



A Real-Time Detection Framework for On-Tree Mango Based on SSD Network

Qiaokang Liang^{1,2,3}, Wei Zhu^{1,2,3}, Jianyong Long^{1,2,3}(✉),
Yaonan Wang^{1,2,3}, Wei Sun^{1,2,3}, and Wanneng Wu^{1,2,3}

¹ College of Electrical and Information Engineering,
Hunan University, Changsha 410082, China

{Qiaokang, Zxdzhuwei, lji_cv, yaonan, wei_sun,
wuwanneng}@hnu.edu.cn

² Hunan Key Laboratory of Intelligent Robot Technology in Electronic
Manufacturing, Hunan University, Changsha 410082, China

³ National Engineering Laboratory for Robot Vision Perception and Control,
Changsha 410082, China

Abstract. On-tree fruit detection in orchards is important for yield estimation, mapping and automatic harvesting in modern agriculture. This paper proposes a real-time detection framework for on-tree mango based on SSD (Single shot Multi Box Detector) network, a state-of-the-art object detection algorithms based on deep learning. The mango image dataset used in this paper was gathered from outdoor mango orchards. Firstly, the dataset was annotated and converted to a trainable dataset for SSD network. Secondly, the author designed new sampling strategies and image distortions at the image pre-processing stage to optimize data augmentation techniques. Moreover, the default box proposal methods of SSD network were improved by redesigning the shapes of default boxes on multiple feature maps according to our own dataset. Finally, to explore which classification network is most suitable for mango detection, an experiment was presented to compare the detection performance of SSD network with the VGG16 and ZFNet as base network respectively. Almond dataset was also used to verify our proposed method. Experimental results demonstrated that, with optimization of data augmentation techniques and default box proposals, our improved VGG16-based SSD network can achieve higher performance than Faster R-CNN in on-tree mango detection, with F1 score of 0.911 at 35 FPS for 400×400 input image, which is a real-time detection.

Keywords: Mango · Detection · SSD · Deep learning

1 Introduction

Nowadays, on-tree fruit detection in orchards for yield estimation and mapping plays a more and more important role in modern agriculture with the rapid development of computer vision techniques. Compared to manual counting, vision-based automated detection methods are more efficient and save human resource [1]. With the help of computer vision technology, we can get more accurate information about on-tree fruit, like its size, maturity, location and so on. Once obtain this information, we can estimate

crop yield and harvest automatically using robots [2]. In this paper, we focus on on-tree mango detection in outdoor orchard.

There are many prior works related to fruit detection. Handcraft features was used to encode visual attributes and extract features for mango detection in [3]. In fact, feature encoding is unique to a specific dataset and image quality. Based on computer vision techniques, segmentation technique was utilized in [1] to segment the fruit region from the input image and counted fruits. This approach is tested on apples images with large size and colorful appearance collected from different websites. Fusion data of multiple sensors and machine vision algorithms were employed for feature detection and extraction in [4].

Recently, deep convolutional neural networks (CNNs) based methods have achieved the state-of-the-art results for object detection [5–7], both with high accuracy and detection speed.

The dataset we used was released by [8] and contained more than 1,900 images with on-tree mangoes in outdoor orchard. There are many challenges with our mango dataset: (1) there are so many mangoes that are blocked by branches, leaves or mangoes. (2) Most images are pretty dark and mangoes are similar to the background. (3) Compared to an input image, with the resolution of 500×500 , mangoes are too small and the resolution of the largest is about 70×70 . These reasons made it more difficult to detect on-tree mangoes with traditional methods. Figure 1 presents examples.

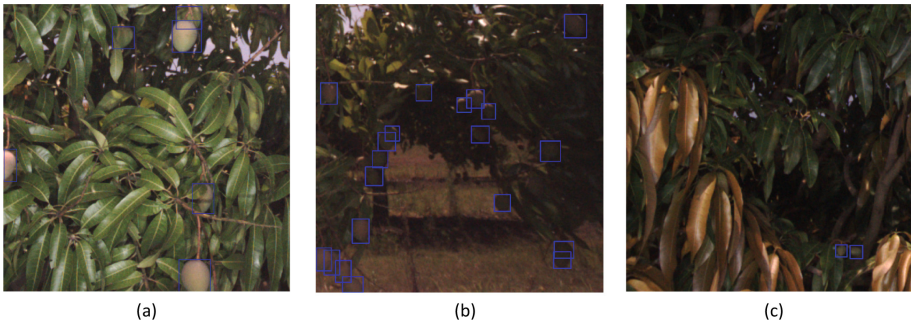


Fig. 1. Examples of mango images. We annotated the ground truth boxes of mangoes in blue. Image (a) shows the instances that mangoes are blocked by leaves or mangoes. Image (b) shows the instances of dark images in which mangoes are pretty similar to background and mangoes overlapping between each other. Mangoes are too small in image (c). (Color figure online)

To address these issues, we applied deep-learning-based SSD network [7] to detect on-tree fruits. Our main contributions are summarized as follows:

- i. We deployed the state-of-the-art object detection framework, SSD, for on-tree mango detection in orchards and got high performance both in the aspects of accuracy and speed.
- ii. We demonstrated how to optimize data augmentation techniques and default box proposals of SSD network to improve detection performance.

- iii. We relabeled the dataset and transformed it to a trainable dataset in VOC2007 style. The updating dataset is available here: https://pan.baidu.com/s/1pdTyVq9PIbhhkR2k_4Tl5zA.

The rest of the paper is organized as follows: Sect. 2 introduces the related works for object detection. Section 3 describes the methodology details. The details of our experiment are in the Sect. 4 and the results are in the Sect. 5. We make a conclusion in Sect. 6.

2 Related Works

Over the last decade, fruit detection has utilized hand engineered features to extract features related to the object in original input image for classification and position [3, 4]. This type of feature extraction method that requires human intervention heavily depends on prior knowledge of the feature designer and is inefficiency. Moreover, it's difficult to adapt to various datasets gathered from complex environments in reality.

More recently, with the development of deep learning technology, deep convolutional neural networks (CNNs) have been successfully applied to the field of image classification, detection and so on. The advantage of CNNs is that they can automatically extract features from the input image through self-learning. Deep-learning-based object detection model can extract more abundant features and its feature expression ability is stronger.

At present, the object detection methods based on deep learning are mainly divided into two categories: candidate region based models and regression based models. The first appeared is detection model based on region candidates, in which the candidate regions are extracted from the detection region preparing for subsequent procedures of feature extraction and classification. Typical representatives are: R-CNN [9], SPP-net [10], Fast R-CNN [11], Faster R-CNN [5], and R-FCN [12]. All the above models have been developed on the basis of the previous generation, and the accuracy is also getting higher. But there is still much room for improvements in terms of speed.

So there comes the regression-based detection model. It is necessary to delineate the default box in a certain way in advance so that the relationship between the prediction box, the default box, and the ground truth object box can be established and used for training. Typical representatives are: YOLO [13] and SSD, which is faster than R-CNN series method. Although the speed has been greatly improved, reaching 45 FPS, YOLO has relatively large classification and positioning errors, and its generalization ability is also weak. YOLOv2 [6] has made great improvements in aspects of data input, network structure and positioning methods and is better than YOLO. SSD draws on the advantages of both YOLO and Faster-RCNN methods, and gets higher performance than both, with mAP reaching 74.3% at 59 FPS on VOC2007 dataset.

Previously, object detection model based on deep learning has been applied to fruit detection. Faster R-CNN was employed to detect fruits in trees in orchard [8]. With the use of transfer learning and data augmentation, [8] get high accuracy on apple and mango detection. Similar to this, Faster R-CNN was also used to detect mangos in [14]. Different from [8], Faster R-CNN used in [14] combined with a novel multi-sensor

framework and a multiple viewpoint approach and got satisfied result on yield estimation. Due to the limitations of Faster R-CNN itself, the detection accuracy and speed have yet to be improved.

In this paper, we applied SSD network to detect on-tree mangoes in outdoor orchard. With the high-performance SSD algorithm, our mango detection can be more accurate and faster.

3 Methodology

3.1 Network Architecture

SSD network uses regression methods for detection, putting the two tasks of positioning and classification into one network. Besides, Lots of improvements were added to increase accuracy and speed. On the one hand, compared with Faster R-CNN, SSD network does not resample pixels or features for bounding box hypotheses, which makes a great improvement in the aspect of detection speed. On the other hand, SSD implemented the anchor method of extracting candidate area of Faster R-CNN on feature maps of various scale and made it more effective and accuracy to detect objects of various scales. Figure 2 illustrated the SSD network architecture.

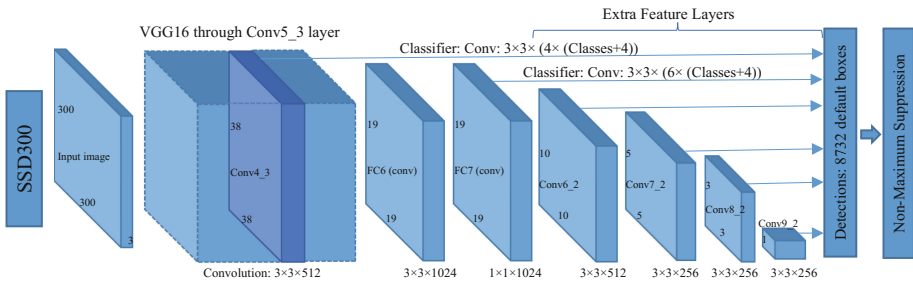


Fig. 2. SSD network architecture. The VGG16 network [15] is used as base network and its fully connected layers (FC6, FC7) were converted into convolutional layers. Besides, SSD designed 4 extra feature layers connected to the end of VGG16. Each extra convolutional layer output a feature map and used as an input for prediction. These extra layers together with Conv4_3 and FC7 layer predict the offsets to default boxes of various scales and aspect ratios and their associated confidences by small convolutional filters. When the resolution of the input image is 300×300 , we called the network SSD300.

Figure 3 shows a brief description of the mango detection process. The input image passes through multiple convolution layers to obtain feature maps. At each location of the feature map, the SSD network generates 4 or 6 default boxes and then uses a series of small convolutions to predict. By setting the overlap rate threshold, a non-maximal suppression (NMS) method is used to remove duplicate bounding boxes. Finally, we only keep the bounding boxes that have greater confidence than the threshold, so we get the final detection results.

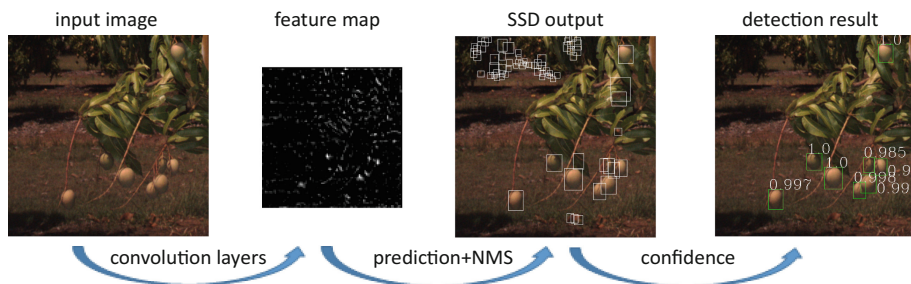


Fig. 3. Forward process of mango detection.

3.2 Data Augmentation

Sampling Strategies. Data augmentation used in [8] has improved fruit detection performance greatly with Faster R-CNN network. As for SSD, the network itself uses a more extensive sampling strategy for data augmentation, which improves 8.8% mAP. The sampling strategies of SSD are described as following [7]:

- Use the entire original input image.
- Sample a patch so that the minimum jaccard overlap with the objects is 0.1, 0.3, 0.5, 0.7, or 0.9.
- Randomly sample a patch.

The size of each sampled patch and minimum jaccard overlap are important, especially for the small object detection. After passing through a series of convolution and pooling layers of VGG16 network, the size of feature map becomes 1/16 of the size of the original image. It's harmful for our mango detection, because the sizes of most mangoes are smaller than 64×64 pixels, which is 4×4 pixels after passing through VGG16. So if we set a small sample ratio and minimum jaccard overlap, mangoes can be better highlighted after the sampled patch being reset to a fixed size.

First of all, we calculated the distribution of mangoes' shapes on the training dataset and found out the extreme values of mangoes' sizes. Then we set the minimum and maximum sampling ratio according to the extreme values and the sizes of the input images. We designed a total of six groups of sampling ratios. Each group has the same minimum and maximum sampling ratios in general, but the overlap thresholds are different.

Table 1 summarized our configurations of sampled patch. Min ration and max ratio are the minimum and maximum sampling ratio respectively and overlap is the minimum jaccard overlap. Numbers 1 to 6 indicate different groups. The groups with smaller overlap threshold can be used to sample small mangoes. In contrast, the groups with larger overlap threshold can be used to sample large mangoes. In addition, the overlap thresholds should be selected so that each sampling ratio can obtain positive examples, which can also reduce the difference in quantity between the positive and negative examples.

Table 1. The configurations of sampled patch.

Configuration	1	2	3	4	5	6
Min ratio	0.05	0.05	0.05	0.05	0.1	0.2
Max ratio	0.6	0.6	0.6	0.6	0.6	0.6
Overlap	0.02	0.05	0.1	0.3	0.5	1

SSD network keeps the overlapped part of the ground truth box if the center of it is in the sampled patch and used them for training. The objects in images in the VOC dataset are larger, so these default parameters are set to be relatively large. As our mangoes are small, we need small sampling ratios and overlap rates. The design of different-scale sampled patches and multiple minimum jaccard overlap rates makes it more effective to detect objects of various scales with SSD network.

Distortions. Some photo-metric distortions are also used in SSD, including brightness, contrast and saturation. When we use SSD network to detect a specific dataset, we can adjust relative parameters to improve detection performance. For example, most images in our fruit dataset are dark, we can change brightness and contrast according to our own dataset, which can help to extract more abundant features.

In addition to the original distortion methods of the SSD network, we also added Gaussian blur method in the image preprocessing stage, which makes the detection model more robust.

Each distortion item has a parameter representing the probability of being implemented. We set the probability threshold to 0.5, which means that each input image has a probability of 0.5 to be implemented distortion operations.

3.3 Default Box Proposals

SSD network generates default boxes of various scales and aspect ratios from different-scale feature maps (Conv4_3, FC7, Conv6_2, Conv7_2, Conv8_2 and Conv9_2 in Fig. 2). For each default box out of k at a given location, we predict c class scores and 4 offsets relative to the original default box shape using small convolution filters with the kernel of 3×3 . For layer Conv4_3, Conv8_2 and Conv9_2, there are 4 boxes at a given location, and 6 boxes in other layers. Finally, there are a total of $(c + 4)k$ filters for each specific location and $(c + 4)kmn$ outputs for a $m \times n$ feature map.

SSD network matches the default box to ground truth box by setting the overlap threshold. The minimum and maximum sizes of the default boxes generated from different feature maps are different, which is contributing to detect more objects of various scales. The minimum and maximum sizes of the default boxes for each feature map are set by the scale ratio parameter. The scale ratio of default boxes is defined as:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), k \in [1, m]. \quad (1)$$

Where s_{\min} is 0.2 and s_{\max} is 0.9 by default, meaning the lowest layer has a scale of 0.2 and the highest layer has a scale of 0.9, and all layers in between are regularly

spaced. M indicate m feature maps, here m is 6. For a specific feature map, the scale ratios are between s_k and s_{k+1} . The minimum sizes and maximum sizes of default boxes for each feature map are defined as:

$$size_{\min} = \dim_{\min} \times s_k, \quad (2)$$

$$size_{\max} = \dim_{\min} \times s_{k+1}, k \in [1, m]. \quad (3)$$

Where \dim_{\min} is the minimum dimension of input image and it's 300. Note that the Eqs. (1) to (3) is suitable for layer FC7 to Conv9_2. For layer Conv4_3, the minimum scale ratio and the maximum scale ratio is 0.1 and 0.2 respectively.

For objects with small changes in size, we can adjust these default boxes to achieve better detection performance. The mangoes we want to detect are generally small, so the default box should be set smaller. In the previous section, we have calculated the distribution of fruits' shapes. We use these statistics to calculate the size of the default box on each feature map. For layer Conv4_3, which is mainly used to detect most of mangoes, we set the minimum and maximum sizes of the default boxes to 20 and 60 pixels respectively, which covers most sizes of fruits. For other layers, we set s_{\min} to 0.1 and s_{\max} to 0.6 in Eq. (1). Thus these default boxes can match more ground truth bounding box and increase positive examples.

3.4 Loss Function

The overall objective loss function of SSD network is a weighted sum of the localization loss (loc) and the confidence loss (conf), which is defined as following:

$$L(x, c, l, g) = \frac{1}{N} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)). \quad (4)$$

Where N is the number of matched default boxes, c is the number of class, and α is the weight term of localization loss. The localization loss is a Smooth L1 loss between the predicted box (l) and the ground truth box (g) parameters.

For small object detection, the confidence loss is more important than localization loss. By default, the weight term α is set to 1 by cross validation. Here we set it to 0.5 to adapt our fruit dataset. Our experimental results proved that the F1 score can increase by 0.3% by decreasing the weight term.

4 Experimental Setup

4.1 Dataset

The mango dataset we trained and tested is a part of dataset released by [8], which contain apple, mango and almond datasets. The original images in mango dataset were in 48-bit format. We converted the images to 24-bit format with the image processing software, ImageJ, so that we can train the images in SSD network. Although the dataset was annotated and tested with Faster R-CNN by [8], there were still some errors with

annotations. In addition, many images don't contain any mango. Annotation errors can affect not only training but also detection result.

In order to minimize the impact of annotation errors, we removed the images without any mango and relabeled the mango dataset with the annotation software [16]. We found that the main annotation error is missing ground truth annotations of mangoes, especially those in dark images, which is difficult even for human to identify. Therefore, we increased contrast of images during the annotation process. When annotated the mangoes that missing annotation, we added training samples, in fact, which is extremely beneficial to the training model. And for testing stage, the evaluation results is more accurate. In this paper, all detection results were obtained from test dataset.

4.2 Evaluation Criteria

In the evaluation system of object detection, there is a parameter called IOU (intersection-over-union), which is the intersection of the test results and ground truth over their union. IOU is defined as:

$$IOU = \frac{area(P) \cap area(G)}{area(P) \cup area(G)}. \quad (5)$$

Where area (P) is the area of the bounding box of prediction, while area (G) is the area of the bounding box of ground truth.

When the IOU of detection result and ground truth bounding box is greater than the threshold, we regard the detection result as ground truth and call it true positive, which represent those instances that are originally mangoes and are predicted as mangoes. On PASCAL-VOC challenges, the threshold is set to 0.5. In this paper, however, we set the threshold to 0.2, which is sufficient for small object detection and a fruit mapping application [8]. Figure 4 shows the different result between 0.2 and 0.5 of IOU on the same detection model.

For better reporting results and comparisons, we used the F1 score, also used in [8], to report our results. F1 score is a comprehensive evaluation result of accuracy and recall rate, which is defined as:

$$F1 = 2 \times \frac{P \times R}{P + R}. \quad (6)$$

Where P is the precision rate and R is the recall rate of prediction results. They are defined as follows:

$$P = \frac{\text{the number of detected mangoes}}{\text{the number of detected objects}}. \quad (7)$$

$$R = \frac{\text{the number of detected mangoes}}{\text{the number of ground truths}}. \quad (8)$$

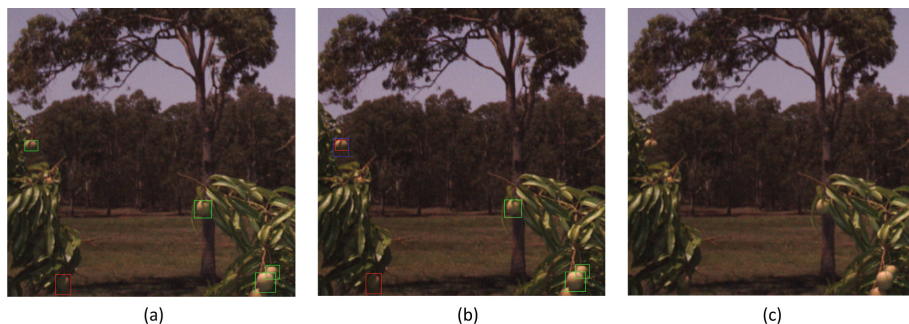


Fig. 4. Results of different IOU threshold for mango detection. The threshold of the image (a) is 0.2 and the image (b) is 0.5. The image (c) is the original image. Although the predicted bounding box (red bounding box in the top-left part of image (a)) is pretty different from the ground truth (blue), we still consider it a true positive (green bounding box in the top-left part of image (b)). (Color figure online)

5 Results

5.1 Optimization

Thanks to the optimization of data augmentation and default box proposals, mango detection performance has greatly improved. Table 2 reported the detection results of our optimization. No matter what kind of image input size, the detection accuracy has improved a lot. Especially for SSD300 network, the F1 score has increased by nearly 10%. The resolution of mango image is 500×500 . So when we resize the image to 300×300 , the mangoes have been reduced in size. This further increases the difficulty of mango detection. The input sizes of SSD400 and SSD500 network are close to that of original image, so better detection performance can be obtained without optimization. Our optimized SSD400 network outperforms the original SSD500 network. The best F1 score was obtained from our SSD500 network, which is 1% higher than the original SSD500 network.

To further verify the effectiveness of our proposed method, we made more experiments on almond dataset, which only contains 572 images with the resolution of 300×300 . We use 375 images for training, 103 for testing and others for validation. The results of almond detection are also shown in Table 2. Because there are fewer training samples, the overall F1 score is relatively low. But the results still reflects the advantages of our method. Through our optimization, detection performance can be significantly improved. The accuracy of SSD400 network can nearly reach the level of SSD500 network.

Table 2. Performance of our improved SSD network

Network	Mango			Almond		
	SSD300	SSD400	SSD500	SSD300	SSD400	SSD500
Original SSD	0.787	0.864	0.903	0.728	0.753	0.771
Our SSD	0.884	0.911	0.912	0.762	0.779	0.783

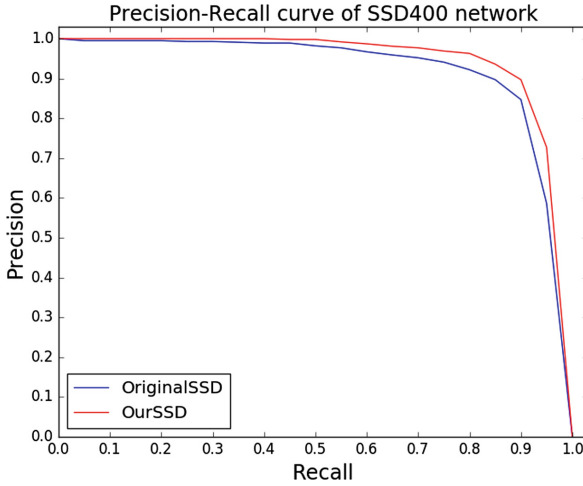


Fig. 5. Precision-Recall curve of original SSD400 network and our SSD400 network for mango detection.

To more visually compare the performance of the SSD network before and after the optimization, we plotted the PR (Precision-Recall) curve of mango detection, as shown in Fig. 5. We can see clearly that the precision-recall curve of our optimized network encloses the curve of the original SSD network, which means that our optimized network is better than the original network. Figure 5 further proved the effectiveness of our proposed method.

Both SSD400 and SSD500 network are better than SSD300 network. Note that SSD400 and SSD500 network got similar performance after optimization. In other words, high performance can be obtained with small input resolution by implementing our optimization techniques of the detection network. Although the SSD400 and SSD500 network have the similar accuracy, the SSD400 network has greater advantages in the aspects of speed and memory usage.

SSD network with its default configurations can be used to detect many kinds of objects of different scales. When we apply SSD network to a specific detection task, like our fruit detection, the data augmentation techniques and default box proposals can be optimized to make SSD network work better for the detection task. Here our experimental results proved it.

5.2 Various Classification Networks

For the image classification tasks based on deep learning, the performance of classification networks of various structures and depths is also different in the aspects of detection accuracy and speed. Similarly, when the SSD is connected to different classification models as base network, different performance will be obtained. In this paper, we experiment with ZFNet [17] and VGG16 for mango detection, which contains 5 and 13 convolution layers respectively. The results are shown in Table 3.

Table 3. Mango detection performance of various classification networks.

Network	Precision	Recall	F1	FPS
ZFNet (0.4)	0.947	0.690	0.798	50
ZFNet (0.5)	0.967	0.627	0.761	50
VGG16	0.920	0.903	0.911	35

We set the input image size to 400×400 . The value in parentheses following ZFNet represents the corresponding confidence threshold.

According to Table 3, although ZFNet is faster than VGG16, the F1 score of ZFNet is much worse than VGG16. The detection confidence threshold of both ZFNet and VGG16 is set to 0.5. As we can see, the recall rate of ZFNet is nearly 30% lower than VGG16, which leads to a very low F1 score of ZFNet. The recall rate of ZFNet increased by 6%, and the F1 score increased by 4%, when the confidence threshold is 0.4. This also shows that the performance of ZFNet is not as good as that of VGG16 in fruit detection.

VGG16 contains more convolution layers than ZFNet, making it more capable in feature abstraction. The higher the layer is, the more semantic it can express, which is more advantageous for detection and classification tasks. Although the VGG16 network is deeper and requires more computations and higher memory usage, it still achieves real-time detection with high accuracy.

5.3 Various Detection Networks

Among the object detection algorithms based on deep learning, Faster R-CNN and SSD are currently the most widely used networks. In this paper, we have also compared the performance of Faster R-CNN and SSD and used VGG16 as the base network for both algorithms. We use F1 score and FPS as indicators to compare SSD network and Faster R-CNN.

Table 4. Performance of various detection networks

Method	Mango	Almond	FPS
Faster R-CNN [8]	0.908	0.775	14
Our SSD300	0.884	0.762	50
Our SSD400	0.911	0.779	35
Our SSD500	0.912	0.783	24

Table 4 shows the comparisons between SSD and Faster R-CNN. Both of our SSD400 and SSD500 network outperform Faster R-CNN in the aspects of accuracy and speed. As we can see that, Faster R-CNN is the slowest in the aspect of detection speed, about 14 FPS, which is mainly due to its architecture and anchor mechanism. SSD300 network performs best in the aspect of speed, but its accuracy is not as good as other algorithms. In comparison, SSD400 is the best choice because it performs very



Fig. 6. Visualization results of mango detection. Green boxes are true positives, red boxes are false positives and blue boxes are false negatives. The results of image (a), (b), (e) and (f) are ideal, without any false positive and false negative. There are many mangoes being detected in image (b), (c) and (d), which are either blocked by branches, leaves or mangoes, or only partially captured by the camera. The two mangoes are small in image (e), but they are still detected, which is largely due to our new sampling strategies. Image (f) presents detection results of dark image. Image (g), (h) and (i) give the results of almond detection. (Color figure online)

well both in terms of speed and accuracy. With the higher accuracy than Faster R-CNN, our SSD400 network achieved a real-time detection.

All networks were trained on the single GTX 1080ti GPU, using cuda8.0 and cuDNN5.1.

5.4 Visualization of Detection Results

As mentioned in Sect. 1, there are many difficulties in detecting on-tree mangoes. First of all, because of the differences of the density of the leaves and the degree of camera shooting angle, which resulting in different light distribution, the obtained images are also very different. This requires that our detection network have a high generalization ability and robustness. The blocking problem is also one of the great challenges we faced with. Under natural conditions, mangoes are blocked by leaves, branches or between mangoes, which can greatly affect the detection performance and positioning accuracy. However, we got satisfying results on fruit detection with our improved SSD network. Figure 6 presents some visualization results of mango and almond detection.

6 Conclusion

In this paper, we present a real-time detection framework for on-tree mango in the orchard based on the state-of-the-art object detection algorithm, SSD network. Even though there are many difficulties in our dataset for fruit detection, our method achieved high detection performance. Compared with the original SSD network, our proposed framework with the optimization of data augmentation techniques and default box proposal methods is more accurate in on-tree mango detection. Besides, the mango dataset was relabeled and converted to the xml format in VOC2007 style. A study of detection performance of SSD network with various classification networks as base network illustrates that the VGG16 is more suitable than ZFNet for on-tree mango detection. Detection results on almond dataset further prove the effectiveness of our method. Experimental results demonstrate that our method has a more excellent performance than Faster R-CNN and achieves F1 score of 0.911 and real-time detection at 35 FPS with a GPU for mango detection. Note that, however, there are still many challenges in mango detection, such as the large-area overlapping between mangoes and mangoes or mangoes and leaves. Future works will focus on these challenges.

Acknowledgement. This work was supported in part by the National Nature Science Foundation of China (NSFC 61673163), Hunan Provincial Natural Science Foundation of China (2016JJ3045), and Hunan Key Laboratory of Intelligent Robot Technology in Electronic Manufacturing (No. 2018002).

References

1. Syal, A., Garg, D., Sharma, S.: Apple fruit detection and counting using computer vision techniques. In: IEEE International Conference on Computational Intelligence & Computing Research, pp. 1–6 (2015)
2. Kapach, K., Barnea, E., Mairon, R., Edan, Y., Ben-Shahar, O.: Computer vision for fruit harvesting robots – state of the art and challenges ahead. *Int. J. Comput. Vis. Robot.* **3**(1–2), 4–34 (2012)

3. Wang, Q., Nuske, S., Bergerman, M., Singh, S.: Automated crop yield estimation for apple orchards. In: Desai, J., Dudek, G., Khatib, O., Kumar, V. (eds.) *Experimental Robotics*. Springer Tracts in Advanced Robotics, vol. 88, pp. 745–758. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-319-00065-7_50
4. Kadmiry, B., Wong, C.K.: Perception scheme for fruits detection in trees for autonomous agricultural robot applications. In: *International Conference on Image & Vision Computing*, New Zealand, pp. 1–6 (2016)
5. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)
6. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. [arXiv:1612.08242](https://arxiv.org/abs/1612.08242) [cs.CV]
7. Liu, W., et al.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
8. Suchet, B., James, U.: Deep fruit detection in orchards. [arXiv:1610.03677](https://arxiv.org/abs/1610.03677) [cs.RO]
9. Girshick, R., Donahue, J., Darrell, T., et al.: 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)
10. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8691, pp. 346–361. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_23
11. Girshick, R.: Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448 (2015)
12. Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: *Advances in Neural Information Processing Systems*, pp. 379–387 (2016)
13. Redmon, J., Divvala, S., Girshick, R., et al.: 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)
14. Stein, M., Bargoti, S., Underwood, J.: Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors* **16**(11), 1915 (2016). <https://doi.org/10.3390/s16111915>
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *NIPS* (2015)
16. Bargoti, S.: Pychet Labeller - an object annotation toolbox (2016). <https://github.com/acfr/pychetlabeller>
17. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53