little when compared with the strategies of CSS and PPCM. The reason was that in the current test set, the side length

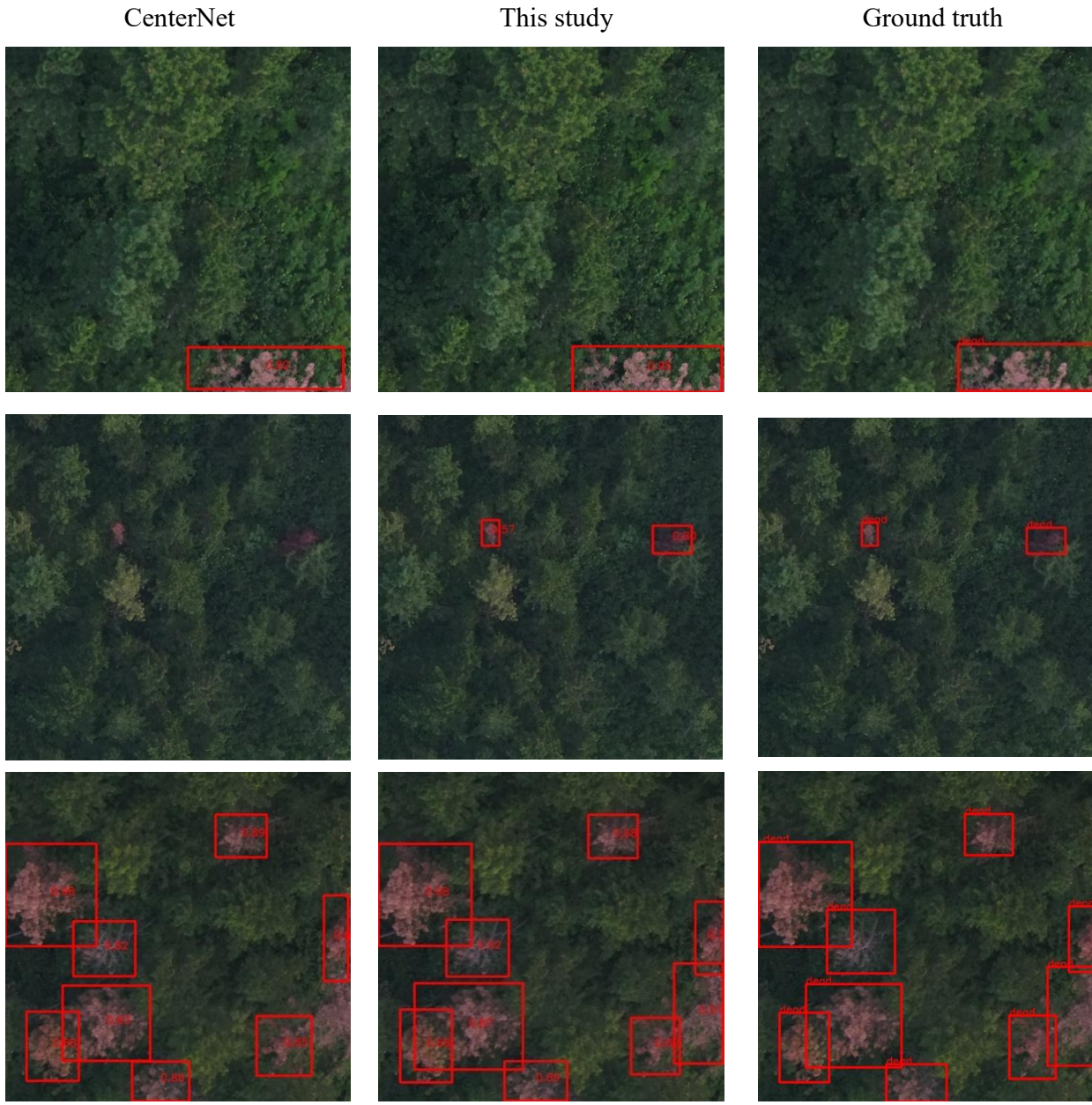


Fig. 6: Sample images of different models: the original CenterNet (Zhou et al. 2019a) model (left), the ECenterNet model (middle), and the ground truth (right).

Table 3: Detection results comparison using ablation and diagnosis architectures.

Method	mAP@[.5]	mAP@[.75]	mAP@[.5, .95]
CenterNet	0.703	0.557	0.498
+SOCP	0.722	0.568	0.507
+SOCP +CSS	0.731	0.599	0.523
+SOCP +CSS+PPCM(Ours)	0.733	0.612	0.547

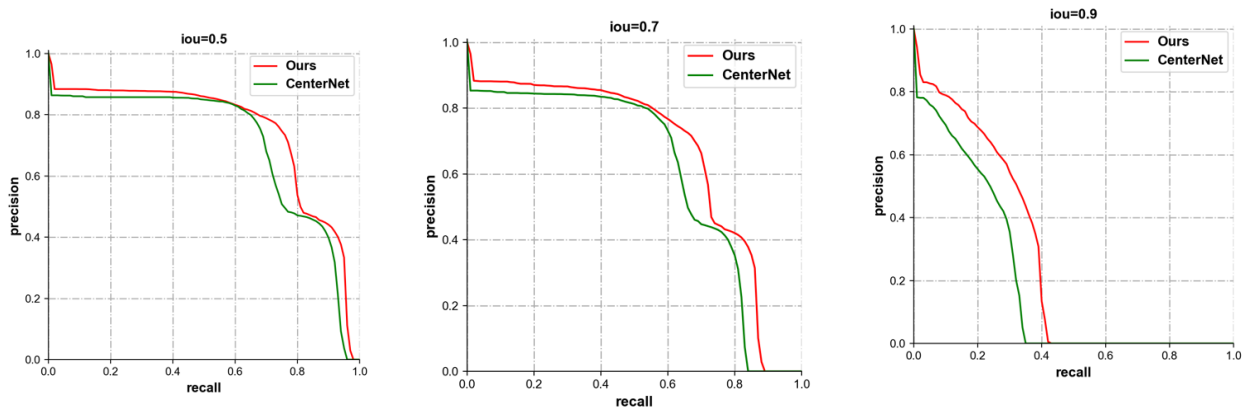
of the cut testing image was mostly between 1000 and 2000 pixels, which was less likely to yield objects with two overlapping centers when rescaled to a size of 128 × 128 pixels. However, it was speculated that with the increase in UAV shooting height, there would be more objects whose center points would coincide with each other, and thus SOCP strategy would play a more important role in inferring the whole big image.

Difficulties in Detection

In the proposed model, the accuracy of detecting the dead plant was higher than that of detecting the infected plant. That was because color differences were greater among infected plants than that of dead plants, while the dead plants were always characterized by pure carmine with small color variances. Fig. 8 shows samples of images including infected and dead plants. It could be clearly seen that even if most of the infected plants were yellow, their shades were different from each other. Some infected yellow plants were mixed with some red leaves, such as the infected plants in Fig. 8 (a, c), while the color of other infected plants was different from each other, such as the infected plants in Fig. 8 (b, d). All of these variations of infected plants add difficulty in detection. On the contrary, dead plants with red leaves could be detected easily, resulting in relatively high accuracy.

CONCLUSION

In this paper, we proposed an improved anchor-free object detection method based on CenterNet (Zhou et al. 2019a). The test results showed greater accuracy of the mAP@[.5], mAP@[.75], and mAP@[.5, .95] than the original CenterNet (Zhou et al. 2019a). The CSS was used to accurately locationing before the training stage, while SOCP and PPM were used to get a more suitable positive sample in the training stage. All the CSS, SOCP, and PPM operations helped to improve the detection accuracy. For the whole procedure, no extra parameters were introduced. In other words, the



(a) Class-agnostic precision-recall curves at IoU =0.50.

(b) Class-agnostic precision-recall curves at IoU =0.70.

(c) Class-agnostic precision-recall curves at IoU =0.90.

Fig. 7: Precision-recall curves of different IoU.

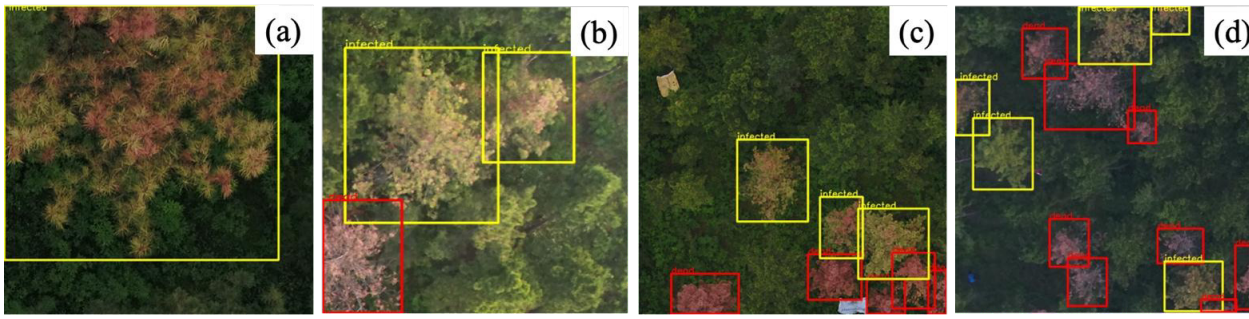


Fig. 8: Samples of images including various infected plants.

accuracy of the proposed model was improved without any increase in running time and model size.

Compared with the original CenterNet, three hyperparameters were added in this paper: one was the IoU threshold of 0.9 in CSS strategy, and the other two were the number of neighborhood pixels (8 in this method) in PPCM strategy and IoU threshold (in this method, it was 0.2). These parameters were empirical values. To explore the precise values of these parameters, Neural Architecture Search (NAS) technology can be used in future searches.

With the development of optimizing technologies, this model can be continuously improved with fewer computing resources, lower costs, and faster inference speeds. In the near future, there will be a model, which adopts a deep learning method for diseased plant detection on a UAV device, to enable researchers for fast and accurate detections. At that time, UAV can transfer the detection results to a ground receiving station in a timely manner during its flight, and researchers can use these results to prevent diseases from spreading in forests or do further studies.

ACKNOWLEDGMENT

This research was financially supported by the National Key Research and Development Program of China (No.2018YFD0600201). The authors are very grateful to the Lingyuan Forestry Bureau for assisting in the data collection process

REFERENCES

Cao, L. 2015. The research progress on machine recognition of plant diseases and insect pests. Chinese Agricultural Sci. Bull., 31: 244-249.

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. 2018. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Machine Intell., 40: 834-848.

Chiu, S.F. 1993. Investigations on Botanical Insecticides in South China: An Update. Botanical Pesticides in Integrated Pest Management, Rajahmundry, India, pp. 134-137.

Dai, J., Li, Y., He, K. and Sun, J. 2016. R-FCN: Object detection via region-based fully convolutional networks. Agronomy, 61: 379-387.

Dash, J.P., Watt, M.S., Pearse, G.D., Heaphy, M. and Dungey, H.S. 2017. Assessing very high-resolution UAV imagery for monitoring forest health during a simulated disease outbreak. ISPRS J. Photogr. Remote Sens., 131: 1-14.

Everingham, M. and Williams, C. 2010. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Part 1 – Challenge & Classification Task, International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification. Springer-Verlag, Germany, pp. 117-176.

Fuentes, A., Yoon, S., Kim, S.C. and Park, D.S. 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17: 2022.

He, K., Zhang, X., Ren, S. and Sun, J. 2016. Deep residual learning for image recognition. IEEE Conf. Comp. Vision Pattern Recog., 19(1):770-778.

Krizhevsky, A., Sutskever, I. and Hinton, G. 2012. ImageNet Classification with Deep Convolutional Neural Networks, NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems, 3-6 December, Lake Tahoe Nevada, Curran Associates, Redhook, NY, USA, pp. 1097-1105.

Kshirsagar, V.P., Baviskar, M.R. and Gaikwad, M.E. 2011. Face Recognition Using Eigenfaces. Proceedings of 2011 3rd International Conference on Computer Research and Development, 11-13 March 2011, Shanghai, China, IEEE, Piscataway, NJ, pp. 319-323.

Law, H. and Deng, J. 2020. CornerNet: Detecting Objects as Paired Keypoints. International Journal of Computer Vision, 128: 642-656.

Lin, T.Y. and Dollar, P. 2016. Ms coco API. https://github.com/pdollar/coco.

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P. 2017. Focal loss for dense object detection. IEEE Trans. Pattern Analy. Mach. Intel., 601: 2999-3007.

Lin, T.Y., Maire, M., Belongie, S., Hays, J. and Zitnick, C.L. 2014. Microsoft COCO: Common objects in context. Europ. Conf. Comp. Vision, 75: 740-755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C. 2016. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision, Springer International Publishing, New York, pp. 21-37.

Long, J., Shelhamer, E. and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Machine Intell., 39: 640-651.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. 2016. You only look once: unified, real-time object detection, 2016 IEEE Conf. Comp. Vision Pattern Recog., 121: 779-788.

Ren, S., He, K., Girshick, R. and Sun, J. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Machine Intell., 39: 1137-1149.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A. and Bernstein, M. 2015. ImageNet large scale visual recognition challenge. Int. J. Comp. Vision, 115: 211-252.

Szegedy, C., Wei, L., Jia, Y., Sermanet, P. and Rabinovich, A. 2015. Going deeper with convolutions. IEEE Conf. Comp. Vision Pattern Recog., 11: 1-9.

Tian, Z., Shen, C., Chen, H. and He, T. 2020. FCOS: Fully convolutional one-stage object detection. Conf. Comp. Vision Pattern Recog., 5: 13.

Zhou, X., Wang, D. and Krhenbühl, P. 2019a. Objects as Points. arXiv:1904.07850: 12.

Zhou, X., Zhuo, J. and Krähenbühl, P. 2019b. Bottom-Up Object Detection by Grouping Extreme and Center Points. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:1901.08043, pp. 850-859.