Lightweight sandy vegetation object detection algorithm based on attention mechanism

Zhongwei Hua,[1,2] Min Guan[2]

[1]Applied Technology College of Soochow University, Suzhou; [2]Academy for Engineering and Technology Fudan University, Shanghai, China

Abstract

This paper proposes a lightweight sandy vegetation object detection algorithm based on attention mechanism to solve the object detection task in the harsh sandy environment. We reduce the number of model parameters by the lightweight design of the anchor-free object detection algorithm model, thereby reducing the model inference time and memory cost. Specifically, the algorithm uses a lightweight backbone network to extract features and linear interpolation in the neck network to achieve multi-scale. Model algorithm compression is performed by depthwise separable convolution in the head network. At the same time, the channel attention mechanism is added to the model to optimise the algorithm further. Experiments have proved the superiority of the algorithm, the mAP in the training effect is 76%, and the prediction time per frame is 0.0277 seconds. It realises the efficiency and accuracy of the algorithm operation in the desert environment.

Correspondence: Zhongwei Hua, Applied Technology College of Soochow University, Suzhou, China.
E-mail: zwhua@suda.edu.cn

Key words: sandy vegetation; object detection; lightweight.

Acknowledgments: this study was sponsored by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province, China. (NO.22KJB520034).

Conflict of interest: the authors declare no potential conflict of interest.

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

Received for publication: 27 June 2022.
Revision received: 14 November 2022.
Accepted for publication: 15 November 2022.

Journal of Agricultural Engineering 2023; LIV:1471
doi:10.4081/jae.2023.1471

Introduction

Soil desertification is one of the world's important ecological and environmental problems. According to data provided by the United Nations, in the past 50 years, the world's land threatened by desertification accounted for 40% of the world's total area, about 3.6 billion hectares. In Africa alone, there are 36 countries facing desertification. The global economic loss due to desertification has reached 42 billion US dollars yearly (Michael *et al.*, 2018). With the advancement of science and technology, more and more intelligent devices based on computer vision are used to study and control deserts. Sandy vegetation has the functions of resisting sand burial, fixing quicksand, and preventing wind erosion, which plays a decisive role in restoring desert ecology. At present, most sandy vegetation is left to fend for itself. Our team is researching the exploration and maintenance of sandy vegetation through intelligent mobile robot technology to improve the survival rate of sandy vegetation and survey the distribution of sandy vegetation. Accurate and real-time perception of sandy vegetation is the prerequisite and technical basis for safe movement, obstacle avoidance, exploration, maintenance, and other work of robots in the sand. However, it is difficult to collect data sets due to the harsh desert environment. At the same time, the desert temperature is high, which limits the energy consumption, operating speed, and scale of network models of smart devices. In addition, factors such as the shapes of sandy vegetation species are similar, and the same colour of vegetation and background cannot be ignored. The existing object detection algorithms cannot deal with these challenges well (Figure 1).

Therefore, it is indispensable to study the object detection algorithm for sandy vegetation. Object detection technology is a technology to identify the category and position of target objects in an image, which plays a fundamental and vital role in computer vision (Zhang *et al.*, 2022). Before deep learning was widely used, traditional object detection techniques used a sliding window mechanism to select areas that may contain target objects and applied feature extraction algorithms such as SIFT to each obtained window. After that, the machine learning classification algorithm is used to classify each window, and the position information and category information of each box is obtained.

The object detection algorithm based on deep learning divides into two-stage and one stage according to whether there is a candidate frame generation stage. The two-stage object detection determines the candidate frame in advance to further improve the detection accuracy, among which the representative algorithms are the Fast-RCNN (Girshick, 2015) algorithm and the Faster-RCNN (Ren *et al.*, 2015) algorithm. However, because of the design of a network for extracting candidate frames, the increase in the number of parameters in this part makes the training and inference time of the model correspondingly longer. On the contrary, the one-stage object detection no longer extracts the candidate frame;

that is, the detection problem is transformed into a regression problem, among which the representative algorithms are the YOLO (Redmon and Farhadi, 2018) series algorithm and the SSD (Liu *et al.*, 2016) algorithm.

Both two-stage and one-stage object detection algorithms rely on predefined anchor boxes, which have always been considered the key to detection. Although the detection effect of this kind of algorithm is remarkable, the anchor frame-based detector still has some shortcomings, such as challenging hyperparameter setting, sensitivity to the aspect ratio of the anchor frame, imbalance of positive and negative templates, and complexity of calculation results (IoU). Therefore, anchor-free object detection algorithms such as CornerNet (Law and Deng, 2018), CenterNet (Duan *et al.*, 2019), and FCOS (Tian *et al.*, 2019b) are gradually becoming popular. CornerNet predicts the heat map of the upper left and lower right corners corresponding to all instances of the same category through a single convolutional neural network and the embedding vector corresponding to each corner point. The embedding vector combines the two corner points of the same target to form a detection frame. Although it is an anchor-free object detection algorithm, it is still carried out based on the corners of the anchor box, and the prediction is made through the corner point information of the anchor box. CenterNet focuses on the central point. Each heatmap corresponds to the centre point of a category. The centre point is represented by a Gaussian kernel, thus penalising the values around the target centre point relatively mildly. The model is trained through a separate convolutional neural network regression target centre point, size, and offset loss, and finally outputs the category and bounding box information, which truly realises the object detection algorithm without anchor boxes. FCOS directly predicts the category of the feature point and the distance from the pixel to the bounding box through the fully convolutional neural network on the feature map. The model uses FPN to layer the feature map to predict objects of different scales, thereby avoiding the situation of predicting multiple overlapping boxes under the same feature map.

In addition, to adapt to different operating conditions and hardware platforms, especially for mobile terminals, network model design tends to be lightweight, such as knowledge distillation, network pruning, and more refined lightweight backbone network design. Lightweight processing reduces the number of parameters in the fully connected layer of the model and reduces the complexity of the model while ensuring that the loss of model accuracy is small. At the same time, without significantly increasing the training cost, the calculation of the convolutional layer is optimised to improve the inference time of the model prediction. The more popular lightweight backbone networks are SqueezeNet (Iandola *et al.*, 2016), ShuffleNet (Zhang *et al.*, 2018) series and MobileNet (Howard *et al.*, 2017) series and other algorithm structures. Among them, SqueezeNet adopts a well-designed structure of compression and expansion. The MobileNet series uses methods such as more efficient depthwise separable convolution and residual structure. The ShuffleNet series proposes the operation of channel shuffling, which further reduces the computational load of the model.

Although there is no object detection case for sandy vegetation, many researchers have studied the object detection technology of other plants. Tian *et al.* (2019a) improved YOLOV3 using DenseNet to propose a YOLOV3-dense-based apple detection algorithm. Birrell *et al.* (2020) proposed an object detection method for iceberg lettuce based on the YOLO network. Kestur *et al.* (2019) proposed a MangoNet based on a fully convolutional deep CNN architecture for mango detection. Williams *et al.* (2019) proposed a fully convolutional neural network based on VGG 16 Net for semantic segmentation of kiwifruit tree canopy images. Yu *et al.* (2019) studied various algorithms for lawn weed recognition based on deep convolutional neural networks and obtained DetectNet with the best results. Dyrmann *et al.* (2016) proposed a Fully CNN- based model for detecting individual weeds in wheat fields obscured by leaves.

Different from the normal farmland environment, whether it is the object detection of plants or fruits, the object detection of sandy vegetation faces many other challenges, such as the similar shape between vegetation types, the same colour of vegetation and background, and harsh operating environment. Therefore, we propose a lightweight sandy vegetation object detection algorithm based on an attention mechanism (Figure 1).

Materials and Methods

As shown in Figure 2, the object detection algorithm consists of four parts: backbone network, attention module, neck network, and head network. First, the algorithm extracts the features of the input image in the backbone network, then suppresses the irrelevant feature information through the attention module, then realizes the fusion of features in the neck network, and finally performs classification and regression prediction on the feature map through the detection head network.

Backbone

The design of the backbone network in this paper is based on the structure of ShuffleNetV2 (Ma *et al.*, 2018). As shown in Figure 3, this module divides the network into two branches before inputting feature maps, one branch is equally mapped, and the other branch undergoes three consecutive convolutions. The first convolution is followed by Batch Normalization and ReLU operations, the second convolution is followed by Batch Normalization, and the third convolution is also followed by Batch Normalization and ReLU operations. The input and output channels are kept consistent. Finally, the results of the two branches are combined through the Concat operation, and the channel shuffling operation is performed to ensure that the information is fully integrated.

The specific design of each layer of the backbone network is shown in Figure 4. The input of the backbone network is a 320×320 image, which first goes through a layer of convolution and maximum pooling operations, then goes through the ShuffleNet V2 module and outputs three feature maps with different scales. Finally, Conv5 convolution is added before the global pooling operation. Then, for the downsampling operation, directly set the convolution stride to 2, the feature map is halved, and the number of channels is multiplied. The Stage2 layer contains No. 0-3 modules. The design of module 0 consists of two branches, one branch is input into the ReLU function after two consecutive convolution normalization operations, and the other branch is subjected to the three consecutive convolution processes described above. No design is made for branch 1 in modules 1-3, and branch 2 still maintains the same operation as module 0, only modifying the convolution step size. Stage 3 and Stage 4 are also similar structures as above.

Attention module

In order to improve the detection effect of sandy vegetation objects in the same colour backgrounds, we introduce the SENet (Hu *et al.*, 2018) attention module. As a well-known attention mechanism, SENet mainly selects the more important feature channels according to the corresponding weights of different channels by mining the relationship between feature channels, that is, the focus of attention distribution, thereby improving the perfor-

mance of the model. By adding this module to the input and output parts of the backbone network, the algorithm can quickly learn the main information of interest and sup- press irrelevant information to improve the convergence and accuracy of the network.

As shown in Figure 5, after the input image size is mapped from h×w×c_1 to h×w×c_2, first we perform feature compression according to the spatial dimension. Turn each two-dimensional feature channel into a number that has a global receptive field to some extent, and the output dimension matches the number of input feature channels. That is, the statistical information between channels is generated through the global average pooling operation and the global information is compressed into channel information.

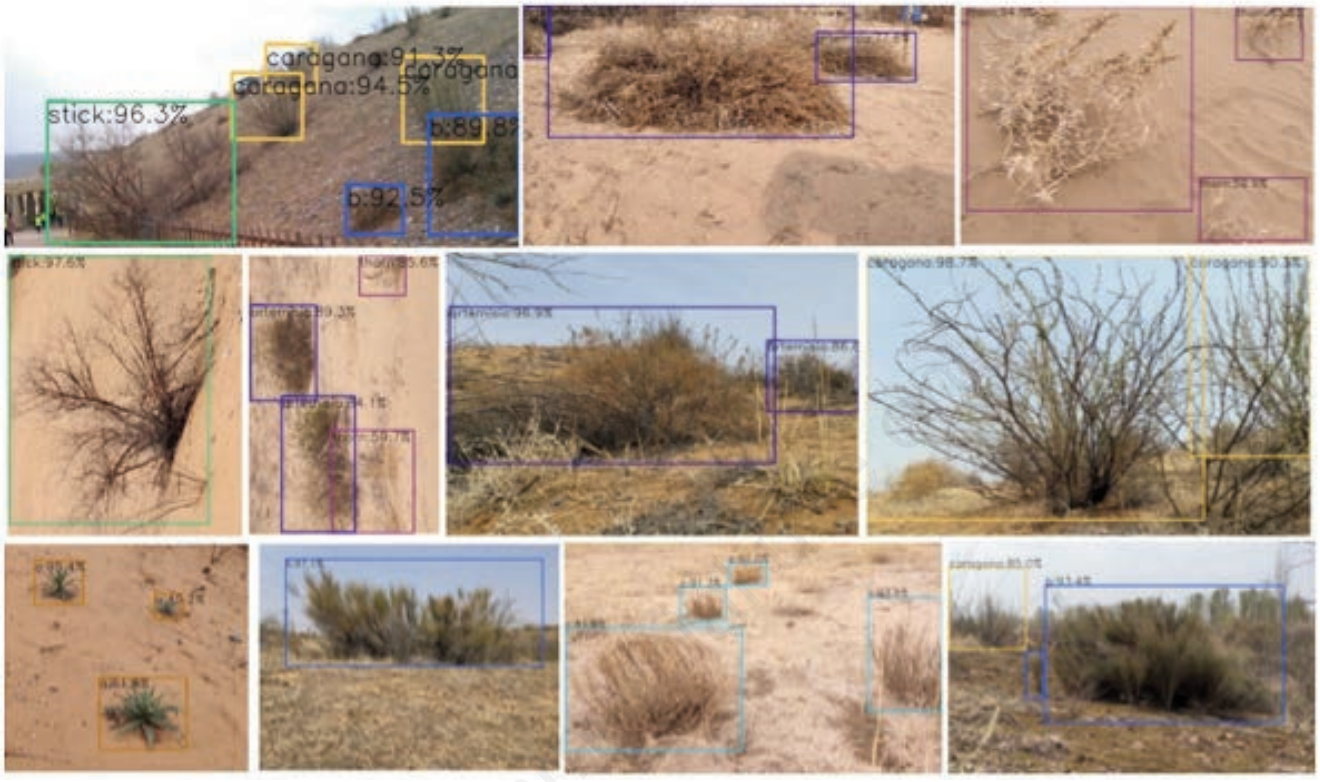

Figure 1. The object detection results of the proposed method for sandy vegetation.

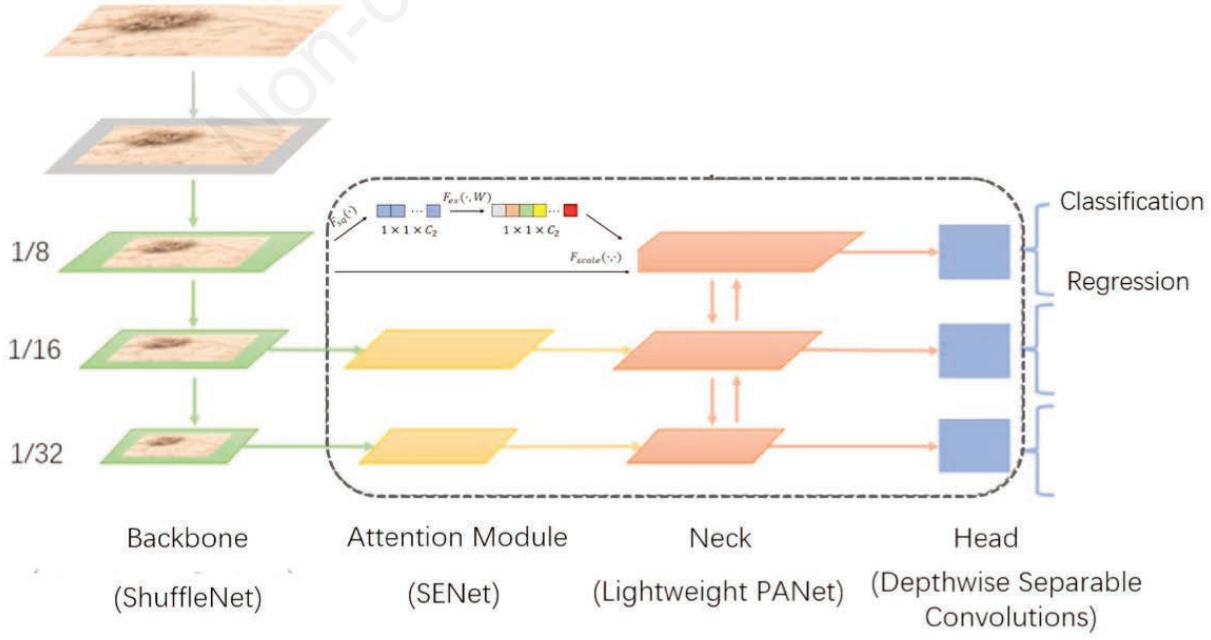

Figure 2. Structure of the proposed method.

Then, in order to better obtain the weight of each channel to the prediction result, the nonlinear relationship between channels is mined as much as possible through the sigmoid function, and the dimensionality reduction operation is realised through two fully connected layers. Finally, the weights of each feature channel after feature selection obtained at different scales are weighted to the previous features channel by channel through multiplication to complete the reweighting of the original features in the channel dimension.

Neck

After obtaining the feature maps of different scales output by the attention mechanism module, it is passed as input to the neck network. The design of the neck network refers to the structure of PANet (Path Aggregation Network) (Liu *et al.*, 2018) for semantic segmentation. As shown in Figure 6, the neck network contains two path-ways, top-down and bottom-up, and enhances the expressive ability of the entire network by transferring network features at different levels, which contains rich information. The feature map first realizes the alignment of feature channel dimensions through a 1×1 convolution kernel, and then uses linear interpolation to complete the process of upsampling and downsampling to ensure the lightweight of the model, removing redundant convolution operations.

Head

The head network learns the parameters for image classification and regression during the training process and returns the corresponding results during the prediction process. The design of the FCOS-based structure for lightweight optimization of the object detection head network structure is shown in Figure 7.

We remove the shared weight part of the FCOS detection head

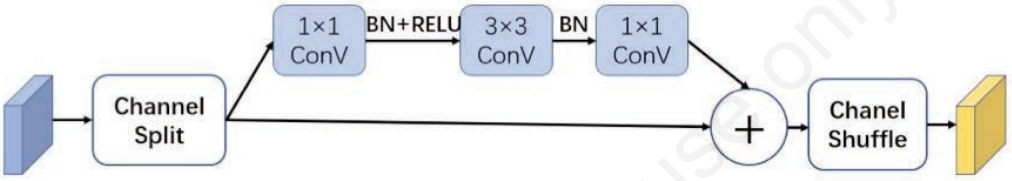

Figure 3. ShuffleNet V2 module.

Layer		Size	Kernel	Stride	Channel
Image		320 × 320	-	-	3
Conv1		160 × 160	3 × 3	2	24
MaxPool		80 × 80	3 × 3	2	
Stage 2	Stage2_0	40 × 40	$\begin{bmatrix}3\times3,\ 24\\1\times1,\ 58\end{bmatrix}\begin{bmatrix}1\times1,\ 58\\3\times3,\ 58\\1\times1,\ 58\end{bmatrix}$	$\begin{bmatrix}2\\1\end{bmatrix}\begin{bmatrix}1\\2\\1\end{bmatrix}$	116
	Stage2_x		$\begin{bmatrix}1\times1,\ 58\\3\times3,\ 58\\1\times1,\ 58\end{bmatrix}\times3$	1	
Stage 3	Stage3_0	20 × 20	$\begin{bmatrix}3\times3,\ 116\\1\times1,\ 116\end{bmatrix}\begin{bmatrix}1\times1,\ 116\\3\times3,\ 116\\1\times1,\ 116\end{bmatrix}$	$\begin{bmatrix}2\\1\end{bmatrix}\begin{bmatrix}1\\2\\1\end{bmatrix}$	232
	Stage3_x		$\begin{bmatrix}1\times1,\ 116\\3\times3,\ 116\\1\times1,\ 116\end{bmatrix}\times7$	1	
Stage 4	Stage4_0	10 × 10	$\begin{bmatrix}3\times3,\ 232\\1\times1,\ 232\end{bmatrix}\begin{bmatrix}1\times1,\ 232\\3\times3,\ 232\\1\times1,\ 232\end{bmatrix}$	$\begin{bmatrix}2\\1\end{bmatrix}\begin{bmatrix}1\\2\\1\end{bmatrix}$	464
	Stage4_x		$\begin{bmatrix}1\times1,\ 232\\3\times3,\ 232\\1\times1,\ 232\end{bmatrix}\times3$	1	
Conv5_		10 × 10	[1 × 1]	1	1024

Figure 4. Structure of the backbone network.

and replace the original ordinary convolution form with depthwise separable convolution. The depthwise separable convolution is divided into channel-by-channel convolution and point-by-point convolution. In the channel-by-channel convolution process, each convolution kernel only calculates one channel, and the number of convolution kernels remains the same as the number of channels in the previous layer. Assuming that there are M channel features and the size of the convolution kernel is $D_K \times D_K$, the parameter quantity after channel-by-channel convolution is $D_K \times D_K \times M$. Then perform pointwise convolution. At this time, the size of the convolution kernel is 1×1×M, which is aligned with the number of channels in the previous layer. The number of convolution kernels is N, and the operated feature maps are added in the channel direction. Finally, the parameter quantity of point-by-point convolution is M×N. The total parameter amount of the depthwise separable convolution is the accumulation of the above two parts. Compared with the ordinary convolution, the parameter amount is significantly reduced. As shown in Equation 1, the parameter amount of ordinary convolution is W times that of using depthwise separable convolution.

$$W = \frac{D_K \times D_K \times M \times N}{D_K \times D_K \times M + M \times N} \quad (1)$$

Loss function

This paper uses Generalized Focal Loss (GFL) (Li *et al.*, 2020) as the loss function, which has two specific forms: Quality Focal Loss (QFL) and Distribution Focal Loss (DFL). The QFL function jointly represents the classification score and the bounding box quality prediction score, which ensures the consistency of training and prediction. Based on Focal Loss, it changes the label information to a continuous value from 0 to 1.

$$QFL(\sigma) = -|y - \sigma|^{\beta}((1 - y)\log(1 - \sigma) + y\log(\sigma)) \quad (2)$$

where $|y - \sigma|^{\beta}$ represents the scale factor part, the representation is the power function of the absolute distance between σ and y, and β is a hyperparameter generally set to 2.

The DFL function optimises the probability of the two positions closest to the label y, one left and one right so that the network can quickly focus on the distribution of the adjacent areas of the object position.

$$DFL(S_i, S_{i+1}) = -((y_{i+1} - y)\,log(S_i) + (y - y_i)\,log(S_{i+1})) \quad (3)$$

QFL and DFL can be jointly expressed as one formula, assuming that the predicted probabilities of values y_l and y_r are p_{yi} and p_{yr}, respectively, the final predicted result is $\hat{y} = y_l p_{y_l} + y_r p_{y_r}$ the GT label is y, and $|y - \hat{y}|^{\beta}$ is used as a scaling factor, then FGL can be expressed as formula 4:

$$GFL(p_{y_l}, p_{y_r}) = -|y - (y_l p_{y_l} + y_r p_{y_r})|^{\beta}((y_r - y)\log(p_{y_l}) + (y - y_l)\log(p_{y_r})) \quad (4)$$

Experiments

Datasets

In order to verify the effectiveness and advancement of the algorithm in this paper, we went to the sandy field to collect 500 images of common sandy plants and marked them as a sandy vegetation dataset. In addition, we augmented the dataset with Mosaic (Bochkovskiy *et al.*, 2020) data augmentation techniques and finally formed 4000 annotated sandy plant images. That is, based on scaling and randomly cropping a single image, the four separately processed images are stitched together into a new image. This can further enrich the feature information and scale of the dataset. Finally, we set 3500 images as the training set and the remaining 500 images as the test set.

Implementation details

In this paper, we will not only use the representative two-stage object detection algorithm Faster-RCNN (Ren *et al.*, 2015), the one-stage object detection algorithm YOLOv3 (Redmon and Farhadi, 2018), and anchor-free object detection algorithm CenterNet (Duan *et al.*, 2019) and FCOS (Tian *et al.*, 2019b) as the comparison groups, but also compare the two currently popular lightweight object detection algorithms YOLOv3-Tiny (Redmon and Farhadi, 2018) and NanoDet (RangiLyu, 2021). The experimental hardware is a computer with an i7 processor, 32G memory, and a 3070 graphics card.

Results

The specific performance of the algorithm is shown in Table 1, where mAP represents the average accuracy of algorithm detection, the model size represents the memory size of the trained model, and time cost represents the model's prediction and inference time for a picture. Although the detection accuracy of the

Table 1. Comparison results of various object detections.

Methods	mAP	Size of the model	Time cost(s)
Faster-RCNN	0.8130	330.5	1.76262
YOLOv3	0.7332	492.9	0.2017
CenterNet	0.7532	115.6	1.4047
FCOS	0.7540	257.6	0.2043
YOLOv3-Tiny	0.5802	33.4	0.1452
NanoDet	0.7259	3.2	0.0310
Ours (without SENet)	0.7388	3.4	0.0206
Proposed method	0.7648	3.5	0.0277

Faster-RCNN algorithm is the highest, at the same time, the speed is also the slowest due to the two-stage inference. On the other hand, the YOLOv3 algorithm model requires the largest memory. Compared with the above two methods, the anchor-free algorithm has a relatively small space and memory, which is very suitable for the requirements of embedded devices or mobile robots. It can be clearly seen from Table 1 that the advantages of a lightweight network are obvious; not only is the memory required very small, but the real-time performance is also good. Among them, NanoDet has the smallest memory, but under the premise that the memory occupied by our algorithm remains similar to that of NanoDet, the accuracy and inference time are ahead of NanoDet and YOLOv3-tiny. In addition, ablation experiments show that adding an SE attention module can effectively improve the accuracy of sandy vegetation object detection while slightly increasing the consumption of memory and reasoning time.

As shown in Figure 8, the desert vegetation recognition accuracy of the YOLOv3-tiny algorithm is low. Missing objects occurred in all 4 images, and false detections occurred in the third image. The NanoDet algorithm misses detection in the fourth image and also has false detection in the third image. In contrast, our proposed object detection algorithm has the best recognition accuracy of sandy vegetation.

Conclusions

The network model in this paper is designed to reduce the amount of model parameters by lightweight design of the anchor-free object detection algorithm model, thereby reducing the model inference time and the memory cost. The detection performance of the algorithm no longer depends on the setting of hyperparameters such as anchor box size, aspect ratio and quantity information, and can be combined with multi-scale network structures. It can also be well adapted to candidate objects with large changes in shape, and there is no need to set the shape and size of the anchor box for different detections, which further improves the generalisation ability of the detection algorithm. On this basis, the model accuracy is further improved through the attention mechanism. At the same time, it ensures the efficiency and accuracy of detecting sandy vegetation in desert environment.

References

Birrell S., Hughes J., Cai J.Y., Iida F. 2020. A field-tested robotic harvesting system for iceberg lettuce. J. Field Robot. 37:225-45.

Bochkovskiy A., Wang C.Y., Liao H.Y.M. 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Duan K., Bai S., Xie L., Qi H., Huang Q., Tian Q. 2019. Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, 6569-78.

Dyrmann M., Karstoft H., Midtiby H.S. 2016. Plant species classification using deep convolutional neural network. Biosyst. Eng. 151:72-80.

Girshick R. 2015. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 1440-8.

Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Hu J., Shen L., Sun G. 2018. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 7132-41.

Iandola F.N., Han S., Moskewicz M.W., Ashraf K., Dally W.J., Keutzer K. 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360.

Kestur R., Meduri A., Narasipura O. 2019. Mangonet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. Eng. Appl. Artific. Intell. 77:59-69.

Law H. Deng J. 2018. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European conference on computer vision (ECCV), 734-50.

Li X., Wang W., Wu L., Chen S., Hu X., Li J., Tang J., Yang J. 2020. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Adv. Neur. Informat. Proc. Syst. 33:21002-12.

Liu S., Qi L., Qin H., Shi J., Jia J. 2018. Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 8759-68.

Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y., Berg A.C. 2016. Ssd: Single shot multibox detector. In: European conference on computer vision. Springer. 21-37.

Ma N., Zhang X., Zheng H.T., Sun J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV), 116-31.

Michael C., Charles H., James R., Stefan S., Graham V.M., 2018. World atlas of desertification.

RangiLyu 2021. Nanodet: Super fast and high accuracy lightweight anchor-free object detection model. Available from: https://github.com/RangiLyu/nanodet.

Redmon J., Farhadi A. 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren S., He K., Girshick R., Sun J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neur. Informat. Proc. Syst. 28.

Tian Y., Yang G., Wang Z., Wang H., Li E., Liang Z. 2019a. Apple detection during different growth stages in orchards using the improved yolo-v3 model. Comput. Electron. Agric. 157:417-26.

Tian Z., Shen C., Chen H., He T. 2019b. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF international conference on computer vision, 9627-36.

Williams H.A., Jones M.H., Nejati M., Seabright M. J., Bell J., Penhall N.D., Barnett J.J., Duke M.D., Scarfe A.J., Ahn H.S., 2019. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. Biosyst. Eng. 181:140-56.

Yu J., Sharpe S.M., Schumann A.W., Boyd N.S. 2019. Deep learning for image-based weed detection in turfgrass. Eur. J. Agron, 104:78-84.

Zhang H., Yu H., Yan Y., Wang R. 2022. Gated domain-invariant feature disentanglement for domain generalizable object detection. arXiv:2203.11432v1

Zhang X., Zhou X., Lin M., Sun J. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 6848-56.