

Improving Apple Detection Using RetinaNet



Zhen Ma  and Nianqiang Li 

Abstract With the rapid modernization of agriculture and the increasing demand for fruits, automated fruit picking tasks are particularly important in fruit production. Computer vision-based fruit target detection is one of the key technologies. Traditional fruit detection methods are limited by the fact that the digital images captured by cameras are susceptible to light, and there may be overlap and occlusion between the fruit and the leaves, which are common situations that can greatly affect traditional fruit detection methods. With the development of deep learning techniques, many target detection techniques have emerged. To improve the accuracy and detection speed of fruit detection, this paper adopts and improves RetinaNet, using MobileNetV3 as one of the feature extraction network, which greatly reduces the inference time of detection models in embedded devices. In order to improve the detection accuracy of the small target of fruits, this paper makes some improvements to the feature extraction network and feature pyramid network in the network, and optimizes the size of anchors with a clustering algorithm. Through experiments, it is shown that the improved RetinaNet algorithm proposed in this paper has high accuracy in apple detection task and better robustness in dark light, overlapping and occlusion situations.

Keywords RetinaNet · MobileNetV3 · Apple detection

1 Introduction

China is a large fruit producing country, and the use of automated fruit picking equipment can greatly reduce human and material resources, and an important prerequisite for this mechanical task is to have accurate machine vision-based fruit detection and positioning technology. Conventional inspection methods identify fruits based on fruits' features such as shapes, colors and textures [1–4]. The detection speed of these methods is usually fast, but the pictures taken by the camera are not ideally

Z. Ma · N. Li (✉)
University of Jinan, Jinan 250022, China
e-mail: ise_linq@ujn.edu.cn

images that contain only complete fruits, and often encounter complex situations such as overlapping fruits, leaves obscuring fruits, and large differences between individual fruits, which can make traditional detection methods very difficult.

Deep learning is one of the frontier technologies in machine learning and artificial intelligence research, and deep learning techniques have brought revolutionary advances in machine learning and computer vision. Target detection refers to separating the background region from the region of interest and determining the classes and location of the region of interest in the input of an unknown image. In recent years, due to the breakthrough of deep learning technology in target detection [5, 6], there have been many scholars applying this technology to fruit detection. The current target detection methods based on deep convolutional networks are mainly divided into two categories, one category is two-stage detection algorithms such as RCNN [7], Fast-RCNN [8], Faster-RCNN [9], Mask-RCNN [10]; the other category is one-stage detection algorithms such as SSD [11], YOLO [12–14], RetinaNet [15].

Bargoti et al. [16] used Fast-RCNN network for fruit detection, and to reduce the computational effort, they split the original high-resolution image and detected each piece separately. They also used flip shift and color space transformation to enhance the dataset and reduce the occurrence of overfitting cases. However, Fast-RCNN is a two-stage detection algorithm, which has a long detection time and is not optimized for small targets. Tian et al. [17] used a modified YOLO-V3 network for apple detection, and modified YOLO-V3 using DenseNet to improve the feature extraction capability of the network.

Compared with the two-stage detection algorithms, the one-stage detection algorithms skip the region proposal stage and complete the prediction of target class and target localization simultaneously in the convolutional network, which greatly speeds up the detection speed and are more suitable for real-time detection of fruits. However, one-stage detection algorithms usually use the mechanism of dense sampling of candidate regions, which can lead to the occurrence of category imbalance [18], i.e., the quantity of negative samples is much larger than the quantity of positive samples, making the training process affected and thus reducing the accuracy of target detection.

In RetinaNet, He et al. [15] proposes a new loss function, Focal Loss, which is characterized by its small impact on the loss function for easy samples and still maintains a high loss for hard samples, thus making the training process more stable and improving the efficiency and accuracy of the detector.

In summary, this paper adopts RetinaNet as the target detection framework and uses MobileNetV3 as its feature extraction network for the application scenario of this paper. In order to improve the accuracy of small target detection, this paper improves the RetinaNet network structure by making a series of modifications to the feature extraction network and FPN so that the low-level features of the image samples have better semantic information of the high-level features. The anchors parameters in the original RetinaNet are not applicable to the application scenario of this paper, so a better anchors parameter is calculated by the K-means distance algorithm to improve the detection accuracy and increase the recall.

The algorithm proposed in this paper takes into account the better accuracy while completing the lightweight of the model, so that the embedded device can complete the real-time orchard apple detection in real time and be fully prepared for the subsequent fully automated fruit picking task.

2 RetinaNet Structure and Improvement

RetinaNet is a unified target detection network consisting of a feature extraction network, a feature pyramid network and two sub-networks. It improves the accuracy of target detection, especially in the detection of small objects. This paper improves on it, and its network structure is shown in Fig. 1. The backbone network mainly obtains the feature map of the whole input image through a series of convolutional operations. two sub-networks classify and localize the target image to be detected based on the output of the backbone network, respectively.

2.1 Feature Extraction Network

MobileNets [19] is based on a streamlined architecture that uses deeply separable convolutions to build lightweight deep neural networks. The network introduces two simple global hyper parameters that effectively balance between latency and accuracy. In this paper, MobileNetV3 is used as the base feature extraction network

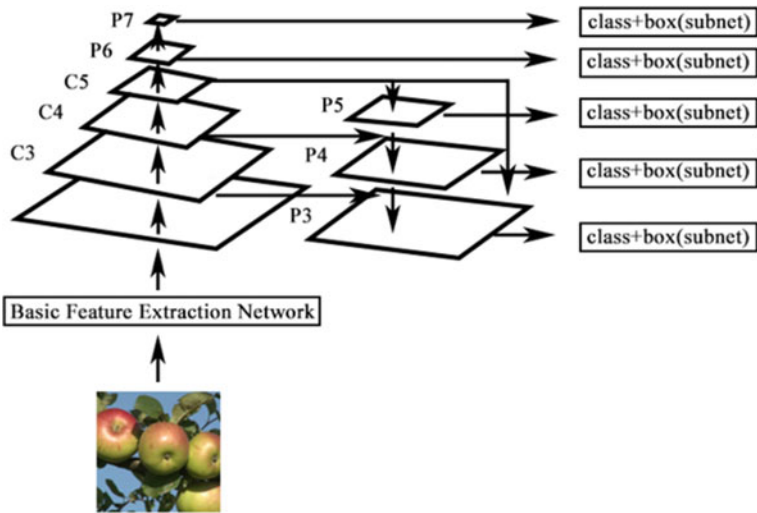


Fig. 1 Overall structure of improved RetinaNet

of RetinaNet for practical application scenarios to shorten the inference time of the model and to achieve the real-time detection task of apples in orchards using embedded devices.

2.2 *Feature Pyramid Network*

Feature Pyramid Network (FPN) [20] has been a fundamental component in multi-scale target detection and can cope well with target detection tasks of different sizes. The high-level features of image samples contain rich semantic information, but it is difficult to predict the location of the target accurately due to low resolution. In contrast, the low-level features of image samples have less semantic information, so that they can accurately contain the location information of objects due to their high resolution. According to this feature, FPN fuses the feature maps of different layers, enabling better recognition of small objects. However, in the FPN module of RetinaNet, after multiple convolution and upsampling operations, the semantic information of the higher-level features of the image samples is difficult to reach the lower-level feature layers, making the lower-level features used to detect small objects lack some semantic information of the higher-level features.

In the application scenario of this paper, for the problem of low detection accuracy in the detection task of small targets such as apples, the C5 feature layer of the feature extraction network is stacked with the P3 feature layer of the FPN after 4 times upsampling, and the stacked feature channels are compressed and fused to the original number of feature channels by a 1×1 convolutional layer to improve the semantic information of the high-level features contained in the low-level features of the image samples.

2.3 *Focal Loss*

In the process of object detection algorithm training, there is a class imbalance problem, the most serious of which is the positive and negative sample imbalance, i.e., the number of negative samples is often larger than the number of positive samples. In some two-stage object detection methods, like Faster R-CNN, a significant portion of negative samples is first filtered out using RPN, and then a deep neural network is used to make accurate category detection and position regression for each candidate frame. In contrast, one-stage target detection methods do not distinguish between positive and negative samples in advance, and directly performs category detection and position regression on the pre-set anchors. So, in general one-stage detection algorithm is faster without high accuracy. To solve this problem, He et al. mentioned Focal Loss in the RetinaNet algorithm.

The traditional cross-entropy loss function is shown as follows:

$$CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1 - p) & \text{otherwise} \end{cases} \quad (1)$$

where $y \in \{\pm 1\}$ refers to the manually labeled classes and $p \in [0, 1]$ is the probability that the model predicts the classes $y = 1$.

For simplicity, we let

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \quad (2)$$

Then we can obtain the formula as follows:

$$CE(p, y) = \log(p_t) \quad (3)$$

To solve the problem of classes imbalance caused by the number of negative samples being much larger than the number of positive samples, we can introduce a weighting factor α , α is defined as follows:

$$\alpha = \begin{cases} \alpha & \text{for class } 1 \\ 1 - \alpha & \text{for class } -1 \end{cases} \quad (4)$$

We can treat α as a hyper parameter and calculate the optimal value by cross-validation method. The loss function then becomes as follows.

$$CE(p_t) = -\alpha_t \log(p_t) \quad (5)$$

In order to better distinguish between easy examples and hard examples, the detector focuses more on hard examples.

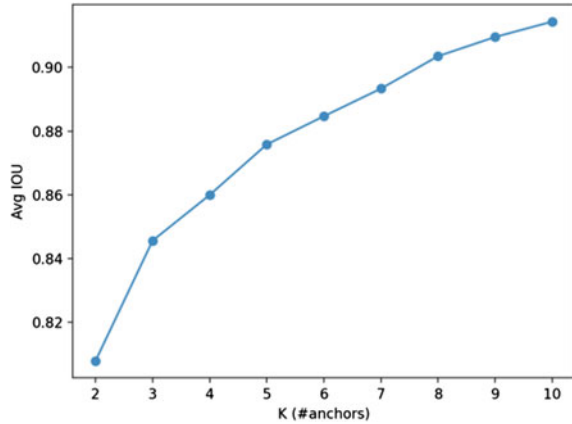
Focal Loss introduces a tunable focusing parameter ($\gamma > = 0$), and the final formula for Focal Loss is shown below.

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t) \quad (6)$$

For the dataset used in this paper, we set γ to be 2 to get better results for the trained model.

2.4 Improving Anchor Using Clustering Algorithm

By using the anchors mechanism [9], the computational effort in the training phase is greatly reduced. Since the setting parameters of anchors can vary between different datasets, the parameters of anchors are recalculated by using the K-means clustering algorithm for the application scenario of this paper, which make the model of this

Fig. 2 Clustering result

paper work better on the fruit dataset. In this paper, the average IOU in each case is calculated by using the K-means distance algorithm between K belonging to (2,10) anchors, and the calculation results are shown in Fig. 2.

It can be seen that the slope of the anchors-average IOU curve changes significantly when the number of anchors is 3. Therefore, when the size of 3 anchors is selected, the complexity of training can be reduced while the accuracy of the model can be taken into account. At this time. The three anchors' sizes are $[27 \times 27, 36 \times 36, 46 \times 46]$. The anchors sizes obtained by clustering the ground truth bounding box are closer to the true values, which make it easier to fit the model to the true position of the target, thus reducing the training difficulty of the model.

3 Experimental Results and Analysis

3.1 Experimental Data

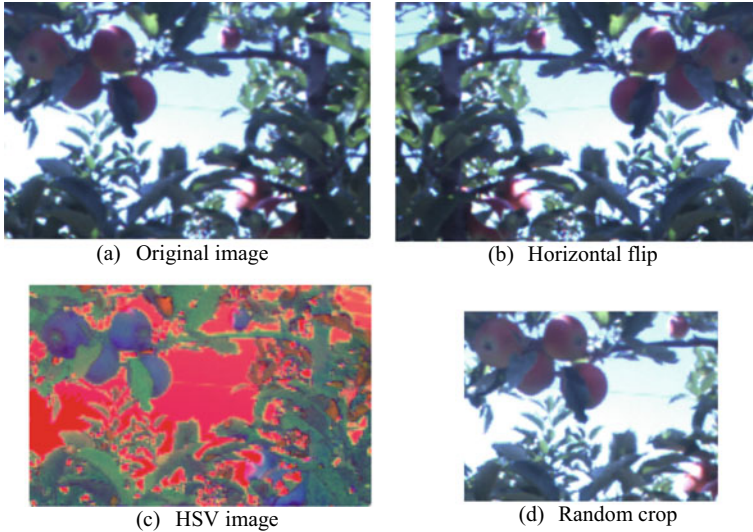
The dataset used in this paper is published by Suchet Bargoti and James Underwood in [21] and can be downloaded at [22]. This paper uses its apple dataset. The dataset provides circular annotations for the fruits, which is converted into a rectangular box representation containing four vertices to better fit the network parameters of this paper (Table 1).

3.2 Data Augmentation

Since this dataset has only more than 1000 apple images, it is easy to overfit during the training process. For this reason, we used the following methods to augment the

Table 1 Apple dataset parameters

Set	Raw image size	Image size	Number of image
train	1616 × 1232	202 × 308	896
val	1616 × 1232	202 × 308	112
test	1616 × 1232	202 × 308	112
train + val	1616 × 1232	202 × 308	1008

**Fig. 3** Data augmentation

dataset to enhance the robustness of the model. (1) Convert all images to HSV color space to enhance the contrast between foreground and background in apple images. (2) Flip all images horizontally to expand the original dataset by a factor of 2. (3) Randomly crop the images during training by randomly cropping 60–90% part and scaled to the size needed by the network. After the above operations, the dataset is expanded to 3 times of the previous size, which greatly reduces the occurrence of overfitting and enhances the generalization of the model. Figure 3 shows the images after data augmentation.

3.3 Evaluation Criteria

In this paper, F1-score is used as the evaluation index of the target detection model. F1-score is the harmonic average of *Recall* and *Precision*. The formulas of *Recall* and *Precision* are as follows:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

Among them, TP is True Positives, which means that the sample is divided into positive samples and the allocation is correct. FP is FN , that is, False Negatives, which means that the sample is divided into negative samples but the allocation is wrong. FN is False Negatives, which means that the sample is divided into negative samples but the allocation is wrong. Thereby, the calculation formula of F1-score is obtained, as shown below:

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall} \quad (9)$$

The higher the F1-score, the more robust the model.

3.4 Experimental Results

In this paper, we conduct comparison experiments by using different detection algorithms on the same datasets, and the experimental results are shown in Table 2. By improving the RetinaNet network architecture and using MobileNetV3 as its feature extraction network, the detection speed is greatly improved, from 8 to 37FPS, and the F1-score is also improved by 20%.

Figure 4 shows the detection results of the improved algorithm for some test samples. In these test result plots, some common cases that are unfavorable to Apple detection are shown. For example, the picture in the upper left corner in Fig. 4, the cyan colored apples are extremely similar in color to the large green leaves in the background. In the top right image, the light is blocked and in dim light, a similar situation to the top left picture occurs, where the apples largely blend in with the background. In the two pictures at the bottom of Fig. 4, there is an overlap between apples and the leaves obscure the apples. Although these common unfavorable situations above can bring great impact on the target detection, however, the improved RetinaNet in this paper still plays a good effect, not only detecting the apples in the images, but also accurately labeling their positions in the images.

Table 2 Experimental results

Model	Backbone	F1
Faster-RCNN	ResNet50	0.878
RetinaNet	ResNet50	0.865
Proposed	MobileNetV3	0.946

Fig. 4 Detection result

3.5 Experimental Analysis

The original RetinaNet uses Focal Loss as the loss function, and its F1-score is not much different from Faster-RCNN under the premise of guaranteeing the detection speed, but as a two-stage detection algorithm, the Faster-RCNN model is relatively large and not well able to accomplish the real-time target detection task. Therefore, this paper adopts the lightweight RetinaNet detection model and uses MobileNetV3 as its feature extraction network, which is faster and can complete the real-time apple detection task. For the detection of the small target of apple, the detection accuracy is greatly improved by improving the RetinaNet network structure, which makes the semantic information of the high-level features well integrated with the low-level features, and by using the K-means clustering algorithm to calculate the size of the anchors suitable for this dataset. For complex and common unfavorable cases, such as dark light, overlap and occlusion, the algorithm in this paper still has excellent performance. However, this algorithm also has some shortcomings, such as some detection targets are missed. In the next work, we will continue to improve this model and add other datasets to further improve the robustness and accuracy of the detection model.

References

1. Wang, Z., Walsh, K. B., Verma, B.: On-tree mango fruit size estimation using RGB-D images. *Sensors* **17**(12), 2738 (2017)
2. Payne, A.B., Walsh, K.B., Subedi, P.P., Jarvis, D.: Estimation of mango crop yield using image analysis–segmentation method. *Computers Electron Agric* **91**, 57–64 (2013)
3. Nanaa, K., Rizon, M., Abd Rahman, M.N., Ibrahim, Y., Abd Aziz, A. Z.: Detecting mango fruits by using randomized hough transform and back propagation neural network. In: 2014 18th International Conference on Information Visualisation, pp. 388–391. IEEE (2014)
4. Rizon, M., Yusri, N. A. N., Kadir, M. F. A., bin Mamat, A. R., Abd Aziz, A. Z., Nanaa, K.: Determination of mango fruit from binary image using randomized Hough transform. In: Eighth International Conference on Machine Vision (ICMV 2015), vol. 9875, pp. 987503. International Society for Optics and Photonics (2015)
5. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Object detection with deep learning: a review. *IEEE Trans, Neural Networks Learn. Syst.* **30**(11), 3212–3232 (2019)
6. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: a survey. *Int. J. Comput. Vision* **128**(2), 261–318 (2020)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
8. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint [arXiv:1506.01497](https://arxiv.org/abs/1506.01497) (2015)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
11. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer, Cham (2016)
12. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
13. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
14. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767). (2018)
15. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the Ieee International Conference on Computer Vision, pp. 2980–2988 (2017)
16. Bargoti, S., Underwood, J.P.: Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.* **34**(6), 1039–1060 (2017)
17. Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z.: Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **157**, 417–426 (2019)
18. Oksuz, K., Cam, B. C., Kalkan, S., Akbas, E.: Imbalance problems in object detection: a review. *IEEE Trans. Pattern Anal. Machine Intell.* (2020)
19. Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., Adam, H.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)
20. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)

21. Bargouti, S., Underwood, J.: Deep fruit detection in orchards. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3626–3633. IEEE (2017)
22. ACFR FRUIT DATASET: <http://data.acfr.usyd.edu.au/ag/treecrops/2016-multifruit/>. Last accessed 8 May 2021