Journal of Food Measurement and Characterization (2024) 18:7067–7076
https://doi.org/10.1007/s11694-024-02717-1

ORIGINAL PAPER

On-tree fruit detection system using Darknet-19 based SSD network

Diwakar Agarwal[1] · **Anuja Bhargava**[1]

Received: 15 May 2024 / Accepted: 19 June 2024 / Published online: 28 June 2024

Abstract
An accurate and reliable on-tree fruit detection system is crucial for automatic yield estimation, mapping, and harvesting. This paper presents a deployment of the state-of-the-art deep object detection model, the Single Shot Detector (SSD) for detecting mango fruits on a tree canopy in an open mango orchard. The proposed approach utilizes the deep learning model Darknet-19 as a feature extractor and SSD for detection and localization in the object detection framework. The network's training, validation, and testing are performed on the publicly available annotated dataset 'Mango YOLO'. The data augmentation applied to training images consists of the random and central extraction of image patches and corresponding horizontal reflection. Experimental results show that the proposed network achieved an average precision of 92.43% in detecting on-tree mango fruits in the test dataset.

Keywords Agriculture · Deep learning · Fruit Detection · Mango · Object detection · Transfer learning

Introduction

Today, computer vision and artificial intelligence have found a large number of applications in various disciplines including industry, medicine, life science, defense, etc. According to recent studies, the use of deep neural networks has also pushed the agriculture sector to a higher level where precision farming and automatic cultivation and harvesting are possible. Previous fruit detection methods [1–5] relied on hand-engineered features such as shape, color, and texture, which were extracted for further use by several classifiers to classify between fruit and non-fruit regions. However, these methods so designed for a specific dataset may not work for other datasets as different datasets have different appearances, backgrounds, and illumination. The capability of these methods to generalize new datasets depends on selecting a sufficient number of distinguishing features. Thus, the lack of a trainable features extractor necessitates the development of deep Convolutional Neural Network (CNN)-based fruit detection methods. The CNN understands input images by learning their complex and sophisticated abstract features such as patterns, edges, etc., at multiple scales of spatial resolution. These methods are generalized well to real orchard scenes, robust to heavy occlusion, and capable of detecting multiple overlapping fruits. In the dimension of deep learning for fruit detection, the extensively used framework exists in two categories- two-stage detectors and single-stage detectors.

Generally, a two-stage detector proposes the probable regions likely to contain fruits and then feeds these regions to the second stage for classification. Typically used two-stage models are OverFeat [6], Region-CNN (R-CNN) [7], Spatial Pyramid Pooling Net (SPPNet) [8] MultiBox [9], Fast R-CNN [10], Faster R-CNN [11], and Mask-RCNN [12]. On the other hand, the SSD and YOLO-based detectors eliminate the region proposal stage and rely on a single CNN to localize bounding boxes and estimate class probabilities. This makes the object detector faster than the R-CNN. Some commonly used single-stage models are SSD [13], You Only Look Once (YOLO) [14], YOLOv2 [15], RetinaNet [16], YOLOv3 [17], YOLOv4 [18], etc.

Although the aforementioned models are comparable to each other in terms of speed and accuracy, still there is a scope for improvement in dealing with occlusion and scale variation issues while on-tree fruit detection. In the advancement in object detection, the Vision Transformer

✉ Anuja Bhargava
anuja.bhargava@gla.ac.in

Diwakar Agarwal
diwakar.agarwal@gla.ac.in

[1] Department of Electronics & Communication, GLA University, Mathura, India

(ViT) has emerged as a vision model that has linear computational complexity and exceptional global perception capabilities [19]. Recently, the Swin Transformer was used in [20] to detect the grape bunches and it was concluded that the ViT outperforms SSD and YOLO in terms of precision and detection accuracy in heavy occlusion scenarios.

Liang et al. [21] implemented single-stage detection of on-tree mango fruits based on VGG16 which is more sophisticated and less precise. Later approaches used the YOLO network for more accurate detection but at the cost of an increased number of layers and complexity. This paper also proposes the single-stage detection for mango fruits that uses less complex pre-trained CNN Darknet-19 as a feature extractor to provide multi-scale features to the SSD. The box regression is used for bounding box localization in input images. The major contributions are summarized as follows:

- Deployed the Darknet-19 in SSD for the detection of on-tree mango fruits in an outdoor orchard.
- Optimized the anchor boxes at different scales of feature maps used for training to improve the detection performance.
- The proposed detector is robust to occlusion and varying illumination conditions.

In this paper, for ease of reading and writing, several complex terms are written in abbreviated form which are listed in Table 1. The rest of the paper is organized as follows: Sect. 2 presents the work related to the proposed approach. The methodology is explained in Sect. 3. Section 4 presents the details of network and model training. Section 5 briefs the parameters used for performance evaluation. Section 6 provides the experimental results and the conclusion is given in Sect. 7.

Table 1 List Of abbreviations

Abbreviation	Definition
SSD	Single Shot Detector
YOLO	You Only Look Once
CNN	Convolutional Neural Network
R-CNN	Region-CNN
SPPNet	Spatial Pyramid Pooling Net
mAP	mean Average Precision
NMS	Non-Maximal Suppression
VOC	Visual Object Classes
SGDM	Stochastic Gradient Descent with Momentum
IoU	Intersection over Union
FP	False Positive
FN	False Negative
TP	True Positive
TN	True Negative

Related work

This section presents the deep CNN-based state-of-the art methods which are available as two-stage and single-stage fruit detectors. The automatic on-tree fruit detection aids the farmers in accurate yield estimation and mapping so that the resources can be used efficiently and save the laborer task as much as possible [22]. In the literature, R-CNN and its variants are mostly used in several two-stage fruit detection algorithms. Gao et al. [23] proposed the use of Faster R-CNN to detect fully-occluded and partially-occluded apple fruits under natural illumination. The mean Average Precision (mAP) of 87.9% was achieved on augmented 12,800 images obtained from 800 images. Jia et al. [24] also proposed using Mask R-CNN to detect apple fruits. The ResNet and DenseNet were integrated to extract probable fruit regions for further classification by Mask R-CNN. The mAP of 97.31% was achieved on 120 test images. Fu et al. [25] implemented ZFNet and VGG16-based Faster R-CNN model on the images of apples captured in both daytime and night-time. The mAP of 89.3% was achieved with the VGG16 network.

On the other hand, a single-stage fruit detector is based on SSD and YOLO deep networks. Liang et al. [21] designed an SSD with ZFNet and VGG16 as base networks for on-tree fruit detection. The mAP of 92% was achieved with the VGG16 network. Koirala et al. [26] proposed a 'MangoYOLO' network which was based on the integrated features of YOLOv3 and YOLOv2(tiny) for mango detection and achieved the mAP of 98.3%. Sozzi et al. [27] implemented six versions of YOLO to find a suitable model for the detection of bunches of white grapes. The YOLOv4-tiny was observed as the best model both in terms of speed and accuracy. Zhang et al. [28] proposed a deep network 'OrangeYolo' based on YOLOv3 for the detection of orange fruits. The mAP of 95.7% was achieved in comparison to YOLOv3 (90.5%), YOLOv4 (91.1%), and YOLOv5 (91.7%). Wang et al. [29] proposed the 'ShuffleNet v2' an improved version of YOLOv5 for the detection of litchi fruits. The mAP of 93.9% was achieved on 275 images of the test dataset. Similarly, Cao and Yuan [30] proposed an improved YOLOv4 network 'YOLOv4-LightC-CBAM' for mango detection and achieved the mAP of 95.39%. Chandana et al. [31] proposed the 'MangoYOLO5' which is much lighter than the original YOLOv5 for mango detection and achieved the mAP of 3.0% more than YOLOv5 (91.7%). Another application of YOLOv4 was seen in Lai et al. [32] where the model was utilized to detect ripe fresh fruit bunches for the extraction of palm oil from a tree and achieved the mAP of 87.9%. Quach et al. [33] proposed the use of YOLOv8 for tomato detection and achieved the mAP of 91.0%.

Table 2 Details Of extra added layers

Name	Stride	Filter size	Number of filters	Type	Activations
Conv14	Stride [1 1] with no padding	$1 \times 1 \times 512$	128	2-D Convolution	$16 \times 16 \times 128$
leaky14	-	-	-	Leaky ReLU with scale 0.1	$16 \times 16 \times 128$
Conv15	Stride [2 2] with padding [1 1 1 1]	$3 \times 3 \times 128$	256	2-D Convolution	$8 \times 8 \times 256$
leaky15	-	-	-	Leaky ReLU with scale 0.1	$8 \times 8 \times 256$
Conv16	Stride [1 1] with no padding	$1 \times 1 \times 256$	128	2-D Convolution	$8 \times 8 \times 128$
leaky16	-	-	-	Leaky ReLU with scale 0.1	$8 \times 8 \times 128$
Conv17	Stride [2 2] with no padding	$3 \times 3 \times 128$	256	2-D Convolution	$3 \times 3 \times 256$
leaky17	-	-	-	Leaky ReLU with scale 0.1	$3 \times 3 \times 256$
Conv18	Stride [1 1] with no padding	$1 \times 1 \times 256$	128	2-D Convolution	$3 \times 3 \times 128$
leaky18	-	-	-	Leaky ReLU with scale 0.1	$3 \times 3 \times 128$
Conv19	Stride [2 2] with no padding	$3 \times 3 \times 128$	256	2-D Convolution	$1 \times 1 \times 256$
leaky19	-	-	-	Leaky ReLU with scale 0.1	$1 \times 1 \times 256$
Conv20	Stride [2 2] with padding [1 1 1 1]	$1 \times 1 \times 256$	128	2-D Convolution	$1 \times 1 \times 128$
Leaky20	-	-	-	Leaky ReLU with scale 0.1	$1 \times 1 \times 128$

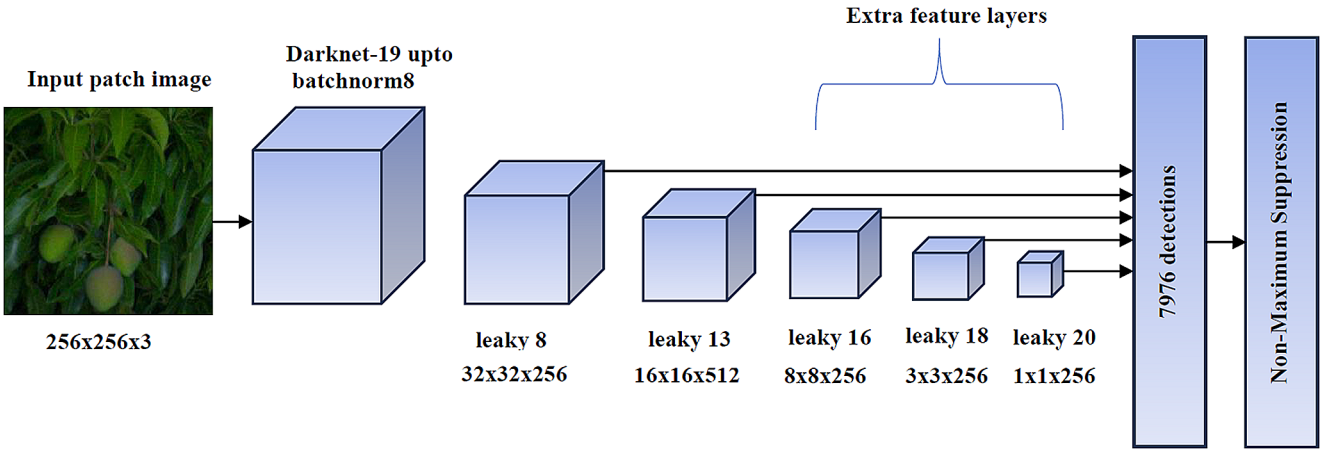

Fig. 1 Proposed SSD network including Darknet-19 as the base network

Methodology

SSD object detection framework

The single-stage detectors implement the object detection framework in a single network that includes feature extraction, classification, localization, and making the bounding box around the detected fruit in an image. In the proposed framework, the pre-trained CNN 64-layers Darknet-19 [15] adopted through transfer learning is used as a base network for feature extraction. The transfer learning allows the reusing of existing parameters i.e., convolutional weights from the model pre-trained by the large dataset in the working model. For SSD, the layers after 'leaky13' are removed and 14 other layers are added at the end as detailed in Table 2. These layers are progressively decreasing in size and are available for predicting the bounding boxes of different scales and aspect ratios. The feature maps are obtained from the layers 'leaky8', 'leaky13', 'leaky16', 'leaky18', and 'leaky20' as shown in Fig. 1. Before training the SSD, a fixed set of default bounding boxes generally 4 or 6 is predefined for each class at each location of a feature map with varying scale and aspect ratio.

In this work, 6 default boxes are defined for 'leaky8' and 'leaky13' whereas, 4 default boxes are defined for 'leaky16', 'leaky18', and 'leaky20'. If the size of a feature map is $M \times N$ and the number of default boxes associated with each location is B then, a feature map can contribute $M \times N \times B$ detections per class. Thus, in the proposed object detection framework all feature maps contribute 7976 detections for a single class 'Mango'. Moreover, a 3×3 convolutional window at each location of a feature map predicts all class confidence scores with four offset values (w, h, Δw, Δh) of a bounding box relative to default box positions. The final output image is obtained when all predicted bounding boxes pass to the Non-Maximal Suppression (NMS) where redundant boxes are reduced with

Table 3 Proposed scale values and aspect ratios at feature maps for computing default boxes

Feature map	Scale	Aspect ratio
leaky8	0.0261	[1, 2, 1/2, 3, 1/3]
	0.0459	1
leaky13	0.0657	[1, 2, 1/2, 3, 1/3]
	0.0855	1
leaky16	0.1053	[1, 2, 1/2]
	0.1251	1
leaky18	0.1449	[1, 2, 1/2]
	0.1647	1
leaky20	0.1845	[1, 2, 1/2]
	0.2043	1

the aim to keep one box with a high confidence score for each object.

Scale and aspect ratio of default boxes

The scale and aspect ratio are two major factors that need to be pre-computed to propose the dimension of default boxes. In general, the scale of an object is defined as the proportion of the height and width of an object with respect to the height and width of the image in which it is contained. For example, if the height of a fruit is 97 pixels and the image size is 256×256 pixels then, it means that this object occupies 0.3789 units of height in the image or 37.89% of the total image height. Among the feature maps, the 'leaky8' has a smaller receptive field that can detect small objects and thus has a minimum scale value. Whereas, the 'leaky 20' has a larger receptive field to detect large objects and possess a maximum scale value.

In this work, the minimum and maximum between all widths and heights of all on-tree mangoes are considered as the minimum scale (s_{min}) and maximum scale (s_{max}) respectively. Then, by applying the arithmetic progression (1) between s_{min} and s_{max}, total of 10 scale values are computed for in-between feature maps as provided in Table 3.

Two scale values are defined for each feature map. The i^{th} scale value is given by

$$s_i = a + (i-1)\,d \tag{1}$$

where, $a = s_{min}$ and d is the difference between two consecutive scale values. The aspect ratios [1, 2, 1/2, 3, 1/3] are chosen for 6 default boxes, and aspect ratios [1, 2, 1/2] are chosen for 4 default boxes. Therefore, using scale and aspect ratio the dimension of default boxes in terms of width (w) and height (h) is computed as $256 \times s_k$ where, s_k is the scale value of the k^{th} feature map.

Network and Model Training

This section covers the details of the database, training, validation, testing sets, and training parameters.

Database

The database used in this work is adopted from Koirala et al. [26]. It contains 1730 night-time JPEG images (612×512×3) of mango canopies and corresponding annotation XML files. The files and directories in the database are structured similarly to the PASCAL VOC (Visual Object Classes) dataset which does not require changing the available script to read the annotations. There is a total of 15,132 fruit annotations where each annotation is defined by the minimum (top-left corner) and maximum (bottom-right corner) spatial coordinate of a bounding box around the fruit as shown in Fig. 2.

Training, validation, and test set

Data augmentation techniques increase the size of the database by increasing the variation among images. It also increases the generalizability of the model to the test dataset

(a) (b) (c)

Fig. 2 Some exemplar tree canopy images with corresponding annotations from the database

and reduces the chance of overfitting. In this work, to make the working dataset, four random patches, and one central patch are extracted from each image in the database. This makes the dataset comprised of 8650 patches which is 5 times larger than the original database. Randomly selected 60% of total patches i.e., 5190 patch images comprise the training set which is used to train the SSD. The next 10% of total patches i.e., random 865 patch images comprise the validation set which is used to provide an evaluation of the model ability while tuning the model parameters and the remaining 2595 patches comprise the test set.

The training and validation set undergoes two types of transformation. Firstly, the color transformation where the colors of each patch are altered by randomly adjusting the values of hue, saturation, brightness, and contrast in the HSV color space. Secondly, a box-labelling preserving 2D random affine transformation which includes horizontal reflection and scaling. A threshold value of 0.5 is defined that takes care of the overlap between the reflected bounding box and the centered view of the reflected patch. If the amount of overlap exceeds the threshold, then the bounding box is clipped to the rectangular border within the dimension of the reflected patch. All reflected patches and corresponding bounding boxes are rescaled to 256×256. In contrast, no color transformation is applied to the test set except rescaling to the size 256×256. To train the SSD, the Stochastic Gradient Descent with Momentum algorithm (SGDM) [34] is applied by considering the following hyperparameters as provided in Table 4.

The SGDM is a widely used optimization algorithm in machine learning that updates the training parameters to minimize the loss by achieving the global minima of the loss function. The algorithm computes the gradient of the function using mini-batches of the training dataset at each iteration in the direction of the negative gradient of the loss function. Consider, θ_l is the current position of a point on the trajectory of the loss function $E(\theta)$ at the l^{th} iteration then, the new position θ_{l+1} at $(l+1)^{th}$ iteration is computed by incorporating the change updated by $\alpha \nabla E(\theta_l)$ and the previous change scaled by the momentum value γ as given by (2).

$$\theta_{l+1} = \theta_l - \alpha \nabla E(\theta_l) + \gamma(\theta_l - \theta_{l-1}) \tag{2}$$

Table 4 The values of training parameters

Training parameters	Value
Initial learning rate	1e-5
Learn rate drop period	5
Learn rate drop factor	0.1
L2 regularization	0.0001
Shuffle	every epoch

where, θ is the parameter vector, l is the iteration number, α is the learning rate, $\nabla E(\theta)$ is the gradient of the loss function $E(\theta)$, and γ is the value of momentum.

Performance evaluation

There are several parameters upon which the performance of an SSD network relies. Specifically, class confidence score and Intersection over Union (IoU) are two important parameters.

Class confidence score

The confidence score is a numeric value between 0 and 1 that defines the confidence or probability of the detection of each class by the SSD. In this work, the class confidence score threshold is chosen as 0.5 which means that the SSD will discard those detections which have a confidence score less than the threshold value.

Intersection over union

An IoU is a ratio of the overlap area to the area of union between the ground-truth bounding box and the predicted bounding box. It is also termed as Jacard index which varies between 0 (no overlap) and 1 (full overlap). The value of IoU is pre-defined before training and is used by NMS to eliminate redundant boxes around an object and select the strongest box in the final result. Any detection with IoU greater than the pre-defined threshold is considered as a positive detection otherwise, considered as a negative detection. In the proposed approach, the IoU is set to 0.5 for better detection of relatively small-size mango fruit.

Evaluation metrics

Several metrics are computed in the literature to evaluate the performance of a fruit detection method. In this work, Precision, Recall, F1-score, and mAP are computed to evaluate the efficacy of the proposed model. The positive and negative detections also depend on various factors other than the NMS threshold. For example, a False Positive (FP) could occur due to the similarity between fruit and foliage, and sometimes the False Negative (FN) occurs when the actual present fruit is not detected due to heavy occlusion and poor illumination. On the other hand, the correct predictions result in True Positive (TP) and True Negative (TN). So, by considering TP, FP, and FN, the Precision, Recall, and F1-score are defined as.

1) Precision (P)

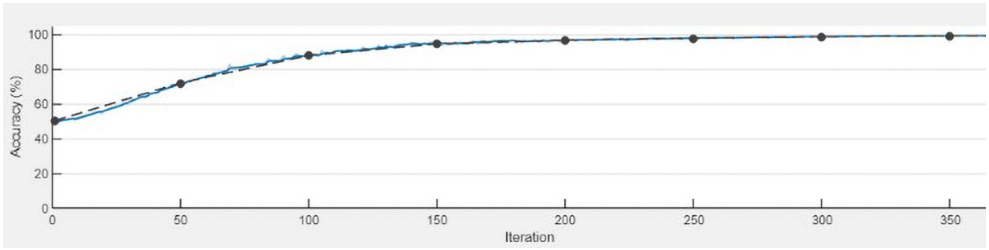

Fig. 3 Training accuracy with respect to the number of iterations

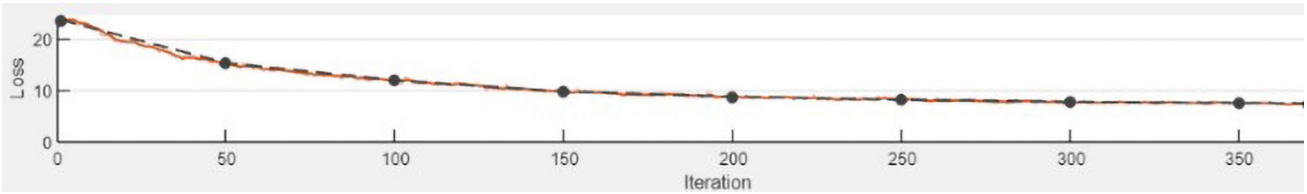

Fig. 4 Training loss with respect to the number of iterations

It is defined as the proportion of true positives in total predicted positives (3).

$$P = \frac{TP}{TP + FP} \tag{3}$$

2) Recall (Rec)

It is defined as the proportion of true positives in total ground-truth positives (4).

$$Rec = \frac{TP}{TP + FN} \tag{4}$$

3) F1-score (FS)

Any object detection model achieves either high precision or high recall as a trade-off exists between precision and recall. Therefore, the F1-score is chosen as the best metric that evaluates the predictions resulting from the SSD. It measures the average proportion of true positives and is computed as the harmonic mean of precision and recall as given by (5). The value of F1-score varies between 0 and 1 which is located on the Precision-Recall curve where the value of precision and recall is highest and identical.

$$FS = \frac{2 \times P \times Rec}{P + Rec} \tag{5}$$

4) Mean average precision

It is defined as the average of precisions computed for all test images (6).

$$mAP = \frac{P}{N} \tag{6}$$

where, N is the total number of test images.

Experimental results and discussion

This section presents the results of training and testing of the proposed SSD network on the datasets and limitations of the work.

Training results

The SSD deep network is trained on the training dataset for a total of 5 epochs. Considering the mini-batch size of 71, the network took around 73 iterations to be trained in 1 epoch. Figure 3 shows a plot of the training accuracy against the number of iterations for 365 iterations. It can be seen that the training of the model started with an accuracy of 52% in the first epoch and reached 72.5% on the 50th iteration. Thereafter, the accuracy rises steadily and reaches 99.0% on the 250th iteration in the 4th epoch. From this point, the training accuracy is slightly improved to 99.01% and remains unchanged for subsequent iterations and epochs. Also, the validation accuracy reaches 99.06% which ensures the model's generalization capability to the validation dataset. Furthermore, the loss encountered while training the model is also plotted against the number of iterations as shown in Fig. 4. It can be seen that in the first epoch 23% loss is incurred by the model as the model has not been exposed to many training images. The training loss then gradually decreased and reached 7.55% while the validation loss reached 7.49% as the number of iterations increased.

Testing results

The effectiveness of the proposed SSD is evaluated on the pre-processed test dataset that comprised 2595 patch images. Considering both NMS and IoU of 0.5 and mini-batch size

of 35, the trained SSD provides the dimensional parameters of the predicted bounding boxes which include the spatial coordinate of the top-left corner, width, and height. Figure 5 shows the bounding boxes around the detected mangoes with the corresponding confidence score as the output of the SSD. These predicted boxes are the strongest bounding boxes among multiple detections for each mango which possesses a high confidence score. The predicted bounding boxes are then compared with corresponding ground-truth boxes to compute FP, TP, and FN. Thus, for each testing patch the precision, recall, F1-score, and mAP are computed by following (3), (4), (5), and (6). Furthermore, the variation of precision with respect to recall can be visualized graphically with the help of the Precision- Recall curve as shown in Fig. 6. It is noticed that the detector achieves good detection results with 82.42% precision and 85.87% recall. These provide the F1-score of 84.10% and mAP of 92.43%. Moreover, to check the efficacy of the proposed fruit detector, a comparison with the best performances of existing fruit detection networks based on mAP is also provided in Table 5. It can be seen that the proposed SSD using Darknet-19 as a base network achieves 0.43% higher mAP than VGG16-based SSD as utilized by Liang et al. [21]. It also achieves 4.53% and 3.13% higher mAP in comparison to VGG16-based Faster R-CNN as utilized by Gao et al. [23] and Fu et al. [25] respectively.

Besides mean average precision, the detection time is also an utmost parameter for evaluating the detection performance. The proposed method took 0.253 s to detect mangoes in a tree canopy patch of resolution 256×256. This time is 0.012 s and 0.072 s more than the time taken by Gao et al. [23] (0.241 s) and Fu et al. [25] (0.181 s) respectively with the resolution of 1920×1080. The reason is that Darknet-19 has more convolutional layers than VGG16 that resulted in large size of trained weights. However, Darknet-19 extracts high level features with increased mAP at the expense of the computational time.

Limitations

The low value of mAP so obtained by the proposed method is the result of some limitations, namely, FP detections, FN

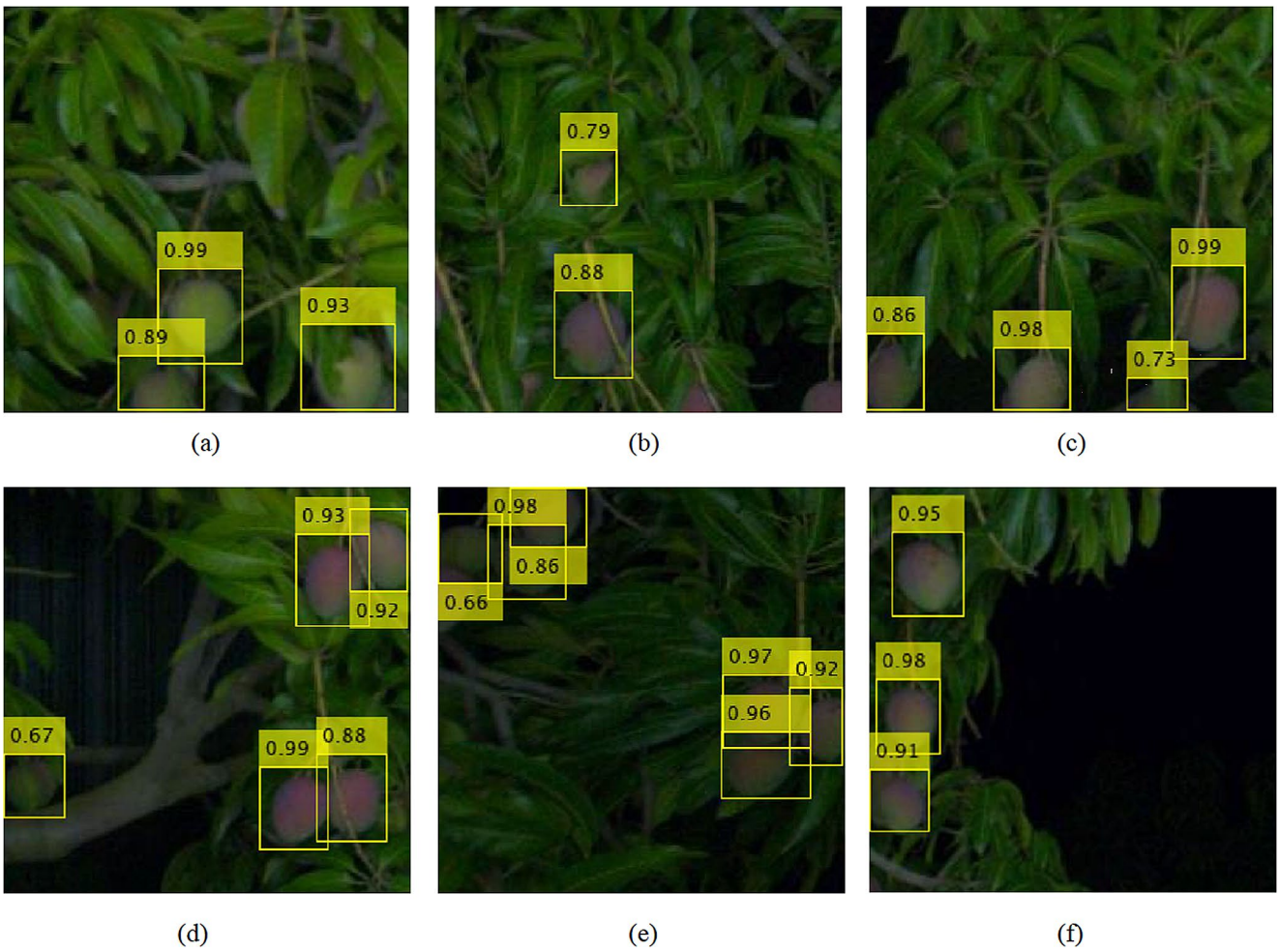

Fig. 5 Detection results by the proposed trained SSD

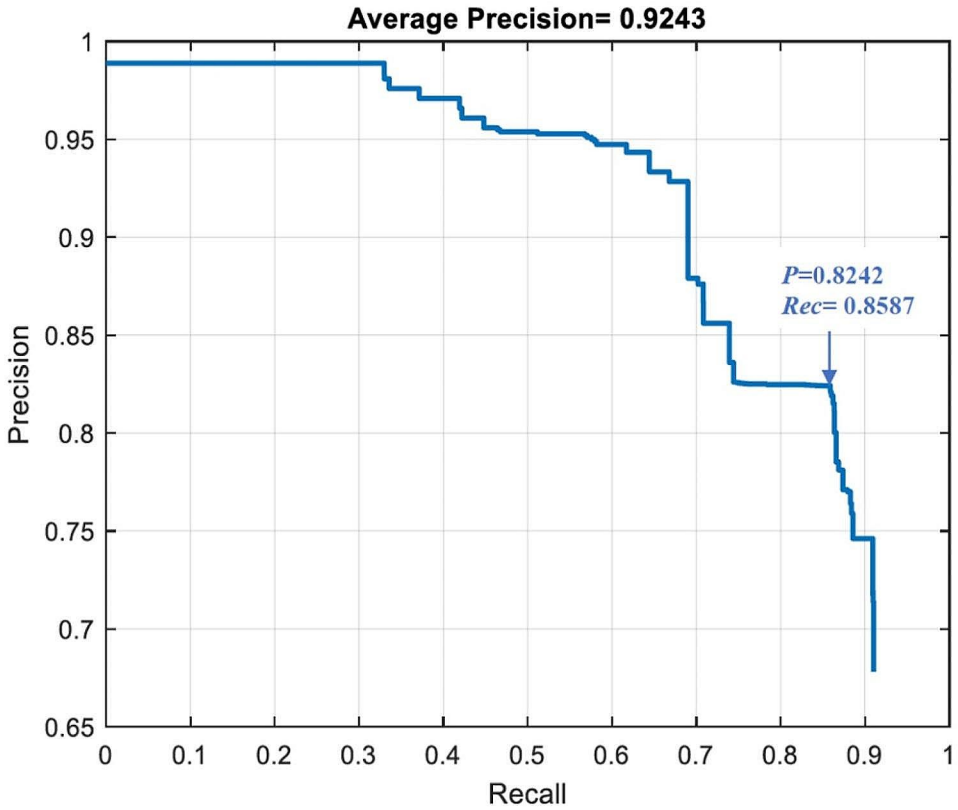

Fig. 6 Precision-Recall curve of the proposed SSD for mango detection

Table 5 Comparison of the proposed SSD with existing fruit detection networks

Research study	Type of fruit and database	Base network	Input image size	mAP (%)
Liang et al. [21]	Mango [22]	VGG16	400×400	92.0
Gao et al. [23]	Apple (manually created)	VGG16	1920×1080	87.9
Fu et al. [25]	Apple (manually created)	VGG16	1920×1080	89.3
Proposed network	Mango [26]	Darknet-19	256×256	92.43

detections, and group detections. These failure cases are shown in Fig. 7. Since the dataset consists of night-time images, the main cause of FP and FN is the inability of the model to distinguish between leaves and mangoes specially the green mangoes. This can be overcome up to some extent by an adequate training of the network with more subtle features extracted from the training patches. Furthermore, the FP detections are also result from the predefined low IoU threshold value. If the IoU is very low then, even the detection overlapping is slightly more than the threshold, it may lead to full mango detection.

Similarly, FN arises if the value of IoU is high then, despite the detection overlapping is more than 50% the mango may be rejected as the overlapping could not reach the set IoU value. Also, the proposed method provides a single detection for the bunch of mangoes. This is mainly due to the NMS algorithm which is used by the object detector to reject multiple redundant true positives. The rejection of multiple detections is the nature of NMS and this affects the performance of the model when individual fruits have to be detected in a bunch. Hence, the FN increases.

Conclusion

This paper presented a deployment of the SSD network for the detection of on-tree mango fruits based on the deep CNN Darknet-19 as a base network. The scale and aspect ratios of the feature maps are also proposed which are utilized to compute the dimensions of default boxes closer to the dimensions of ground-truth boxes. Also, training of the proposed SSD by the data-augmented input images makes the network robust to varying illumination conditions. The experimental results reveal the detection capability of Darknet-19 which is comparable to the state-of-the-art network VGG16. The fruit detection results from the proposed SSD can be used in yield mapping of a large open orchard. However, it is limited to only one face of a tree. As future aspects, it is expected to use 3D technology to detect the fruits from unseen parts of a tree canopy so that the number of fruits can be counted to achieve actual yield estimation.

Fig. 7 Failure cases of the proposed work (a) FP (b) FN (c) group detection

Also, there will always be a scope for improvement that can be achieved by overcoming the overlapping of fruits and heavy occlusion by leaves.

Funding This research did not receive any specific grant from funding agencies in the public, commercial or not-for-profit sectors.

Data availability Data can be available on request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

Informed consent Not applicable.

References

1. S. Naik, B. Patel, Machine vision based fruit classification and grading – a review. Int. J. Comput. Appl. **170**(9), 22–34 (2017)
2. U.O. Dorj, M. Lee, S.S. Yun, An yield estimation in citrus orchards via fruit detection and counting using image processing, *Computers and electronics in agriculture*, vol. 140, pp. 103–112, 2017. https://doi.org/10.1016/j.compag.2017.05.019
3. Z. Iqbal, M.A. Khan, M. Sharif, J.H. Shah, M.H. ur Rehman, K. Javed, An automated detection and classification of citrus plant diseases using image processing techniques: A review, *Computers and electronics in agriculture*, vol. 153, pp. 12–32, 2018. https://doi.org/10.1016/j.compag.2018.07.032
4. T.H. Liu, R. Ehsani, A. Toudeshki, X.J. Zou, H.J. Wang, Detection of citrus fruit and tree trunks in natural environments using a multi-elliptical boundary model. Comput. Ind. **99**, 9–16 (2018). https://doi.org/10.1016/j.compind.2018.03.007
5. W. Xu, H. Chen, Q. Su, C. Ji, W. Xu, M.S. Memon, J. Zhou, Shadow detection and removal in apple image segmentation under natural light conditions using an ultrametric contour map, *Biosystems engineering*, vol. 184, pp. 142–154, 2019. https://doi.org/10.1016/j.biosystemseng.2019.06.016
6. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, 2013, arXiv preprint arXiv:1312.6229. https://doi.org/10.48550/arXiv.1312.6229
7. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81
8. K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in European Conference on Computer Vision. Springer, 2014, pp. 346–361. https://doi.org/10.1007/978-3-319-10578-9_23
9. C. Szegedy, S. Reed, D. Erhan, D. Anguelov, S. Ioffe, Scalable, high-quality object detection, 2014, arXiv preprint arXiv:1412.1441. https://doi.org/10.48550/arXiv.1412.1441
10. R. Girshick, Fast R-CNN, in IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. https://doi.org/10.1109/ICCV.2015.169
11. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
12. K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in IEEE International Conference on Computer Vision, 2017, pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.322
13. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single Shot MultiBox Detector, in *Computer Vision – ECCV 2016*: 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part I 14, Springer International Publishing, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
14. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, in IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788
15. J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271. https://doi.org/10.1109/CVPR.2017.690
16. T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in IEEE International Conference on Computer Vision, 2017, pp. 2999–3007. https://doi.org/10.1109/ICCV.2017.324

17. J. Redmon, A. Farhadi, YOLOv3: an incremental improvement, 2018, arXiv preprint arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767
18. A. Bochkovskiy, C. Wang, H. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020, arXiv preprint arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934
19. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, In Proceedings of the IEEE/CVF international conference on computer vision, pp. 10012–10022, 2021
20. J. Wang, Z. Zhang, L. Luo, W. Zhu, J. Chen, W. Wang, SwinGD: A robust grape bunch detection model based on swin transformer in complex vineyard environment, *Horticulturae*, vol. 7, no. 11, pp. 492, 2021. https://doi.org/10.3390/horticulturae7110492
21. Q. Liang, W. Zhu, J. Long, Y. Wang, W. Sun, W. Wu, A real-time detection framework for on-tree mango based on SSD network, in Intelligent Robotics and Applications: 11th International Conference, ICIRA 2018, Newcastle, NSW, Australia, Proceedings, Part II 11, Springer International Publishing, 2018, pp. 423–436. https://doi.org/10.1007/978-3-319-97589-4_36
22. S. Bargoti, J. Underwood, Deep fruit detection in orchards, in IEEE international conference on robotics and automation (ICRA), 2017, pp. 3626–3633
23. F. Gao, L. Fu, X. Zhang, Y. Majeed, R. Li, M. Karkee, Q. Zhang, Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN, *Computers and Electronics in Agriculture*, vol. 176 pp. 105634, 2020. https://doi.org/10.1016/j.compag.2020.105634
24. W. Jia, Y. Tian, R. Luo, Z. Zhang, J. Lian, Y. Zheng, Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot, *Computers and Electronics in Agriculture*, vol. 172, pp. 105380, 2020. https://doi.org/10.1016/j.compag.2020.105380
25. L. Fu, Y. Majeed, X. Zhang, M. Karkee, Q. Zhang, Faster R–CNN–based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. Biosyst. Eng. **197**, 245–256 (2020). https://doi.org/10.1016/j.biosystemseng.2020.07.007
26. A. Koirala, K.B. Walsh, Z. Wang, C. McCarthy, Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO', Precision Agriculture, **20**, pp. 1107–1135, 2019. https://doi.org/10.1007/s11119-019-09642-0
27. M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, F. Marinello, Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms, *Agronomy*, vol. 12, no. 2, pp. 319, 2022. https://doi.org/10.3390/agronomy12020319
28. W. Zhang, J. Wang, Y. Liu, K. Chen, H. Li, Y. Duan, W. Wu, Y. Shi, W. Guo, Deep-learning-based in-field citrus fruit detection and tracking. Hortic. Res. **9** (2022). https://doi.org/10.1093/hr/uhac003
29. L. Wang, Y. Zhao, Z. Xiong, S. Wang, Y. Li, Y. Lan, Fast and precise detection of litchi fruits for yield estimation based on the improved YOLOv5 model. Front. Plant Sci. **13**, 965425 (2022). https://doi.org/10.3389/fpls.2022.965425
30. Z. Cao, R. Yuan, Real-Time Detection of Mango Based on Improved YOLOv4, *Electronics*, vol. 11, no. 23, pp. 3853, 2022. https://doi.org/10.3390/electronics11233853
31. P. Hari Chandana, P. Subudhi, R. Vara Prasad Yerra, MangoYOLO5: a fast and Compact YOLOv5 Model for Mango Detection, in *Computer Vision and Machine Intelligence. Lecture Notes in Networks and Systems*, vol. 586, ed. by M. Tistarelli, S.R. Dubey, S.K. Singh, X. Jiang (Springer, Singapore, 2023), pp. 719–731. https://doi.org/10.1007/978-981-19-7867-8_57
32. J.W. Lai, H.R. Ramli, L.I. Ismail, W.Z.W. Hasan, Real-time detection of ripe oil palm fresh fruit bunch based on YOLOv4. IEEE Access. **10**, 95763–95770 (2022). https://doi.org/10.1109/ACCESS.2022.3204762
33. L.D. Quach, K.N. Quoc, A.N. Quynh, H.T. Ngoc, N.T. Nghe, Tomato Health Monitoring System: Tomato classification, detection, and counting System based on YOLOv8 model with Explainable MobileNet models using Grad-CAM++. IEEE Access. (2024). https://doi.org/10.1109/ACCESS.2024.3351805
34. Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, *Neural Comput*., vol. 1, no. 4, pp. 541–551, 1989. https://doi.org/10.1162/neco.1989.1.4.541