ORIGINAL RESEARCH **Open access**

Improving Instance Segmentation Quality for Green Gram Pod Detection via a Point-Based Rendering Framework

Andrei Popescu[1*], Mihai Ionescu[1], Elena Stan[2], Cristina Radu[3]

Abstract

The rise of Artificial Intelligence, deep learning, and contemporary computer vision techniques has been the primary force behind breakthroughs in agriculture. Latest detection models that support real-time processing directly on edge devices effectively solve many farming issues, including identifying diseases, pests, or insects, and determining crop maturity stages (fruits and vegetables). Today's harvesting machines and robotic fruit pickers heavily depend on the detection strength of these algorithms. Several detection methods have been introduced and evaluated in the literature, achieving solid results in mean average precision. Nevertheless, today's agricultural applications need not only strong mean average precision but also fast inference times. This work presents a Detectron2-powered framework integrated with PointRend (Point-based Rendering) that achieves superior, high-quality pixel-level instance segmentation when identifying and locating green gram pods or Mung Bean (*Vigna radiata*) in real-world field environments, generating sharp and refined boundaries for precise positioning of the green gram pods. Experimental results show that the proposed framework outperforms the well-known Mask R-CNN model, achieving higher mean average precision and noticeably better detection quality.

Keywords Object detection, Instance segmentation, Detectron2, Pointrend, Green gram pods

*Correspondence:
Andrei Popescu
andrei.popescu@usamv.ro
[1] Department of AI for Sunflower and Corn Seed Germination Prediction, Faculty of Agriculture, University of Bucharest, Bucharest, Romania
[2] Department of Precision Drainage and Soil Water Retention Modeling, Faculty of Agriculture, University of Agricultural Sciences Cluj-Napoca, Cluj-Napoca, Romania
[3] Department of Smart Fertilization Algorithms, Faculty of Agriculture, Polytechnic University of Bucharest, Bucharest, Romania

Introduction

Recent advancements in Artificial Intelligence (AI), cutting-edge computer vision methods, and the growing availability of spatial and temporal agricultural field data, satellite images, and economical cloud computing infrastructure have led to significant progress in promoting sustainable farming. Current farming strategies emphasize the smart use of available resources to increase productivity and deliver improved financial gains. This approach is referred to as Precision Agriculture (PA). Keeping track of crop health is crucial for producing superior yields.

The growing interest in precision agriculture has motivated many researchers and engineers (specializing in ML/DL and computer vision) to design effective models that deliver exact identification, detection, classification, and positioning of key crop elements needed to gather important growth parameters. Early uses of deep learning in agriculture primarily addressed classification challenges in crop classification. These included disease classification [1], pest classification [2], detection [3], grading and classification of fruits by maturity stage, such as coffee beans [4], and beehive monitoring [5].

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which permits non-commercial use, distribution, reproduction, and adaptation in any medium or format, provided that appropriate credit is given to the original author(s) and source, a link to the license is provided, and any changes made are indicated. Adapted material must be distributed under the same license.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

The arrival of advanced computer vision algorithms enabled computers to classify farm produce and accurately mark their positions within images, using either bounding boxes or polygons [6].

Even though numerous systems have been developed for fruit and vegetable detection to enable real-time automation during harvest, there remains a strong need for pixel-level precision in detecting and recognizing specific fruits and vegetables.

Before the widespread adoption of machine learning, recognizing and sorting fruits and vegetables relied mostly on conventional image processing methods. The standard classical segmentation methods include three main categories: region-based, threshold-based, and edge-based. These approaches were dominant and commonly applied in detection systems for identifying, locating, and classifying various crops. One technique proposed by Thendral *et al.* [7] used two strategies, edge and color segmentation, for oranges. The color-based approach achieved 85% accuracy in detecting oranges. An automatic threshold-based technique [8] for detecting insect damage in soybeans used hyperspectral imaging and achieved 91.7% accuracy on insect-affected samples. Another threshold-based solution [9] employed the OTSU algorithm for depth-based segmentation and identification of clustered tomatoes, attaining a recognition accuracy of 87.9%. A frequently used traditional image processing strategy centers on edge detection. Modified Canny edge detection was applied by Monir Rabby *et al.* [10] for fruit detection and classification (Apples and Oranges) through color and shape feature extraction. Muruganandam *et al.* [11] conducted a comparative evaluation of multiple edge detection algorithms, including Prewitt, Laplacian, and Sobel, for rice disease and pest identification.

Moreover, the authors built two CNN-based models to illustrate the superiority of deep learning over classical image processing for segmentation tasks. Researchers quickly recognized the importance of developing segmentation methods that provide improved object separation with minimal processing requirements. A watershed segmentation model [12] applied to cotton leaves demonstrated faster computation, reaching a segmentation success rate of 97% relative to manual leaf area extraction. Clustering-based segmentation methods have the advantage of producing reliable results rapidly. Two widely recognized unsupervised clustering techniques are K-means and Fuzzy C-means. An adaptive K-means clustering method [13] for diseased tomato leaf segmentation achieved better precision and performance compared to standard K-means, DBSCAN, and mean-shift approaches. A cognitive fuzzy C-means (CFCM) algorithm [14] effectively separates diseased and healthy areas in rice plants and apple trees. Multiple CNN-backbone algorithms have been suggested in studies for crop monitoring tasks, such as early disease identification [15]. The most established architecture for instance segmentation across agriculture and other areas is Mask R-CNN [16, 17], which effectively addresses the challenges posed by overlapping objects.

In scenarios requiring real-time fruit and vegetable detection, the Yolo family of detectors and Single Shot Multi-Box Detectors (SSDs) are generally preferred to Faster R-CNN. Research presented by Tian *et al.* [18] demonstrates real-time apple detection using an improved YOLOv3 model. In a similar vein, the work described by Lamb and Chuah [19] proposes an optimized SSD detector for strawberries that runs on a Raspberry Pi.

Green gram ranks among the most popularly consumed pulses. Apart from its rich nutritional profile, it is recognized for enhancing soil fertility and nitrogen content. Research also confirms that farmers derive significant economic benefits from applying modern technologies in green gram production. Innovations such as deep learning and computer vision demonstrate significant potential to improve the quality of agricultural output. A review of existing studies found no dedicated work focused solely on pixel-level detection and instance segmentation of green gram pods. This gap likely stems from inherent difficulties, such as pod color blending with the background, which complicate detection.

Additionally, the compact size of the pods presents further obstacles for accurate object detection. The current study focuses on constructing a specialized polygon-annotated dataset of green gram pods and creating a Detectron2-based framework that incorporates a rendering technique (PointRend) to upgrade the leading Mask R-CNN algorithm, resulting in higher accuracy (mAP) and superior instance segmentation quality for green gram pods. The key contributions of this work are:

© 2024 The Author(s). | Published by Agro Intelligence Research Press

• The study introduces the first dataset featuring polygon-annotated images of green gram pods, supporting better identification, detection, and localization of green pods with stronger instance segmentation under difficult field conditions.

• The work produced a Detectron2-based framework utilizing deep learning foundation models and advanced computer vision tools to raise the standard of instance segmentation.

• The proposed solution delivers not only better segmentation precision and accuracy but also elevates the overall mask quality, facilitating the extraction of subtle details and sharper boundary definitions for green gram pods. (Segmentation + improvement in segmentation quality)

The remaining parts of the paper are divided into four sections. Section 2 outlines the materials and methods employed, covering dataset characteristics, annotation procedures, augmentation strategies, experimental configuration, evaluation measures, and model designs. Section 3 presents an extensive summary of training and validation outcomes, including predictions on new data. Section 4 offers a focused discussion, with a comparative evaluation of the findings and main observations from the experiments, and draws connections to comparable studies in the literature. Section 5 wraps up the paper by recapitulating the essential results and observations and outlining possible avenues for future investigation.

Materials and Methods

Dataset details

The images used for the experiments were collected from a local agricultural field in Vijayapur district, Karnataka. These raw photographs were taken with a smartphone camera at various times throughout the day, covering morning, afternoon, evening, and nighttime conditions. The collection further includes scenes with occluded pods to promote robust performance across different practical situations. In the preprocessing phase, all images were resized to 1024 × 1024 to strike a suitable balance between effective detection and computational efficiency. Multiple examples from the dataset, shown in **Figure 1**, depict green gram pods against intricate, similarly colored backgrounds, making their detection and precise positioning notably difficult.

© 2024 The Author(s). | Published by Agro Intelligence Research Press



Figure 1. Sample raw dataset images consisting of green gram pods collected from the field.

Table 1 outlines the dataset used in this research to train an instance segmentation model for field-based green gram pod detection. It contains 1215 training images, each containing 3720 training instances of green grams. The validation portion consists of 270 images with 1512 green gram instances. The division between the training and validation image sets is 82% and 18%, respectively.

Table 1. Dataset details indicating contents of training and validation sets.

SL. No.	Dataset split	Image resolution	Number of instances	Number of images
1.	Training	1024×1024	3720	1215
2.	Validation		1512	270

Annotation details

To assemble the raw green gram pod images, a visit was made to a nearby farm in the Vijayapur district of Karnataka. A smartphone featuring a 48 Megapixel camera served as the capture tool, helping maintain suitability for deployment on edge-level mobile devices. The unprocessed images originally measured 1846×4000 (portrait orientation) or 4000×1846 (landscape orientation). Before training, these were prepared and scaled down to 1024×1024 resolution. Polygon annotations were created using the LabelMe tool and exported in the Common Objects in Context (COCO) JSON format. Figure 2 displays representative dataset images in both unannotated and annotated states.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

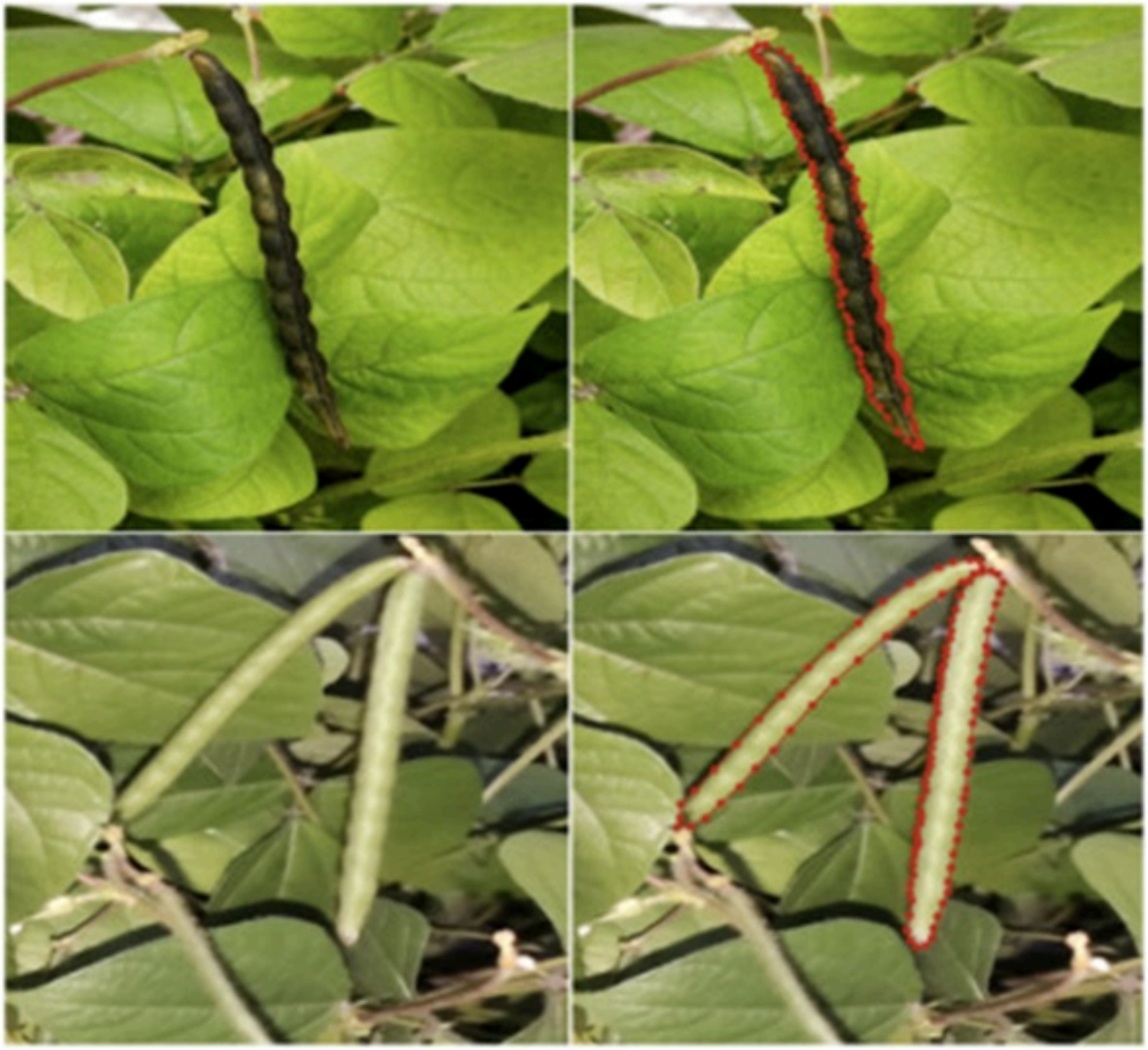


Figure 2. Green gram sample images from the dataset before annotation (left) and after annotation (right).

Images on the left side of the figure lack markings, while those on the right feature polygon outlines with red-dotted boundaries encircling the green gram pods.

Image augmentations

Detectron2 incorporates a dedicated dataset mapper that introduces transformations to increase dataset volume and mitigate overfitting during training. Applied techniques encompass resizing (800×600), random brightness adjustments (0.8 to 1.8), random contrast variations (0.6 to 1.3), random saturation shifts (0.8 to 1.4), random rotation (90°), random lighting changes (0.7), and random vertical flip (probability of 0.4). In addition to expanding the dataset, these modifications promote strong generalization when models operate in real-world production environments.

Experimental setup

Experiments were conducted using the Detectron2 [20] library, a modular, highly extensible object detection framework built on PyTorch [21]. It facilitates a wide range of computer vision operations, including object detection, keypoint

© 2024 The Author(s). | Published by Agro Intelligence Research Press

detection, and multiple segmentation categories (Semantic, Instance, and Panoptic). Training was conducted on the Google Colaboratory Pro platform [22], which provides essential resources, including a Tesla P100-PCIE-16GB GPU. The development machine was a MacBook Air running macOS Mojave, with Python 3.7 as the primary language. Supporting libraries included cv2, Pillow, and NumPy. The training hyperparameters included 2000 epochs, batch sizes of 2, 4, or 8 images per batch, learning rates of 1e−3 or 1e−2, and augmentation methods for random brightness, contrast, lighting, saturation, rotation, and vertical flip.

Performance evaluation metrics

The model's assessment relied on key indicators, including mean average precision (mAP), average recall (AR), and F1-score. Central to this process is the intersection-over-union (IoU) parameter, which plays a vital role in detection and localization evaluations. IoU values range from 0 to 1 and measure the overlap between the ground-truth and predicted bounding boxes. Larger overlaps signify improved model capability. Detections are valid only if the overlap exceeds the chosen IoU threshold; lower values classify them as invalid. Calculations include true positive (TP), true negative (TN), false positive (FP), and false negative (FN) counts. Precision is the proportion of TPs relative to the total of TPs and FPs, as defined in **Eq. 1**.

$$Precision\ \ (P) = \frac{TP}{TP+FP} \quad (1)$$

Recall represents the proportion of TPs against the total of TPs and FNs, detailed in **Eq. 2**.

$$Recall\ \ (R) = \frac{TP}{TP+FN} \quad (2)$$

F1-score combines precision and recall, formulated in **Eq. 3**.

$$F1 - score = \frac{2\ \times (P^{*}R)}{P+R} \quad (3)$$

This score offers a practical compromise between precision and recall.

Average Precision (AP) corresponds to the area under the precision-recall curve and is given by **Eq. 4**.

$$AP\ = \int_{0}^{1} P\left(R\right) \mathrm{d}R \quad (4)$$

Mean average precision is the primary indicator of overall object detection accuracy. It averages the AP scores across all n classes using **Eq. 5**.

$$mAP = \ \frac{1}{n} \sum_{i=1}^{n} AP_{i} \quad (5)$$

In this context, n stands for the total number of classes in the dataset.

Model architectures

© 2024 The Author(s). | Published by Agro Intelligence Research Press

PointRend [23] provides a versatile approach for adapting established detection models to produce semantic or instance segmentation results. Mask R-CNN is a popular two-stage instance segmentation model. Extending Faster R-CNN, it adds a dedicated mask prediction head that operates alongside standard bounding-box outputs. **Figure 3** presents the conventional Mask R-CNN structure utilized for instance segmentation [24].

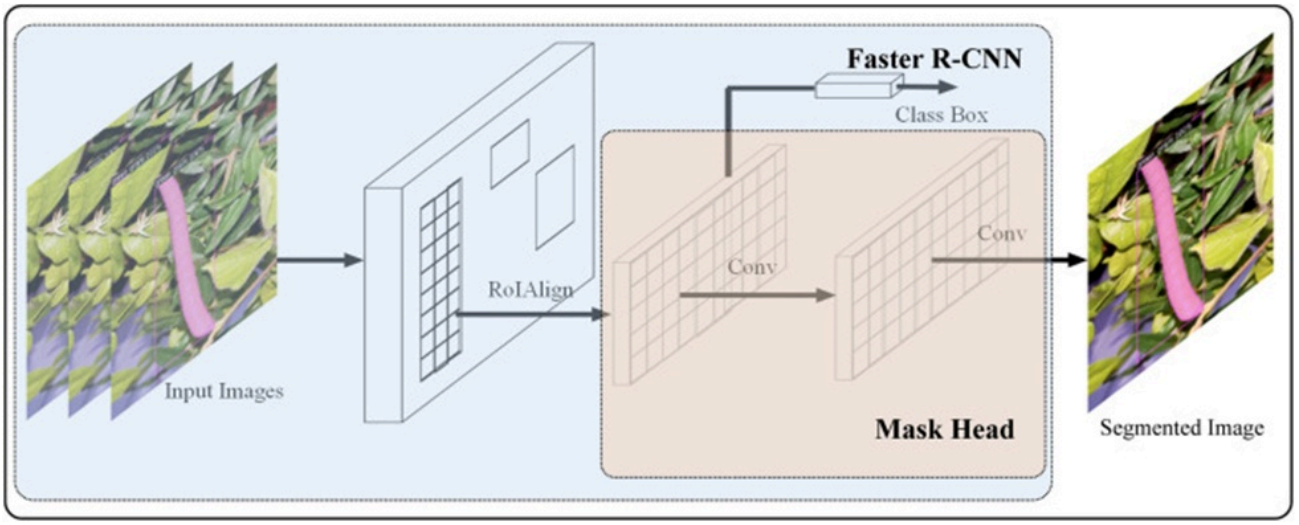


Figure 3. Detectron2's Mask R-CNN architecture for instance segmentation.

The RoIPool layer in Faster R-CNN applies rough spatial quantization to extracted features, often causing misalignment that disrupts accurate pixel-level correspondence. Mask R-CNN addresses this by using the RoIAlign layer, ensuring precise alignment and feature mapping. A mask head containing two convolutional layers then runs in parallel to the bounding box branch.

Outputs from the Mask R-CNN model feature predicted bounding boxes carrying class probabilities and boundary labels around green gram pods. For achieving refined segmentation of these pods, the framework illustrated in **Figure 4** advances Mask R-CNN by focusing on boundary enhancement for instance segmentation. It substitutes the original mask head (three FC layers: FC1, FC2, FC3) with a point-based mask head paired with a coarse head (two FC layers: FC1, FC2). Coarse features drawn from the backbone CNN feed into PointRend, which refines them into detailed features via point selection on uncertain points (N).

© 2024 The Author(s). | Published by Agro Intelligence Research Press

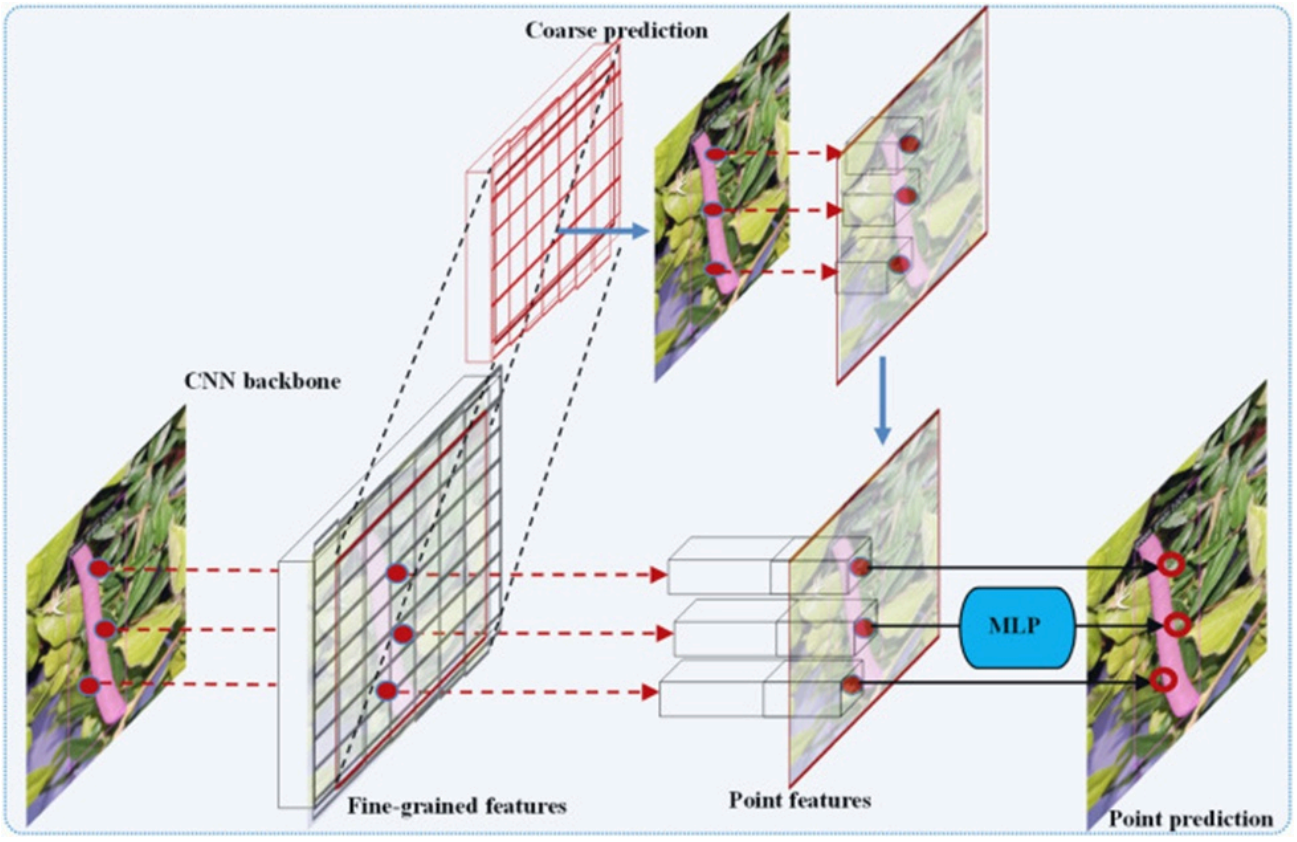


Figure 4. PointRend architecture for Instance segmentation, indicating point proposal and selection for improved class boundaries.

PointRend generates smooth boundaries via bilinear upsampling, boosting prediction resolution with a modest set of boundary points associated with the green gram pods.

Point selection strategy

The process for enhancing segmentation masks unfolds in successive stages. Neural networks first produce a rough segmentation mask for each green gram pod in the image. Uncertainty in these masks is then quantified at the pixel level using measures like entropy or confidence values. Areas showing high uncertainty reflect greater ambiguity in the initial predictions. Point sampling subsequently targets these unclear zones—typically pod edges or regions with sharp visual transitions—for detailed improvement. A structured grid sampling selects a defined number of points requiring refinement, focusing on boundary areas. Refining these points ultimately yields clear, accurate segmentation masks around the pods. This workflow substantially raises overall segmentation standards. The full point selection process appears in **Figure 5**.



Figure 5. Point selection strategy adopted in PointRend for improving the quality of the segmentation mask.

Loss function

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Three primary loss elements guide the training process: segmentation loss (SL), point-wise loss (PL), and the combined total loss. This total loss arises from a weighted summation of SL and PL.

Segmentation loss functions at the individual pixel level to calculate discrepancies between the model's predicted probability distributions and the corresponding ground truth labels for every pixel. Cross-entropy (CE) loss is the standard function applied in segmentation and is expressed in **Eq. 6**:

$$L_{seg} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \cdot \log\left(p_{i,c}\right) \quad (6)$$

Where:

• pi,c indicates the predicted probability that pixel i belongs to class c

• yi,c represents the actual ground truth label for pixel i in class c

• N equals the total number of pixels across the image

• C equals the total classes (set to C=1 here)

Point-wise loss assesses the success of mask refinement specifically at the chosen points inside each instance mask. Smaller values here lead to more precise object edges and reveal greater detail at those locations.

For each selected point m within an instance mask, this loss applies a function such as Smooth L1, as defined in **Eq. 7**:

$$L_{point} = \frac{1}{|M|} \sum_{m \in M} \mathrm{SmoothL1}\left(\hat{y}_m, y_m\right) \quad (7)$$

Where:

• |M| stands for the complete count of selected points

• ŷm denotes the refined prediction value for point m

• ym indicates the ground truth label at point m

The overall total loss forms through the weighted merging of SL and PL, following **Eq. 8**:

$$L_{total} = \lambda \cdot L_{seg} + (1 - \lambda) \cdot L_{point} \quad (8)$$

Where:

• λ serves as the adjustable hyperparameter managing emphasis between SL and PL depending on project needs

• Lseg refers to the segmentation loss

• Lpoint refers to the point loss

Deeper review indicates that losses in the PointRend model break down as:

© 2024 The Author(s). | Published by Agro Intelligence Research Press

(1) Region of interest (RoI) head: Includes RoI classification loss (L_{cls}^{ROI}) and RoI localisation loss (L_{loc}^{ROI}). L_{cls}^{ROI} evaluates the correct class assignment to predicted bounding boxes, while L_{loc}^{ROI} measures the closeness of those boxes to actual object positions.

(2) Region proposal network (RPN): Features RPN classification loss (L_{cls}^{RPN}) and RPN localisation loss (L_{loc}^{RPN}). L_{cls}^{RPN} determines effectiveness at classifying anchors as foreground or background. L_{loc}^{RPN} tracks accuracy in localizing proposed regions.

(3) Mask head: Contains mask loss (Lmask), which judges the accuracy of the produced binary masks.

(4) Point mask head: Holds point loss (LPoint), which shows the capability in handling uncertain points inside masks via cross-entropy calculations.

The total loss (LTotal) equals the weighted sum of these separate components and is recorded after each iteration. **Eq. 9** illustrates the overall loss for PointRend. Meanwhile, Mask R-CNN employs the same losses listed above, except it omits any dedicated term for the point head.

$$L_{Total} = L_{cls}^{ROI} + L_{loc}^{ROI} + L_{cls}^{RPN} + L_{loc}^{RPN} + L_{mask} + L_{point} \quad (9)$$

Results and Discussion

Model training and evaluation

Training and validation performance were compared between the standard Mask R-CNN (using its regular head) and the PointRend version (incorporating both point and standard heads). Detailed visual reviews confirm that PointRend delivers superior boundary quality—smooth yet crisp—surrounding the identified green gram pods. Building instance segmentation models for green gram remains challenging due to the strong visual similarity between the background and target objects (green gram pods). The proposed system's outputs were examined at the pixel level, including bounding boxes and corresponding pixel-level segmentation masks.

Evaluation of mean average precision (mAP)

Training of the PointRend model for detection and instance segmentation on the specialized dataset was performed in Google Colab. Two leading segmentation approaches underwent testing. The first involved Mask R-CNN+FPN across three backbone variants: ResNet101 (R101-FPN), ResNeXt101 (X101-FPN), and ResNet50 (R50-FPN). The second used PointRend+FPN with matching backbones and added the point head to support detailed predictions. Tabulated data enabled thorough model comparison. Metrics centered on mean average precision (mAP) computed across IoU thresholds from 0.5 to 0.95 at 0.05 intervals. AP50 reflects average precision at IoU=0.5, and AP75 at IoU=0.75. APM and APL cover mean average precision for medium instances (322 < area < 962) and large instances (area > 962). Evaluation considered both object positioning and boundary prediction. **Table 2** summarizes mAP figures for different configurations. For Mask R-CNN bounding box mAP, X101-FPN led with 52.25%, ahead of R101-FPN at 50.96% and R50-FPN at 48.49%. Over 2000 epochs, X101-FPN training lasted roughly 23.30 minutes, exceeding R101-FPN (12.05 minutes) and R50-FPN (10.03 minutes). **Table 3** presents segmentation mAP for Mask R-CNN, where X101-FPN reached 39.48%, followed by R101-FPN at 36.84% and R50-FPN at 33.07%, along with corresponding AP50, AP75, APM, and APL figures.

Table 2. Training results for the Mask R-CNN model (bounding box) in terms of mAP(best results are in bold face).

SL. No.	Baselines	Training time (hh: mm: ss)	Mean average precision (mAP) in%

© 2024 The Author(s). | Published by Agro Intelligence Research Press

			AP_L	AP_M	AP_{75}	AP_{50}	mAP
1.	**X101-FPN**	00:23:30	**78.34**	**45.82**	**59.51**	**67.73**	**52.25**
2.	R101-FPN	00:12:05	80.50	42.81	58.64	65.57	50.96
3.	R50 -FPN	00:10:03	78.31	39.76	55.37	65.21	48.49

Table 3. Training results for the Mask R-CNN model (segmentation) in terms of mAP(best results are in boldface).

SL. No.	Baselines	Training time (hh: mm: ss)	Mean average precision (mAP) in%				
			AP_L	AP_M	AP_{75}	AP_{50}	mAP
1.	**X101-FPN**	00:23:30	**67.67**	**26.79**	**46.48**	**63.70**	**39.48**
2.	R101-FPN	00:12:05	64.20	24.01	41.90	62.79	36.84
3.	R50 -FPN	00:10:03	61.76	18.93	37.21	58.02	33.07

Tables 4 and 5 display bounding box and segmentation mAP outcomes for the PointRend model.

Table 4. Training results for the PointRend model (bounding box) in terms of mAP(best results are in bold face).

SL. No.	Baselines	Training Time (hh: mm: ss)	Mean Average Precision (mAP) in%				
			AP_L	AP_M	AP_{75}	AP_{50}	mAP
1.	**X101-FPN**	00:42:54	**80.40**	**45.04**	**61.80**	**68.52**	**53.43**
2.	R101-FPN	00:20:46	80.89	43.74	58.41	67.30	51.79
3.	R50 -FPN	00:15:18	76.75	41.14	56.77	65.17	48.35

Table 5. Training results for the PointRend model (segmentation) in terms of mAP(best results are in bold face).

SL. No.	Baselines	Training time (hh: mm: ss)	Mean average precision (mAP) in%				
			AP_L	AP_M	AP_{75}	AP_{50}	mAP
1.	**X101-FPN**	00:42:54	**74.03**	**33.13**	**54.63**	**67.49**	**46.15**
2.	R101-FPN	00:20:46	74.33	29.68	53.41	65.81	44.45
3.	R50 -FPN	00:15:18	72.64	26.44	50.04	66.61	42.89

Figure 6 contains mAP evaluation plots. **Figures 6a and 6b** illustrate Mask R-CNN results for bounding box and segmentation, whereas **Figures 6c and 6d** show PointRend outcomes for the same tasks.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

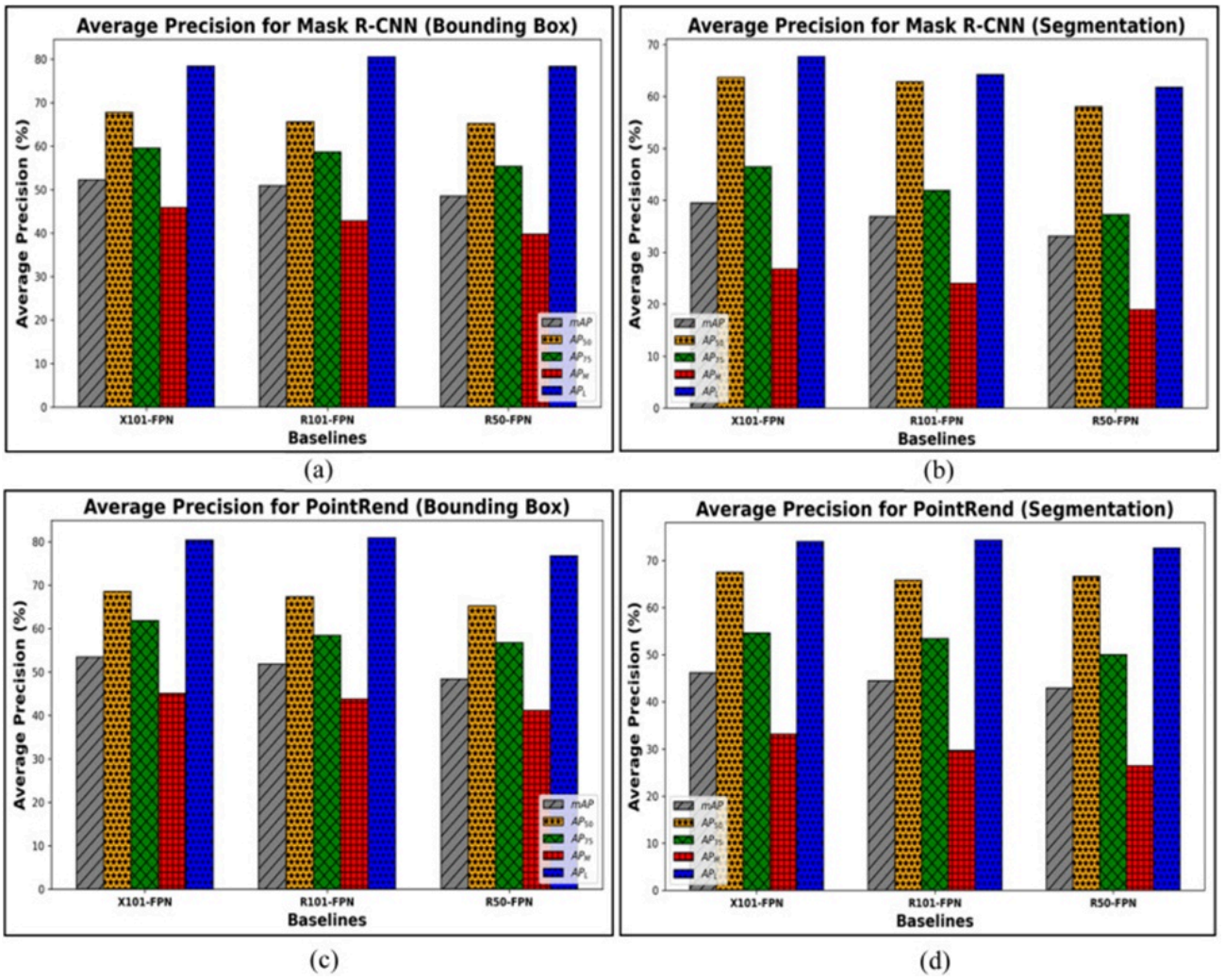


Figure 6. Plots showing variation in mean average precision (mAP) for (a) Mask R-CNN (Bounding box), (b) Mask R-CNN (Segmentation), (c) PointRend (Bounding box), and (d) PointRend (Segmentation).

Evaluation of average recall (AR)

Average recall measurements show the extent to which the trained model handles false negatives (FN) arising from its detections. Assessments rely on AR calculated with a cap of 10 detections per image (ARmax=10), applied separately to bounding box and segmentation outputs.

AR values for instance segmentation with the Mask R-CNN model are presented in **Tables 6 and 7** for the bounding box and segmentation tasks. As shown in **Table 6**, the X101-FPN backbone achieved the highest AR of 55.80%, outperforming the remaining baselines at 55.00% and 53.30%. **Table 7** indicates that X101-FPN achieved 43.10%, exceeding R101-FPN (42.20%) and R50-FPN (37.60%). Both tables also report figures for ARmax=100, ARM, and ARL.

Table 6. Training results for the Mask R-CNN model (bounding box) in terms of (AR) (best results are in bold face).

SL. No.	Baselines	Training time (hh: mm: ss)	Average recall (AR) in%			
			AR_L	AR_M	$AR_{max=100}$	$AR_{max=10}$
1.	**X101-FPN**	00:23:30	**84.30**	**54.70**	**61.10**	**55.80**

© 2024 The Author(s). | Published by Agro Intelligence Research Press

2.	R101-FPN	00:12:05	84.40	51.50	59.10	55.00
3.	R50 -FPN	00:10:03	84.10	55.10	61.20	53.30

Table 7. Training results for the Mask R-CNN model (segmentation) in terms of (AR) (best results are in bold face).

SL. No.	Baselines	Training time (hh: mm: ss)	Average recall (AR) in%			
			AR_L	AR_M	$AR_{max=100}$	$AR_{max=10}$
1.	**X101-FPN**	00:23:30	**72.30**	**36.30**	**45.60**	**43.10**
2.	R101-FPN	00:12:05	70.20	35.10	44.40	42.20
3.	R50 -FPN	00:10:03	67.70	31.30	40.90	37.60

Similarly, **Table 8** summarizes bounding-box AR performance for the PointRend model. The X101-FPN and R101-FPN backbones both reached a peak of 57.00%, while R50-FPN attained 52.30%. The table also includes the ARmax, ARM, and ARL metrics.

Table 8. Training results for the PointRend model (bounding box) in terms of (AR (best results are in bold face).

SL. No.	Baselines	Training time (hh: mm: ss)	Average recall (AR) in%			
			AR_L	AR_M	$AR_{max=100}$	$AR_{max=10}$
1.	X101-FPN	00:42:54	85.55	55.20	62.00	57.00
2.	**R101-FPN**	00:20:46	**86.20**	**55.30**	**62.50**	**57.00**
3.	R50 -FPN	00:15:18	82.70	55.40	61.60	52.30

Table 9 covers segmentation AR outcomes, where X101-FPN led with 49.40%, followed by R101-FPN (49.10%) and R50-FPN (48.00%). Additional entries for ARmax=100, ARM, and ARL are provided for the PointRend segmentation case.

Table 9. Training results for the PointRend model (segmentation) in terms of (AR) (best results are in bold face).

SL. No.	Baselines	Training time (hh: mm: ss)	Average recall (AR) in%			
			AR_L	AR_M	$AR_{max=100}$	$AR_{max=10}$
1.	**X101-FPN**	00:42:54	**77.10**	**47.00**	**54.00**	**49.40**
2.	R101-FPN	00:20:46	77.10	44.00	52.20	49.10
3.	R50 -FPN	00:15:18	76.30	46.90	53.90	48.00

F1-scores derived from the mAP and AR results for both models (using only top-performing backbones) are compiled in **Table 10** across bounding box and segmentation categories. Findings confirm that PointRend yields improved outcomes thanks to its point head, which supports more precise point-level predictions than the conventional Mask R-CNN head.

Table 10. Calculation of (F1-score) for Mask R-CNN and PointRend (Bounding Box and Segmentation) using mAP and AR (best results are in bold face).

© 2024 The Author(s). | Published by Agro Intelligence Research Press

SL. No.	Model	Segmentation evaluation (%)			Bounding box evaluation (%)		
		F1-score	AR	mAP	F1-score	AR	mAP
1.	Mask R-CNN	37.82	36.30	39.48	53.45	54.70	52.25
2.	**PointRend**	**46.57**	**47.00**	**|46.15**	**54.30**	**55.20**	**53.43**

Figure 7 features AR graphs for bounding box and segmentation across the two models. **Figures 7a and 7b** track bounding box AR shifts for Mask R-CNN and PointRend, whereas **Figures 7c and 7d** display segmentation AR trends. Charts show notably higher AR scores for bounding boxes than for segmentation results. PointRend demonstrated superior performance in both evaluation categories.

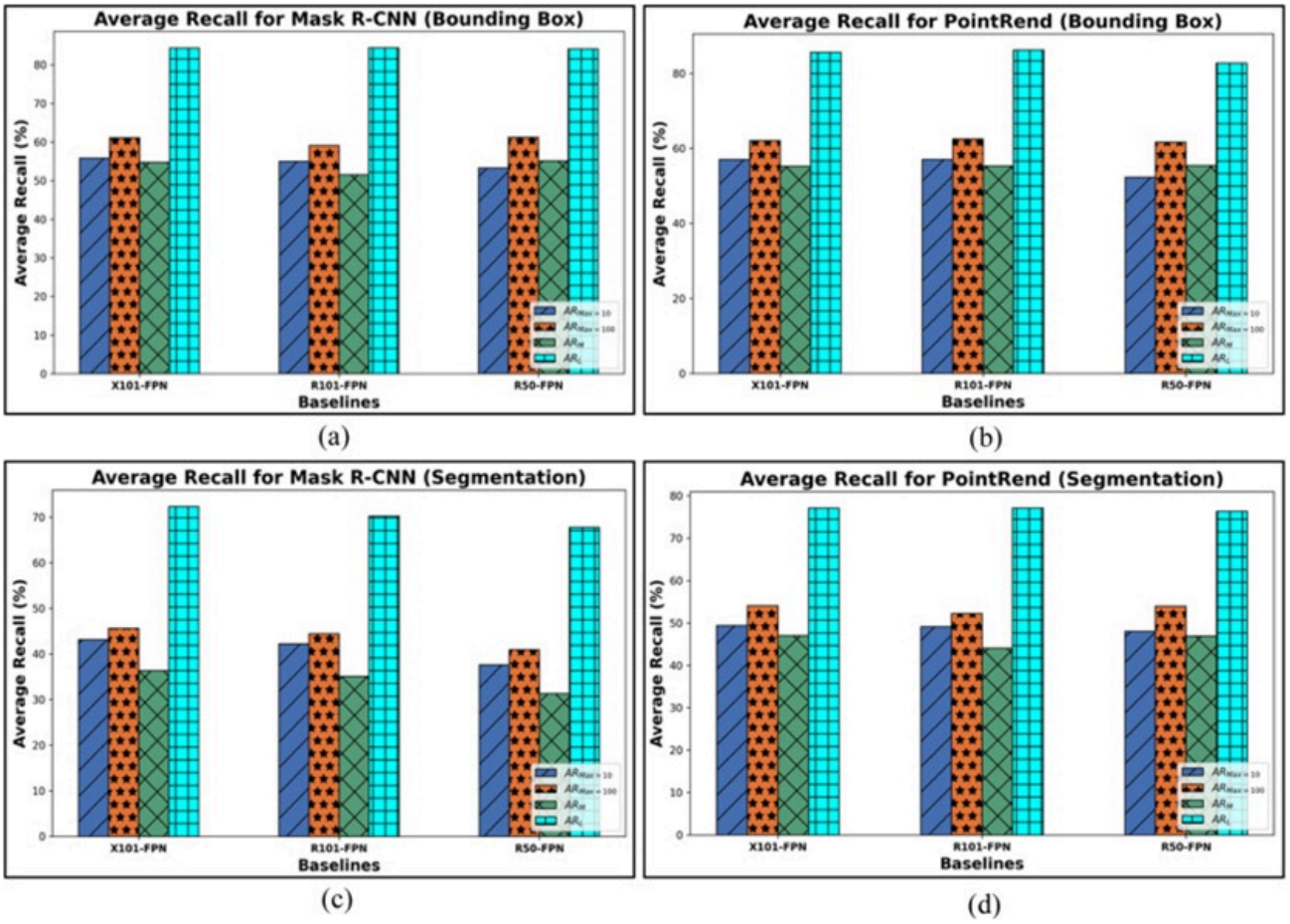


Figure 7. Plots showing variation in average recall (AR) for (a) Mask R-CNN (Bounding box), (b) PointRend (Bounding box), (c) Mask R-CNN (Segmentation), and (d) PointRend (Segmentation).

Figure 8 shows F1-scores alongside mAP and AR for Mask R-CNN and PointRend. **Figure 8a** addresses bounding box data, and **Figure 8b** addresses segmentation data.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

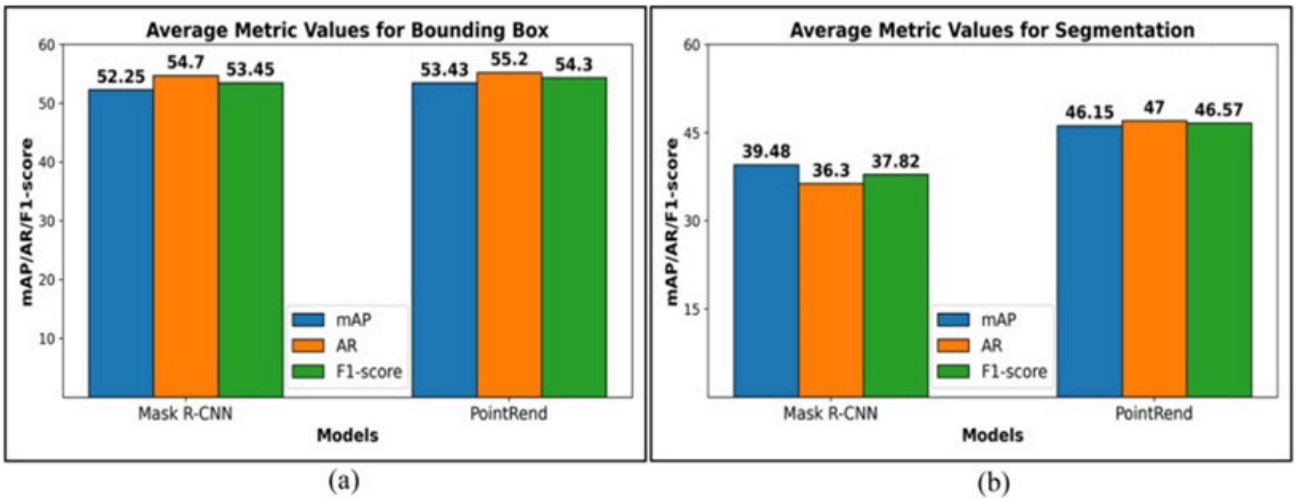


Figure 8. Variation of mAP, AR, and F1-score for (a) Mask R-CNN and PointRend (Bounding box) and (b) Mask R-CNN and PointRend (Segmentation).

Figure 9 illustrates validation outcomes achieved by the PointRend model. These outcomes highlight the value of the rendering strategy in strengthening both detection accuracy and instance segmentation refinement.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

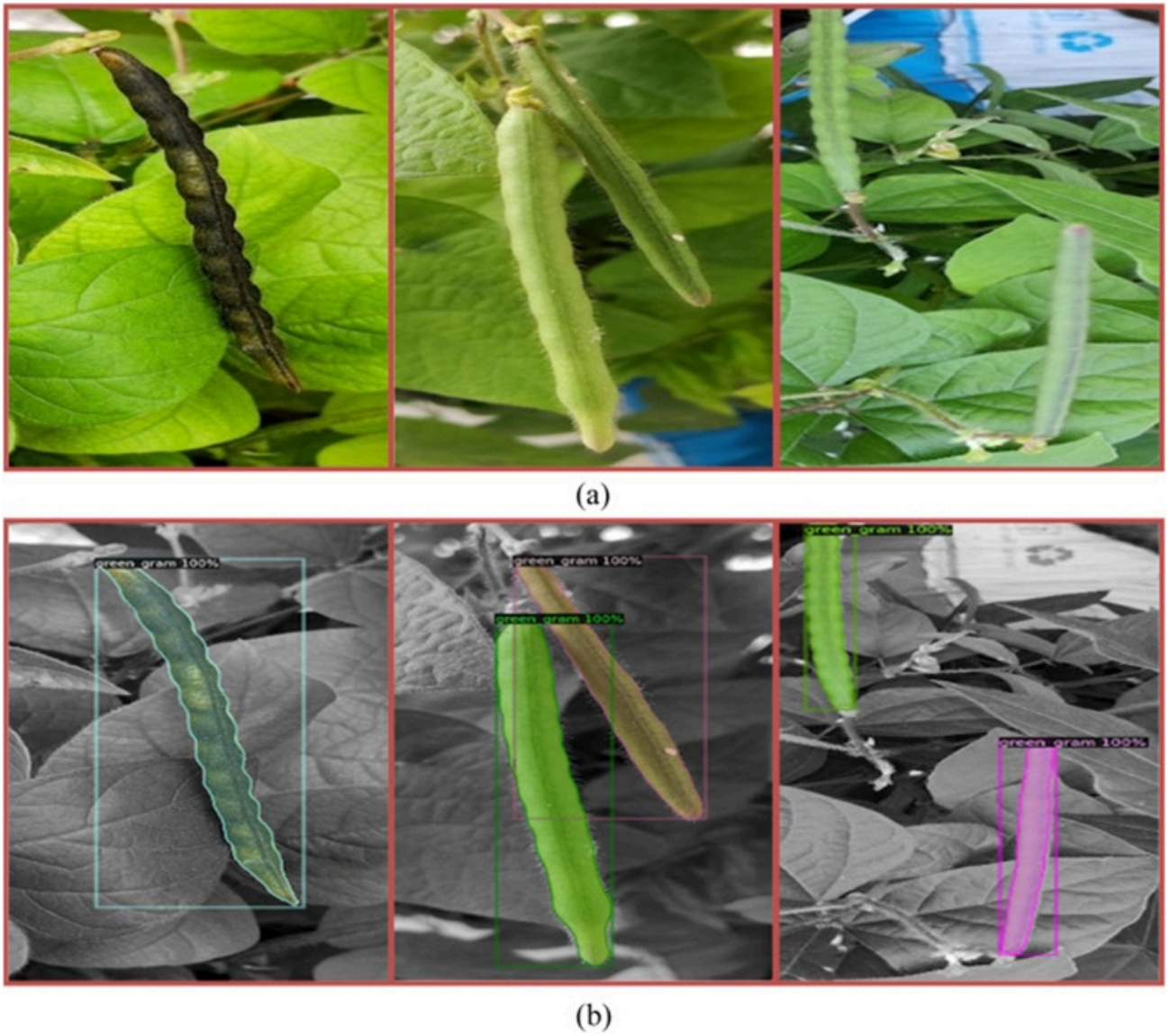


Figure 9. The model's performance on sample images from the validation set includes: (a) Input images and (b) Corresponding model-predicted images with bounding boxes and pixel-level class boundaries.

The upper images show raw validation samples, while the lower images show model-generated outputs. Images on the right emphasize bounding-box placement alongside pixel-level segmentation.

Compared with Mask R-CNN, the elevated confidence scores, compact bounding boxes, and refined smooth-sharp pod outlines demonstrate substantial promise. Additional validation involved testing on previously unseen images, confirming the model's ability to produce high-quality instance segmentation, marked by delicate, fluid, and distinct boundaries around green gram pods.

Figure 10 provides a visual summary of model behavior on unseen data. To keep things clear, three sample inputs (labeled 1, 2, and 3) appear in **Figures 10a-10c**. PointRend predictions exhibit noticeably cleaner boundary definition than those from Mask R-CNN. Zoomed sections uncover finer structural details. The figure also shows the false positives (FPs) and false negatives (FNs) generated by each model. In summary, PointRend exceeds Mask R-CNN by delivering enhanced instance segmentation, with smooth, distinct edges encircling green gram pods.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

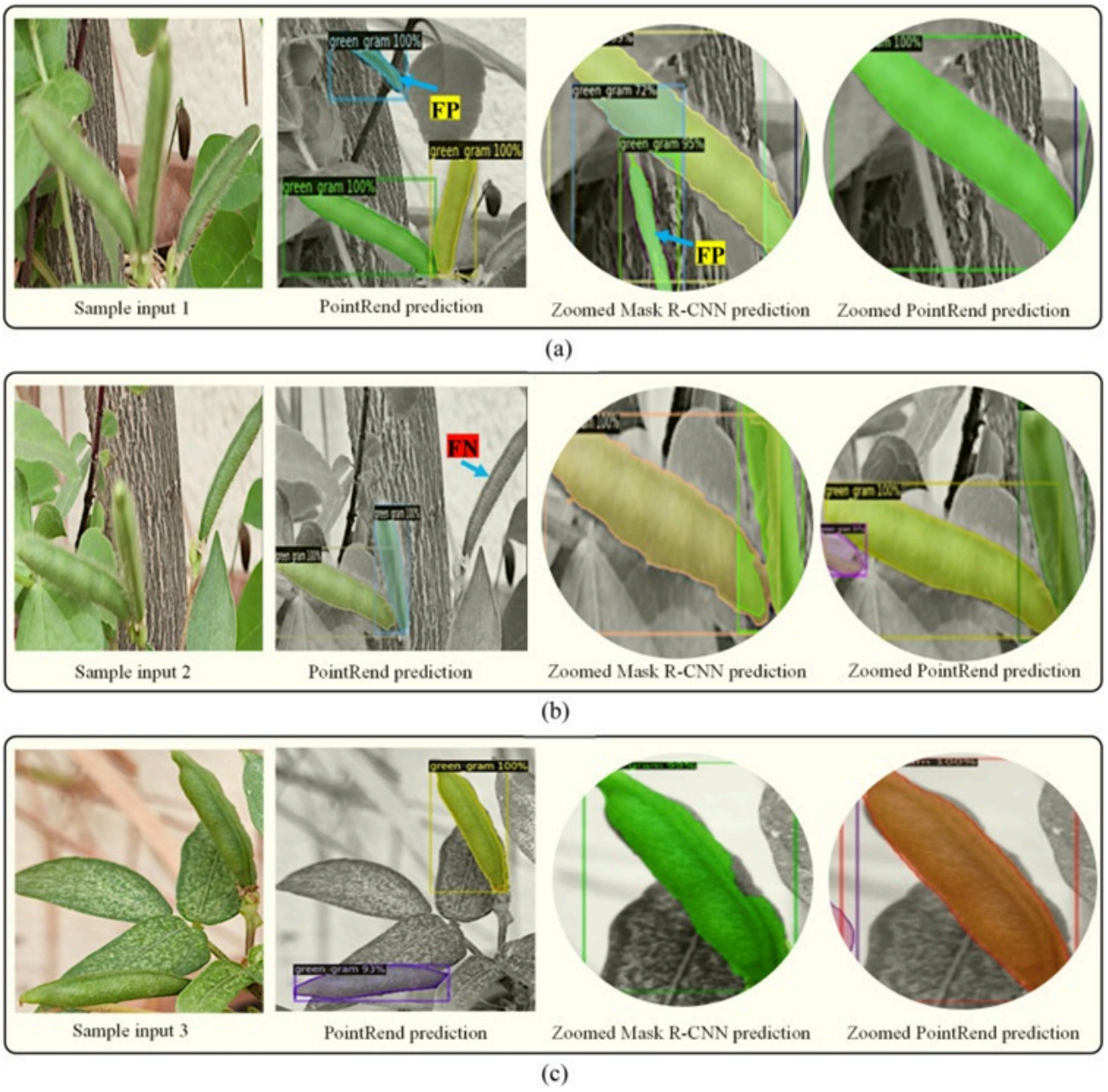


Figure 10. The visual evaluation of the Mask R-CNN and PointRend trained models on unseen images. The sequence followed is: sample input image-PointRend Prediction-Zoomed Mask R-CNN prediction-Zoomed PointRend Prediction, where: (a) Sample input 1, (b) Sample input 2, and (C) Sample input 3.

Loss function evaluation

After completing the training runs, recorded loss values were reviewed using the components introduced in Section 2.6.2. To focus the discussion, only PointRend-related losses are included here. Altogether, seven separate loss terms were examined.

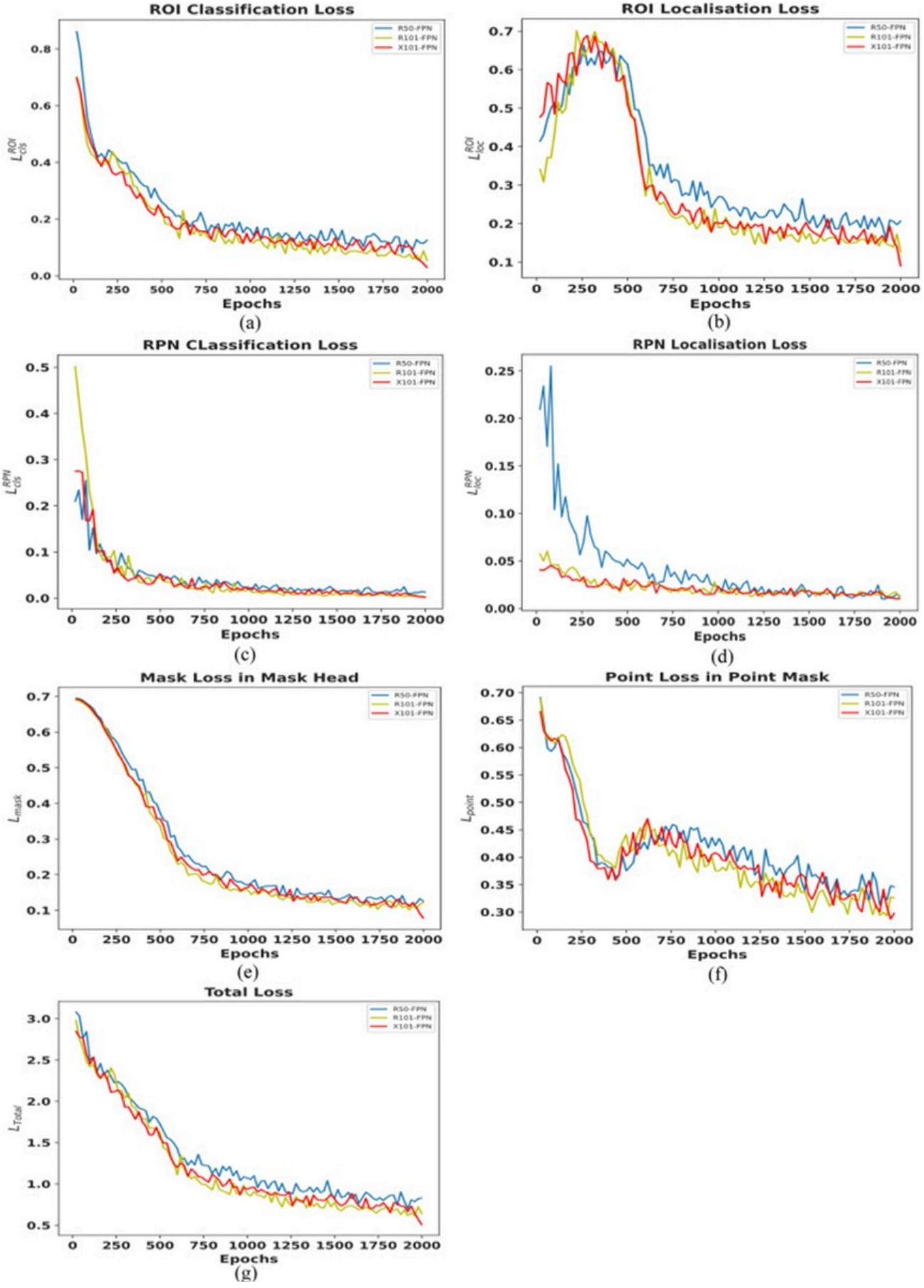
The RoI head contributes two fundamental terms: RoI classification loss (L_{cls}^{ROI}) and RoI localization loss (L_{loc}^{ROI}). The RPN supplies the RPN classification loss (L_{cls}^{RPN}) and RPN localization loss (L_{loc}^{RPN}). Remaining terms consist of mask loss from the mask head (Lmask) and point loss from the point mask head (Lpoint). A weighted combination of all terms forms the total loss (LTotal).

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Studying these individual losses offers critical insight into model behavior across the training cycle. Such analysis helps reduce mistakes in class assignments, spatial positioning, and boundary refinement.

In addition, loss patterns support the identification of optimal hyperparameter settings that further enhance overall model performance. Visual trends in the losses are shown in **Figure 11** for the R50-FPN, R101-FPN, and X101-FPN configurations. **Figure 11a** charts RoI classification loss progression over epochs. This loss steadily declines and stabilizes by the 2000th epoch with final values of R50-FPN=0.1257, R101-FPN=0.0538, and X101-FPN=0.0292. The X101-FPN backbone produced the lowest RoI classification loss (0.0292) compared to the others. For RoI localisation loss shown in **Figure 11b**, X101-FPN (0.09) again surpassed R50-FPN (0.1253) and R101-FPN (0.2067). RPN classification and localisation losses are plotted in **Figures 11c and 11d**. Mask loss and point loss appear in **Figures 11e and 11f**, with the overall total loss in **Figure 11g**. Detailed inspection confirms that X101-FPN consistently delivers the strongest results among the tested backbones.

© 2024 The Author(s). | Published by Agro Intelligence Research Press



© 2024 The Author(s). | Published by Agro Intelligence Research Press

Figure 11. Plot indicating the loss function evaluation (a) RoI Classification Loss, (b) RoI Localisation Loss, (c) RPN Classification Loss, (d) RPN Localisation Loss, (e) Mask Loss, (f) Point Loss, and (g) Total Loss.

Findings from the PointRend model across both training and testing stages clearly establish its dominance over the Mask R-CNN model in overall effectiveness. As highlighted earlier, this research marks the initial attempt to apply advanced rendering techniques, such as segmentation, using green gram pods as the specific case study. The initial PointRend model improved segmentation performance on the COCO and Cityscapes datasets, achieving an average precision of 38.2% at a top image size of 224×224. By comparison, the current framework achieved a higher mAP of 46.15% on larger 1024×1024 images and produced noticeably superior instance segmentation quality compared to Mask R-CNN's 39.48%.

Research by Liu *et al.* [25] addressed a related problem: spotting cucumber fruits against complex backgrounds that blend with the fruit itself. Their work strengthened instance segmentation via Mask R-CNN and delivered impressive results against alternatives (Yolo V2, Yolo V3, and basic Mask R-CNN), but paid limited attention to sharpening mask edge precision. A different effort by Yao *et al.* [26] explored and contrasted models such as Mask R-CNN, Mask Scoring R-CNN, and Cascade R-CNN with several backbone options for segmenting peach fruit diseases. Separately, Wang *et al.* [27] proposed a YOLOv5 approach for greenhouse cucumber detection that relied on color-space attributes. **Table 11** provides a side-by-side performance comparison of the current models (Mask R-CNN and PointRend) with comparable literature methods for peach disease and cucumber fruit segmentation. Observations confirm that the suggested PointRend model exceeds both the tested Mask R-CNN and prior literature models across bounding box and segmentation mean average precision. The new PointRend system achieved a 39.10% gain in bounding-box AP50 and a 40.80% increase in segmentation AP50. In addition to stronger mean average precision scores, PointRend stands out for its enhanced segmentation quality. The ResNeXt101-FPN backbone configuration, applied across Mask R-CNN and PointRend trials, delivered the top results.

Table 11. Comparison of the proposed work with a similar implementation from the literature (best results are in bold face).

SL. No.	Reference	Dataset category	Agricultural commodity studied	Task performed	Model/Network	Bounding box AP50	Segmentation AP50
1	[26]	Peach Disease Image Dataset (PDID)	Detection of peach diseases	Segmentation	Cascade R-CNN	0.450	0.243
					Mask Scoring R-CNN	0.301	0.279
					Mask R-CNN	0.294	0.267
2	[27]	Custom dataset	Detection of cucumber fruits	Object detection (Bounding Box)	YOLO V5	0.550	–
4	Proposed approach	Custom dataset	Green gram pod identification	Segmentation with boundary refinement	Mask R-CNN	0.677	0.637
					PointRend	0.685	0.675

Conclusion

Current computer vision applications in farming require precise identification and positioning of various crops, including field, fruit, and vegetable crops, to support automation that reduces manual effort and time for producers. For this reason, detection systems must achieve pixel-level accuracy to ensure flawless operation. The presented framework meets this

© 2024 The Author(s). | Published by Agro Intelligence Research Press

need by creating a PointRend-powered model using the Detectron2 platform for instance segmentation and by illustrating clear boundary improvements through point-driven, refined forecasting. Moreover, side-by-side evaluations of Mask R-CNN setups (with three different backbone options) against the PointRend model confirm the advantages of the new design. Results show that the PointRend approach yields higher mean average precision scores (covering both bounding boxes and segmentation) and well-defined, fluid class outlines around green gram pods in actual field conditions. Positive results on test samples (previously unseen images) indicate strong adaptability of the rendering strategy. Compared to standard Mask R-CNN, the PointRend model achieves higher accuracy (a 2.26% mAP gain for bounding box detection and a 16.90% mAP increase for segmentation) and higher-quality boundaries around green gram pods.

The technique introduced here is the first to evaluate a rendering-driven instance segmentation system focused on green gram pods—a challenging small-object detection task—within real field settings, where difficulties arise from the near-identical appearance of pods and surrounding leaves. The suggested method supplies more reliable object (pod) localization than the tested Mask R-CNN, standard Mask R-CNN, and Mask Scoring R-CNN versions, paired with refined segmentation masks. Accurate field-level pod detection enables dependable yield prediction, supporting better organization of post-harvest processes and ultimately raising farmer earnings while promoting more sustainable farming. Upcoming studies intend to broaden the dataset by adding further field photographs (including those with diseased pods) to increase model strength and resilience.

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 09 May 2023 Revised: 14 Jul 2023 Accepted: 31 Jul 2023
Published online: 25 November 2024

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at https://creativecommons.org/licenses/by-nc-sa/4.0/.

References

1. Bhujel A, Khan F, Basak JK, Jaihuni M, Sihalath T, Moon BE, et al. Detection of gray mold disease and its severity on strawberry using deep learning networks. J Plant Dis Prot. 2022. https://doi.org/10.1007/s41348-022-00578-8.

2. Thenmozhi K, Reddy US. Crop pest classification based on deep convolutional neural network and transfer learning. Comput Electron Agric. 2019;164:104906. https://doi.org/10.1016/j.compag.2019.104906.

3. Math RKM, Dharwadkar NV. Deep learning and computer vision for leaf miner infestation severity detection on muskmelon (Cucumis melo) leaves. Comput Electr Eng. 2023;110:108843. https://doi.org/10.1016/j.compeleceng.2023.108843.

4. Huang N, Chou D, Lee C, Wu F, Chuang A, Chen Y, et al. Smart agriculture: Real-time classification of green coffee beans by using a convolutional neural network. IET Smart Cities. 2020;2(4):167-72. https://doi.org/10.1049/iet-smc.2020.0068.

5. Kaplan Berkaya S, Sora Gunal E, Gunal S. Deep learning-based classification models for beehive monitoring. Ecol Inform. 2021;64:101353. https://doi.org/10.1016/j.ecoinf.2021.101353.

6. Barré P, Stöver BC, Müller KF, Steinhage V. LeafNet: A computer vision system for automatic plant species identification. Ecol Inform. 2017;40:50-6. https://doi.org/10.1016/j.ecoinf.2017.05.005.

7. Thendral R, Suhasini A, Senthil N. A comparative analysis of edge and color-based segmentation for orange fruit recognition. In: Proceedings of the International Conference on Communication and Signal Processing. 2014. p. 463-6. https://doi.org/10.1109/ICCSP.2014.6949884.

8. Ma Y, Huang M, Yang B, Zhu Q. Automatic threshold method and optimal wavelength selection for insect-damaged vegetable soybean detection using hyperspectral images. Comput Electron Agric. 2014;106:102-10. https://doi.org/10.1016/j.compag.2014.05.014.

9. Xiang R, Jiang H, Ying Y. Recognition of clustered tomatoes based on binocular stereo vision. Comput Electron Agric. 2014;106:75-90. https://doi.org/10.1016/j.compag.2014.05.006.

10. Rabby MKM, Chowdhury B, Kim JH. A modified canny edge detection algorithm for fruit detection and classification. In: Proceedings of the 10th International Conference on Electrical and Computer Engineering (ICECE). 2018. p. 237-40. https://doi.org/10.1109/ICECE.2018.8636811.

11. Muruganandam P, Tandon V, Baranidharan B. Rice crop diseases and pest detection using edge detection techniques and convolution neural network. In: Bansal JC, Engelbrecht A, Shukla PK, editors. Computer Vision and Robotics. Singapore: Springer; 2022. p. 49-64. https://doi.org/10.1007/978-981-16-8225-4_5.

12. Niu C, Li H, Niu Y, Zhou Z, Bu Y, Zheng W. Segmentation of cotton leaves based on improved watershed algorithm. In: Li D, Li Z, editors. Proceedings of the Computer and Computing Technologies in Agriculture IX.. Springer International Publishing; 2016; 478: 425-36. https://doi.org/10.1007/978-3-319-48357-3_41.

13. Tian K, Li J, Zeng J, Evans A, Zhang L. Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm. Comput Electron Agric. 2019;165:104962. https://doi.org/10.1016/j.compag.2019.104962.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

14. Sampathkumar S, Rajeswari R. An automated crop and plant disease identification scheme using cognitive fuzzy C-means algorithm. IETE J Res. 2020:1-12. https://doi.org/10.1080/03772063.2020.1780163.

15. Math RM, Dharwadkar NV. Early detection and identification of grape diseases using convolutional neural networks. J Plant Dis Prot. 2022;129:521-32. https://doi.org/10.1007/s41348-022-00589-5.

16. Afzaal U, Bhattarai B, Pandeya YR, Lee J. An instance segmentation model for strawberry diseases based on Mask R-CNN. Sensors. 2021;21(19):6565. https://doi.org/10.3390/s21196565.

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

18. Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput Electron Agric. 2019;157:417-26. https://doi.org/10.1016/j.compag.2019.01.012.

19. Lamb N, Chuah MC. A strawberry detection system using convolutional neural networks. In: Proceedings of the IEEE International Conference on Big Data (Big Data). 2018. p. 2515-20. https://doi.org/10.1109/BigData.2018.8622466.

20. Wu Y, Kirillov A, Massa F, Lo W. Detectron2. GitHub; 2019. https://github.com/facebookresearch/detectron2.

21. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. Curran Associates, Inc; 2019;32:8024-35. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

22. Bisong E. Google Colaboratory. In: Bisong E. Building machine learning and deep learning models on Google Cloud Platform. Apress; 2019. p. 59-64. https://doi.org/10.1007/978-1-4842-4470-8_7.

23. Kirillov A, Wu Y, He K, Girshick R. PointRend: Image segmentation as rendering. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2019. p. 9796-805. arXiv:1912.08193.

24. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. 2017. p. 2961-9. https://doi.org/10.1109/ICCV.2017.322.

25. Liu X, Zhao D, Jia W, Ji W, Ruan C, Sun Y. Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access. 2019;7:139635-42. https://doi.org/10.1109/ACCESS.2019.2942144.

26. Yao N, Ni F, Wu M, Wang H, Li G, Sung WK. Deep learning-based segmentation of peach diseases using convolutional neural network. Front Plant Sci. 2022;13. https://doi.org/10.3389/fpls.2022.876357.

27. Wang N, Qian T, Yang J, Li L, Zhang Y, Zheng X, et al. An enhanced YOLOv5 model for greenhouse cucumber fruit recognition based on color space features. Agriculture. 2022;12(10):1556. https://doi.org/10.3390/agriculture12101556.

© 2024 The Author(s). | Published by Agro Intelligence Research Press