Pattern Recognition Letters 147 (2021) 206–211

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Deep learning-based apple detection using a suppression mask R-CNN☆

Pengyu Chu[a], Zhaojian Li[a,*], Kyle Lammers[a], Renfu Lu[b], Xiaoming Liu[c]

[a] Department of Mechanical Engineering, Michigan State University, East Lansing, MI, 48824, USA
[b] Agricultural Research Service (ARS), U.S. Department of Agriculture (USDA), East Lansing, MI, 48824, USA
[c] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824, USA

ARTICLE INFO

Article history:
Received 2 October 2020
Revised 6 March 2021
Accepted 19 April 2021
Available online 5 May 2021

Keywords:
Vision system
Fruit detection
Deep learning
Robotic harvesting
Image segmentation

ABSTRACT

Robotic apple harvesting has received much research attention in the past few years due to growing shortage and rising cost in labor. One key enabling technology towards automated harvesting is accurate and robust apple detection, which poses great challenges as a result of the complex orchard environment that involves varying lighting conditions and foliage/branch occlusions. This letter reports on the development of a novel deep learning-based apple detection framework named Suppression Mask R-CNN. Specifically, we first collect a comprehensive apple orchard dataset for "Gala" and "Blondee" apples, using a color camera, under different lighting conditions (overcast and front lighting vs. back lighting). We then develop a novel suppression Mask R-CNN for apple detection, in which a suppression branch is added to the standard Mask R-CNN to suppress non-apple features generated by the original network. Comprehensive evaluations are performed, which show that the developed suppression Mask R-CNN network outperforms state-of-the-art models with a higher F1-score of 0.905 and a detection time of 0.25 second per frame on a standard desktop computer.

1. Introduction

Fruit harvesting is highly labor-intensive and cost-heavy; it is estimated that the labor needed for apple harvesting alone is more than *10 million worker hours* annually, attributing to approximately 15% of the total production cost in U.S[1]. Growing labor shortage and rising labor cost have steadily eroded the profitability and sustainability of the fruit industry. Furthermore, manual picking activities constitute great risks of back strain and musculoskeletal pain to fruit pickers due to repetitive hand motions, awkward postures when picking fruits at high locations or deep in the canopy, and ascending and descending on ladders with heavy loads [2]. Therefore, there is an imperative need for the development of robotic mass harvesting systems to tackle labor shortage, lower human injury risks, and improve productivity and profitability of the fruit industry.

The first and foremost task in robotic harvesting is apple detection, which identifies apples in the area of interest and provides targets for the robot to perform subsequent actions. Due to the low cost of cameras and the tremendous advances in computer vision [3], image-based apple detection systems have gained great popularity in robotic fruit harvesting since the late 1980s. However, robust apple detection in the presence of complex tree structures, varying lighting conditions, and foliage/branch occlusions is a challenging task.

Several methods have been proposed to address the above challenges. For example, a simple thresholding method is developed in [4,5] to generate a binary image with smoothing filters that eliminate noise and irrelevant details. The large segmented regions are then recognized as fruits. This method is easy to implement but it is susceptible to varying lighting conditions. A circular Hough Transform is also proposed to obtain binary edge images along with a matrix of votes on the detection candidates [6,7]. This approach works well with a simple background but is less applicable in a complex structured environment, such as in a dense fruit orchard. Another idea is to combine shape and texture of the fruit to generate a richer set of feature representations [8–11]. By comparing the differences between fruit and leaves in texture, specific fruit or vegetable like broccoli are then detected. However, this method relies on hand-crafted features and is sensitive to lighting conditions and occlusions.

With rapid advancements in deep learning in recent years, deep neural networks (DNNs) have found great successes in object detection and semantic image segmentation [12,13]. DNN-based

☆ Handle by Editor Sudeep Sarkar.

* Corresponding author.
E-mail address: lizhaoj1@msu.edu (Z. Li).

https://doi.org/10.1016/j.patrec.2021.04.022
0167-8655/

methods can learn feature representations automatically without the need of feature hand-engineering. For example, Dias et. al[14]. used a combination of convolutional neural network (CNN) and support vector machine (SVM) to extract features of apple blossoms in a complex background, which shows a good performance of 0.822 F1-score. More recently, region-based convolutional neural network (R-CNN) has gained great popularity in object detection [15]. R-CNN utilizes regions of interest produced by selective search [16] and then regresses bounding box location with classification. A subsequent work, faster R-CNN [17], adds region a locating method instead of selective search to improve its performance.

Despite the aforementioned developments, accurate apple perception to support robotic harvesting in real orchard environments remains a great challenge. Existing methods either provide insufficient accuracy [18,19] or are based on simple structured orchards with little occlusion and stable lighting conditions [14,20]. As such, the goal of this study is to develop a robust and accurate apple detection framework to support robotic harvesting in real orchard environment. Towards this end, we collected a comprehensive dataset from two commercial orchards for two varieties of apples with distinct colors under various lighting conditions. Furthermore, we extended the well-known Mask R-CNN [21] with a suppression network, hereinafter referred to as suppression Mask R-CNN, to improve detection performance. Performance evaluations for apple detection were then conducted to compare the proposed suppression Mask R-CNN with state-of-the-art models.

The contributions of this work are summarized as follows:

1. We collect and process a comprehensive orchard dataset with multiple apple varieties under various lighting conditions in real orchard environment.
2. We develop a new deep network, suppression Mask R-CNN, to remove false detections due to occlusion and thus increase the accuracy and robustness of apple detection.
3. Extensive evaluations show that the proposed suppression Mask R-CNN achieves state-of-the-art performance.

The remainder of this paper is organized as follows. Section 2 reviews the existed state-of-the-art work. Section 3 presents the orchard data collection and processing. The suppression Mask R-CNN is then detailed in Section 4. Experiments are performed in Section 5 to evaluate the proposed framework with comparisons to state-of-art approaches. Finally, Section 6 concludes the paper with discussions on future work.

2. Related work

Several state-of-the-art deep learning-based apple detection approaches have been developed. In particular, DaSNet [19], a deep convolutional neural network that exploits the techniques of spatial pyramid pooling and gate feature pyramid network, is proposed for apple detection. It uses a lightweight residual network as its backbone to achieve improved computational efficiency. Although DaSNet has a decent performance (0.832 F1-score) and a lightweight overhead, the algorithm is only trained and validated on a dataset that contains a single apple variety with good lighting. YOLOv3 [18], another lightweight network that combines Region Proposal Network (RPN) and classification network into a single architecture, is applied in [22] for apple detection. While the network offers a fast detection rate, it has a relatively low F1-score of 0.817. Mask R-CNN [21], a popular object detection algorithm, is also deployed for apple detection [23]. The Mask R-CNN is a two-stage detector that involves a RPN and a classification network. The former searches the location of region of interest (ROI), whereas the latter predicts the class of ROI and regresses the bounding box of the ROI candidates. The Mask R-CNN is successfully applied to apple detection in [23] with promising performance demonstrated.

Fig. 1. Six sample images from the collected dataset: (a)-(c) Gala apples under overcast, back lighting, and direct lighting conditions, respectively; and (d)-(e) Blondee apples under overcast, back lighting, and direct lighting conditions, respectively.

However, the dataset they use only has one apple variety with good lighting conditions, making the results less compelling. In this paper, we use a comprehensive orchard database that contains multiple apple varieties under various lighting conditions. Further, we develop a novel Suppression Mask R-CNN that has superior performance as compared to the aforementioned approaches.

3. Data collection and processing

In this study, apple images of "Gala" and "Blondee" varieties were taken in two commercial orchards in Sparta, Michigan, USA during the 2019 harvest season. The two apple varieties have distinct color characteristics; "Gala" apples are red over a yellow background, while "Blondee" apples have a smooth yellow skin (see Fig. 1). A RGB camera with a resolution of 1,280x720 was used to take images of apples at a distance of $1 \sim 2$ meters to the tree trunk, which is the typical range of harvesting robots [24]. The images were collected across multiple days to cover both cloudy and sunny weather conditions. In a single day, the data were also collected at different times of the day, including 9:00am in the morning, noon, and 3:00pm in the afternoon, to cover different lighting angles: front-lighting, back-lighting, side-lighting, and scattered lighting. When capturing images, the camera was placed parallel to the ground and directly facing the trees to mimic the harvesting scenario. A total of 1,500 images were captured where two sample images are shown in Fig. 1.

We next processed the collected raw orchard images into formats that can be used to train and evaluate deep networks. Specifically, apples in the images were annotated by rectangles using VGG Image Annotator [25] and the annotation was then compiled into the human-readable format. Compared to polygon and mask annotations, rectangular annotation used here accelerates data preparation, particularly in dense images like our dataset. The annotated dataset was then split into training, validation, and test subsets with the apple quantities of 10,530, 4,203, and 4,795, respectively.

4. Suppression mask R-CNN

This section describes the development of a new deep learning-based apple detection approach that systematically combines a DNN backbone and a RGB feature-based suppression network. As shown in Fig. 2, the proposed suppression Mask R-CNN consists of two parts: a feature learning backbone from Mask R-CNN [21] and a feature suppression end. The former is used to learn apple features and generate region proposals. In the meantime, due to the

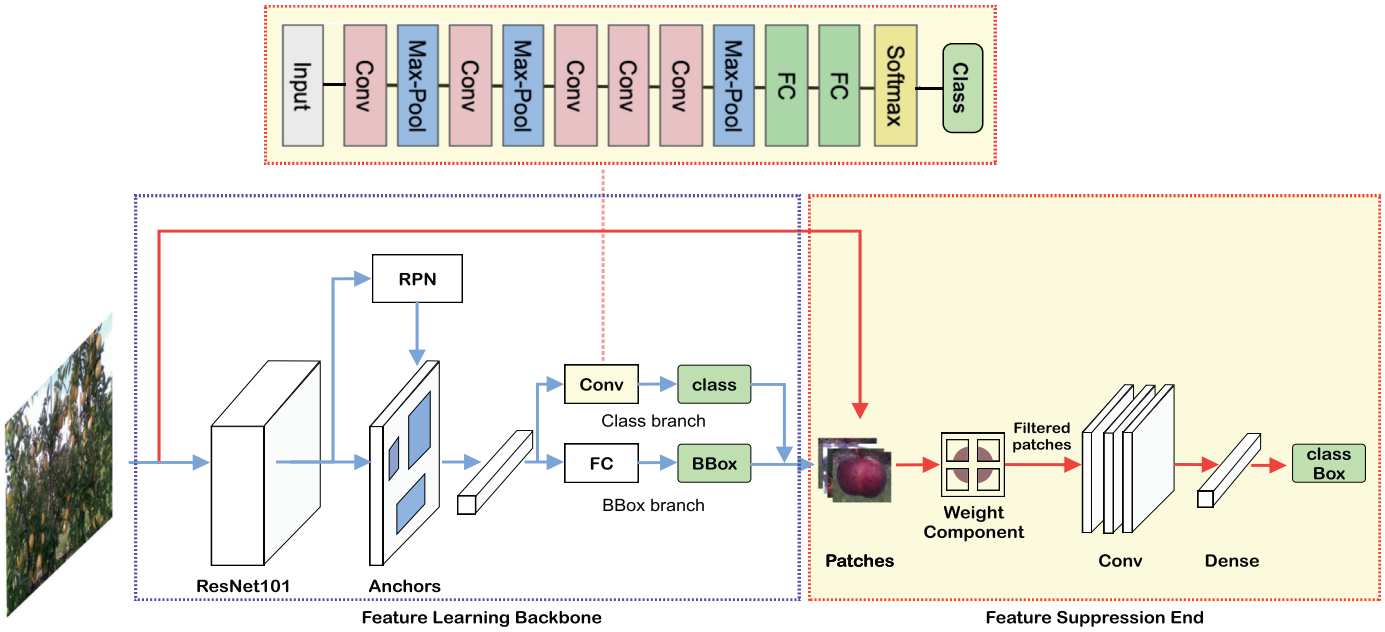

Fig. 2. Structure of the suppression Mask R-CNN. It consists of a feature learning backbone and a feature suppression end. The feature learning backbone is a deep network to learn apple features while the feature suppression end, consisting of a weighting component and a shallow ConvNet, is used to filter non-apple regions.

foliage and branch occlusions, it will also learn foliage and branch features that can cause false detection. As such, we introduce a suppression network to filter non-apple features to improve detection performance by exploiting a combination of clustered features and convoluted features. These two networks are trained separately to avoid generating similar feature maps. We next discuss the two networks in more details.

4.1. Feature learning backbone

The feature learning network uses the Mask R-CNN backbone [21] and follows Mask R-CNN's two-stage learning procedures with two modifications. First, the convolutional backbone in Mask R-CNN is used for feature extraction over an entire image, and is applied as the network backbone for bounding-box recognition. In this study, we instantiate feature learning backbone with ResNet-101-FPN [21] as its backbone. ResNet101 outperforms other single ConvNet mainly because it maintains strong semantic features at various resolution scales. Even though ResNet101 is a deep network, the residual blocks and dropouts function help it avoid gradient vanishing and exploding problems. Then similar to [21], we use a Region Proposal Network (RPN) [17] to generate object regions. RPN is a small convolutional network which can convert feature maps into scored region proposals around where the object lies. These proposals with certain height and width are called anchors, which are a set of predefined bounding boxes. The anchors are designed to capture the scale and aspect ratio of specific object classes and are typically determined based on object sizes in the dataset. In the second stage, class and box offset are predicted by virtue of Faster R-CNN [17] that applies bounding box classification and regression in parallel. As shown in Fig. 2, another network is employed to take the proposed regions from the first stage and assign them to specific areas of a feature map obtained at the second stage. After scanning these areas, the network generates object classes and bounding boxes simultaneously [21].

Second, for improving the recall or true detection of our algorithm, we introduce a convolutional structure (as shown in Fig. 2) in the class branch to learn additional feature representations. The features condensed from the Mask R-CNN backbone and fully connected layers may have lost considerable details of apples. Since images have many occlusions in our dataset, the deep network can treat some partial foliage features as apple features. These additional feature representations will enable the identification of certain regions in an image as an occluded apple or foliage. Furthermore, we freeze the layers in the ResNet101 backbone and train this class branch independently in case there are many overlaps compared to our main network.

4.2. Feature suppression end

After the feature learning step, bounding boxes of apple candidates are obtained. The image patches inside the bounding boxes are then fed into a feature suppression end to remove mis-labeled candidates. Since the feature learning backbone may have learned wrong inference features like leaves with apple-like shapes, the purpose of this suppression network is to avoid that non-apple regions flow into the last decision layer.

Specifically, the suppression network consists of a weighting component and a shallow ConvNet. The weighting component is a 2x2 grid clustering layer that aims to determine apple regions in terms of apple pixel counts. The motivation is that in our annotated dataset, each apple is annotated in the center of a bounding box and occupies the major area in that bounding box. Even though the canopies always partially occlude the apple, the pixels corresponding to the apple are still in the majority. Based on our observation of dataset, the four regions $(a, b, c, d$ as shown in Fig. 3-(3)) generally contain most apple pixels. Therefore, as shown in Fig. 3, we divide each bounding box in the training dataset into four regions, $a, b, c, d,$ as a 2x2 grid. The four regions a, b, c, d is, respectively, located near the left top, right top, left bottom, and right bottom with a margin of 5% pixels to the box edges. Furthermore, we use K-means clustering [26] to group similar pixels and obtain several clusters. After clustering, we label each pixel with its class number i, $i = 1, 2, 3, \ldots n$, with n being the pre-specified cluster numbers (In our experiments, we use $n = 3$). Since the class associated with the most pixels will correspond to the apple region, we select the "apple" region from the four grids and define its pixel counts as N^a, N^b, N^c, and N^d, respectively. We will then

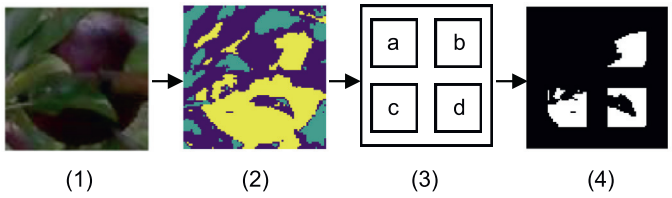

Fig. 3. Illustration of the proposed weighting scheme: (1) sliced image inside the bounding box of a detected apple; (2) pixel clustering using K-means with $k = 3$ where each cluster is shown in one of the three colors; (3) image partitioning into 4 regions and counting pixel numbers of each cluster in the 4 grids; and (4) apple pixel determination by assigning the pixels corresponding to the cluster with most pixel counts in the 4 grids as apple pixels.

set the apple region pixels as 1 whereas other pixels are assigned to zero. A sample output is shown in Fig. 3. The weighting component keeps the objective information and generates an output with only apple pixels, which makes it more efficient to train feature suppression network that we will discuss later. The other merit of weighting component is that if the previous network recognizes a leaf as an apple, only leaf pixels are treated as objectives and flow to next ConvNets. That makes suppression network easy to discriminate apple and non-apple objectives.

The second component is a shallow convolutional network that is used to learn apple features based on filtered patches generated by the weighting component. Compared to the feature learning backbone, the features to learn in this shallow network is less. Only three convolution layers (3x3x32, 3x3x32, 3x3x64) associated with pooling layers (17x17x32, 7x7x32, 2x2x64) and ReLU as activation are used to fit the discrimination function. Two additional dense layers are employed to flatten feature maps and produce decision. This network has a total of 45,153 trainable parameters. The detailed architecture is described in Fig. 2. With the help of feature suppression end, we suppress non-apple class flowing into the decision layer and it does not significantly increase inference time since the depth of the feature suppression end is small. The proposed feature suppression end can be viewed as a filter to efficiently reduce false alarms.

4.3. Loss functions

Since we train the feature learning backbone and the suppression network separately, we define two loss functions as follows. For the feature learning backbone, we use the same loss function with Mask R-CNN [21], which defines a multi-task loss on each sampled region of interest as $L_{backbone} = L_{cls} + L_{box}$, where L_{cls} and L_{box} are, respectively, classification loss and bounding box loss defined as:

$$L_{backbone} = \frac{1}{N_{cls}} \Sigma_i L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{box}} \Sigma_i p_i^* \cdot L_{box}(t_i, t_i^*) \quad (1)$$

$$L_{cls}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i), \quad (2)$$

where p_i and p_i^* are, respectively, the predicted probability and ground truth of anchor i; t_i and t_i^* are, respectively, predicted coordinates and ground-truth coordinates; N_{cls} and N_{box} are normalization terms of batch size and number of anchor locations; the loss function L_{box} is the L1-smooth function [27]; and λ is a parameter that controls the balance between the classification loss and the bounding box loss [28]. In our network, we use $\lambda = 1$ as we assign equal weights to the two losses.

For feature suppression end, we define L_{end} as the average binary cross-entropy loss. For a patch associated with ground-truth class, L_{end} is defined as:

$$L_{end} = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})], \quad (3)$$

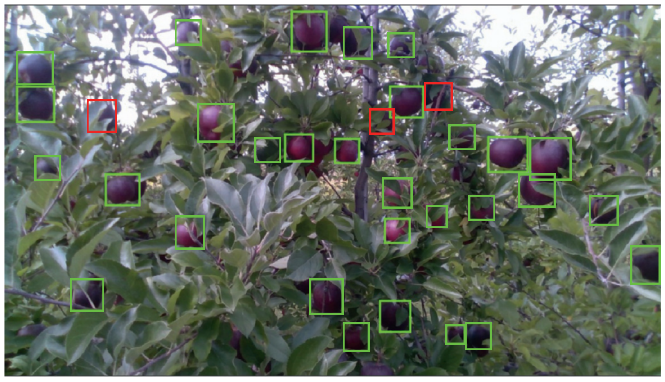

Fig. 4. An example of Gala apple detection using our suppression Mask R-CNN. It shows that the majority of apples are detected (green bounding boxes) but there are still 3 apples missed (red bounding boxes) due to heavy occlusion. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where y is the ground truth and $\hat{y}$ is the prediction.

5. Experiment results

5.1. Implementation

In this section, we evaluate the efficacy of the suppression Mask R-CNN with the processed data as discussed in Section 3. The network hyper-parameters, including the momentum, learning rate, decay factor, training steps, and batch size, are set as 0.9, 0.001, 0.0005, 934, and 1, respectively, through cross-validation. The input image size is 1,280x720, which is aligned with the camera resolution. To better analyze the training process, we set up 100 epochs for training. We exploit a pre-trained model on COCO dataset [29] to warm-start the training process and it generally only needs 50 epochs to converge. A detection example is shown in Fig. 4, where green boxes represent correctly identified apples while red boxes represent missed detection.

To quantitatively evaluate the detection performance, we use performance metrics including precision, recall and F1-score for algorithm evaluation. All detection outcomes are divided into four types: true positive (TP), false positive (FP), true negative (TN), and false negative (FN), based on the relation between the true class and predicted class. Then precision (P) and recall (R) are defined as follows:

$$P = \frac{TP}{TP + FP} R = \frac{TP}{TP + FN} \quad (4)$$

Then F1-score is defined based on precision and recall as follows:

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (5)$$

Note that the suppression network offers a tradeoff between recall and precision, that is, aggressive suppression will lead to higher precision but lower recall rate. This tradeoff can be controlled by adjusting two confidence thresholds th_1 in the class branch network and th_2 in the feature suppression end. Then we tune both confidence thresholds during the inference process to obtain the best recall and precision of our entire model. Fig. 5 shows the Pareto plot, where each point represents the performance of a combination of th_1 and th_2. From the Pareto front (blue solid lines) in Fig. 5, we choose two "best" configurations C_1 and C_2, among which C_1 represents a better F1-score 0.905 whereas C_2 achieves a better of recall rate of 0.939. The detection performance with C_1 has 10% increase in precision and 0.4% increase in recall whereas 1.6% increase in precision and 1.3% increase in recall are

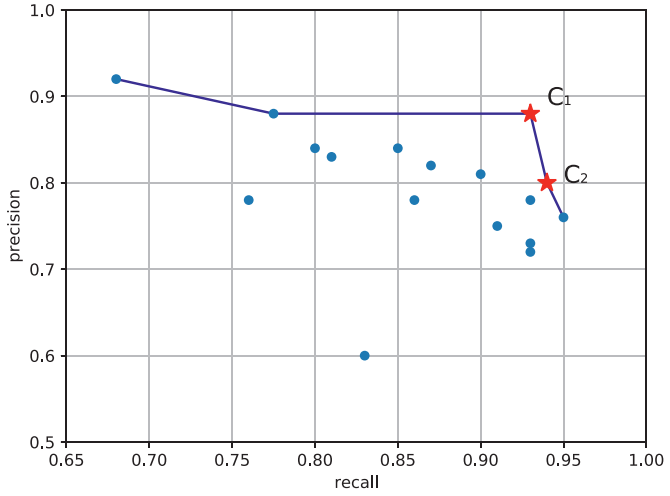

Fig. 5. The Pareto plot of recall-precision on different combinations of th_1 and th_2.The Pareto front is shown in blue solid lines and the two configurations used to compare with the state-of-art networks (see Table 1) are shown in red stars. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

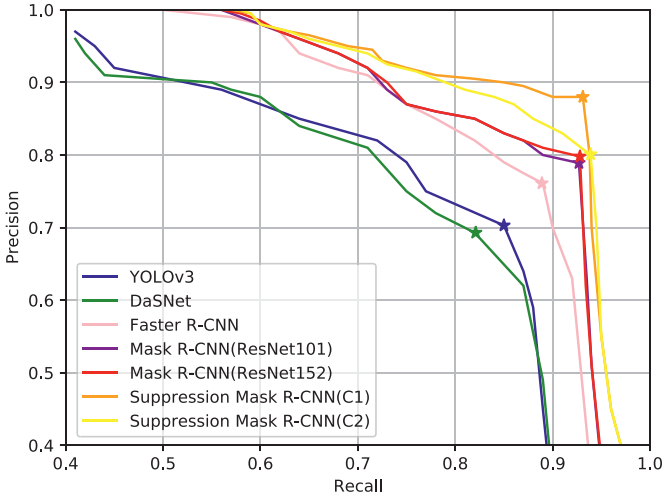

Fig. 6. The plot of recall-precision curves on different approaches. Our proposed Suppression Mask R-CNN networks (in both configurations) outperform the state-of-the-art algorithms.

Table 1
Performance comparison between the state-of-the-art networks and our proposed Suppression Mask R-CNN with two parameter configurations (C_1 and C_2).

	Precision	Recall	F1-score
YOLOv3	0.703	0.860	0.773
DaSNet	0.693	0.821	0.751
Faster R-CNN	0.761	0.889	0.820
Mask R-CNN(ResNet101)	0.789	0.927	0.852
Mask R-CNN(ResNet152)	0.798	0.928	0.858
Suppression Mask R-CNN(C_1)	**0.880**	0.931	**0.905**
Suppression Mask R-CNN(C_2)	0.801	**0.939**	0.864

achieved with configuration C_2. These results demonstrate that in both cases, our integrated class branch and the suppression end approach improve the true detection and the C_2 configuration significantly reduce false fruit detection rates.

5.2. Comparison with the state-of-the-arts

In order to fully evaluate the performance of the proposed approach, we compare our approach with the state-of-the-art apple detection algorithms based on our comprehensive image dataset. The algorithms that we compare with include *YOLOv3* [22], *DaSNet* [19], *Faster R-CNN* [30], and *Mask R-CNN* [23]. These approaches are trained and evaluated on the same training data and test data. For Mask R-CNN, we consider two configurations: ResNet101 backbone and ResNet152 backbone. The recall-precision curves of these approaches are shown in Fig. 6. Furthermore, the precision, recall and F1-score are shown in Table 1. It can be seen in Fig. 6 and Table 1 that the proposed Suppression Mask R-CNN has superior performance as compared to the existing approaches.

Fig. 7. Detection results on different apple varieties under various lighting conditions: (a)-(c) detection on Gala apples under overcast, back lighting, and direct lighting conditions, respectively; and (d)-(e) detection on Blondee apples under overcast, back lighting, and direct lighting conditions, respectively.

5.3. Evaluation on different apple varieties and lighting conditions

In addition, we also evaluate our model in different sub-datasets. Specifically, we separate the whole dataset into several sub-datasets based on apple variety and lighting conditions. The evaluations are summarized in the Table 2 and results are shown

Table 2
Performance evaluation on subset of the data with different apple varieties as well as different lighting conditions.It can be seen that similar performance are obtained in Gala and Blondee apples while back lighting can slightly decrease the performance.

	Dataset					
	Category		Lighting Condition			Total
	Gala	Blondee	Overcast	Direct Lighting	Back Lighting	
Number	3357	1438	3356	959	480	4,795
Precision	0.87	0.89	0.89	0.89	0.84	0.88
Recall	0.93	0.93	0.93	0.93	0.93	0.93
F1-score	0.90	0.91	0.91	0.91	0.88	0.91

in 7. The results show that our model has a better performance for Blondee apples than for Gala. Compared to back lighting conditions, the detection of our model reaches a higher precision under overcast or direct lighting conditions, which indicates that artificial lighting may be helpful for further improving the performance and it will be investigated in our future work.

6. Conclusion

In this study, we collected a comprehensive apple dataset for two varieties of apples with distinct yellow and red colors under different lighting conditions from the real orchard environment. A novel suppression Mask R-CNN was developed to robustly detect apples from the dataset. Our developed feature suppression network significantly reduced false detection by filtering non-apple features learned from the feature learning backbone. Our suppression Mask R-CNN demonstrated superior performance, compared to state-of-the-art models in experimental evaluations.

Our future work will include the incorporation of depth information in the network design to further improve the detection performance. Furthermore, foliage and branches detection will be developed to provide necessary contextual information for the robot to maneuver, e.g., avoiding colliding with tree branches. Lastly, we will also investigate whether artificial lighting augmentation can enhance the detection performance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] K. Gallardo, P. Galinato, 2012 Cost Estimates of Establishing, Producing, and packing Red Delicious Apples in Washington, FS099E, 2012.
[2] F.A. Fathallah, Musculoskeletal disorders in labor-intensive agriculture, Appl Ergon 41 (6) (2010) 738–743 Special Section: Selection of papers from IEA 2009., doi:10.1016/j.apergo.2010.03.003.
[3] D.I. Patrício, R. Rieder, Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review, Comput. Electron. Agric. 153 (2018) 69–81.
[4] D.C. Slaughter, R.C. Harrell, Color vision in robotic fruit harvesting, Transactions of the ASAE 30 (4) (1987) 1144–1148.
[5] P.W. Sites, M.J. Delwiche, Computer vision to locate fruit on a tree, Transactions of the ASAE 31 (1) (1988) 257–0265.
[6] D. Whittaker, G. Miles, O. Mitchell, L. Gaultney, Fruit location in a partially occluded image, Transactions of the ASAE 30 (3) (1987) 591–0596.
[7] M. Benady, G.E. Miles, Locating melons for robotic harvesting using structured light, Paper-American Society of Agricultural Engineers (USA) (1992).
[8] W. Qiu, S. Shearer, Maturity assessment of broccoli using the discrete fourier transform, Transactions of the ASAE 35 (6) (1992) 2057–2062.
[9] M. Cardenas-Weber, A. Hetzroni, G.E. Miles, Machine vision to locate melons and guide robotic harvesting, Paper-American Society of Agricultural Engineers (USA) (1991).
[10] P. Levi, A. Falla, R. Pappalardo, Image controlled robotics applied to citrus fruit harvesting, in: 7th International Conference on Robot Vision and Sensory Controls, Zurich (Switzerland), 2–4 Feb 1988, IFS Publications, 1988.
[11] J. Zhao, J. Tow, J. Katupitiya, On-tree fruit recognition using texture properties and color data, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2005, pp. 263–268.
[12] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, C. McCool, Deepfruits: a fruit detection system using deep neural networks, Sensors 16 (8) (2016) 1222.
[13] S. Bargoti, J.P. Underwood, Image segmentation for fruit detection and yield estimation in apple orchards, J. Field Rob. 34 (6) (2017) 1039–1060.
[14] P.A. Dias, A. Tabb, H. Medeiros, Apple flower detection using deep convolutional networks, Comput. Ind. 99 (2018) 17–28.
[15] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE Trans Pattern Anal Mach Intell 38 (1) (2015) 142–158.
[16] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int J Comput Vis 104 (2) (2013) 154–171.
[17] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.
[18] J. Redmon, A. Farhadi, Yolov3: an incremental improvement, arXiv preprint arXiv:1804.02767 (2018).
[19] H. Kang, C. Chen, Fruit detection and segmentation for apple harvesting using visual sensor in orchards, Sensors 19 (20) (2019) 4599.
[20] D. Bulanon, T. Kataoka, Fruit detection system and an end effector for robotic harvesting of fuji apples, Agricultural Engineering International: CIGR Journal 12 (1) (2010).
[21] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[22] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, Z. Liang, Apple detection during different growth stages in orchards using the improved YOLO-V3 model, Comput. Electron. Agric. 157 (2019) 417–426.
[23] 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, Comput. Electron. Agric. 172 (2020) 105380.
[24] Z. De-An, L. Jidong, J. Wei, Z. Ying, C. Yu, Design and control of an apple harvesting robot, Biosyst. Eng. 110 (2) (2011) 112–122.
[25] A. Dutta, A. Zisserman, The VIA annotation software for images, audio and video, in: Proceedings of the 27th ACM International Conference on Multimedia, in: MM'19, ACM, New York, NY, USA, 2019. DOI:10.1145/3343031.3350535
[26] K. Krishna, M.N. Murty, Genetic k-means algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29 (3) (1999) 433–439.
[27] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[28] X. Wang, A. Shrivastava, A. Gupta, A-Fast-RCNN: hard positive generation via adversary for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2606–2615.
[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[30] S. Wan, S. Goudos, Faster R-CNN for multi-class fruit detection using a robotic vision system, Comput. Networks 168 (2020) 107036.