Smart Agricultural Technology 5 (2023) 100284

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

O2RNet: Occluder-occludee relational network for robust apple detection in clustered orchard environments

Pengyu Chu [a], Zhaojian Li [a,*], Kaixiang Zhang [a], Dong Chen [a], Kyle Lammers [a], Renfu Lu [b]

[a] Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824, USA
[b] Department of Agriculture (USDA) Agricultural Research Service (ARS), East Lansing, MI 48824, USA

ARTICLE INFO

Editor: Spyros Fountas

Keywords:
Computer vision
Apple detection
Fruit harvesting
Occlusion-aware detection
Transfer learning

ABSTRACT

Automated apple harvesting has attracted significant research interest in recent years because of its great potential to address the issues of labor shortage and rising labor costs. One key challenge to automated harvesting is accurate and robust apple detection, due to complex orchard environments that involve varying lighting conditions, fruit clustering and foliage/branch occlusions. Apples are often grown in clusters on trees, which may be mis-identified as a single apple and thus causes issues in fruit localization for subsequent robotic harvesting operations. In this paper, we present the development of a novel deep learning-based apple detection framework, called the Occluder-Occludee Relational Network (O2RNet), for robust detection of apples in clustered situations. A comprehensive dataset of RGB images were collected for two varieties of apples under different lighting conditions (overcast, direct lighting, and back lighting) with varying degrees of apple occlusions, and the images were annotated and made available to the public. A novel occlusion-aware network was developed for apple detection, in which a feature expansion structure is incorporated into the convolutional neural networks to extract additional features generated by the original network for occluded apples. Comprehensive evaluations of the developed O2RNet were performed using the collected images, which outperformed 12 other state-of-the-art models with a higher accuracy of 94% and a higher F1-score of 0.88 on apple detection. O2RNet provides an enhanced method for robust detection of clustered apples, which is critical to accurate fruit localization for robotic harvesting.

1. Introduction

Driven by rising costs and growing shortages in harvesting labor, robotic apple harvesting has gained increased research attention over the past decade. In the U.S. alone, fruit harvesting requires more than *10 million worker hours* annually, attributing to approximately 15% of the total apple production cost [17]. Mechanization and automation promise next-gen harvesting systems with low operating cost and high efficiency, as well as the ability to assess individual fruit for quality and maturity at the point of harvest [32].

As such, several research groups have been developing robotic harvesting systems [26,63,43,10,73]. Despite progresses, several important challenges in developing a fully functional robotic harvesting system remain, and no commercially-viable systems are yet available in the market. One key challenge that is pointed out by the existing works is efficient and robust fruit detection in the presence of varying light conditions and fruit/foliage occlusions. Indeed, the perception system provides the robot system with information on target fruits, which are first and foremost for subsequent planning and control tasks. In addition, fruit perception techniques have also been used in other applications of interest, including yield estimation and crop health status monitoring [42]. Perception in unstructured orchard environments, however, is a daunting task as a result of variations in illumination and appearance, noisy backgrounds, and clustered environments with occlusions [9]. The goal of this paper is thus to present a novel deep learning-based detection algorithm to convergently address the aforementioned challenges. We show that the developed algorithm is able to achieve state-of-the-art performance. Before describing the technical details, we review relevant backgrounds and state-of-the-art approaches to put our algorithm in better context.

1.1. Image sensing techniques

Vision-based perception schemes can be classified into four categories based on the sensor used: monocular camera scheme, binocular stereovision scheme, laser active visual scheme, and thermal imaging

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

https://doi.org/10.1016/j.atech.2023.100284
Received 11 May 2023; Received in revised form 1 July 2023; Accepted 1 July 2023
Available online 11 July 2023

scheme, which cover both two-dimension imaging schemes and three-dimension imaging schemes [76]. Specifically, the monocular scheme uses a single camera to acquire image data, and it is widely used in fruit harvesting due to its low cost and rich information provided by the RGB images. For instance, in [27], the authors proposed a new LedNet model for apple detection that achieves an accuracy of 85.3%. The main disadvantage of the monocular scheme is that the color images are sensitive to fluctuating illumination.

Different from the monocular camera schemes, the binocular stereovision schemes exploit two cameras separated in a certain distance/angle to obtain two image data on the same scene. The point cloud of fruit can then be constructed through triangulation on extracted features [57]. For instance, [51] used a stereo camera to detect and localize mature apples in tree canopies, and achieved an accuracy of 89.5%. In [67], the authors developed a clustered tomato detection method based on a stereo camera, and the recognition accuracy was 87.9%. Although the stereovision scheme tends to render better results, it suffers from high complexity, long computation time, and uncertainties in stereo matching [20].

On the other hand, the laser active visual schemes obtain three-dimensional features using laser scans, where laser beam reflections are exploited to generate a 3D point cloud based on the time-of-flight principle. The 3D point cloud can then be used to reconstruct the scene. For example, [59] utilized infrared laser scanning devices to recognize cherry on the tree. [71] acquired a total of 200 images for independent 'Fuji' apples and developed an apple recognition method using the near-infrared linear-array structured light for 3D reconstruction. [62] proposed a point cloud based apple detection method using a LiDAR laser scanner and reached a 88.2% overall accuracy on the defoliated tree dataset [62]. Note the defoliated scene is significantly less challenging than the real orchard conditions during the harvest season. Furthermore, the laser point cloud is generally sparse and it is challenging to be used in real-world orchards with dense backgrounds. The high cost and complexity also limit its practical application in agricultural applications.

Finally, the thermal imaging schemes make use of the distinct thermal characteristics of fruit and leaves (e.g., the different temperature distributions) to obtain the visualization of infrared radiation [35]. In [5], citruses are successfully segmented using a thermal infrared camera according to the largest temperature difference in both day and night conditions. An enhanced approach for fruit detection [6] was developed using the combination of the thermal image and the color image. The results showed a promising performance under weak lighting environments. However, in the thermal imaging scheme, the accuracy of recognition is largely affected by the shadow of the tree canopy [55].

Considering the cost, performance, and real-time constraints, our work focuses on the monocular camera scheme, the state-of-art of which will be discussed next.

1.2. Recognition approaches

Image-based fruit recognition approaches can be classified into *feature analysis* approaches and *deep learning-based* approaches, depending on how features are obtained. In *feature analysis* approaches, handcrafted features are first extracted based on the fruit characteristics, and classification approaches are then developed to recognize fruit. [54,53] developed thresholding methods to classify fruit from other background objects using smoothing filters that remove irrelevant noises. The large segmented regions are then recognized as fruits. This method is capable of segmenting fruit regions in simple backgrounds but it is susceptible to varying lighting conditions and complex canopies. [64,3] proposed a circular Hough Transform approach to obtain binary edge images and then used a voting matrix to identify fruits. This approach is sensitive to complex structured environments and it generally fails in a dense scene. In [44,7,31,75], they combined the shape and texture of the fruit to obtain a richer set of feature representations. Then, extracted features between fruit and leaves are compared and contrasted to identify the fruits. However, this method is also sensitive to lighting conditions and occlusions.

On the other hand, deep learning-based approaches have found great successes in object detection and semantic image segmentation [49,2], which can learn feature representations automatically without the need of manual feature engineering. Specifically, Convolutional Neural Networks (CNNs) have showed great advantages in the field of object detection in recent years, making it possible to recognize fruits in complex situations due to its deep extraction of high-dimensional features of objects. R-CNN and its variants Fast R-CNN and Faster R-CNN [19,18,47] have enjoyed particular successes. For instance, [72] deployed the fine-tuned Faster R-CNN using the pre-trained network VGG19 [52] and achieved a precision of 82.4% for apple detection. Modified Inception-ResNet (MI-ResNet) [45] used deep simulated learning for yield estimation to address challenges including the varying degrees of fruit sizes and natural lighting conditions. You Only Look Once (YOLOv3) [46], a representative of the one-stage object detector, uses logistic regression to predict an objectless score for each bounding box. An improved YOLOv3 model [60] was developed to detect apples with a precision of 85.0%. Due to the simple optimization pipeline, YOLO enjoys much faster inference than the aforementioned region-based methods. EfficientDet [58], an augmented variant of YOLOv3, exploits a pyramid network to enable the detection of scaling targets. In [28], they evaluated EfficientDet in their customized apple dataset with a precision of 75.65%, while their proposed model FruitDet reached a precision of 80.78%.

However, the aforementioned deep CNN approaches do not address the challenge of fruit/foliage occlusions in real-world orchards. Towards that end, Compositional Convolutional Neural Network (CompNet) [30] was proposed to detect partially occluded objects. The framework exploits a differentiable fully compositional model that uses occluder kernels to localize occluders (the occluding objects). Bilayer Convolutional Network (BCNet) [29], another model to address the occlusion challenge, applies two Graph Convolutional Network (GCN) layers to separately infer the occluding objects (occluder) and partially occluded instance (occludee). Superior performance was reported on occluded scenarios. In apple detection, various approaches are developed to enhance the performance of deep learning-based models in complex orchards. [21] introduced CBL (Convolutional layers, Batch normalization, Leaky-relu activation function [13]) module and CA (coordinate attention) module into YOLOv5 [24], and finally increased 4.41% in the precision compared to the base model. [68] utlized a customized YOLOv3 to reach a recall of 93.4% for overlapped apples. These two approaches are trying to extract the higher-level features by modifying models to improve the performance. Different from above, [16] took advantage of a depth filter to remove background trees with a RGB-D camera and finally improved apple detection precision by 2.5% on overlapped apples. This paper will model the relationships among overlapped apples and enhance the apple edge features to improve the precision for clustered apples.

1.3. Our contributions

In this paper, we develop a novel Occluder-Occludee Relational Network (O2RNet) to enhance apple detection in the presence of occlusions in clustered apples that are frequently present in real-world orchards. Specifically, we employ ResNet [22] and RPN [47] to extract features of targets and utilize occluder-occludee layers to split candidates into occluder and occludee. Compared to other occlusion models, we only use bounding boxes as labels instead of pixel-level masks that contain more texture and shape information. In addition, we present a new apple

dataset[1] collected in two Michigan apple orchards in multiple harvesting seasons. We evaluate the performance against state-of-the-art object detection models and demonstrate superior performances. The contributions of this paper are highlighted as follows:

1. A comprehensive apple dataset of 1246 images for two varieties of apple under different lighting conditions and occlusion levels were collected from two orchards during two harvesting seasons.
2. A novel Occluder-Occludee Relational Network (O2RNet) was developed for enhanced apple detection in the presence of occlusions due to apple clusters.
3. The O2RNet outperformed 12 state-of-the-art deep learning-based models for apple detection.

2. Materials and data processing

2.1. Image collection and annotation

In this study, apple images were taken in two orchards: the commercial orchard in Sparta, Michigan, USA during the 2019 harvest season and the experimental orchard of Michigan State University in East Lansing, Michigan, USA during the 2021 harvest season. The apples are mainly 'Gala' that are generally red over a green/yellow background (see Fig. 1). An RGB camera (RealSense D435i) with a resolution of 1280×720 was used to take images of apples at a distance of $1-2$ meters from the tree trunks, which is the typical range of harvesting robots [10,73,74]. 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 9am, noon, and 3pm, to cover different lighting angles: front-lighting, back-lighting, side-lighting, and scattered lighting. Furthermore, we also captured clustered apples with different occlusion levels including both foliage and branches occlusion. When capturing images, the camera was placed parallel to the ground and directly facing the trees to mimic the harvesting scenario. For training our model, we split the dataset a total of 1246 images into 934 and 312 as the training sets and the test sets respectively. A few typical sample images for the two varieties of apples under three lighting conditions are shown in Fig. 1.

The acquired raw images were then processed 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 [14], and the annotations were then compiled into the human-readable format. Compared to polygon and mask annotations, rectangular annotations used here would accelerate data preparation for our dataset of dense images. The annotated dataset was then split into training and test subsets with the apple quantities of 10523, and 3995 respectively. The processed image database can be accessed publicly at https://github.com/pengyuchu/MSUAppleDatasetv2.git.

2.2. Transfer learning

We have employed transfer learning to enable faster training and improved performance. Transfer learning is a popular scheme that starts the model development with a pre-trained model on a large-scale dataset and then fine-tunes the model on a customized dataset from the specific domain of interest [77]. For apple detection in this study, we used ImageNet [11] to pre-train each model and only replaced the last fully-connected layers in each model. Since there are objects of apples and alike in ImageNet, the pre-trained models converge faster in our customized apple dataset compared to randomized initial parameters.

[1] The database is open-sourced at https://github.com/pengyuchu/MSUAppleDatasetv2.git.

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

2.3. Performance metrics

For model development and evaluation, conventionally the apple dataset is randomly partitioned into training, validation, and test sets for model training and evaluation, respectively. 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. The precision (P) and recall (R) are defined as follows:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}. \tag{1}$$

The F1-score is then subsequently defined as:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}. \tag{2}$$

To better evaluate the precision between the prediction and the ground truth, we also employ Microsoft Common Objects in Context (COCO) dataset [33] evaluation metrics. Specifically, after the calculation of precision and recall, we calculate the average precision (AP) and average recall (AR) based on different Intersection over Union (IoU) between the prediction and the ground truth. For example, $AP_{IoU=.50}$ or AP_{50} denotes that AP is averaged over $IoU = 0.50$ values, which belongs

to PASCAL VOC metric [48]. We also use $AP_{IoU=.75}$ or AP_{75}, which is a stricter metric for model evaluations. In our study, we use a spectrum of 10 IoU thresholds ranging 0.50 : 0.05 : 0.95 to average over multiple IoUs to obtain a comprehensive set of results.

2.4. Data augmentation

Data augmentation is a method that can be adopted to increase data diversity for achieving robust training and enhanced performance of computer vision models. For example, transformations and rotations are frequently employed to increase the diversity of images from a single source. It has been shown to be a powerful tool in agriculture applications [65,56,12] as it generates distinct data from existing orchard data. This is especially useful for applications with a limited dataset by detecting anomalies in images with different transformations and making it possible to generate new training examples without actually acquiring new data.

Specifically, in the considered application of apple detection in orchards, the collected dataset can only cover a limited set of scenarios. Therefore, we applied several data augmentation techniques [8] on the collected and processed data to enhance the data diversity for improving the inference performance of our models. Specifically, besides geometric transformations including scaling, translating, rotating, reflecting, and shearing, we also applied color space augmentations such as modifying the brightness and contrast to fit different intensities. In addition, we injected Gaussian noises on the collected images by randomly modifying the pixel intensities based on a Gaussian distribution. Furthermore, we applied Mixup by randomly selecting two images from the dataset and blending the intensities of the corresponding voxels of the two images [36]. Filtering is another augmentation approach we applied where we modify the intensities of each pixel using convolution [50]. Specifically, we exploited sharpening [50] to detect and intensify the edges of objects found within the image. We applied these additional augmentation techniques on our dataset and the benefits of data augmentation will be demonstrated in the experiment section.

3. Methodology

In this section, we first present the key challenges of object detection in clustered environments and an overview of the general object detection framework. Based on those, we describe the proposed Occluder-Occludee Relational Network (O2RNet) with explicit occluder-occludee relation modeling. Finally, we specify the objective functions for the entire network optimization, followed by details on the training and inference processes as well as performance evaluation metrics.

3.1. Challenge and main idea

For images with heavy occlusions, multiple overlapping objects captured in the same bounding box can result in confusing object outlines from both front objects and occlusion boundaries. In apple orchards, apple clusters are quite common (see Fig. 2 for a few examples). However, the prediction head design of Faster R-CNN directly regresses the occludee with a fully convolutional network, which neglects both the occluding instances and the overlapping relations between objects. With this limitation, Faster R-CNNs will inevitably omit some occludees due to Non- maximum Suppression (NMS). On the other hand, with a properly tuned threshold, the RPN can propose many candidates after feeding the target features from CNN (see Fig. 3), but the NMS will suppress the nearby bounding boxes and neglect occludees. Motivated by this observation, the proposed O2RNet aims at extending the existing two-stage object detection methods by adding an occlusion perception branch parallel to the original object prediction pipeline. By explicitly modeling the relationship between occluder and occludee, the interactions between objects within the Region of Interest (RoI) region can be well incorporated during the bounding box regression stage.

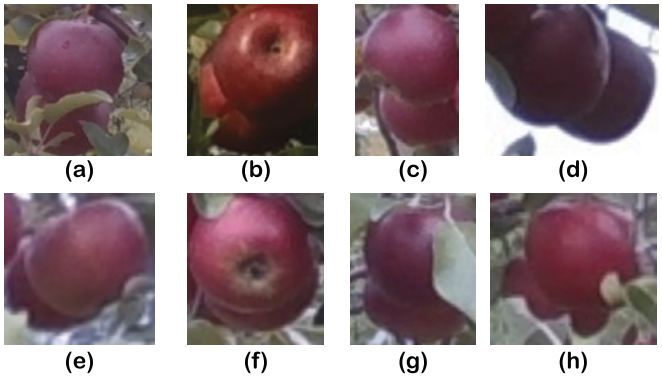

Fig. 2. Eight sample images from the collected dataset show cascaded apples at different occlusion levels: (a)-(d) apples are in the normal occlusion and can be identified by most models; (e)-(h) apples are highly cascaded and would be easily detected as one apple.

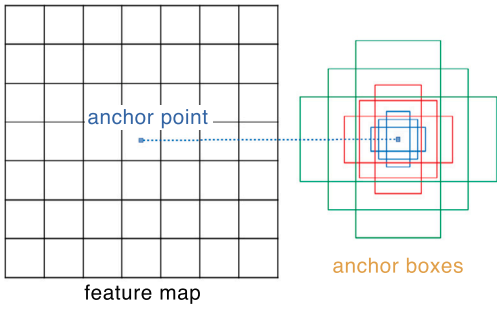

Fig. 3. Illustration of how RPN works. The RPN selects anchor points on the feature map and generates anchor boxes for each point. The anchor boxes are generated based on the two parameters, i.e., scales and aspect ratios.

3.2. O2RNet workflow

As illustrated in Fig. 4, the O2RNet follows the two-stage architecture used in Faster R-CNN [47] and consists of three main parts. First, we use a Residual Network (ResNet) [22] as the backbone for feature learning/extraction over the entire image. Specifically, the ResNet-101-FPN [23] is instantiated as its backbone for feature extraction, as it outperforms other single ConvNets mainly due to its capability of maintaining strong semantic features at various resolution scales. Even though ResNet-101 is a deep network, the residual blocks and dropouts function help it avoid gradient vanishing and exploding problems. Second, an RPN [47] is employed to generate object regions, which is a small convolutional network to convert feature maps into scored region proposals around which the object lies. The generated proposals with a 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 chosen to be consistent with object sizes in the dataset. RPN is mainly used for predicting bounding boxes in Faster R-CNN, but it can also provide enough anchors with different scales that was further exploited in our network as explained in the sequel. Third, an occlusion-aware modeling head with a structure of two classification and regression branches is built for the occluder and occludee for decoupling overlapping relations and segments the instance proposals obtained from the RPN. Compared to the conventional class-agnostic classification, this task is divided into two complementary tasks: occluder prediction using the original classification head and occludee modeling with an additional Feature Expansion Structure (FES), where the occluder predictions provide rich foreground cues like textures and the FES predicts the positions of occluding regions to guide occludee object regression.

More specifically, an input image is first processed by the ResNet backbone to extract intermediate convolutional features for down-

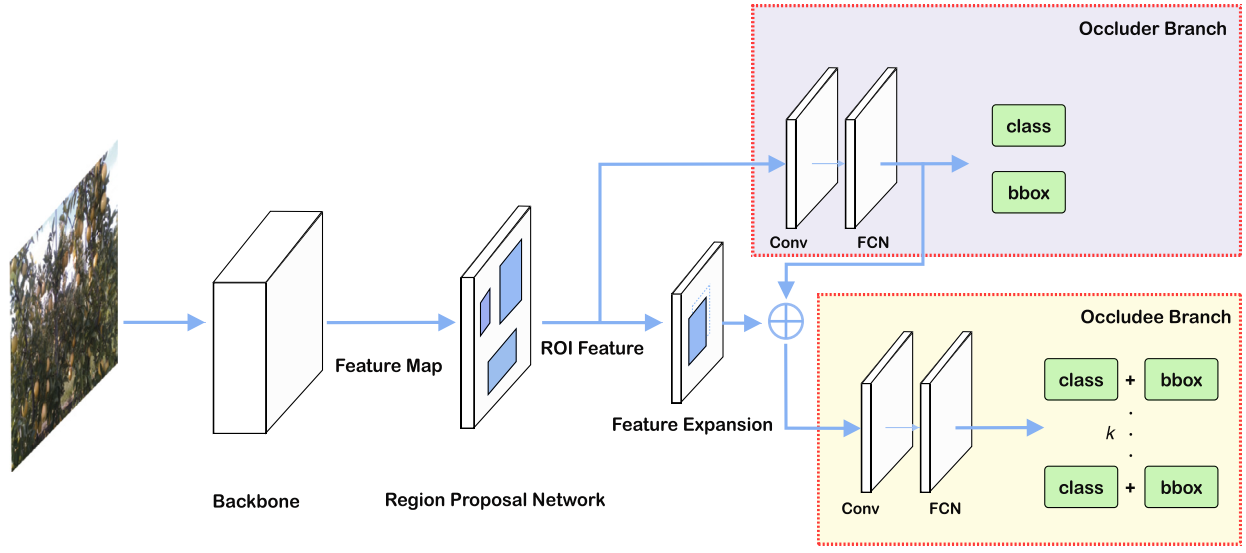

Fig. 4. Network structure of the proposed Occluder-Occludee Relational Network (O2RNet). It consists of a feature learning backbone, RoI feature extraction, and object detection heads with occluder and occludee branches. The Feature Expansion Structure (FES) provides expanded RoI features along with features from the occluder branch to facilitate the detection of occludee.

stream processing. The object detection head (i.e., RPN) then predicts bounding box proposals, which are then consumed by the occlusion perception branches into the occluder branch and the occluee branch. For the occluder branch, we adopt the object detection head in Faster R-CNN [47] to output positions as well as categories for instance candidates and prepare the cropped RoI features for the occludee branch. In the occludee branch, the input consists of both cropped RoI features from the occluder branch and expanded features from FES, which is targeted for modeling occluded regions by jointly detecting boundaries. Essentially, the distilled occlusion features are added to the original input RoI features and passed to the next module. Finally, the occludee branch, which has a similar structure to the occluder branch, predicts the occludee guided by these expanded features and outputs classes and bounding boxes for the partially occluded instances. The occluder-occludee relational modeling is discussed in more details in the following section.

3.3. Occluder-occludee relationship modeling

For highly overlapped apples, in typical Faster R-CNN-based models, the generated region proposals corresponding to the partially occluded ones may be separated into disjoint subregions by the occluder. As such, we employed the FES to obtain boundary features from the occludee, where expansion in each direction extends the potential proposals for the occludee. In our implementation, we expand t steps in k ($k = 8$ in this study) directions from the original RoI proposals, and the expanded RoI proposals contain additional boundary features. The rationale is that irregular occlusion boundaries unrelated to the occludee can cause confusion to the network, which in turn provides essential cues for decoupling occludees from occluders. Therefore, occlusion patterns are explicitly modeled by detecting bounding boxes of the occluders using the occluder detection branch, and since the occludee detection branch jointly predicts bounding boxes for the occludee, the overlap between the two layers can be directly identified as occlusion boundary that can thus be distinguished from the real object bounding boxes. In order to reach this goal, the occluder modeling module is designed as a simple 3×3 convolutional layer followed by one FCN layer, the output of which is fed to the up-sampling layer and one 1×1 convolutional layer to obtain one channel feature map for occludee branch.

3.4. End-to-end learning

As we have two separate detection heads in the occluder and the occludee branches, we define two loss functions in the following way. For the occluder branch, we adopt the loss function used in Faster R-CNN [47], which defines a multi-task loss on each sampled region of interest as

$$L_{Occluder} = L_{cls} + L_{bbox}, \tag{3}$$

where L_{cls} and L_{bbox} are, respectively, classification loss and bounding box loss defined in Faster R-CNN [47].

The final loss L is a weighted sum of the loss from occluder branch and the loss from occludee branch defined as:

$$L = \lambda_1 L_{Occluder} + \lambda_2 L_{Occludee}. \tag{4}$$

Here $L_{Occludee}$ is the occludee branch loss that is the sum of the k expanded proposal losses, i.e.,

$$L_{Occludee} = \sum_{i=0}^{k} (L^i_{cls} + L^i_{bbox}). \tag{5}$$

Here λ_1 and λ_2 are two positive linear weights and $\lambda_1 + k \cdot \lambda_2 = 1$, which are tuned to balance the two loss functions. In our study, λ_1 was tuned to be $\{1.0, 0.75, 0.5, 0.25, 0\}$ on various trials for cross-validation.

3.5. Training and inference

During the training process, we filter out parts of the non-occluded RoI proposals to keep occlusion cases taking up 50% for balanced sampling. SGD with momentum is employed to train the model with $60K$ iterations where it starts with $1K$ constant warm-up iterations. The batch size is set to 2 and the initial learning rate is 0.01 with a weights decay of 0.95. In our study, ResNet-101-FPN is used as the backbone and the input images are resized without changing the aspect ratio, i.e., by keeping the shorter side and longer side of no more than 1200 pixels. For inference, the occludee branch predicts bounding boxes for the occluded target object in the high-score box proposals generated by the RPN, while the occluder branch produces occlusion-aware features as input for the occludee branch. The one with the highest score is then chosen as the output.

Table 1
Performance of O2RNet on the customized apple dataset. The step is from FES, which represents how much features expanded. The evaluation uses AP, AR, and F1-score at the different IoUs.

Model	Step	AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}	F1-Score
O2RNet	**t=1**	**0.511**	**0.945**	**0.935**	**0.351**	**0.938**	**0.803**	**0.864**
	t=2	0.490	0.920	0.900	0.330	0.900	0.770	0.820
	t=3	0.490	0.920	0.904	0.328	0.900	0.770	0.820

Table 2
Model parameters numbers and inference time between the state-of-the-art networks and our proposed Occluder-occludee Relational Network (O2RNet).

Models	Parameters ($\times 10^7$)	Time (ms)
FCOS	2.6	80
YOLOv8	4.2	30
Faster R-CNN (ResNet50)	2.6	75
Faster R-CNN (ResNet101)	4.5	100
EfficientDet-b0	0.4	35
EfficientDet-b1	0.7	50
EfficientDet-b2	0.8	70
EfficientDet-b3	1.2	100
EfficientDet-b4	2.1	160
EfficientDet-b5	3.4	250
CompNet via VGG	1.4	90
CompNet via RPN	4.5	100
O2RNet (ResNet50)	2.6	80
O2RNet (ResNet101)	4.5	110

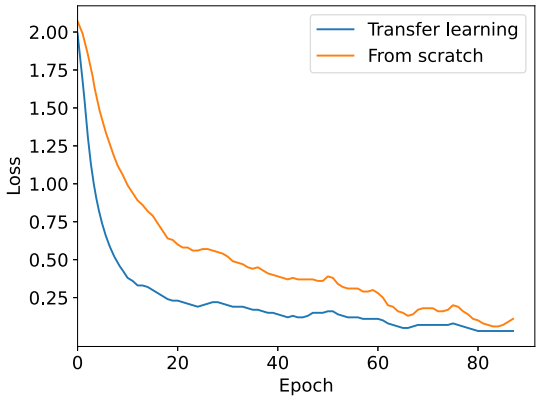

Fig. 5. Training loss comparison between transfer learning and training from scratch on our model (O2RNet). The training loss with transfer learning from ImageNet apparently decreases and converges faster as compared with training from scratch.

4. Results and discussions

4.1. Experimental setup

In this section, we evaluate the efficacy of the proposed O2RNet on the processed data as discussed in Section 2.1. The network hyperparameters, 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 1280×720, which is aligned with the resolution of the camera used in our data collection. To better analyze the training process, we set up 80 epochs for training. We exploit a pre-trained model on the COCO dataset [33], where we train on 2017train (115k images) and evaluate results on both 2017val and 2017test-dev to pre-train model parameters. This pre-trained model generally only takes 50 epochs to converge. By tuning the steps t in FES, different results are obtained and listed in Table 1, which shows that O2RNet with $t = 1$ leads to the best performance.

4.2. Performance comparison and analysis

To accelerate the model training on our customized dataset, we initialized parameters by transfer learning from ImageNet [11]. ImageNet provides large-scale images in different fields (including apples) and large- scale ground truth annotation. During the transfer learning process, our model learned specific characteristics with an effective transfer of features from ImageNet. Compared to randomized parameters, the results (see Fig. 5) shows that our model converges faster as benefited from the pretraining on a large-scale database.

Furthermore, data augmentation is another useful technique to optimize detection performance without increasing inference complexity. We applied five augmentation strategies, including geometric transformations (GTs), color space transformations (CSTs), Gaussian noise injection, mixup and sharpening data augmentation, to extend our dataset. The results are summarized in Table 3. It shows that GTs such as rotation, flipping and scaling – by changing the pixel position of the image and reordering apples in the image – improve the accuracy performance by around 1%. Through changing color illumination and intensity of an image, CSTs also roughly increases the performance by 1%. Due to the sparsity of apples on some images, mixup helps enlarge apple density on the image and enhances the accuracy by 2%. It turns out that Gausian noise and sharpening do not help much, as they try to change textures and increase complexities on the dataset, which generate confusing data and is not suitable for our model. Finally, the augmentation combination of GTs, CSTs and Mixup offers the best enhancement by increasing the accuracy of 4% on our dataset.

To better evaluate the performance of our model, we compare our O2RNet with the-state-of-art object detection methods on our customized apple dataset (see Table 2 for a list of benchmark models and their number of parameters). The performance of these models were tested on an Ubuntu 20.04 with an AMD 3990X 64-Core CPU and a GeForce RTX 3090Ti GPU (24 GB GDDR6X memory). Besides, the libraries Pytorch [41] and Detectron2 [66] were also used to perform these deep learning functions over CUDA 10 [39]. In particular, FCOS and YOLOv8 are representatives of one-stage detectors, achieving consistent improvement and demonstrating their effectiveness by outperforming the SSD method [34] on several public datasets [61,4]. We also evaluate Faster R-CNN and EfficientDet since they are state-of-the-art models with promising performance demonstrated in fruit harvesting-related works [37,69]. We also compare O2RNet with the state-of-the-art occlusion-aware network CompNet [15].

We then use the same experimental setup to train each model and evaluate them on the same apple test dataset. The results are shown in Table 4, which compares the detection precision and recall over different IoUs among the 14 selected models (including our O2RNet). Notably, in addition to FCOS, EfficientDet-b5 and Faster R-CNN achieved decent F1-scores of 0.83 and 0.82, respectively. Two occlusion-aware networks, CompNet and our O2RNet clearly outperform all traditional models with F1-scores of 0.86 and 0.88, respectively, and O2RNet clearly shows superior performance over CompNet. It can be seen that our O2RNet can achieve a great detection performance in the precision and recall and subsequently the F1-score.

In our approach, we focus on advancing the accuracy of detection for clustered apples, which shows significant challenges due to occlusion. To exactly present how our O2RNet improves the detection accuracy, we select a total 832 overlapped apples from the test dataset as the clustered cases. Our results (see Table 5) are then benchmarked against those 12 state-of-the-art methods mentioned before. The em-

Table 3
Performance of O2RNet on the augmented dataset. The geometric transformations consist of rotation, flipping and scaling. The color space transformations consist of brightness and contrast shifting. Finally, all of the augmentation methods are integrated to evaluate the O2RNet.

Augmentation	AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}	F1-Score
Base	0.51	0.92	0.90	0.35	0.91	0.80	0.84
Geometric transformations (GTs)	0.52	0.93	0.91	0.35	0.91	0.80	0.85
Color space transformations (CSTs)	0.52	0.93	0.91	0.35	0.91	0.81	0.85
Gausian noise	0.48	0.91	0.90	0.34	0.91	0.80	0.83
Mixup	0.52	0.93	0.92	0.35	0.92	0.81	0.85
Sharpening	0.52	0.92	0.90	0.35	0.91	0.80	0.84
GTs+CSTs+Mixup	**0.52**	**0.96**	**0.94**	**0.36**	**0.94**	**0.83**	**0.88**
All	0.52	0.94	0.92	0.36	0.92	0.83	0.86

Table 4
Performance comparison of our own models and other 12 state-of-the-art deep learning models on the customized apple dataset.

Models		AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}	F1-score
FCOS [1]		0.48	0.89	0.87	0.34	0.87	0.78	0.80
YOLOv8 [25]		0.48	0.90	0.87	0.32	0.86	0.77	0.81
Faster R-CNN	ResNet50 [38]	0.48	0.89	0.87	0.32	0.87	0.78	0.81
	ResNet101 [38]	0.49	0.94	0.93	0.31	0.84	0.75	0.82
EfficientDet	EfficientDet-b0 [40]	0.45	0.89	0.85	0.30	0.82	0.71	0.77
	EfficientDet-b1 [40]	0.45	0.89	0.86	0.30	0.82	0.72	0.77
	EfficientDet-b2 [40]	0.46	0.89	0.87	0.30	0.82	0.73	0.78
	EfficientDet-b3 [40]	0.49	0.93	0.91	0.32	0.84	0.75	0.81
	EfficientDet-b4 [40]	0.50	0.94	0.92	0.34	0.88	0.78	0.82
	EfficientDet-b5 [40]	0.50	0.95	0.93	0.34	0.88	0.78	0.83
CompNet	CompNet via VGG [70]	0.50	0.94	0.92	0.36	0.94	0.80	0.85
	CompNet via RPN [15]	0.51	0.95	0.94	0.35	0.94	0.80	0.86
O2RNet	O2RNet-ResNet50	0.50	0.93	0.91	0.35	0.91	0.80	0.84
	O2RNet-ResNet101	**0.52**	**0.96**	**0.94**	**0.36**	**0.94**	**0.83**	**0.88**

Table 5
Performance comparison of our own models and other 12 state-of-the-art deep learning models on the 832 clustered apples.

Models		AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}	F1-score
FCOS [1]		0.29	0.71	0.63	0.22	0.69	0.53	0.57
YOLOv8 [25]		0.36	0.76	0.69	0.22	0.71	0.59	0.64
Faster R-CNN	ResNet50 [38]	0.33	0.71	0.65	0.21	0.68	0.59	0.62
	ResNet101 [38]	0.36	0.75	0.69	0.23	0.70	0.61	0.65
EfficientDet	EfficientDet-b0 [40]	0.30	0.71	0.64	0.30	0.65	0.57	0.60
	EfficientDet-b1 [40]	0.30	0.71	0.64	0.21	0.65	0.58	0.61
	EfficientDet-b2 [40]	0.32	0.72	0.66	0.21	0.67	0.59	0.62
	EfficientDet-b3 [40]	0.34	0.74	0.67	0.21	0.68	0.59	0.63
	EfficientDet-b4 [40]	0.36	0.75	0.69	0.23	0.70	0.61	0.65
	EfficientDet-b5 [40]	0.36	0.76	0.69	0.24	0.71	0.61	0.65
CompNet	CompNet via VGG [70]	0.36	0.75	0.69	0.24	0.69	0.61	0.65
	CompNet via RPN [15]	0.38	0.77	0.71	0.26	0.72	0.63	0.67
O2RNet	O2RNet-ResNet50	0.44	0.85	0.80	0.31	0.80	0.71	0.75
	O2RNet-ResNet101	**0.46**	**0.87**	**0.82**	**0.33**	**0.82**	**0.72**	**0.77**

pirical results demonstrate that our method outperforms the existing state-of-the-art methods with an increase of 11%, 9% and 10% in terms of precision, recall and F1-score. Some representative inference results are shown in Fig. 6. This superior performance is indicative of the efficacy of O2RNet in dealing with the clustered cases in the complex apple orchard through enhancing the boundary features of overlapped apples.

5. Conclusion

In this study, we collected a comprehensive apple dataset under different lighting conditions and at various occlusion levels from two orchards. A novel Occluder-Occludee Relational Network (O2RNet) was developed to robustly detect clustered apples from the dataset. Our developed O2RNet significantly reduced false detection and improved the detection rate by embedding relationships between the occluder and the occludee. As a result, our model consistently outperformed 12 other state-of-the-art models when evaluated using our apple image dataset. It was found that transfer learning and data augmentation techniques were useful tools to enhance learning efficiency and model performance.

Our future work will include the incorporation of foliage information in the network design to further improve the detection performance since the current work only focused on clustered apples. Furthermore, branch detection will be developed to provide necessary contextual in-

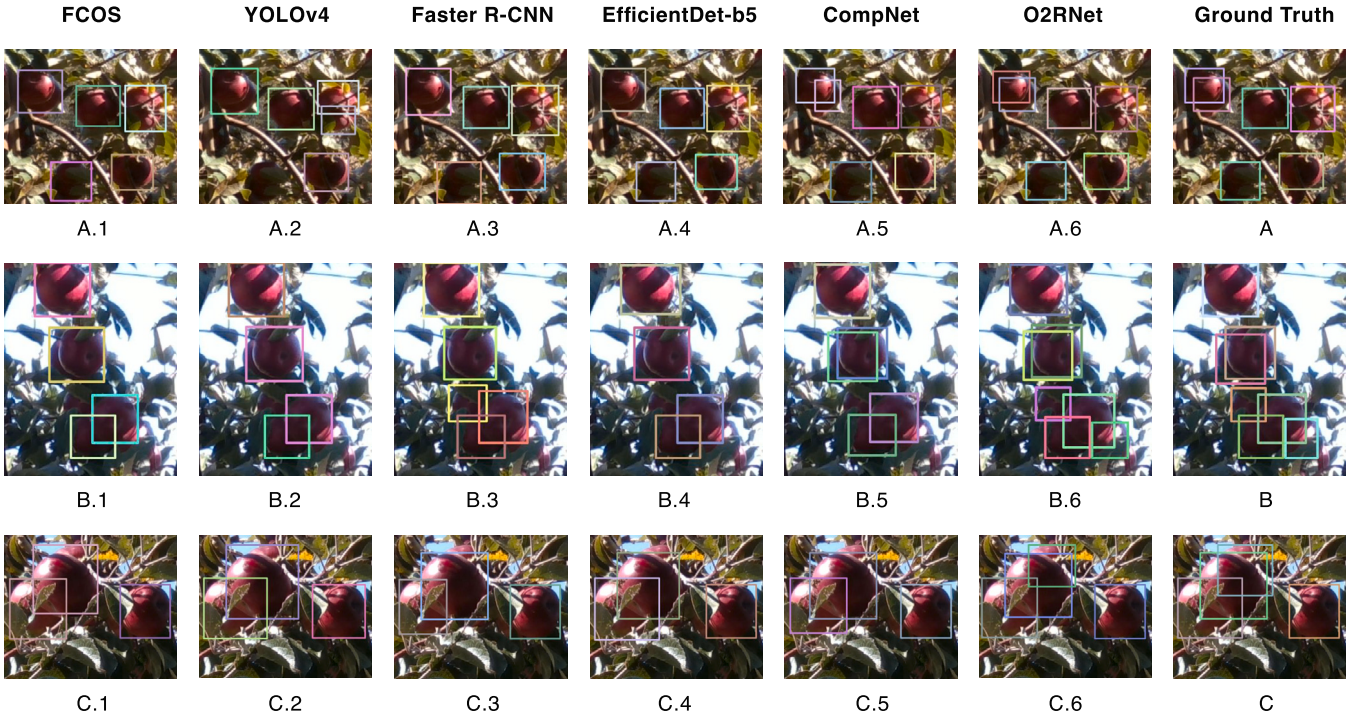

Fig. 6. Results from six models on the various lighting conditions and occlusions.

formation for the robot to maneuver, e.g., avoiding collisions with tree branches. Lastly, we will also investigate whether artificial lighting augmentation can enhance the detection performance.

CRediT authorship contribution statement

Pengyu Chu: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Zhaojian Li:** Resources, Supervision, Writing – review & editing. **Kaixiang Zhang:** Data curation, Writing – review & editing. **Dong Chen:** Writing – review & editing. **Kyle Lammers:** Data curation. **Renfu Lu:** Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgement

This project was partially funded by NSF ECCS 2024649 and the USDA-ARS inhouse project 5050-43640-003-000D. The authors would also like to acknowledge Michigan State University's Teaching and Research Center in Holt, Michigan and Schawllier's Country Basket in Sparta, Michigan for their support for collecting the image data in the orchards. The authors thank Mr. Lingxuan Hao ans Wanqun Yang for helping label the orchard images.

References

[1] A. Ahmad, D. Saraswat, V. Aggarwal, A. Etienne, B. Hancock, Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems, Comput. Electron. Agric. 184 (2021) 106081.

[2] S. Bargoti, J.P. Underwood, Image segmentation for fruit detection and yield estimation in apple orchards, J. Field Robot. 34 (6) (2017) 1039–1060.

[3] M. Benady, G.E. Miles, Locating melons for robotic harvesting using structured light, in: Paper-American Society of Agricultural Engineers (USA), 1992.

[4] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: optimal speed and accuracy of object detection, preprint, arXiv:2004.10934, 2020.

[5] D. Bulanon, T. Burks, V. Alchanatis, Study on temporal variation in citrus canopy using thermal imaging for citrus fruit detection, Biosyst. Eng. 101 (2) (2008) 161–171.

[6] D. Bulanon, T. Burks, V. Alchanatis, Image fusion of visible and thermal images for fruit detection, Biosyst. Eng. 103 (1) (2009) 12–22.

[7] 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).

[8] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, A. Haworth, A review of medical image data augmentation techniques for deep learning applications, J. Med. Imag. Radiat. Oncol. 65 (5) (2021) 545–563.

[9] P. Chu, Z. Li, K. Lammers, R. Lu, X. Liu, Deep learning-based apple detection using a suppression mask r-cnn, Pattern Recognit. Lett. 147 (2021) 206–211.

[10] 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.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei Imagenet, A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[12] L. Divyanth, D. Guru, P. Soni, R. Machavaram, M. Nadimi, J. Paliwal, Image-to-image translation-based data augmentation for improving crop/weed classification models for precision agriculture applications, Algorithms 15 (11) (2022) 401.

[13] A.K. Dubey, V. Jain, Comparative study of convolution neural network's relu and leaky-relu activation functions, in: Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018, Springer, 2019, pp. 873–880.

[14] A. Dutta, A. Zisserman, The VIA annotation software for images, audio and video, in: Proceedings of the 27th ACM International Conference on Multimedia, MM '19, New York, NY, USA, ACM, ISBN 978-1-4503-6889-6, 2019.

[15] S.A. Fennimore, M. Cutulle, Robotic weeders can improve weed control options for specialty crops, Pest Manag. Sci. 75 (7) (2019) 1767–1774.

[16] 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 (2020) 245–256.

[17] K. Gallardo, P. Galinato, 2021 cost estimates of establishing, producing, and packing red delicious apples in washington, in: FS099E, Washington State University Extension Fact Sheet, 2012.

[18] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[19] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[20] M.W. Hannan, T.F. Burks, Current developments in automated citrus harvesting, in: 2004 ASAE Annual Meeting, American Society of Agricultural and Biological Engineers, 2004, p. 1.
[21] Q. Hao, X. Guo, F. Yang, et al., Fast recognition method for multiple apple targets in complex occlusion environment based on improved yolov5, J. Sens. (2023) 2023.
[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[23] 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.
[24] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, K. Michael, J. Fang, C. Wong, Z. Yifu, D. Montes, et al., ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations, Zenodo (2022).
[25] G. Jocher, A. Chaurasia, J. Qiu, YOLO by Ultralytics, https://github.com/ultralytics/ultralytics, Jan. 2023.
[26] H. Kang, C. Chen, Fruit detection and segmentation for apple harvesting using visual sensor in orchards, Sensors 19 (20) (2019) 4599.
[27] H. Kang, C. Chen, Fast implementation of real-time fruit detection in apple orchards using deep learning, Comput. Electron. Agric. 168 (2020) 105108.
[28] F.A. Kateb, M.M. Monowar, M. Hamid, A.Q. Ohi, M.F. Mridha, et al., Fruitdet: attentive feature aggregation for real-time fruit detection in orchards, Agronomy 11 (12) (2021) 2440.
[29] L. Ke, Y.-W. Tai, C.-K. Tang, Deep occlusion-aware instance segmentation with overlapping bilayers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4019–4028.
[30] A. Kortylewski, J. He, Q. Liu, A.L. Yuille, Compositional convolutional neural networks: a deep architecture with innate robustness to partial occlusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8940–8949.
[31] 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, pp. 2–4.
[32] B. Li, A. Zhou, C. Yang, S. Zheng, The design and realization of fruit harvesting robot based on iot, in: 2016 International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2016), Atlantis Press, 2016, pp. 158–161.
[33] 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.
[34] 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, October 11–14, 2016, Proceedings, Part i 14, Springer, 2016, pp. 21–37.
[35] J. Lu, N. Sang, Y. Hu, H. Fu, Detecting citrus fruits with highlight on tree based on fusion of multi-map, Optik 125 (8) (2014) 1903–1907.
[36] Y. Lu, D. Chen, E. Olaniyi, Y. Huang, Generative adversarial networks (gans) for image augmentation in agriculture: a systematic review, Comput. Electron. Agric. 200 (2022) 107208.
[37] M.L. Mekhalfi, C. Nicolò, Y. Bazi, M.M. Al Rahhal, N.A. Alsharif, E. Al Maghayreh, Contrasting yolov5, transformer, and efficientdet detectors for crop circle detection in desert, IEEE Geosci. Remote Sens. Lett. 19 (2021) 1–5.
[38] J.K. Norsworthy, S.M. Ward, D.R. Shaw, R.S. Llewellyn, R.L. Nichols, T.M. Webster, K.W. Bradley, G. Frisvold, S.B. Powles, N.R. Burgos, et al., Reducing the risks of herbicide resistance: best management practices and recommendations, Weed Sci. 60 (SP1) (2012) 31–62.
[39] NVIDIA, P. Vingelmann, F.H. Fitzek, Cuda, release: 10.2.89, https://developer.nvidia.com/cuda-toolkit, 2020.
[40] E.-C. Oerke, Crop losses to pests, J. Agric. Sci. 144 (1) (2006) 31–43.
[41] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.
[42] H.N. Patel, R. Jain, M.V. Joshi, et al., Fruit detection using improved multiple features based algorithm, Int. J. Comput. Appl. 13 (2) (2011) 1–5.
[43] F. Qingchun, Z. Wengang, Q. Quan, J. Kai, G. Rui, Study on strawberry robotic harvesting system, in: 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 1, IEEE, 2012, pp. 320–324.
[44] W. Qiu, S. Shearer, Maturity assessment of broccoli using the discrete fourier transform, Trans. ASABE 35 (6) (1992) 2057–2062.
[45] M. Rahnemoonfar, C. Sheppard, Deep count: fruit counting based on deep simulated learning, Sensors 17 (4) (2017) 905.
[46] J. Redmon, A. Farhadi, Yolov3: an incremental improvement, preprint, arXiv:1804.02767, 2018.
[47] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).
[48] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 658–666.
[49] 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.
[50] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (60) (2019) 1–48.
[51] Y. Si, G. Liu, J. Feng, Location of apples in trees using stereoscopic vision, Comput. Electron. Agric. 112 (2015) 68–74.
[52] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arXiv:1409.1556, 2014.
[53] P.W. Sites, M.J. Delwiche, Computer vision to locate fruit on a tree, Trans. ASABE 31 (1) (1988) 257–0265.
[54] D.C. Slaughter, R.C. Harrell, Color vision in robotic fruit harvesting, Trans. ASABE 30 (4) (1987) 1144–1148.
[55] D. Stajnko, M. Lakota, M. Hočevar, Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging, Comput. Electron. Agric. 42 (1) (2004) 31–42.
[56] D. Su, H. Kong, Y. Qiao, S. Sukkarieh, Data augmentation for deep learning based semantic segmentation and crop-weed classification in agricultural robotics, Comput. Electron. Agric. 190 (2021) 106418.
[57] J. Sun, B. Lu, H. Mao, et al., Fruits recognition in complex background using binocular stereovision, J. Jiangsu Univ.-Nat. Sci. Ed. 32 (4) (2011) 423–427.
[58] M. Tan, R. Pang, Q.V. Le Efficientdet, Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.
[59] K. Tanigaki, T. Fujiura, A. Akase, J. Imagawa, Cherry-harvesting robot, Comput. Electron. Agric. 63 (1) (2008) 65–72.
[60] 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.
[61] Z. Tian, C. Shen, H. Chen, T.He. Fcos, Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9627–9636.
[62] N. Tsoulias, D.S. Paraforos, G. Xanthopoulos, M. Zude-Sasse, Apple shape detection based on geometric and radiometric features using a lidar laser scanner, Remote Sens. 12 (15) (2020) 2481.
[63] S. Wan, S. Goudos, Faster r-cnn for multi-class fruit detection using a robotic vision system, Comput. Netw. 168 (2020) 107036.
[64] D. Whittaker, G. Miles, O. Mitchell, L. Gaultney, Fruit location in a partially occluded image, Trans. ASAE 30 (3) (1987) 591–0596.
[65] Q. Wu, Y. Chen, J. Meng, Dcgan-based data augmentation for tomato leaf disease identification, IEEE Access 8 (2020) 98716–98728.
[66] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2, https://github.com/facebookresearch/detectron2, 2019.
[67] R. Xiang, H. Jiang, Y. Ying, Recognition of clustered tomatoes based on binocular stereo vision, Comput. Electron. Agric. 106 (2014) 75–90.
[68] G. Xuan, C. Gao, Y. Shao, M. Zhang, Y. Wang, J. Zhong, Q. Li, H. Peng, Apple detection in natural environment using deep learning algorithms, IEEE Access 8 (2020) 216772–216780.
[69] B. Yan, P. Fan, X. Lei, Z. Liu, F. Yang, A real-time apple targets detection method for picking robot based on improved yolov5, Remote Sens. 13 (9) (2021) 1619.
[70] S.L. Young, G.E. Meyer, W.E. Woldt, Future directions for automated weed management in precision agriculture, in: Automation: The Future of Weed Control in Cropping Systems, Springer, 2014, pp. 249–259.
[71] B. Zhang, W. Huang, C. Wang, L. Gong, C. Zhao, C. Liu, D. Huang, Computer vision recognition of stem and calyx in apples using near-infrared linear-array structured light and 3d reconstruction, Biosyst. Eng. 139 (2015) 25–34.
[72] J. Zhang, M. Karkee, Q. Zhang, X. Zhang, M. Yaqoob, L. Fu, S. Wang, Multi-class object detection using faster r-cnn and estimation of shaking locations for automated shake-and-catch apple harvesting, Comput. Electron. Agric. 173 (2020) 105384.
[73] K. Zhang, K. Lammers, P. Chu, Z. Li, R. Lu, System design and control of an apple harvesting robot, Mechatronics 79 (2021) 102644, https://doi.org/10.1016/j.mechatronics.2021.102644.
[74] K. Zhang, K. Lammers, P. Chu, N. Dickinson, Z. Li, R. Lu, Algorithm design and integration for a robotic apple harvesting system, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2022, pp. 9217–9224.
[75] 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.
[76] Y. Zhao, L. Gong, Y. Huang, C. Liu, A review of key techniques of vision-based control for harvesting robot, Comput. Electron. Agric. 127 (2016) 311–323.
[77] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2020) 43–76.