Computers and Electronics in Agriculture 214 (2023) 108369

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Peach ripeness classification based on a new one-stage instance segmentation model

Ziang Zhao [a,*], Yulia Hicks [a], Xianfang Sun [b], Chaoxi Luo [c]

[a] School of Engineering, Cardiff University, Cardiff CF243AA, United Kingdom
[b] School of Computer Science and Informatics, Cardiff University, Cardiff CF244AG, United Kingdom
[c] College of Plant Science and Technology, Huazhong Agricultural University, Wuhan 430070, China

ARTICLE INFO

Keywords:
Peach
Ripeness classification
Instance segmentation
Attention mechanism
One-stage model

ABSTRACT

Peach instance segmentation is a crucial part to locate peaches and classify their ripeness stages to build an automatic peach harvesting or monitoring machine. This paper proposes a large and high-quality peach dataset called NinePeach, and a new one-stage instance segmentation model. The NinePeach dataset aims to reproduce real-world field conditions, encompassing various factors that can significantly influence the accuracy of peach detection, such as varying natural light intensity, instances of multiple fruit adhesion, and occlusion caused by stems and leaves. This is the largest and the most varied peach dataset among publicly available peach datasets to our best knowledge. Our proposed one-stage segmentation model does not require Region Proposal Network (RPN) to generate bounding box proposals, it directly identifies object instances by their centre locations and sizes and predict their category at the same time. The proposed model incorporates channel attention and spatial attention mechanisms to enhance object detection capabilities in crucial channels and spatial locations. Experimental results show that the state-of-the-art Mask RCNN performs 69.91% average precision (AP) with Swin-T backbone, our model surpasses it with the same backbone, achieving the highest 72.12% AP, and delivering more precise mask and boundary predictions. Specifically, our model is capable of accurately detect peaches under various conditions, such as peaches partially obscured by leaves, peaches partially exposed or overlapped. These advancements present promising prospects for the application of this technology to other fruits or crops.

1. Introduction

Peach is a kind of widely popular fruit that predominantly grown in Asia and Europe, with China being the largest producer of peaches in the world in both 2021 and 2022, with a production volume of around 16 million metric tons. The European Union is the second largest producer, with a production of about 2.9 million metric tons in the same period (USDA Foreign Agricultural Service, 2022). So far, the harvesting of peaches has mainly relied on manual labor, which demands substantial human and material resources, resulting in labor-intensive and time-consuming processes that can be costly and inefficient. In an effort to automate and mechanize the monitoring and collection of peaches, automatic classification of peaches according to their ripeness stage is an essential element. Accurate identification of peach ripeness stage plays a crucial role not only in ensuring precise peach yield prediction, effective field management, and optimal crop production, but also in achieving high quality of peach postharvest consumption and marketing. The stage of harvested peach is a key factor that significantly affects its shelf life and market value.

In the past few years, researchers have developed many methods to help detect fruit and classify its ripeness stage automatically. There are primarily two approaches for identifying fruit ripeness stage: destructive and non-destructive methods. Destructive methods utilize indices that are based on internal attributes, such as titratable acidity, soluble solids content, and total soluble solids, to determine fruit ripeness. Shinya et al. (2013) conducted research on peach ripeness by analyzing features such as fruit mass, soluble solids content, ground skin color, spectral absorbance difference at 670 nm and 720 nm index, as well as fruit/flesh firmness and uniaxial compression strength. Usenik et al. (2014) identified the ripening stage of four plum cultivars by the measurement of plums peel, flesh color, soluble solids content and firmness and the sensorily evaluation on eating quality. Azodanlou et al. (2004) applied a novel concept using solid phase microextraction (SPME) and measurement of total volatile compounds to distinguish between various stages

* Corresponding author.
E-mail address: zhaoz60@cardiff.ac.uk (Z. Zhao).

https://doi.org/10.1016/j.compag.2023.108369
Received 28 June 2023; Received in revised form 28 September 2023; Accepted 25 October 2023
Available online 31 October 2023

of strawberry ripeness. However, such methods can be laborious and costly, as they require expensive equipment and cause the destruction of the fruit. In contrast, non-destructive methods are more economical and efficient, such as RGB imaging, multispectral imaging, and near-infrared hyperspectral imaging, enabling the identification of fruit ripeness without damaging the fruit. An optical imaging based method was formulated to assess different external properties on the identification of four successive banana maturity stages (Xu et al., 2022) Tan et al. (2010) evaluated the oil palm ripeness and oil content by the use of color features from images captured by an RGB camera.

It is noteworthy that deep learning recently has been used as a non-destructive method to identify and classify the fruit ripeness stage using fruit images as inputs. Firstly, fruit ripeness classification stands as one of the popular domains where deep learning methods have found widespread application within the agricultural sector. (Kamilaris & Prenafeta-Boldú, 2018). In this kind of task, there is usually only one fruit in an image, as well as only one target that models are trained to predict. Four different ripeness stages of banana were classified using proposed convolution neural network (CNN) and compared with the state-of-the-art CNNs using transfer learning (Saranya et al., 2021). Suharjito et al. (2021) proposed a new data augmentation method called "9 angle crop" and created a mobile application to classify the ripeness levels of oil palm fresh fruit bunch using many lightweight CNNs like MobileNets (Howard et al., 2017; Sandler et al., 2019), EfficientNet (Tan & Le, 2020) and NASNet (Zoph et al., 2018). Besides, a number of other research studies employed vanilla and customized CNNs to predict different fruit ripeness stages. For instance, AlexNet (Krizhevsky et al., 2012) was utilized for tomatoes (Das & Singh Yadav, 2020), DenseNet (Huang et al., 2018) and Inception-v3 (Szegedy et al., 2015) for mulberries (Miraei Ashtiani et al., 2021), and VGG (Simonyan & Zisserman, 2015) for grapes (Ramos et al., 2021).

Secondly, based on fruit ripeness classification, some studies have focused on using deep learning methods to detect the locations of multiple fruits in one image. The four coordinate positions (called bounding boxes) of all fruit in the image are the targets to be predicted by training models. A kiwifruit detection system was developed for field images using the Faster R-CNN (Ren et al., 2016) and implemented using ZFNet (Zeiler & Fergus, 2013), which mitigates the subjectivity and limitations associated with manually selected kiwifruit features (Fu et al., 2018). Xiao et al. (2021) employed a two-step approach to detect apples, first utilizing Fast-RCNN (Girshick, 2015) to predict the locations of apples and then utilizing GoogLeNet (Szegedy et al., 2014) to predict apple ripeness. YOLOv3 (Redmon & Farhadi, 2018) is an efficient object detection model along with its previous YOLO variants (Redmon et al., 2016; Redmon & Farhadi, 2016). Therefore, it was frequently utilized to detect fruit like strawberry (Zhou et al., 2021) and green cucumber (Bai et al., 2022). Besides, a number of other studies were based on YOLOv3 to improve detection performance. For example, DenseNet was used to replace the original transport layer of YOLOv3, resulting in the network detection of apples at different maturity stages with better performance than the original YOLOv3 (Tian et al., 2019). Liang et al. (2020) proposed a method to detect litchi fruits and stems at nighttime environment, which adopted YOLOv3 to locate the anchor boxes of litchi fruits and then employed U-Net (Ronneberger et al., 2015) to perform fruiting stems segmentation.

Thirdly, some research has concentrated on the pixel-level and instance-level classification of fruit ripeness by segmenting the pixels of the fruit in an image. As the pioneer of instance segmentation models, Mask RCNN (He et al., 2017) and its modified version were most frequently adopted to segment fruit instances. Santos et al. (2019) demonstrated that Mask RCNN can effectively detect, segment, and track grape clusters, which exhibits significant variability in shape, colour, size, and compactness. A strawberry fruit detector was constructed based on Mask RCNN to overcome the difficulties of poor universality and robustness for traditional machine vision algorithms (Yu et al., 2019). In the same way, Pérez-Borrero et al. (2020) proposed a methodology for instance segmentation of strawberries using a modified version of Mask RCNN. A Mask RCNN was enhanced to accept dual-mode data fusion of RGB and depth images for a robust visual recognition for fruit and stem of cherry tomatoes (Xu et al., 2022). Hameed et al. (2022) proposed a score-based mask edge improvement of Mask-RCNN to segment fruit and vegetable images in a supermarket environment. A Mask RCNN was modified by fusing an attention module into the backbone network to enhance its feature extraction ability to precisely segment apples in an orchard (Wang & He, 2022). Similarly, Jia et al. (2022) proposed a green overlapped apples segmentation network, which extended Mask RCNN by adding an attention mechanism to prediction head for focusing more on the informative pixels but also suppressing the noise.

Furthermore, there are also some other different models used for fruit instance segmentation. An edge-guided based fruit segmentation model EdgeSegNet was proposed, which included modules specially designed to locate potential target areas and sharpen the edges (Sheng et al., 2023). Jia et al. (2021) designed an anchor-free model FoveaMask for segmentation of green fruits, in which a position attention module is introduced into the embedding mask branch to aggregate the effective information pixels and improve robustness ability.

It can be seen that whilst the above research has achieved some success in predicting fruit ripeness there are still some unresolved issues. First, fruit images with pixel-level ripeness annotations are essential to train a fruit ripeness classification model, while collecting and labelling them are time-consuming and laborious tasks and only few datasets are publicly available. Second, a lot of the current fruit instance segmentation models are based on Mask RCNN. As a two-stage segmentation method, Mask RCNN first generates instance bounding boxes using a detector and then classifies each pixel within every box. Compared with one-stage methods that produce pixel-wise classification maps and cluster them into instances, two-stage methods usually have better segmentation performance but typically produce a great number of proposal boxes, resulting in long inference time and large computational resources usage. Specifically, two-stage methods are not suitable for real-time instance classification or segmentation task which usually runs on mobile and embedded devices, as the resources and latency are highly restricted.

This work presents a peach dataset and a one-stage attention-based peach instance segmentation method to perform accurate peach localization in natural orchard environments. The main contributions of this research include:

1. A new large high-quality annotated peach dataset called NinePeach, which contains images of nine cultivars of peach in different ripeness stages under natural illumination.
2. A new one-stage and anchor-free instance segmentation model to detect peaches and classify their ripeness stage simultaneously. The proposed model removes the need for region proposal network (RPN) and the design of anchor generators. Channel attention and spatial attention are deployed in our model to enhance the perception ability.
3. The new model performs at the highest 72.12 % average precision (AP), surpassing the state-of-the-art Mask RCNN with the 69.91 % AP and produces more precise and smooth boundary predictions.

2. Materials and methods

2.1. Peach images collection

Peach images were collected from the experimental orchard in Huazhong Agricultural University, Wuhan, China. The collection was conducted during May to June 2022 and included nine cultivars of peaches: Dahongpao, Qingfeng, Chunmei, Chunmi, Chunxue, Songsen, Maotao, Youpantao, and XiahuiNo5. The image capture device used was a smartphone whose specifications are detailed in Table 1. The images

Table 1
Specifications of the mobile device used.

Device Specifications	
System OS	Android 11
CPU	Octa-core (1x3.2 GHz Kryo 585 & 3x2.42 GHz Kryo 585 & 4x1.80 GHz Kryo 585)
Main Camera Sensor	Sony IMX598(1/2″)
Focal Length	4.7 mm

were originally captured and stored in JPEG format at a resolution of 4000 × 3000 pixels.

The camera was positioned at a distance of 30–50 cm from the peach and captured images from various angles. It is worth noting that all the peach images were acquired under natural lighting conditions and in real-world production settings, where the peaches exhibited diverse physical configurations. These configurations include but are not limited to isolated peaches, peaches that are in close proximity to one another, peaches that are partially obscured by leaves or stalks, and peaches that are illuminated from the opposite side. Samples of images from each kind of peach are presented in Fig. 1.

2.2. Peach images annotation

A total of 3849 images were selected to form a dataset, representing nine cultivars of peaches that have been classified into three distinct stages of ripeness: unripe, semi-ripe, and ripe. Two annotators were independently in charge of carrying out the image labeling process and one reviewer would make decisions when it comes to controversial cases. All cultivars of peach were annotated individually.

The instance category distribution of 3849 peach images is presented in Fig. 2. Some cultivars lack ripe stage images due to objective

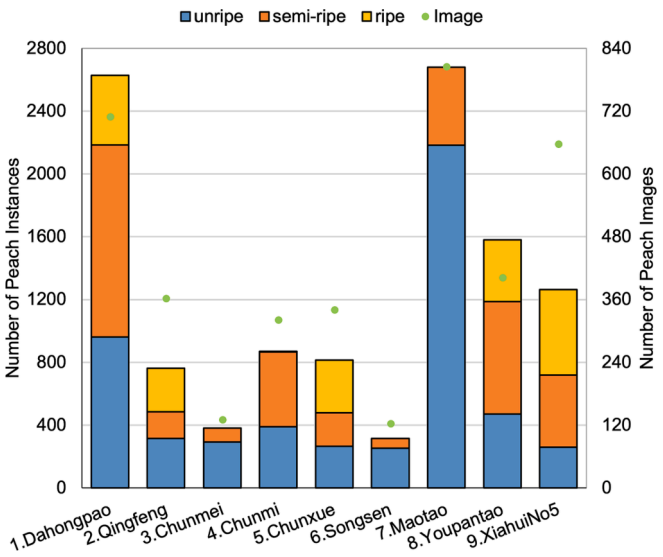


Fig. 2. Instance category distribution of our peach collection.

conditions. For example, there are relatively fewer Chunmi and Songsen trees, and their ripe fruits are dropped by weather or picked by animals, whilst Maotao takes longer time to become ripe than other cultivars, which exceeded our collection schedule. Therefore, "long-tail" phenomenon exists in our dataset, which is discussed below.

Similar to the MS COCO (Lin et al., 2014), three different sizes of objects are defined based on the instance area using the following criteria: small area ($\leq 32^2$), medium area ($32^2 \sim 96^2$) and large area ($>96^2$).

Each image was manually labeled using Label-Studio (Tkachenko

Fig. 1. Samples of our dataset contains 9 cultivars of peaches. All images are collected from different angles under natural light.

et al., 2020), generating a ground truth labelled image that contains individual segmentation masks of all peaches depicted in the image. The labeling process adhered to a rigorous standard, which involved generating a precise mask for each peach captured in the image, even in challenging scenarios where peaches may had appeared nearly imperceptible due to distance, occlusions, or their proximity to the image boundaries. A sample of annotation is shown in Fig. 3.

To reduce the computational requirements of the models, the images were resized to 1024 × 768 pixels. For every cultivar of peach, the images were randomly split with a ratio of 7:3 for training and validation sets respectively. Then the individual training sets and validation sets were combined to form a total training set of 2690 images and a total validation set of 1159 images. To alleviate the "long-tail" problem, we increased the number of semi-ripe and ripe instances by oversampling 750 randomly selected images that did not contain unripe instances to make the category distribution more balanced. The used data augmentation methods included random angle rotation, random jitter, and random flipping. The instance category distribution of our dataset can be seen at Table 2. Thus the balanced dataset called NinePeach was created to contain 3240 images for training and 1359 images for validation.

Table 2
The instance category distribution of our dataset.

	Original		Balanced	
Category	Train	Validation	Train	Validation
Unripe	3669	1717	3669	1717
Semi-ripe	2768	1140	3312	1307
Ripe	1403	589	1689	737
Total	7840	3446	8679	3761

2.3. Proposed model

Our proposed model was designed to simultaneously segment instance masks and predict their categories using full instance mask annotations as supervision instead of the bounding boxes of masks. Specifically, the proposed model is anchor-free and gets rid of the bounding box prediction, which would reduce much calculation and resource consumption. The architecture of the proposed model is illustrated in Fig. 4. The model consists of three parts: a backbone, a feature pyramid network, and a shared detection head following the pipeline from SOLOv2 (Wang et al., 2020).

In contrast to Mask RCNN, our model does not rely on a region proposal network (RPN) to generate proposals. Specifically, our model directly identifies object instances by their centre locations and sizes. To determine object locations, the input image would be divided into a uniform grid of size $S \times S$, resulting in S^2 possible center location class. If the centre of an object falls within a grid cell, then that cell is responsible for predicting the object's semantic category and segmenting its instance. Later in the article, we also demonstrate that our model outperforms original baseline due to embedding of the convolutional block attention, which enables the model to focus on objects in key channels and spatial locations.

2.3.1. Feature extraction (Backbone + FPN)

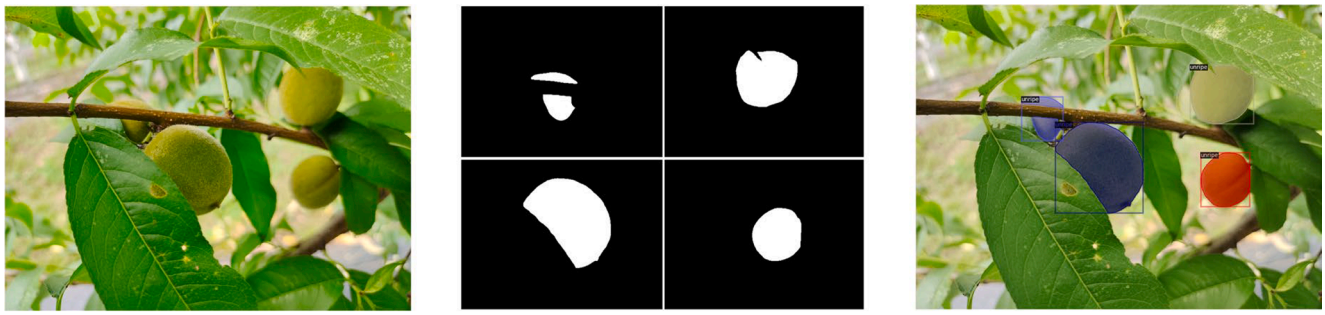
Image feature extraction is the process of identifying and extracting relevant information or features from an image. Our model follows the paradigm from Lin et al. (2018), where the feature extraction part is made up of two parts: a backbone and a feature pyramid network. The underlying module extracts the low-level features such as edges and angles, while the high-level features are fed into a classifier to determine the object category. ResNet50/101 and Swin-T are used as backbone in our proposed model. ResNet (He et al., 2016) was proposed to solve the image classification task. It overcame the vanishing gradient problem that can occur in very deep neural networks by introducing residual blocks. The residual blocks allow information to flow directly from earlier layers to later layers without being affected by intermediate layers. This makes it easier for ResNet to learn useful features from the input image. Swin Transformer (Swin-T, Liu et al., 2021) is a recently proposed neural network for visual recognition tasks that has shown strong performance on several benchmarks. Swin-T uses a hierarchical architecture where image patches are progressively downsampled to multiple scales. This allows Swin-T to capture both local and global features in an image, which can be important for visual recognition tasks. Additionally, Swin-T incorporates a shifted window mechanism that improves the processing of spatially adjacent patches, further enhancing its ability to capture fine-grained details. Swin-T has also been shown to have strong generalization ability, which means it can learn to recognize objects even when they are presented in new or unusual contexts. This is important for real-world applications where images may be taken under varying conditions. The output of backbone is made of a set of feature maps at four different resolutions.

The Feature Pyramid Network (FPN, Lin et al., 2017) was introduced to extend the backbone network, which is especially effective for the detection of targets at different scales. FPN works by taking the feature maps produced by backbone at different levels of the network, and building a feature pyramid that includes high-level features with strong semantics, as well as low-level features with strong spatial information. The final output of the FPN consists of a set of feature maps at four resolutions.

Overall, the ResNet/Swin-T with FPN are powerful architectures for image feature extraction, as they leverage the strengths of both ResNet/Swin-T and FPN to extract high-level and low-level features from the input image and combine them to accurately detect objects at different scales.

2.3.2. Detection head (Kernel Branch + Feature Branch)

Given the output of pyramid network, the detection head consists of two branches: kernel branch and feature branch, accepting each pyra-



(a) (b) (c)

Fig. 3. Illustration of the image annotation. (a) Original image. (b) Individual instance masks. (c) Annotated image.

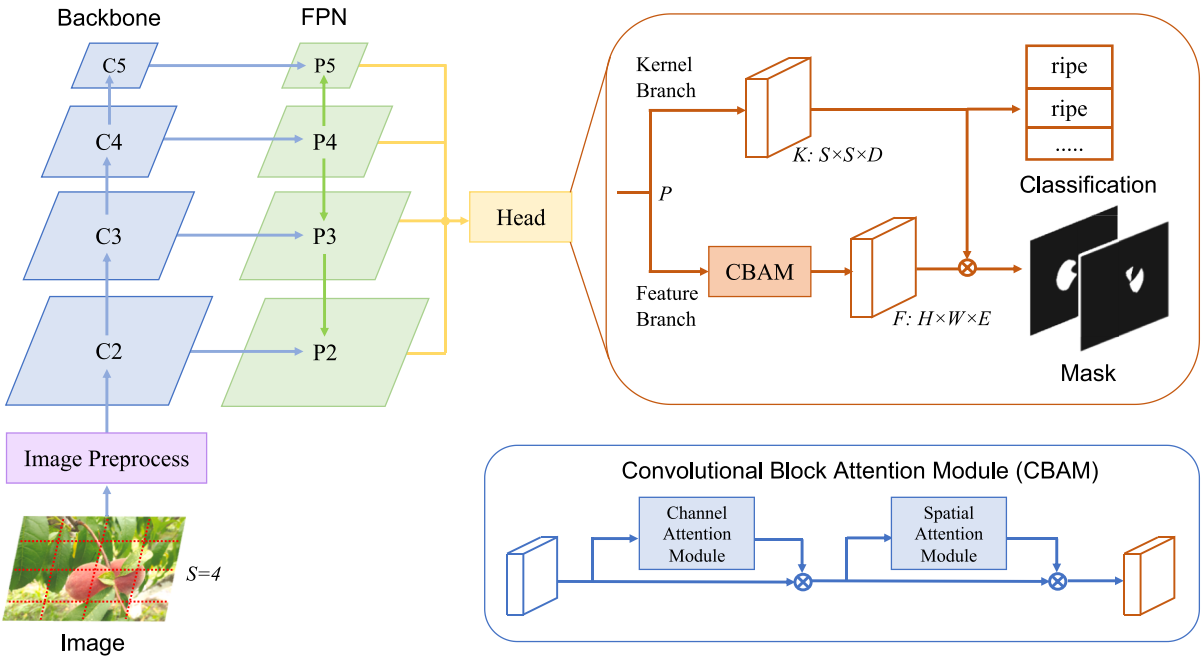


Fig. 4. The architecture of proposed model. It consists of three parts: a backbone, a feature pyramid, and a shared detection head.

mid feature as input. The output of feature pyramid is noted as P.

In the kernel branch, P is resized into a shape of $S \times S \times C$, and then a series of 4 convolution layers and a final $3 \times 3 \times D$ convolution layer are used to produce the kernel $K \in \mathbb{R}^{S\times S\times D}$. It should be noted that in the first of the four convolution layers, two additional input channels are concatenated which contain pixel coordinates normalized to the range of $[-1, 1]$ following CoordConv (Liu et al., 2018). In each grid, the kernel branch predicts D-dimensional outputs, which indicate the predicted convolution kernel weights. The final stage of the kernel branch involves the use of two convolution layers to generate predictions for the kernel and category, the last convolution layer used to predict category is a deformable convolution layer (DCN, Dai et al., 2017) . The weights of the detection head are shared among different levels.

In the mask feature branch, P is firstly passed through Convolutional Block Attention Module (CBAM, Woo et al., 2018). CBAM introduced attention to design network architecture, which consists of Channel Attention Module and Spatial Attention Module. These two modules use max pooling and average pooling to get feature information from channels and spatial locations, please refer to Woo et al. (2018) for more details. By connecting Channel Attention Module and Spatial Attention Module, CBAM enables the model to increase its expressive ability, focus on important features and suppress unimportant ones. CBAM does not change the shape of the input features, therefore the shape of the output of CBAM remains the same as P. Then, feature pyramid fusion is applied to learn a unified and high-resolution mask feature representation. This is achieved through multiple stages of convolution layers, group normalization, ReLU activation, and $2\times$ bilinear upsampling, the FPN features (P2 to P5) are scaled into 1/4 of original image size. Similar to the use of CoordConv in kernel branch, normalized coordinates are also catenated with FPN feature P5, enabling model's position sensitivity. A final 1×1 convolution layer is applied on scaled features (P2 to P5) to generate mask feature $F \in \mathbb{R}^{H\times W\times E}$.

Here we set the D from the kernel branch equal to E, implying that the predicted kernel is for a 1×1 convolution. After the mask kernel $K_{i,j}$ from the kernel branch and mask feature F from the mask branch are obtained, a dynamic convolutional operation is employed to generate the instance mask of S^2 channels corresponding to $S \times S$ grids. The operation can be written as:

$$M_{i,j} = K_{i,j} * F \tag{1}$$

where $K_{i,j} \in \mathbb{R}^{1\times 1\times E}$ is the convolution layer kernel predicted by the kernel branch, and $M_{i,j} \in \mathbb{R}^{1\times H\times W}$ is the mask prediction containing only one instance whose centre is at grid cell (i, j). For example, if D and E are set equal to 4, the mask branch would generate an output with a shape of H × W × 4. The kernel branch would generate an output with a shape of S × S × 4, which can be viewed as S^2 1x1 convolution kernels whose depths are 4, the dynamic convolutional operation would use two outputs above to get the predicted mask. At last, the predicted mask would be post-processed to get the peach instance segmentation results.

2.4. Model training

2.4.1. Loss function

In this paper, the proposed model only generates the predictions of peach categories and peach masks. To simultaneously consider the performance of both predictions, the loss function is designed to consist of two major components: the classification loss L_{class} and the mask loss L_{mask}, and λ is the weight factor of mask loss.

$$L = L_{class} + \lambda L_{mask} \tag{2}$$

where L_{class} is the focal loss (Lin et al., 2018) for semantic category classification and L_{mask} is the dice loss (Sudre et al., 2017) for mask prediction.

The L_{class} is calculated as follows:

$$L_{class} = -\alpha(1-\boldsymbol{p})^{\gamma} log(\boldsymbol{p}) \tag{3}$$

where α is set to 0.25 and γ is set to 2.0 in our study. $\boldsymbol{p}$ is the probability of the predicted instance. Sigmoid operation is used in calculating $\boldsymbol{p}$.

The L_{mask} is calculated as follows:

$$L_{mask} = 1 - \frac{2\left|\Sigma_{x,y}\left(\boldsymbol{p}_{x,y} \bullet \boldsymbol{q}_{x,y}\right)\right|}{\Sigma_{x,y}\boldsymbol{p}_{x,y}^2 + \Sigma_{x,y}\boldsymbol{q}_{x,y}^2} \tag{4}$$

where $\boldsymbol{p}_{x,y}$ and $\boldsymbol{q}_{x,y}$ refer to the value of pixel located at (x, y) in predicted mask $\boldsymbol{p}$ and ground truth mask $\boldsymbol{q}$.

2.4.2. Training details

We train our proposed model and Mask RCNN on the NinePeach dataset. ResNet 50/101 and Swin-Transformer are used as backbone networks. For ResNet50/101 backbone, the batch size is set to 16 with 27 K iterations in all, and the initial learning rate is set to 0.005 and divided by 10 at iteration 18 K and 24 K. For Swin Transformer backbone, the batch size is set to 4 with 54 K iterations in all. The initial learning rate is set to 0.005 and divided by 10 at iteration 36 K and 48 K. Additionally, we also train our model on nine individual peach datasets separately to validate the generalization ability of our model. ResNet50 is used as backbone network, the batch size is set to 16 with 10 K iteration, and the initial learning rate is set to 0.005 and divided by 10 at iteration 6 K and 8 K. We used stochastic gradient descent (SGD) optimizer, weight decay 0.0001, momentum 0.9. The learning rate is warmed up for the first 1000 iterations, then updates according to the StepLR method.

The backbone is initialized with pre-trained weights on ImageNet (Krizhevsky et al., 2012) and all convolution layers in the detection head are initialize with normal distribution. The data augmentation strategies used in training contain random horizontal flip, resizing the input images such that the shortest side is one of 640, 672, 704, 736, 768 or 800 pixels while the longest is at most 1333. The number of grids for four feature map levels is (40, 36, 24,16). The loss weights for L_{class} are set as {unripe:1.0, semiripe:1.5, ripe:2.0} to pay more attention on categories with fewer instances. The λ of the loss function L is set to 3 during training.

2.5. Model inference

2.5.1. Evaluation metrics

The average precision (AP) and average recall (AR) are frequently used to measure the performance of segmentation models. The definitions of precision and recall are:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

where TP is the number of cases that the target is peach and is correctly detected, FP is the number of cases that the target is not a peach, but it is wrongly detected, and FN is the number of cases that the target is a peach, but it is not detected.

AP is a standard measure for measuring the sensitivity of the network to a target object and is an indicator that reflects the global performance of the network. The higher the AP value, the better the detection accuracy of the proposed model is. Following the criterion of MS COCO, mean average precision (mAP) is used as our primary metric to evaluate the model performance, which is calculated by averaging 10 Intersection over Union (IoU) thresholds ranging from 0.50 to 0.95 across all categories. Additionally, the AP values for IoU = 0.50 and 0.75, and for individual category are also computed.

2.5.2. Inference details

The data augmentation strategy used in inference is only resizing the input images such that the shortest side is 800 pixels while the longest is at most 1333 pixels. During the inference, preprocessed input image would be passed through the backbone network, the feature pyramid network, and the detection head to generate two predictions. The first prediction from kernel branch includes the predicted category scores and predicted mask kernels, while the second prediction from feature branch includes predicted mask features. Then the predicted mask kernels are utilized to perform a convolution operation on the predicted mask feature to generate predicted soft masks followed a sigmoid operation, with the value range being [0,1]. A threshold of 0.5 is used to convert predicted soft masks to binary masks. It is noted that the final category scores are calculated by pixel-wise multiplication of the predicted category scores with binary masks, followed by division by the count of binary masks. Then we keep top 500 predictions and remove the redundant predicted masks via non-maximum suppression (NMS). Finally, the predicted masks would be reshaped and interpolated to original image size.

3. Experiments and results

3.1. Experiments

In this study, experiments are based on Detectron2 (Wu et al., 2019) and have been carried out using Python 3.9.13 and PyTorch 1.13 on a computer with the specifications shown in Table 3.

We show that our proposed model achieves competitive results compared to Mask RCNN in quantitative evaluation on NinePeach dataset. Then, we provide a detailed ablation study of the detection head and class loss weights. We also separately train our model on individual peach datasets to explore the generalization ability of the model. Finally, the segmentation results are visualized, and the computational parameters are calculated. We highlight the best result in following tables to better understand the model performance.

3.2. Results

3.2.1. Main results

We train our model and state-of-the-art Mask RCNN using on NinePeach dataset, then compare their instance segmentation performance. Fig. 5 illustrates the training loss and periodic evaluation (14 checkpoints for Res50/101 and 6 checkpoints for SWIN) results of our proposed models with different backbones, the losses are converged and evaluation results are stabilized at the end of the training. The results are presented in Table.4. Our proposed model with a SWIN-T backbone achieves the highest AP of 72.12 % in all experiments. Besides, our model outperforms Mask RCNN on overall AP when using the same backbone.

Firstly, with increasing backbone complexity and capacity, performance gains are progressively enhanced. For example, our model increases about 1.66 % and 5.79 % AP when changing ResNet50 to ResNet101 and Swin-T. This observation means the FPN and the detection head need more representative features generated by a stronger backbone as the condition for segmentation.

Secondly, our model has a lower AP_{75} and a higher AP_{50} than Mask RCNN, which indicates that our model is stricter when outputting predictions. This suggests a slight confidence reduction which is caused by the pixel-wise calculation on predicted category scores with binary masks in the inference phrase.

Thirdly, compared to Mask RCNN which has relatively higher AP_{small} and AP_{medium}, our model tends to have better performance in predicting large peach instances, which is similar to related work (Yu et al., 2019). We indicate this situation benefits from the mask feature fusion in the mask feature branch, which fuses features of different scales to get a unified and high-resolution feature representation.

Finally, our proposed model outperforms Mask RCNN on every category AP, which demonstrates it has better segmentation performance. Notably, both models are relatively good at predicting ripe peach instances. We believe the complexity of segmenting ripe instances

Table 3
Specifications of the computer used for experiments.

Device Specifications	
System OS	Cent OS 7
CPU	Inter Xeon Gold 6152 @2.1 GHz
Graphics	Nvidia Tesla V100
Memory	32.0 GB

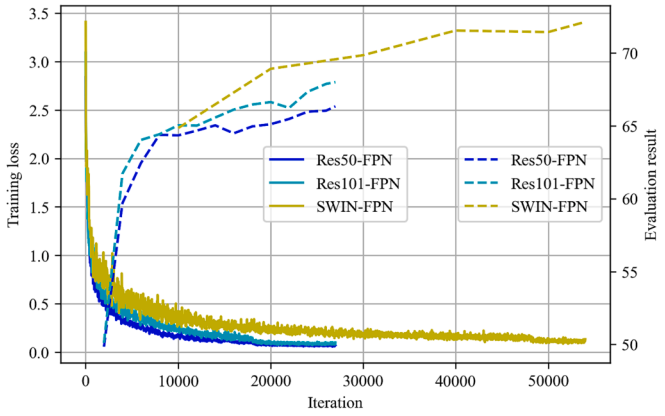


Fig. 5. The training loss and evaluation results of our model.

is reduced because the ripe peach not only has conspicuous color to distinguish, but also appears alone usually as a result of falling naturally and fruit thinning artificially.

3.2.2. Ablation results

We conduct a series of ablations to investigate the impact of different components in detection head and different loss weights in loss function on segmentation performance.

Detection Head. The detection head plays a critical role in our model. Table 5 shows the ablation result on the component of the detection head. Compared to vanilla baseline, adding coordinates and replacing the last convolution with deformable convolution attains 2.11 % and 2.47 % AP gains. Besides, adding the CBAM for stronger spatial perception ability gives significant 4.55 % AP improvement. Our proposed model leverages above three components and improve the baseline by 5.66 % AP.

Class Loss Weights. As the category distribution of the dataset is imbalanced, different weights for different categories are needed to reduce the imbalance. Table 6 shows some ablation results on different loss weights set for three categories. The loss weight settings {unripe:1.0, semiripe:1.5, ripe:2.0} demonstrated the best performance, emphasizing the importance of specific losses and thereby enhancing model performance. However, overly imbalanced weight settings {1:2:3} which pays much more attention on semiripe and ripe instances deteriorates model performance.

3.2.3. Individual model training and evaluation results

We separately train our proposed model with Res50-FPN backbone on nine individual peach datasets and evaluated the model with the same backbone trained using NinePeach on nine individual peach datasets. The results are shown in Table 7. The evaluation results of the model trained using NinePeach performs better compared to those of models trained using individual peach datasets. The average AP improvement stands at 21.05 %, with Songsen showing the most significant enhancement at 36.61 %. We indicate that after merging the datasets, not only the distribution of peach categories becomes more balanced, but also the model has more data samples for learning the characteristics and patterns of peaches in different ripeness stage, thus improving the generalization ability.

3.2.4. Visualization

We visualize the peach segmentation performance of our proposed model in Fig. 6. As shown in Fig. 6(a), besides the easy cases when the peaches are fully visible and can be segmented accurately, our model is capable of detecting peaches in more complex cases. Specifically, when the peaches overlap with each other or are partially obscured by tree branches or leaves, our model still performs well on identifying them accurately. The good segmentation performance shows the feasibility of the dynamic convolution operation in the detection head, of which two operators are mask features and mask kernels that both are learned from the output of the feature pyramid network. It is worth noting that our model not only detects multiple peaches of varying sizes within a single image accurately, but also generates almost as smooth boundary as the ground truth, benefitting from fused and high-resolution and mask feature representation after CBAM operation. Fusing features of different scales that merges the information of peaches of varied sizes to a unified feature enables the model to make predictions of varying sizes at the same time. The high-resolution mask feature brings larger predicted masks which means negligible loss when reshaping them back to

Table 5
Ablation on different components of the detection head.

Model	AP	AP_{75}	AP_{50}	AP_{unripe}	$AP_{semiripe}$	AP_{ripe}
Vanilla	59.81	77.00	62.48	53.52	58.03	67.88
+Coord	61.92	76.22	64.54	57.63	58.57	69.57
+DCN	62.28	74.56	63.83	60.39	57.42	69.04
+CBAM	64.36	76.57	66.60	61.91	61.32	69.84
Our model	**65.47**	**77.29**	**68.13**	**63.42**	**62.31**	**70.67**

Table 6
Ablation on different weights for class loss.

Weights	AP	AP_{50}	AP_{75}	AP_{unripe}	$AP_{semiripe}$	AP_{ripe}
1.0: 1.0: 1.0	65.02	77.86	67.53	63.94	59.97	71.18
1.00:1.25:1.75	63.59	75.79	65.70	61.64	59.47	69.66
1.0: 1.5: 2.0	**66.33**	**78.59**	**68.95**	**65.21**	**61.93**	**71.85**
1.00:1.75:2.25	65.33	77.64	67.71	63.89	60.49	71.62
1.0: 2.0: 3.0	63.06	74.14	65.60	60.75	58.10	70.34

Table 7
Individual model training and evaluation results on nine peach datasets.

	Individual training			Evaluation		
Peach	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
1.Dahongpao	39.52	57.74	38.44	57.48	69.92	59.69
2.Qingfeng	48.96	66.73	48.66	76.77	84.30	79.84
3.Chunmei	37.92	56.41	38.19	72.78	82.70	77.12
4.Chunmi	46.55	59.38	45.74	49.38	56.64	51.07
5.Chunxue	50.81	68.38	52.68	73.87	82.98	79.30
6.Songsen	32.60	59.77	31.54	69.21	76.17	74.75
7.Maotao	46.06	62.08	46.65	53.90	61.13	55.69
8.Youpantao	38.34	57.47	37.13	61.71	73.62	64.84
9.XiahuiNo5	54.59	70.33	55.32	69.76	78.52	72.02
Average	43.93	62.03	43.82	64.98	74.00	68.26

Table 4
Instance segmentation AP on NinePeach dataset.

Model	Backbone	AP	AP_{50}	AP_{75}	AP_{small}	AP_{medium}	AP_{large}	AP_{unripe}	$AP_{semiripe}$	AP_{ripe}
Mask RCNN	Res50-FPN	65.02	75.53	70.93	17.47	36.10	77.31	63.98	59.65	71.43
	Res101-FPN	66.02	75.91	71.76	13.01	33.93	78.29	64.92	60.25	72.99
	SWIN-FPN	69.91	83.11	**76.26**	**24.26**	**45.81**	76.47	64.57	64.09	76.08
Our Model	Res50-FPN	66.33	78.59	68.95	13.92	32.03	76.57	65.21	61.93	71.86
	Res101-FPN	67.99	77.73	70.84	11.21	32.41	78.75	65.76	62.39	75.84
	SWIN-FPN	**72.12**	**83.76**	75.49	11.52	40.25	**82.19**	**68.24**	**69.26**	**78.87**

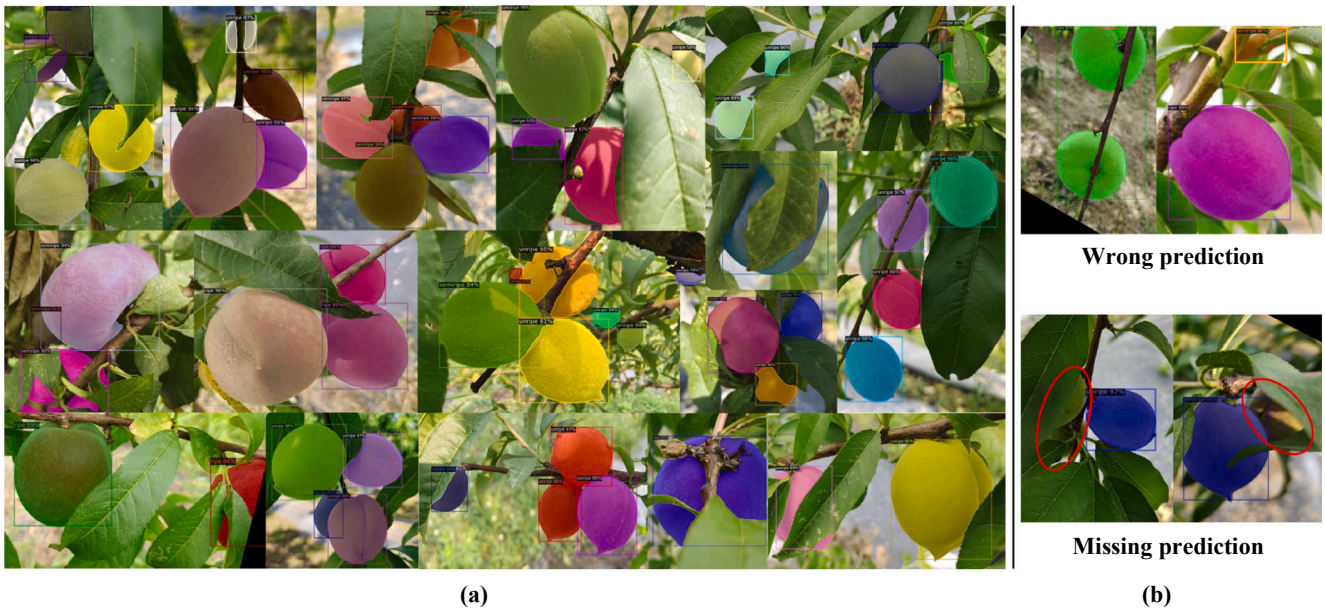


Fig. 6. (a) Examples of proposed model segmentations. (b) Examples of non-accurate proposed model.

original sizes.

However, there are few cases we called wrong prediction and missing prediction output by our model as shown in Fig. 6(b). Wrong prediction means that two or more objects are wrongly predicted to be one object, or a part of background part is wrongly predicted as a peach. We assume wrong prediction occurs when some parts in the image have similar features with each other or with known category feature, which makes the model regard them as the same object or target objects. One the other hand, if peaches are too obscured to be discovered or look like background because of misleading light conditions, the model tends to ignore them or treat them as the background, resulting in the problem of missing prediction in these scenarios.

Furthermore, we compare the peach segmentation performance between our model and Mask RCNN in Fig. 7. The red and blue boxes are used to emphasize the difference. In case Fig. 7(a), Mask RCNN ambiguously predicts the leaf as a part of the peach, while our model can segment the peach without leaf clearly. It can be observed that our model produces more precise and smooth boundary predictions than Mask RCNN. Fig. 7(b) shows a challenging case where a peach is occluded by leaves and stalks at the same time. Our model segments the peach almost perfectly, it accurately detects the peach in most of the regions, especially those along the tricky boundaries, while Mask RCNN cannot clearly segment the boundaries between peach and leaves and stalks, producing much more inaccurate and incomplete predictions. In Fig. 7(c), Mask RCNN predicted one peach separated by a leaf as two individual peaches, whilst our model predicted the separated parts as a one object.

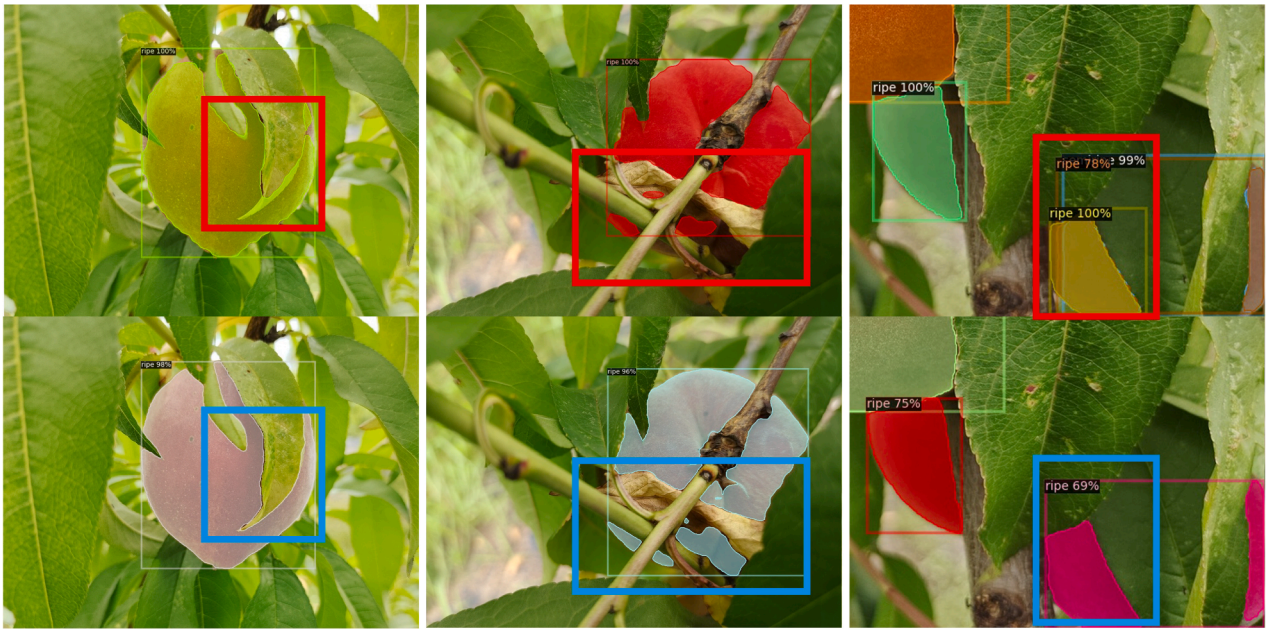


Fig. 7. Segmentation comparison between Mask RCNN and our proposed model. Top is Mask RCNN, below is our model.

3.2.5. Computational parameters

We compare the number of learnable parameters, the number of floating-point operations, inference speed and maximum GPU memory usage during training between our proposed model and Mask RCNN using the same backbone ResNet50-FPN. The results are shown in Table 8.

Learnable Parameters(Params). The learnable parameters are the weights and biases associated with the model's layers, which are learned during the training process and are used to make predictions. The number of learnable parameters in a model is an indicator of its capacity. Our model has 2.24 M more parameters than Mask RCNN, most of which are introduced by convolution layers. This means that our model is more complex and requires more data for training.

Floating-point Operations(FLOPs). FLOPs refers to the computational workload or complexity of the model during inference, which is calculated based on the operations performed in the model's layers such as matrix multiplications, convolutions, and activations. As FLOPs is not a well-defined concept and it is dependent on specific input data (FAIR, 2021), the built-in analyze function from Detectron2 was adopted to calculate the FLOPs. Our model has more 38.5G FLOPs than Mask RCNN, showing that our model has higher computational complexity.

Inference Time. Here we report the pure inference time in a batch, which specifically measures the time it takes for the model to compute predictions once the input data is fed into the model. Our model is 25 % faster than Mask RCNN during inference, which indicates our model is relatively faster to execute.

Maximum GPU Memory Usage. The maximum GPU memory usage indicates the peak memory consumption by the model during the training progress, which is the highest value of "max_mem" in the log. Our model saves 2593 M GPU memory than Mask RCNN, which makes it possible to train with larger batch and run on edge devices.

In summary, Mask RCNN has fewer Params and FLOPs but longer inference time and more GPU memory usage as a result of abundant anchors generated during training and inference. Despite having more Params and FLOPs, our model manages to keep inference time and GPU usage relatively low. It maintains better accuracy and precision than Mask RCNN while delivering results faster. Our model is able to perform a larger number of FLOPs quickly, striking a fine trade-off between performance and complexity. This efficiency can be attributed to the detection head that is anchor-free and shared between different feature map levels, which allows our model to maximize computational power while minimizing memory requirements and enables our model can be potentially deployed on GPUs with limited memory capacities.

4. Discussion

4.1. The details of NinePeach dataset

To the best of our knowledge, there is no official standard for classifying the ripeness of peaches on trees. With the cooperation with a botanist specializing in peach, we determine the peach ripeness into three stages subjectively. The only criteria we set is that annotators must choose their first judgment when meeting ambiguous cases. Similar to other large datasets, NinePeach dataset also has a long-tail phenomenon, which refers to a situation where few categories have a high frequency of occurrence, while the majority of categories have relatively few instances, forming a "long tail" in the distribution curve. We additionally oversampled the images to increase the number of instances of fewer categories and set different weights for different categories to alleviate this problem. The improved dataset has a balanced category distribution, facilitating the training of a large and well-performing peach instance segmentation model.

Table 8
The comparison of capacity and complexity of our proposed model and Mask RCNN.

Model	Params	FLOPs	Inference Time	Max Memory
Our Model	46.17 M	213.4 ± 0.2G	0.09 s	8542 M
Mask RCNN	43.93 M	174.9 ± 1.0G	0.12 s	11135 M

4.2. The limitations of the proposed model

Our proposed model demonstrates accurate peach detection capabilities, even when peaches are obstructed by tree branches or leaves. However, in few cases where certain regions within the image exhibit similar features with each other or with known category features, our model may generate false prediction, and missing prediction occurs when peaches are too obscured or look like background due to lighting conditions. We attribute these unreliable predictions to the larger receptive field of our model and the misleading illumination conditions of the image.

The incorporating of CBAM has led to a noteworthy 4.55 % increase in AP, but it has also augmented the complexity of the model, with the extensive use of convolution operations resulting in an elevation of both learnable parameters and floating-point operations. The potential improvement directions of our method are to reduce unreliable predictions and reduce computational complexity.

5. Conclusion

Precise identification of peach ripeness stage plays a crucial role in developing automated harvesting systems for large peach orchards, as it enhances picking efficiency and reduces production costs. Motivated by this, a high-quality peach dataset called NinePeach and a one-stage peach instance segmentation model were constructed in this paper. The NinePeach dataset comprises a total of 4599 peach images, categorized into three distinct stages of ripeness: unripe, semi-ripe, and ripe. This dataset aims at reproducing the actual situation in the field, including images with factors like different intensity of natural light, multi-fruit adhesion, and occlusion caused by stems and leaves.

Our proposed one-stage peach instance segmentation model does not require an RPN to generate bounding box proposals. The prediction of masks is obtained through dynamic convolution operations on the mask feature and kernel feature outputted from two branches. Channel attention and spatial attention are considered to enhance the ability of detecting objects in key channels and spatial locations, which brings a significant positive impact on model performance. Benefits from the anchor-free and memory-friendly design, our proposed model achieves a delicate balance between model performance and complexity, manifested by the fact that it utilizes fewer GPU resources while delivering faster and better predictions compared to Mask RCNN.

At present, the released large peach dataset provides a foundation for further peach-related studies and reduces their workload. The proposed model is able to accurately detect peaches and generate smooth boundaries of them, even in some cases where peaches are occluded, which establishes a robust basis for further work like peach pick point estimation and peach diseases monitoring. These advances create opportunities for offering practical solutions for farmers, applying this technology to other fruits or crops and considering the ever-evolving nature of agriculture.

In future research, a lightweight neural network will be explored for efficient feature extraction to enhance real-time performance in peach detection. The next step is to address the issue of unreliable predictions, reducing the computational complexity of the model for potential deployment in mobile or embedded applications.

CRediT authorship contribution statement

Ziang Zhao: Methodology, Software, Validation, Visualization, Writing – original draft, Data curation. **Yulia Hicks:** Methodology, Writing – review & editing, Supervision. **Xianfang Sun:** Methodology,

Writing – review & editing, Supervision. **Chaoxi Luo:** Resources, 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

The NinePeach dataset is available at: https://drive.google.com/drive/folders/1vCSoqVGWhy4pvyEVlW-oLH8uN2S_nUKo?usp=share_link

Acknowledgment

We thank Mr. Shuai Chen and Mr. Zhezheng Zeng for taking the images. Also, we express our special gratitude to Ms. Chenjing Zhao for her valuable help with annotating the images. We appreciate the computational resources provided by Advanced Research Computing at Cardiff (ARCCA).

References

Azodanlou, R., Darbellay, C., Luisier, J.-L., Villettaz, J.-C., & Amadò, R., 2004. Changes in flavour and texture during the ripening of strawberries. Eur. Food Res. Technol., 218, 2, 167–172. Scopus. https://doi.org/10.1007/s00217-003-0822-0.

Bai, Y., Guo, Y., Zhang, Q., Cao, B., Zhang, B., 2022. Multi-network fusion algorithm with transfer learning for green cucumber segmentation and recognition under complex natural environment. Comput. Electron. Agric. 194, 106789 https://doi.org/10.1016/j.compag.2022.106789.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable Convolutional Networks (arXiv:1703.06211). arXiv. http://arxiv.org/abs/1703.06211.

Das, P., & Singh Yadav, J.P., 2020. Transfer Learning based Tomato Ripeness Classification. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 423–428. https://doi.org/10.1109/I-SMAC49090.2020.9243345.

FAIR. (2021). fvcore: A Minimalist Computer Vision Library. GitHub. https://github.com/facebookresearch/fvcore.

Girshick, R., 2015. Fast R-CNN. arXiv:1504.08083 [Cs]. http://arxiv.org/abs/1504.08083.

Fu, L., Feng, Y., Majeed, Y., Zhang, X., Zhang, J., Karkee, M., Zhang, Q., 2018. Kiwifruit detection in field images using Faster R-CNN with ZFNet. IFAC-PapersOnLine 51 (17), 45–50. https://doi.org/10.1016/j.ifacol.2018.08.059.

Hameed, K., Chai, D., Rassau, A., 2022. Score-based mask edge improvement of Mask-RCNN for segmentation of fruit and vegetables. Expert Syst. Appl. 190, 116205 https://doi.org/10.1016/j.eswa.2021.116205.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. In: IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.322.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861 [Cs]*. http://arxiv.org/abs/1704.04861.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K.Q., 2018. Densely Connected Convolutional Networks (arXiv:1608.06993). arXiv. http://arxiv.org/abs/1608.06993.

Jia, W., Zhang, Z., Shao, W., Hou, S., Ji, Z., Liu, G., Yin, X., 2021. FoveaMask: A fast and accurate deep learning model for green fruit instance segmentation. Comput. Electron. Agric. 191, 106488 https://doi.org/10.1016/j.compag.2021.106488.

Jia, W., Zhang, Z., Shao, W., Ji, Z., Hou, S., 2022. RS-Net: Robust segmentation of green overlapped apples. Precis. Agric. 23 (2), 492–513. https://doi.org/10.1007/s11119-021-09846-3.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016.

Krizhevsky, A., Sutskever, I., & Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25. https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Liang, C., Xiong, J., Zheng, Z., Zhong, Z., Li, Z., Chen, S., Yang, Z., 2020. A visual detection method for nighttime litchi fruits and fruiting stems. Comput. Electron. Agric. 169, 105192 https://doi.org/10.1016/j.compag.2019.105192.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S., 2017. Feature Pyramid Networks for Object Detection. arXiv. https://doi.org/10.48550/arXiv.1612.03144.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P., 2018. Focal Loss for Dense Object Detection. arXiv. https://doi.org/10.48550/arXiv.1708.02002.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision – ECCV 2014. Springer International Publishing, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.

Liu, R., Lehman, J., Molino, P., Such, F.P., Frank, E., Sergeev, A., & Yosinski, J., 2018. An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution (arXiv:1807.03247). arXiv. http://arxiv.org/abs/1807.03247.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B., 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows (arXiv:2103.14030). arXiv. http://arxiv.org/abs/2103.14030.

Miraei Ashtiani, S.-H., Javanmardi, S., Jahanbanifard, M., Martynenko, A., Verbeek, F.J., 2021. Detection of Mulberry Ripeness Stages Using Deep Learning Models. IEEE Access 9, 100380–100394. https://doi.org/10.1109/ACCESS.2021.3096550.

Pérez-Borrero, I., Marín-Santos, D., Gegúndez-Arias, M.E., Cortés-Ancos, E., 2020. A fast and accurate deep learning method for strawberry instance segmentation. Comput. Electron. Agric. 178, 105736 https://doi.org/10.1016/j.compag.2020.105736.

Ramos, R.P., Gomes, J.S., Prates, R.M., Simas Filho, E.F., Teruel, B.J., dos Santos Costa, D., 2021. Non-invasive setup for grape maturation classification using deep learning. J. Sci. Food Agric. 101 (5), 2042–2051. https://doi.org/10.1002/jsfa.10824.

Redmon, J., & Farhadi, A., 2016. YOLO9000: Better, Faster, Stronger (arXiv:1612.08242). arXiv. https://doi.org/10.48550/arXiv.1612.08242.

Redmon, J., & Farhadi, A., 2018. YOLOv3: An Incremental Improvement (arXiv:1804.02767). arXiv. https://doi.org/10.48550/arXiv.1804.02767.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection (arXiv:1506.02640). arXiv. https://doi.org/10.48550/arXiv.1506.02640.

Ren, S., He, K., Girshick, R., & Sun, J., 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497 [Cs]. http://arxiv.org/abs/1506.01497.

Ronneberger, O., Fischer, P., & Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597 [Cs]. http://arxiv.org/abs/1505.04597.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C., 2019. MobileNetV2: Inverted Residuals and Linear Bottlenecks (arXiv:1801.04381). arXiv. https://doi.org/10.48550/arXiv.1801.04381.

Santos, T., Souza, L., Santos, A., & Avila, S. (2019). *Grape detection, segmentation and tracking using deep neural networks and three-dimensional association*.

Saranya, N., Srinivasan, K., Kumar, S.K.P., 2021. Banana ripeness stage identification: A deep learning approach. J. Ambient Intell. Human. Comput. https://doi.org/10.1007/s12652-021-03267-w.

Sheng, X., Kang, C., Zheng, J., Lyu, C., 2023. An edge-guided method to fruit segmentation in complex environments. Comput. Electron. Agric. 208, 107788 https://doi.org/10.1016/j.compag.2023.107788.

Shinya, P., Contador, L., Predieri, S., Rubio, P., Infante, R., 2013. Peach ripening: Segregation at harvest and postharvest flesh softening. Postharvest Biol. Technol. 86, 472–478. https://doi.org/10.1016/j.postharvbio.2013.07.038.

Simonyan, K., & Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [Cs]. http://arxiv.org/abs/1409.1556.

Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., & Cardoso, M.J., 2017. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations (Vol. 10553, pp. 240–248). https://doi.org/10.1007/978-3-319-67558-9_28.

Suharjito, Elwirehardja, G.N., & Prayoga, J.S., 2021. Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. Comput. Electron. Agric., 188, 106359. https://doi.org/10.1016/j.compag.2021.106359.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A., 2014. Going Deeper with Convolutions (arXiv:1409.4842). arXiv. https://doi.org/10.48550/arXiv.1409.4842.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). *Rethinking the Inception Architecture for Computer Vision* (arXiv:1512.00567). arXiv. https://doi.org/10.48550/arXiv.1512.00567.

Tan, M., & Le, Q.V., 2020. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (arXiv:1905.11946). arXiv. https://doi.org/10.48550/arXiv.1905.11946.

Tan, Y. A., Low, K. W., Lee, C. K., & Low, K.S., 2010. Imaging technique for quantification of oil palm fruit ripeness and oil content. Eur. J. Lip. Sci. Technol., 112, 8, 838–843. Scopus. https://doi.org/10.1002/ejlt.201000020.

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z., 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agric. 157, 417–426. https://doi.org/10.1016/j.compag.2019.01.012.

Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N., 2020. Label Studio: Data labeling software. Label Studio. https://labelstud.io/.

USDA Foreign Agricultural Service. 2022. Global leading peach and nectarine producing countries in 2021/2022 (in 1,000 metric tons). Statista. https://www.statista.com/statistics/739329/global-top-peaches-and-nectarines-producing-countries/.

Usenik, V., Stampar, F., Kastelec, D., 2014. Indicators of plum maturity: When do plums become tasty? Scientia Horticulturae 167, 127–134. https://doi.org/10.1016/j.scienta.2014.01.002.

Wang, X., Zhang, R., Kong, T., Li, L., & Shen, C., 2020. SOLOv2: Dynamic and Fast Instance Segmentation (arXiv:2003.10152). arXiv. https://doi.org/10.48550/arXiv.2003.10152.

Wang, D., He, D., 2022. Fusion of Mask RCNN and attention mechanism for instance segmentation of apples under complex background. Comput. Electron. Agric. 196, 106864 https://doi.org/10.1016/j.compag.2022.106864.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I.S., 2018. CBAM: Convolutional Block Attention Module (arXiv:1807.06521). arXiv. http://arxiv.org/abs/1807.06521.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R., 2019. *Facebookresearch/detectron2* [Python]. Meta Res. https://github.com/facebookresearch/detectron2.

Xiao, B., Nguyen, M., Yan, W.Q., 2021. Apple Ripeness Identification Using Deep Learning. In: Nguyen, M., Yan, W.Q., Ho, H. (Eds.), Geometry and Vision. Springer International Publishing, pp. 53–67. https://doi.org/10.1007/978-3-030-72073-5_5.

Xu, P., Fang, N., Liu, N., Lin, F., Yang, S., Ning, J., 2022. Visual recognition of cherry tomatoes in plant factory based on improved deep instance segmentation. Comput. Electron. Agric. 197, 106991 https://doi.org/10.1016/j.compag.2022.106991.

Yu, Y., Zhang, K., Yang, L., Zhang, D., 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. Comput. Electron. Agric. 163, 104846 https://doi.org/10.1016/j.compag.2019.06.001.

Zeiler, M.D., & Fergus, R., 2013. Visualizing and Understanding Convolutional Networks (arXiv:1311.2901). arXiv. http://arxiv.org/abs/1311.2901.

Zhou, X., Lee, W.S., Ampatzidis, Y., Chen, Y., Peres, N., Fraisse, C., 2021. Strawberry Maturity Classification from UAV and Near-Ground Imaging Using Deep Learning. Smart Agric. Technol. 1, 100001 https://doi.org/10.1016/j.atech.2021.100001.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q.V., 2018. Learning Transferable Architectures for Scalable Image Recognition (arXiv:1707.07012). arXiv. http://arxiv.org/abs/1707.07012.