ORIGINAL RESEARCH **Open access**

Instance Segmentation of Pepper Plants for Automated Phenotyping under Field Conditions

Sara Ben Youssef[1*], Amal Trabelsi[1], Karim Boudiaf[2], Nabil Jebali[3]

Abstract

A significant share of resources in vegetable breeding firms is allocated to phenotyping. The progress of computer vision technologies now permits the digitization of these operations, yielding enhancements in both efficiency and quality. Phenotyping, however, is commonly conducted in open fields or greenhouses, where environmental factors and lighting are constantly changing. This lack of uniform conditions creates difficulties when attempting to automatically identify the relevant objects in images—a crucial initial step in phenotyping. Conventional image analysis approaches have been found to lack the robustness needed to cope with such variability. In contrast, deep learning models have, over recent years, proven capable of detecting and internalizing salient features that are more resilient and reflective of the core patterns, thereby managing diverse and variable conditions effectively. In the present study, we offer a deep learning-driven solution for the instance segmentation of peppers post-harvest under field settings. The approach is implemented and validated on three pepper varieties: Blocky Bell, Jalapeño, and Lamuyo. For each variety, we compare the performance of this novel method with an earlier solution rooted in classical image processing, aiming to demonstrate the advantage of deep learning-based instance segmentation over traditional techniques as a first phenotyping step. Deep learning-based instance segmentation models surpass the results of classical image processing algorithms across all three pepper varieties: in Blocky Bell, mAP improves from 0.63 to 0.97; in Jalapeño, from 0.39 to 0.52; and in Lamuyo, from 0.67 to 0.97.

Keywords Deep learning, Phenotyping, Instance segmentation, Peppers, Breeding, Mask RCNN

*Correspondence:
Sara Ben Youssef
sara.benyoussef@inat.ucar.tn
[1] Department of AI for Barley and Durum Wheat Drought Resistance, Faculty of Agriculture, University of Tunis, Tunis, Tunisia
[2] Department of Machine Learning for Soil Salinity Mapping, Faculty of Agriculture, University of Sousse, Sousse, Tunisia
[3] Department of Smart Irrigation Scheduling, Faculty of Agriculture, University of Sfax, Sfax, Tunisia

Introduction

To sustain a world population expected to reach 10 billion by 2050, it is imperative to increase food production and reduce waste. A principal goal, therefore, is to breed crops—be they cereals, orchard trees, or vegetables—for greater yields and enhanced tolerance to both biotic and abiotic stresses.

Vegetable breeding enterprises rely on digital phenotyping to swiftly and precisely assess a plant's physical traits and identify the genetic factors underlying those traits. This capability underpins data-informed choices when selecting superior varieties aligned with consumer requirements. Nonetheless, gathering adequate phenotypic data with both precision and cost-effectiveness remains a persistent hurdle: even with the widespread availability of affordable phenotyping devices such as smartphones and mobile cameras, acquiring images in a standardized fashion continues to require considerable time from breeding teams, driving up costs.

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

Digitizing this workflow using computer vision to derive phenotyping features, such as vegetable dimensions and shape, paves the way for greater efficiency and higher quality. That said, phenotyping is often performed directly in the field or greenhouse, precluding the strict image standardization found in other industrial domains. This lack of uniformity poses a challenge for automatically distinguishing objects of interest from their surroundings (image segmentation), as classical image analysis methods falter when confronted with shifting illumination, shiny surfaces, cluttered backgrounds, and intricate scenes [1, 2]. Traditional segmentation pipelines generally follow a rigid sequence—color thresholding, morphological transformations, and geometric filtering, among others—to extract instances from source images. Even where parameters for these steps are selected adaptively (e.g., informed by preliminary image analyses), managing the full scope of biological and environmental variation described is exceedingly difficult in practice. A sturdy, precise, and extensible instance segmentation pipeline is needed to build efficient digital phenotyping workflows, allowing development efforts to focus on trait extraction rather than generic computer vision challenges.

Segmentation built upon deep learning frameworks is rapidly emerging as a standard industrial practice. Training and running deep learning models has become increasingly feasible computationally, and contemporary networks can produce remarkable results with comparatively modest training sets. An additional strength of this paradigm lies in its tolerance for non-uniform imaging settings, provided the training data adequately represent the expected range of image variation. In light of this, shifting to deep learning segmentation appears to be a logical progression for refining digital phenotyping pipelines.

Instance segmentation is the computer vision task of detecting and segmenting each distinct object in an image. Performing instance segmentation in the wild—that is, in authentic, uncontrolled settings—introduces several challenges: object occlusion, differences in scale and size, variations in shape and visual appearance, and complex backgrounds. Furthermore, compiling large, varied, and annotated datasets to train instance segmentation models for real-world applications can be demanding. Labeled data may be scarce, especially for uncommon or highly specific object categories, hindering the model's capacity to generalize and perform robustly. Another obstacle involves extending the model's capabilities to novel object classes. Models trained on a fixed set of classes often struggle when encountering previously unseen categories. Tailoring these models to accommodate new classes requires additional data collection, annotation, and fine-tuning, all of which consume substantial time and resources. Confronting these issues requires innovations in computer vision algorithms, encompassing robust feature representations, effective handling of scale and appearance changes, occlusion management strategies, data augmentation methods, and streamlined inference procedures. Equally, access to rich, meticulously annotated datasets that closely mimic real-world conditions can greatly boost the performance of instance segmentation models operating in the wild.

Within the narrower domain of pepper phenotyping, until recently, separating individual peppers in real-world imagery still depended on classical image processing methods. Yet certain conditions reveal the limitations of these classical techniques for this application. Elements such as discoloration or blemishes on the fruit, soil, foliage, ambient lighting, background hues, and overlapping peppers all lead to segmentation inaccuracies. Moreover, adopting classical image processing techniques hampers the solution's scalability. Indeed, a bespoke algorithm must be manually engineered for each pepper variety, requiring substantial work whenever a new variety is introduced into the automated fruit phenotyping system. It is also highly probable that the algorithm will need recalibration whenever an aspect of the image-capture setup changes.

In this paper, we measure the extent to which deep learning instance segmentation outperforms classical methods in a multi-variety pepper phenotyping context. To this end, we design, realize, and validate a deep learning-driven solution for pepper instance segmentation. We confirm the generalization strength of our novel solution on three pepper varieties: Blocky Bell, Jalapeño, and Lamuyo.

Our principal aim with this work is to put forward an instance segmentation solution that is currently deployed in operational, real-world digital phenotyping workflows worldwide. We outline the protocol and pipeline used in these operations and demonstrate that deep learning is an effective approach for reliable instance segmentation under field

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

conditions. Replacing manual phenotyping with a digital counterpart yields savings in both time and expense. Critically, our solution maintains precision and robustness even under variable conditions, making it a credible alternative to current manual phenotyping workflows. To summarize, this paper outlines our effective digital phenotyping pipeline, which can serve as a template for other international fruit phenotyping teams.

The structure of the paper is as follows: Section 2 delivers a review of the state of the art in this area, Section 3 describes the dataset and its preparation. Section 4 details the proposed method for pepper instance segmentation, and Section 5 describes the conducted experiment. Subsequently, Section 6 presents the achieved results, Section 7 discusses these results, and Section 8 concludes the paper.

Related Work

The adoption of digital tools and artificial intelligence to automate visual inspection tasks in agriculture is steadily gaining ground across a wide range of operations. Numerous studies over the past few years have proposed methods to address diverse agricultural challenges. These include approaches for crop disease recognition [3, 4], pest detection and enumeration [5, 6], damage assessment on greenhouse plants [7], and plant classification [8, 9]. Several useful, carefully structured surveys are available to help navigate the state of the art across these varied research directions [1, 2, 10-12].

Sa *et al.* [13] target the segmentation of fruits and peppers for robotic picking, expanding the Fast RCNN architecture and introducing a multi-modal Faster R-CNN model. Tunio *et al.* [14] put forward a deep learning pipeline for fruit detection and segmentation, exploring several strategies. They advance a modified U-Net design and employ transfer learning to improve performance, while also evaluating their model against alternative detection frameworks such as YOLOv4, achieving 98% accuracy. In Bargoti and Underwood [15], a deep learning framework tailored for orchard fruit detection is introduced. The authors build upon a Faster R-CNN base fitted with a region proposal network (RPN) to secure precise object localization and classification. This study demonstrates the deployment of a cutting-edge detection framework, Faster R-CNN, for detecting fruits such as mangoes, almonds, and apples in orchard environments. San *et al.* [16] describe a fruit classification scheme that draws on both color properties and morphological operations. These two feature categories are merged into a unified descriptor and supplied to various classifiers—Support Vector Machines, Random Forest, KNN—tasked with characterizing each instance. The work proposes an efficient fusion of color and shape information, achieving high classification scores for grapes, apples, tomatoes, pears, and cherries. Notably, the number of extracted features varies per object, and the imagery is captured under controlled conditions. Such an approach would likely fail in uncontrolled outdoor settings, where objects overlap and illumination shifts.

A valuable review addresses segmentation techniques applied to assorted fruit use cases within yield contexts [17]. That paper examines many works from differing angles. For example, it offers a summary of acquisition systems spanning RGB and hyperspectral sensors. Various methodological strands are also taken into account: techniques rooted in classical image processing, those based on feature extraction followed by classification, and finally solutions underpinned by deep learning. The shortcomings and constraints of conventional methods vis-à-vis emerging deep learning alternatives are thoroughly discussed.

Certain studies focus specifically on pepper segmentation across different setups—laboratory benches, greenhouses, and real-world conditions. The inconsistent lighting typical of greenhouses or authentic harvesting sites often creates shadows and overexposed areas, making the task more demanding. These investigations may leverage either classical image processing combined with machine learning or more contemporary deep learning-based pipelines.

Gao *et al.* [18] devise a method for estimating chili pepper dimensions. Features are derived from segmented regions, with segmentation based on an algorithm that blends color-space conversions and morphological processing. Rajalakshmi and Subashini [19] present a chili pepper segmentation method that uses X-ray imagery for texture segmentation via Gabor filters. The resulting texture-segmented output is refined through entropy, standard deviation, and range filtering. Once texture analysis is complete, additional features can be mined using statistical techniques. The study

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

employs Gray Level Co-occurrence Matrices and first-order statistics as feature-extraction tools applied to the texture-filtered image. Extracted statistical features are then passed to a pixel-level Support Vector Machine (SVM) classifier. Ostovar *et al.* [20] design a method for segmenting yellow peppers inside greenhouses to support robotic harvesting. Thresholding techniques are standardly applied to separate objects from the image background. Because fruit color shifts over the growing season and illumination is spatially uneven, the authors propose an adaptive, image-dependent threshold selection scheme. Image processing has likewise been used to characterize bell pepper and chili pepper seeds in the work of Olaes *et al.* [21]. Multiple shape and size attributes are extracted after threshold-based segmentation of a grayscale image, which is then optimized via equalization to enhance element contrast. Overlaps are absent, and the background is a homogeneous black, enabling reliable segmentation.

Mohi-Alden *et al.* [22] describe an in-line pepper classification system operating under optimal laboratory conditions. Lighting is tightly controlled, and peppers travel on a conveyor belt singly, thus sidestepping occlusion and overlap difficulties. Peppers are first isolated with classical image processing, then a broad set of features is extracted and reduced through Principal Component Analysis (PCA) to retain the most informative ones. These features feed a MultiLayer Perceptron neural network that sorts the segmented peppers into five categories with fair success. Yet, segmentation itself poses little difficulty here, as the items sit on a white backdrop under uniform lighting.

Other lines of research confront pepper segmentation in greenhouses where conditions are not strictly managed. Illumination varies with weather, and leaves and neighboring peppers also cause occlusions and partial visibility. Different strategies are proposed to address this. For example, Ji *et al.* [23] propose a real-scenario green pepper segmentation method that couples feature extraction with classical image processing and machine learning applied to color images. The images are mapped to Lab color space, where an initial k-Nearest Neighbor (KNN) based segmentation is performed. Feature selection precedes the construction of an input vector that is supplied to a pre-trained supervised Support Vector Machine module. The authors report detection accuracy, though how this value is derived and the quality of the segmentation, as measured by standard metrics such as mAP (mean Average Precision), remain unclear.

More recently, Lu *et al.* [24] present a compelling study contrasting classical image processing with deep learning-driven techniques on a pepper segmentation task. The objective is to synchronize the segmentation with an intelligent robot for pepper picking in actual farm settings. Specifically, the authors curate a dataset of Sichuan peppers (Hanyuan variety) and validate it across various light intensities, object counts, and leaf-induced occlusions. They benchmark traditional segmentation methods (RGB and HSV color spaces, k-means clustering) against deep learning algorithms (U-Net convolutional network, Pyramid Scene Parsing Network, DeeplabV3+ convolutional network), evaluating performance using Intersection Over Union (IoU) and Mean Pixel Accuracy (MPA). The findings demonstrate that the U-Net algorithm markedly surpasses classical techniques, above all when peppers are occluded.

Most recent publications showcase deep learning-centric approaches. Liu *et al.* [25] leverage hyperspectral imaging to improve the automated picking of green peppers in greenhouses. Green peppers pose a tough challenge because their hue can blend with neighboring and overlapping foliage. The authors design a new, sophisticated convolutional neural network tailored to hyperspectral inputs and claim it outperforms segmentation based solely on RGB images within the visible spectrum. Building on the same rationale of exploiting additional information from hyperspectral data, Yu *et al.* [26] propose mounting an optical filter on a regular RGB camera, thereby avoiding the expense of specialized hardware. The filter is intended to provide discriminative features for green pepper detection based on its transmission profile. To optimize this transmission curve, the authors propose the Optical Filter Net, which stacks a depth-wise convolution layer representing the filter, further convolution layers for the RGB sensor channels, and a U-Net for semantic pepper segmentation. The transmission profile is refined during training to maximize segmentation quality. Outcomes are marginally better than those achievable with visible-range color images alone.

In a recent contribution, Cong *et al.* [27] introduce an instance segmentation algorithm that adapts a mask region-based convolutional neural network (Mask RCNN) with transformers and a customized loss function, targeting sweet pepper segmentation in greenhouses using high-resolution imagery. They obtain acceptable results on a single pepper variety

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

that expresses three colors (yellow, red, green), with occlusions and overlaps emerging as the principal performance-limiting factors.

The focus of our research is on developing a practical instance segmentation solution for reliable pepper phenotyping under changing conditions. While many prior works rely on classical image processing pipelines, we demonstrate their inadequacy in highly dynamic environments. Our deep learning-based proposal demonstrates greater robustness to a wide range of variability, including location shifts, background alterations, lighting changes, occlusions, dirt, and sensor noise.

Additionally, our solution operates exclusively on RGB imagery from ubiquitous devices such as smartphones and tablets. This simplicity encourages worldwide adoption in phenotyping routines, obviating the need for complex setups such as hyperspectral cameras or additional optical filters used in earlier studies.

Finally, we highlight the generalization capacity of our proposed solution across pepper varieties and other crops, demonstrating successful results across three distinct pepper types.

Dataset

Annotated data are a prerequisite for training deep learning solutions. Beyond mere availability, the data must faithfully represent the full spectrum of variability encountered in real production environments. In our work, we secured this essential diversity by drawing upon an existing collection of images amassed over multiple acquisition campaigns conducted between 2019 and 2021. These photographs were captured by various personnel at different BASF—Nunhems sites (in Spain and Mexico), using the very smartphones and tablets that breeding teams regularly carry for fieldwork. The device roster included models such as the Samsung Galaxy S7 Edge, Samsung Galaxy J7 Pro, and Samsung Galaxy Tab Active2. Image resolution is 12 MP (exact width and height fluctuate slightly depending on the device). This works out to a resolution per fruit spanning from roughly 20 kP for Jalapeño up to about 500 kP for Lamuyo. It should be underscored that capture distance is neither standardized across segments nor held constant within a segment, so a universal conversion between pixel units and physical measurements cannot be defined for the dataset as a whole.

A capture protocol was instituted to address acquisition variability and deliver a workable solution. Shots must be taken from a top-down viewpoint at a broadly comparable height (which varies by pepper type). The peppers themselves should be laid on their side on the ground with sufficient spacing to keep contact—and thus overlap—to a minimum. Known-size reference markers are placed in the frame during capture to enable subsequent size estimation. For Blocky Bell and Lamuyo, a blue rectangular card sized 10 × 7.5 cm^2 serves this purpose; for Jalapeño, yellow rectangular cards measuring 7 × 2.5 cm^2 set in the background act as the reference.

The final dataset assembled contains images of the three pepper types under investigation. Every image features instances of a single variety only; no image mixes multiple varieties. In total, the dataset holds 1407 Blocky Bell images, 150 Jalapeño images, and 216 Lamuyo images. **Figure 1** presents sample images from each of the three varieties.

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



Figure 1. Examples of images in the dataset. The first row shows images of Blocky Bell, the second row of Jalapeño, and the third row of Lamuyo.

It is worth pointing out that, while the protocol was designed to limit variability, eradicating it remains out of reach. Different technicians capture the photos across different seasons and in different countries. Even with the top-down angle and broadly consistent heights, variation creeps in. Peppers are arranged on the ground with care to avoid overlap, yet some end up oriented vertically or touching, especially in images with many fruits. On top of that, unavoidable variation is introduced by location, by the specific capture device used, and by fluctuating illumination tied to weather conditions and the time of day. **Figure 2** shows examples of peppers standing upright or overlapping.



Figure 2. Examples of cases with vertically placed and overlapping peppers.

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

As noted previously, deep learning-based models depend on ground truth labels for training. Ordinarily, these labels are generated by human annotators working manually. However, producing annotations for instance segmentation is notoriously laborious and slow. Hence, in our study, we explored ways to cut down the annotation workload.

On one front, we deliberately selected only a fraction of images per variety for manual annotation. Rather than labeling every image, we retained a representative subset. As argued earlier, the training corpus needs to encompass the full range of variability observed in real-world settings. Including highly redundant images in the training set provides no new information to the model and wastes effort. We therefore manually select a set of representative images for each variety, ensuring the chosen sample adequately reflects the dataset's full diversity. The selection accounts for factors such as different seasons, locations, illumination, and background conditions, as well as the number of instances per image. The physical appearance of the peppers is also weighed: the full color range, overlapping fruit, fruit orientation, blemishes, soil, leaves, and so on. After this curation, 180 Blocky Bell images, 35 Jalapeño images, and 216 Lamuyo images were singled out for manual annotation.

The quantity of images designated for manual labeling differs markedly from one variety to the next. This stems from the contrasting number of individual instances appearing in each image. Blocky Bell photos, for instance, typically contain anywhere from 1 to 90 fruits per frame. Lamuyo, by contrast, exhibits a much narrower range of just 1 to 20 fruits per image. Jalapeño pictures sit at the other extreme, spanning roughly 50 to nearly 350 instances per frame. Consequently, far fewer Jalapeño images were selected than for the other two varieties. The dense packing of instances in Jalapeño images makes annotation particularly demanding, with annotators spending considerably longer on each such frame than on other types. That said, a single Jalapeño image provides a wealth of training examples, which may allow effective instance segmentation to be learned from fewer images. For Lamuyo, the opposite logic applied: all available images were chosen for manual review, precisely because each image contains so few instances.

Table 1 records, per variety, how many images were selected for annotation alongside the total number of annotated instances. It can be seen that the 35 chosen Jalapeño images occur far more often than those of the other two types. Lamuyo, meanwhile, yields a much lower instance total despite having many more images tagged for annotation. **Figure 3** plots the distribution of instances per image for each variety. The bulk of Lamuyo images contains just 1 to 10 instances. In contrast, within the 35 Jalapeño images, the instance count per frame ranges widely, with many images holding large numbers.

Table 1. Number of annotated images and instances per variety.

Variety	Annotated instances	Annotated images
Blocky Bell	3791	180
Lamuyo	1016	216
Jalapeño	5880	35

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

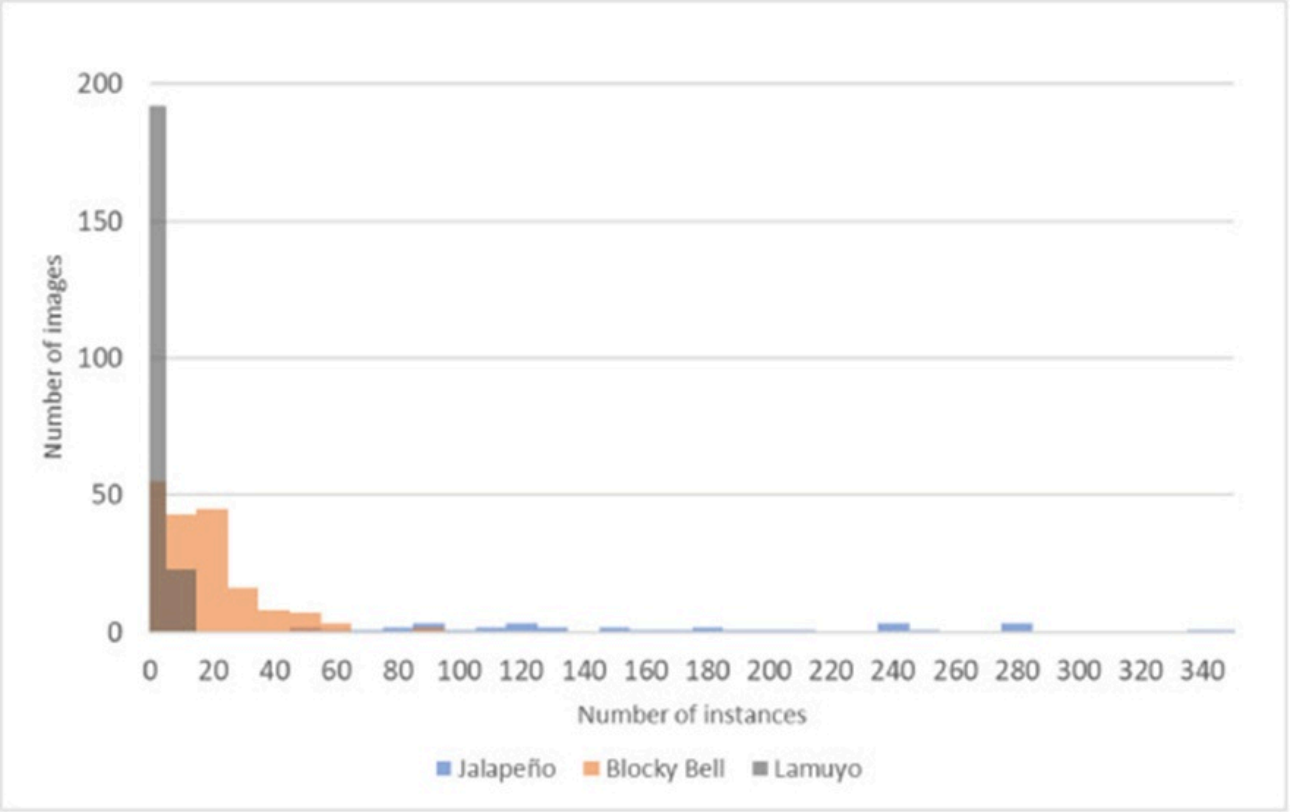

Figure 3. Distribution of the number of instances per image for the three varieties.

On another front, to accelerate annotation and reduce effort, we provide instance segmentation masks generated by the classical image processing algorithm as annotation proposals for Blocky Bell and Jalapeño. For Lamuyo, we instead use the output of the deep learning-based instance segmentation model previously trained on Blocky Bell data. Under this setup, the annotator's job is reduced to correcting masks that inaccurately trace pepper silhouettes, pruning false positives, and inserting masks for any peppers missed. Since the supplied proposals are already fairly accurate, this correction-based workflow places a much lighter burden on annotators than requiring them to draw every annotation from scratch. The overall effort required for annotation is thus substantially diminished.

A protocol was also established for manual annotation work. As shown in **Figure 4**, annotations must hug the pepper outline even when other elements—leaves, soil, stems, or neighboring fruit—partially obscure it. For Blocky Bell and Lamuyo, the stem is to be excluded from instance segmentation. For Jalapeño, the opposite holds: the stem is to be included within the instance. Finally, peppers lying on the image edges are deliberately left unannotated. These edge fruits are regarded as outside the region of interest, typically appearing only in part. We want the instance segmentation model to neither detect nor segment such borderline peppers; hence, they are excluded from the annotation set. The entire protocol is shaped by the requirements of the phenotyping analyses that follow the instance segmentation step.

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



Figure 4. Manual annotation protocol examples. The images shown are regions of bigger images.

Once the corrected annotations are retrieved, the dataset is prepared for training by constructing instance masks. In these masks, background pixels are set to 0, while pixels tied to individual instances carry values unique to those instances. For illustration, all pixels belonging to the first pepper receive the value 1, all those of the second pepper the value 2, and so on.

Figure 5 summarizes the sequence of steps used to obtain the final instance masks that constitute the ground truth for the deep learning-based instance segmentation model. The second column displays sample annotation proposals produced either via the classical image processing pipeline or via earlier versions of the deep learning model. At this juncture, the annotations remain noisy: some instances are incomplete, others are missing, and a few are false positives. The third column shows the images following the annotator's manual adjustments. The fourth column visualizes the corresponding definitive ground truth instance masks.



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

Figure 5. Steps for obtaining ground-truth instance masks for an image. From left to right, original image, annotations proposals to upload to CVAT, corrected annotations, and final instances masks.

Materials and Methods

In this study, a solution for deep learning-based instance segmentation is conceived, implemented, and tested. Instance segmentation involves the separate delineation of every occurrence of a class present within an image. The aim goes beyond merely segmenting all objects of a given class; it also entails distinguishing each object of that class. Here, only one class exists: “pepper.” The output from instance segmentation flows into a trait analysis pipeline that computes yield-related attributes at both the single-fruit level (e.g., size, shape) and the plot level (size distribution, yield estimate). This information is then fused with numerous other traits to support breeding decisions. Earlier, instance segmentation relied on classical image processing methods; in this work, we advance a deep learning-based alternative and demonstrate its clear advantage.

This section begins by describing the baseline method, rooted in a classical image algorithm, which previously handled pepper instance segmentation. Following that, the deep learning methodology is outlined, along with the training process used to produce the models. The section concludes with a description of the evaluation approach and the experiments conducted.

Baseline method

A pepper segmentation model built upon classical image analysis operations was already functioning within a trait analysis workflow. A schematic of the model architecture, accompanied by sample results, appears in **Figure 6**. The input is the original image, and the output consists of the segmentation masks. Parameter settings for each processing stage are tuned separately for different pepper segments.

1. Color thresholding: Two bands in the RGB color space are specified—one for red fruits and the other for yellow fruits. This operation produces an initial candidate mask covering all image pixels judged to belong to fruit.

2. Opening: thresholding routinely leaves behind a multitude of very small artifacts that can sit in proximity to one another (e.g., soil patches tinged reddish or yellowish). A morphological opening is therefore applied to suppress these.

3. Connection: all disconnected regions are split apart into independent instances for the subsequent filtering stages.

4. Size filtering: independent instances whose size falls below a predetermined threshold are likewise considered artifacts and removed.

5. Position filtering: regions touching the image borders are also treated as artifacts and removed via positional filtering (e.g., partially visible items inadvertently captured at the frame edge).

6. Closing: the regions that survive now form a strong candidate for all fruit pixels. However, the earlier opening may have eroded them; a closing operation is performed to restore what was lost partially.

7. Union: Though most true fruit pixels may have been identified by now, they are typically not partitioned into the correct number of fruit instances. This occurs mainly because many fruits rest very close together on the soil and appear fused in the image, making the connection operation insufficient for proper instance separation. A different strategy is required for adequate segmentation; hence, all regions are merged in this step.

8. Region growing: This operation yields a segmentation mask split that more faithfully reflects the individual fruits in the image. The feature guiding region growing is the 2-norm; that is, neighboring points are merged into the same region if

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

the Euclidean distance between their pixel values lies within a fixed threshold. The resulting regions constitute the final segmentation masks for the pepper fruits.

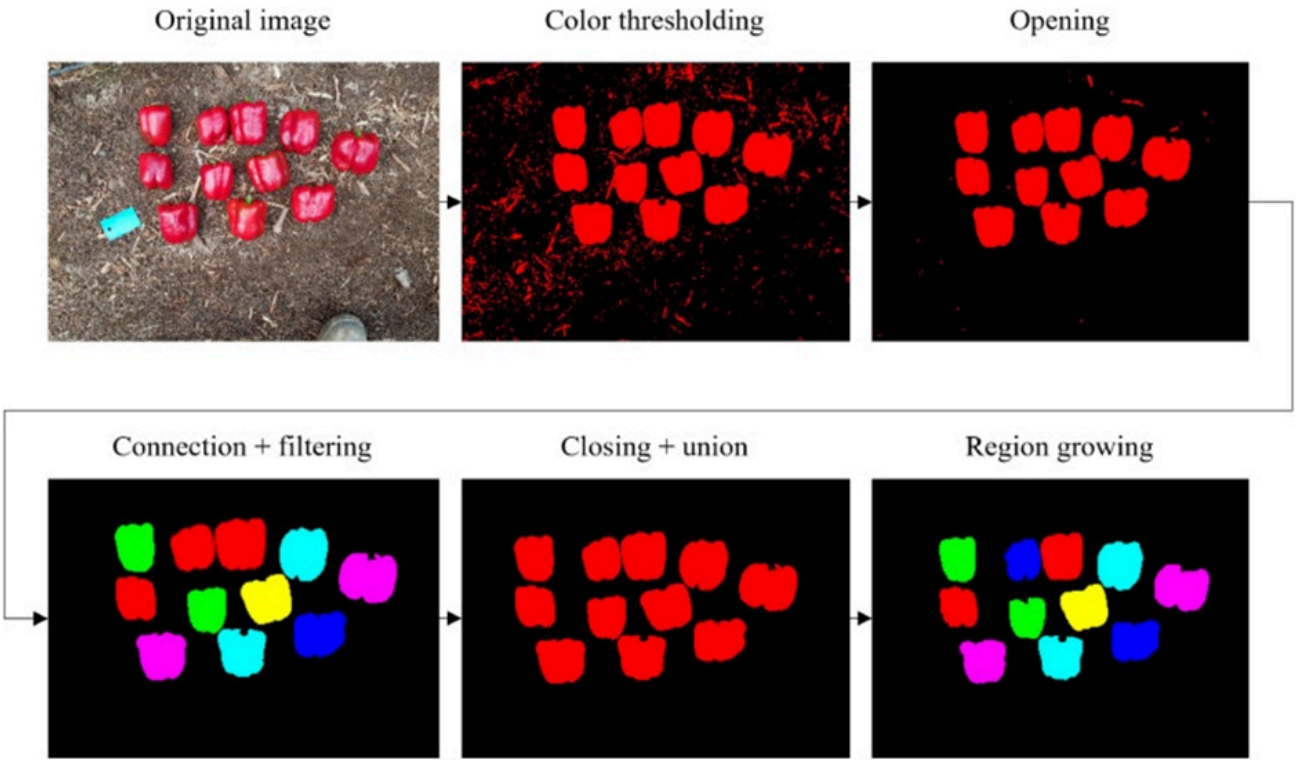

Figure 6. Classical image analysis pipeline example for a Blocky Bell image.

This segmentation model performs sufficiently well to sustain regular data collection. However, it exhibits several issues that require attention to achieve the targeted data quality and to allow seamless extension to new cases:

- Color thresholding is brittle in the face of small departures from ideal conditions, such as reflections, dust, water droplets, or shifts in scene illumination. All such departures are very frequent under practical capture conditions, difficult to control, and hard to counteract. Errors introduced at this initial stage propagate through the rest of the pipeline, exerting a substantial impact.

- The opening, closing, size filtering, and region growing steps are all scale-dependent, requiring the target elements to sit within a particular size bracket. Yet standardizing size is not practical, owing to the crop's biological variability and the realities of image capture.

- Model parameters have to be selected and tuned separately for each segment, because the scenes present different characteristics (e.g., fruit color and relative size, illumination conditions).

While it would certainly be possible to continue elaborating and refining this approach within the classical image analysis framework, the benefits of a deep learning architecture make it the natural choice for an improved segmentation model.

Deep learning-based method

The instance segmentation framework adopted in this study is Mask RCNN [28]. Our decision to use this architecture was informed by a review of the literature, which revealed that Mask R-CNN is the predominant choice for analogous applications reported in the state of the art. Fundamentally, Mask R-CNN represents an evolution of Faster R-CNN [29], a network originally designed for object detection. Because instance segmentation combines the goals of detection and

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

pixel-level segmentation, Mask R-CNN augments the Faster R-CNN architecture with an additional branch that predicts a segmentation mask for each detected object. The overall structure consists of a Region Proposal Network (RPN) that operates on features from a Convolutional Neural Network (CNN) to suggest multiple Regions of Interest (RoI). These RoIs then pass through an alignment step, a classification head, and confidence scoring to yield the final per-class instances. The segmentation branch subsequently produces the mask for every instance.

We built the Mask R-CNN model using the PyTorch library and adopted PyTorch Lightning as a higher-level wrapper to keep the implementation clean and well-structured. More precisely, the model specification was sourced from the Mask R-CNN offering in PyTorch's torchvision library. ResNet50 serves as the backbone feature extractor.

Although most Mask RCNN hyperparameters were left at the default settings provided by the chosen implementation, a select few were modified in particular experiments to better align with the demands of our use case:

• Maximum detections per image across all classes: The factory default caps this at 100. Yet, within our project dataset, there are Jalapeño images with instance counts exceeding 100. To avoid truncation, we lifted the ceiling to 512 for every experiment.

• The hyperparameters dictate how many proposals the Region Proposal Network emits. The standard values stand at 2000 for the training phase and 1000 for testing. To ensure robust detection in images densely populated with instances, we also increased these figures.

For training, the dataset is first split into train, validation, and test subsets, with the split performed per-variety, since an independent model is trained for each variety. This per-variety strategy allows us to fine-tune hyperparameters individually, reflecting the fact that instance counts per image vary substantially across varieties. Running and maintaining separate models for each variety is also more practical, especially since phenotyping operations do not require a single monolithic model spanning all varieties—they are handled separately, as the images never feature mixed varieties. An additional advantage of keeping models distinct is their mutual independence. This guarantees that any future upgrades or new models developed for additional varieties or crops will have no bearing on the instance segmentation models already functioning satisfactorily.

The model learns from pairs of input-output examples from the training and validation splits. Training images drive network parameter optimization, while validation images periodically assess how the model generalizes to data it has not seen. This setup enables monitoring for overfitting and allows selection of the weights from the best epoch, defined as the epoch at which the validation loss attains its minimum value.

As input, the model receives an RGB image drawn from the dataset. The corresponding output is a Python dictionary holding the lists of masks, bounding boxes, and labels. Hence, for each detected instance, the model outputs a single mask, bounding box, and label. This output structure must be generated on the fly during training from the instance masks prepared earlier in the dataset preparation phase. In the context of this project, the label entry in the dictionary is invariably set to "pepper" for every instance. Both inputs and outputs are resized to a uniform resolution of 1024 × 1024 pixels. This size was chosen as a balance: it is compact enough to permit tractable training given the available time and GPU memory, yet sufficiently large to retain the critical information about the peppers. At this resolution, the fruits remained sufficiently large for the model to detect and segment them successfully across all three varieties. Consequently, no image tiling strategy was required.

Before entering the model, the training data may undergo a series of augmentation operations. The transformations employed include flipping, rotation, transposition, scaling, padding, and color perturbation. All augmentations except those affecting color must be applied to both the input image and the output masks simultaneously, because geometric alterations to the input necessitate equivalent alterations to the output. Color perturbations, however, modify only the

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

brightness, contrast, saturation, and hue of the image, leaving the output masks unchanged. Augmentation is introduced to enrich the training distribution and improve the generalization of the trained model.

Beyond augmentation, an additional enhancement that can be leveraged during training is initializing the network with weights transferred from a model pretrained on the COCO dataset [30]. This practice means training does not start from randomly initialized parameters but rather from a set of weights already partway toward an effective solution.

Throughout training, the network employs a Stochastic Gradient Descent (SGD) optimizer configured with a learning rate of 0.001. The batch size is fixed at 4. Experimentation with smaller batch sizes did not yield any performance gains. Larger batch sizes are ruled out by the GPU's memory limit (12 GB on the NVIDIA Titan X Pascal). Every experiment runs for 1000 epochs. However, the final model weights are taken from the epoch with the minimum validation loss, not necessarily the last one. It should be noted that, while there is conceptually only one foreground class ("pepper"), the model's class count is set to 2 because the network internally treats "pepper" and "background" as distinct classes.

Experiments

For the experimental work, the available image dataset for each variety is split into training, validation, and test sets at 70%/15%/15%. **Table 2** presents the resulting image counts per variety within each subset after partitioning.

Table 2. Train, validation, and test split of the dataset per variety. Each column shows the number of images in each image set per variety.

Empty Cell	**Lamuyo**	**Jalapeño**	**Blocky Bell**
Train (70 %)	150	24	125
Validation (15 %)	33	5	27
Test (15 %)	33	6	28
TOTAL	216	35	180

A Mask RCNN model is trained separately for each variety and subsequently tested on that variety's designated test set, yielding three distinct models in total. Augmentation, as detailed above, is active throughout training, and every model is initialized from the COCO-pretrained weights [30]. For the Jalapeño variety specifically, the hyperparameter that determines the volume of proposals emitted by the Region Proposal Network is set to 4000 for training and 2000 for testing. The reasoning is straightforward: Jalapeño images contain a markedly larger number of instances compared to the other two varieties, so the proposal pool must be correspondingly enlarged to ensure all peppers are detected. The results on the respective test sets are reported for all three trained models (one per variety).

To measure the performance gap between the proposed deep learning instance segmentation model and the legacy solution based on classical image processing techniques, we also compute and report results for the earlier solution on the same test sets across all three varieties. This involves running the corresponding classical image-processing algorithm on each test image to produce estimated segmentation masks. Since the test images have ground-truth instance masks, performance figures are derived using the same evaluation protocol as for the deep learning approach.

Beyond this head-to-head comparison, we further analyze the individual contributions of data augmentation and pretrained weight initialization for the Blocky Bell variety. Likewise, we study the effect on Jalapeño results of increasing the hyperparameters controlling the RPN proposal count.

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

The assessment of instance segmentation performance proceeds along two parallel tracks. Quantitatively, we compute numerical metrics by comparing the model's test-set predictions with the ground-truth annotations for the same images. Qualitatively, we conduct a visual inspection by generating output images wherein the predicted instance masks are rendered as colored overlays on the test images.

The principal quantitative metric adopted is Mean Average Precision (mAP) [31]. This metric is standard in both object detection and instance segmentation benchmarks. It captures how faithfully instances are both located and segmented, with an ideal score of 1. A predicted instance qualifies as a true positive if its overlap with a ground truth instance surpasses a specified Intersection over Union (IoU) threshold. The mAP can be computed at a single IoU threshold or averaged across a range of thresholds. In the instance segmentation literature, the widely reported variant is mAP@0.5:0.95, which averages the metric over IoU thresholds from 0.5 to 0.95 inclusive, with increments of 0.05.

Complementing the numerical metrics, we place strong emphasis on visual inspection. For each test image, we render an output image with the predicted instance masks overlaid on the original photograph, assigning each mask a unique color for unambiguous identification. This visualization allows us to directly observe how well the model is performing—whether the estimated masks conform to the requirements stipulated for each variety and what categories of mistakes the model tends to make. Such diagnostic insights are indispensable for guiding future model improvements and can only be gleaned by examining a curated set of examples for each variety.

Results and Discussion

In this portion of the paper, we detail the findings from our investigation. We start by benchmarking the deep learning instance segmentation model against the classical image processing method for each pepper variety. Next, we assess the individual value added by data augmentation and pretrained weight initialization using the Blocky Bell dataset; we regard the conclusions drawn from this dataset as transferable to the other two. Lastly, we investigate how modifying the RPN proposal count hyperparameter influences outcomes in the specific case of Jalapeño.

Deep learning vs classical image processing

Table 3 compiles the mAP@0.5:0.95 scores achieved by the older classical image processing pipeline and the newer deep learning pipeline for all three pepper types.

Table 3. Comparison of the obtained mAP@0.5:0.95 for the three pepper varieties between the classical image processing solution and the deep learning solution.

Empty Cell	**Lamuyo**	**Jalapeño**	**Blocky Bell**
Classical image processing	0.67	0.39	0.63
Mask RCNN	**0.97**	**0.52**	**0.97**

The Mask RCNN-based deep learning method outperforms the classical approach by a wide margin in every case. For Blocky Bell and Lamuyo, the metric values sit very close to 1, indicating nearly flawless segmentation. Jalapeño presents a more complex picture: the deep learning approach still handily beats the classical one, yet its absolute score lags behind the other two varieties. The reason lies in the smaller physical dimensions of Jalapeño fruits, which translate to fewer pixels per instance. While segmentation fidelity may be comparable in qualitative terms, a single mislabeled pixel exerts a much larger relative penalty on a compact area (for instance, 100 pixels) than on an expansive one (for instance, 1000 pixels). Since the evaluation metrics are computed at the pixel level, this discrepancy is amplified. Intersection over Union (IoU) is particularly susceptible to this effect and penalizes smaller objects, thereby also depressing the mean average precision (mAP).

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

Figure 7 presents a side-by-side visual comparison of segmentation outputs from the classical and deep learning pipelines applied to identical images. The classical technique yields acceptable results for sound, blemish-free peppers resting flat against the ground. Yet it struggles considerably when peppers display damage, atypical coloration, surface dirt, or suboptimal positioning. In such challenging scenarios, the deep learning model proves far more capable, owing to its superior ability to generalize across fluctuating conditions. Furthermore, the deep learning approach is markedly better at teasing apart instances that are in physical contact. This is clearly seen in the rightmost Blocky Bell sample and the leftmost Lamuyo sample in the figure: the classical algorithm fails to split closely abutting fruits, whereas deep learning draws a clean boundary between them.



Figure 7. Examples of instance segmentation outputs. Comparison of the output of the same image with the classical image processing approach (left) and the deep learning approach (right). From first row to last row: Blocky Bell, Jalapeño, and Lamuyo. Only a region of the complete image is shown for better visualization.

Turning to inference throughput, timing experiments on an Intel i5-6300U CPU yield an average of 3.65 s per image for the classical approach and 12.5 s per image for the deep learning approach. While the classical pipeline is undeniably faster on equivalent CPU hardware, it is essential to bear in mind that phenotyping operations of this nature are not executed under real-time requirements. A per-image runtime of 12.5 s for the deep learning method is therefore entirely practical. Moreover, the architecture is highly amenable to GPU acceleration: a single GPU brings inference time down to 0.5 s per image.

Contribution of the augmentation and weight initialization

As outlined earlier, the training procedure incorporates recognized best practices from the deep learning development methodology [32]. First, a range of augmentation operations—flip, rotation, transpose, scaling, padding, and color perturbation—is employed. The purpose of these transformations is to cultivate a model with stronger generalization. The final instance segmentation model must deliver robust performance under diverse and unpredictable circumstances: varying backdrops, pepper hues, degrees of surface damage and soiling, uneven lighting, and so on. Introducing random appearance modifications to the training images inflates the dataset and encourages the model to latch onto the most salient and invariant features rather than spurious environmental correlates. Second, instead of starting the optimization

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

from randomly initialized parameters, the Mask RCNN is bootstrapped with weights transferred from a version pretrained on the COCO dataset, positioning the initial point nearer to a desirable minimum.

Table 4 contrasts the results for the Blocky Bell model trained without augmentation and COCO initialization, with each improvement applied in isolation, and with both combined. As one would expect, each enhancement, on its own, has a beneficial impact on the final model's performance. When both are engaged simultaneously, the instance segmentation model achieves its best performance.

Table 4. Value of the mAP@0.5:0.95 for the Blocky Bell model trained with and without the data augmentation and the weight initialization.

Empty Cell	**Blocky Bell**
Mask RCNN	0.92
Mask RCNN + Aug	0.95
Mask RCNN + COCO weights	0.94
Mask RCNN + Aug + COCO weights	**0.97**

Contribution of the RPN number of proposals hyperparameter

For Jalapeño, the typical per-image instance count far outstrips that of the other varieties. To help Mask R-CNN cope with this density and achieve better results, the hyperparameters that determine the number of proposals emitted by the Region Proposal Network are set to 4000 during training and 2000 during testing, departing from the standard defaults of 2000 and 1000, respectively. As Table 5 shows, increasing this hyperparameter provides a clear and positive boost to the Jalapeño instance segmentation model. Inspecting the test images individually confirms that the elevated proposal count directly reduces the frequency of undetected Jalapeño instances.

Table 5. Value of the mAP@0.5:0.95 and R^2 for Jalapeño with different values of the hyperparameter to control the number of RPN proposals.

Empty Cell	**mAP@0.5:0.95**	R^2
Mask RCNN − RPN 2000/1000	0.50	0.96
Mask RCNN − RPN 4000/2000	**0.52**	**0.99**

In fact, a major pain point for both classical and deep learning solutions when applied to Jalapeño peppers has been the outright omission of some instances during segmentation. By dialing up this hyperparameter, we can virtually guarantee that the deep learning solution misses almost no peppers.

Because this issue is especially pronounced for Jalapeño, we probe its influence on this variety's model more deeply. To measure it, we compute the R^2 metric for the Jalapeño model. R^2 is typically a regression metric, but we report it here because mAP alone does not fully convey detection completeness. While mAP jointly captures detection and segmentation quality, R^2 indicates how closely the model's detected pepper count matches the true number of peppers in the images. An ideal score is 1, and the best deep learning model for Jalapeño achieves an R^2 of 0.99, with the caveat that per-image Jalapeño counts range from 85 to 283.

The findings of this study clearly attest to the edge that deep learning holds over classical image processing whenever illumination and the visual properties of the target objects are subject to change. Deep learning algorithms are inherently structured to learn from diverse and intricate data distributions, equipping them to recognize and interpret fine-grained

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

variations and irregular patterns. Classical image processing methods, in contrast, tend to falter when forced to operate beyond the narrowly defined conditions and built-in assumptions for which their rules were handcrafted.

A key advantage of deep learning, evident from our results, is that a single methodological framework suffices across all three varieties. Classical image processing requires significant manual intervention, as it relies on bespoke features and thresholds that must be individually calibrated for each pepper type. This tuning loop is slow and labor-intensive, potentially requiring the trialing and adjustment of scores of parameter combinations for every target domain. For pepper instance segmentation, such fine-tuning was carried out separately for each variety, with additional effort invested to ensure the pipeline held up across different sites, lighting regimes, and background contexts. Any unanticipated shift in lighting, object appearance, or another variable not accounted for during manual feature engineering can trigger erroneous segmentation. Deep learning, by contrast, finds that the same architecture and training protocol prove effective across all three varieties in all the domains encountered, with only minimal hyperparameter tweaking needed. The breadth of the training data provided primarily bounds the reach of a deep learning solution.

Beyond this methodological uniformity, the deep learning solution markedly outstrips its classical counterpart in raw performance. Metric values for the deep learning approach are decisively higher across all varieties. The fundamental reason is that deep learning is far less affected by noise, illumination drift, color shifts, or background changes. It also navigates complex spatial arrangements—for instance, touching or overlapping objects—much more competently.

Furthermore, extending a deep learning solution to new domains is considerably easier. Widening the model's scope to accommodate an additional domain requires only including data from that domain during training, without the manual rule revisions that classical methods demand. The annotation of such new data can be expedited and simplified by the annotation proposal workflow, which builds directly on prior modeling investments.

Cong *et al.* [27] is the most directly comparable method we could identify. Regrettably, the authors have not made their code publicly available, and replicating their approach would be a substantial undertaking given the complexity of the proposed Swin transformer-based network. Their dataset, unlike ours, comprises a single pepper class in three colors. The "sweet peppers" they describe resemble our Blocky Bell and Lamuyo types, for both of which we report an mAP@0.5:0.95 of 0.97. Their reported average precision for sweet pepper segmentation (**Table 3** in that paper) is 0.84, with yellow pepper achieving the best per-class mAP of 0.89 (**Table 4** in that paper). The peppers in their field setting are subject to heavier occlusion than our ground-arranged peppers, rendering their problem somewhat more difficult. It is therefore conceivable that the Cong *et al.* [27] method could yield metrics approaching, or even marginally exceeding, our 0.97, though eclipsing such an already elevated figure would be challenging.

To sum up, deep learning has demonstrated greater robustness across variable conditions than classical image processing and, by extension, is far better suited to the diverse, rapidly evolving environments that characterize plant breeding. This robustness will enable vegetable breeding teams to collect larger datasets more cost-effectively, thereby supporting the development of superior varieties that meet the growing expectations of producers and consumers alike. In the bigger picture, more resilient phenotyping workflows anchored in deep learning models hold the promise of easing the "phenotyping bottleneck" and, ultimately, aiding efforts to nourish a global population projected to grow over the coming decades.

Conclusion

In the present work, a deep learning-driven instance segmentation solution has been implemented and validated across three pepper varieties. A dedicated model was trained for each variety, which delivered marginally better results than a single unified model covering all three. The models developed here are capable of correctly segmenting individual pepper instances under a broad array of conditions—encompassing variations in lighting, soil hue, fruit damage and soiling, camera distance, pepper density, sensor noise, and beyond. It is unequivocally shown that instance segmentation

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

powered by deep learning dramatically surpasses a classical image-processing-based alternative, with deep learning proving especially adept under changing conditions.

Although validation was carried out on three specific pepper varieties, the method can be straightforwardly extended to any other pepper type—or indeed to different fruits entirely—provided the input imagery adheres to the capture protocol set out here (zenithal viewpoint, fruits arrayed on the ground with minimal overlap). It is worth noting that every step in the workflow presented (data curation, annotation, and model training) has been automated through integration with BASF's proprietary model development automation platform. This MLOps and DevOps framework accelerates the conception, testing, and eventual rollout of machine learning and deep learning models.

The models produced by this work are encapsulated in a Python library. They are now in operational use as the initial processing stage in phenotyping pipelines for the three pepper varieties, enabling the estimation of size, shape, weight, and related traits. These Python packages are BASF property and are currently deployed for pepper phenotyping in both field and greenhouse settings operated by BASF in Spain and Mexico.

Future research directions may include developing semi-supervised methods that leverage annotation proposals generated by prior models to inject potentially noisy labels into the training process, thereby bolstering detection metrics while leveraging unannotated images. Such an approach could compress the time and labor needed to train new models on larger image collections while simultaneously lifting performance. It could also make the rapid, efficient onboarding of additional fruit types entirely viable.

Acknowledgements

We want to thank the breeders who, at the time of the data collection, were leading the pepper programs at BASF: Ignacio Susín, Sandra Martínez, and Baptiste Prenveille. They and their teams captured all the images that compose the datasets used in this publication.

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 28 Jan 2023 Revised: 04 Apr 2023 Accepted: 20 Apr 2023
Published online: 25 May 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. Kamilaris A, Prenafeta-Boldú FX. Deep learning in agriculture:A survey. Comput Electron Agric. 2018;147:70–90.

2. Tian H, Wang T, Liu Y, Qiao X, Li Y. Computer vision technology in agricultural automation—A review. Inf Process Agric. 2020;7(1):1–19.

3. Egusquiza I, Picon A, Irusta U, Bereciartua-Perez A, Eggers T, Klukas C, et al. Analysis of few-shot techniques for fungal plant disease classification and evaluation of clustering capabilities over real datasets. Front Plant Sci. 2022;13:813237.

4. Sai Reddy B, Neeraja S. Plant leaf disease classification and damage detection system using deep learning models. Multimed Tools Appl. 2022;81(17):24021–40.

5. Bereciartua-Pérez A, Gómez L, Picón A, Navarra-Mestre R, Klukas C, Eggers T. Insect counting through deep learning-based density maps estimation. Comput Electron Agric. 2022;197:106933.

6. Teng Y, Wang R, Du J, Huang Z, Zhou Q, Jiao L. TD-Det: A tiny size dense aphid detection network under in-field environment. Insects. 2022;13(6):1–14.

7. Gómez-Zamanillo L, Bereciartua-Pérez A, Picón A, Parra L, Oldenbuerger M, Navarra-Mestre R, et al. Damage assessment of soybean and redroot amaranth plants in greenhouse through biomass estimation and deep learning-based symptom classification. Smart Agric Technol. 2023;5:100243.

8. Azlah MAF, Chua LS, Rahmad FR, Abdullah FI, Wan Alwi SR. Review on techniques for plant leaf classification and recognition. Computers. 2019;8(4):77.

9. Argüeso D, Picon A, Irusta U, Medela A, Bereciartua-Pérez A, Fernández S, et al. Few-shot learning approach for plant disease classification using images taken in the field. Comput Electron Agric. 2020;175:105542. https://doi.org/10.1016/j.compag.2020.105542.

10. Singh AK, Ganapathysubramanian B, Sarkar S, Singh A. Deep learning for plant stress phenotyping: trends and future perspectives. Trends Plant Sci. 2018;23(10):883–98.

11. Patrício DI, Rieder R. Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review. Comput Electron Agric. 2018;153:69–81.

12. Zhang Q, Liu Y, Gong C, Chen Y, Yu H. Applications of deep learning for dense scenes analysis in agriculture: a review. Sensors. 2020;20(5):1–33.

13. Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: A fruit detection system using deep neural networks. Sensors. 2016;16(8):1222.

14. Tunio MH, Jianping L, Butt MHF, Memon I, Magsi Y. Fruit detection and segmentation using customized deep learning techniques. In: 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). 2022.

15. Bargoti S, Underwood J. Deep fruit detection in orchards. Proc IEEE Int Conf Robot Autom. 2017:3626–33.

16. San M, San M, Aung MM, Khaing PP. Fruit recognition using color and morphological features fusion. Int J Image Graph Signal Process. 2019;11(10):8–15.

17. Maheswari P, Raja P, Apolo-Apolo OE, Pérez-Ruiz M. Intelligent fruit yield estimation for orchards using deep learning based semantic segmentation techniques—A review. Front Plant Sci. 2021;12:1–18.

18. Gao Y, Li X, Qi K, Chen H. Measurement of chili pepper plants size based on mathematical morphology. IFIP Adv Inf Commun Technol. 2011;345(AICT Part 2):61–70.

19. Rajalakshmi M, Subashini P. Texture based image segmentation of chili pepper X-ray images using Gabor filter. Int J Adv Stud Comput Sci Eng (IJASCSE). 2014;3(3):44–51.

20. Ostovar A, Ringdahl O, Hellström T. Adaptive image thresholding of yellow peppers for a harvesting robot. Robotics. 2018;7(1):11.

21. Olaes EJ, Arboleda ER, Dioses JL, Dellosa RM. Bell pepper and chili pepper classification: an application of image processing and fuzzy logic. Int J Sci Technol Res. 2020;9(2):4832–9.

22. Mohi-Alden K, Omid M, Firouz MS, Nasiri A. A machine vision-intelligent modelling based technique for in-line bell pepper sorting. Inf Process Agric. 2023;10(4):491–503.

23. Ji W, Chen G, Xu B, Meng X, Zhao D. Recognition method of green pepper in greenhouse based on least-squares support vector machine optimized by the improved particle swarm optimization. IEEE Access. 2019;7:119742–54.

24. Lu J, Xiang J, Liu T, Gao Z, Liao M. Sichuan pepper recognition in complex environments: a comparison study of traditional segmentation versus deep learning methods. Agriculture. 2022;12(10):1–17.

25. Liu X, Yu J, Kurihara T, Xu L, Niu Z, Zhan S. Hyperspectral imaging for green pepper segmentation using a complex-valued neural network. Optik. 2022;265:169527.

26. Yu J, Kurihara T, Zhan S. Optical filter net: a spectral-aware RGB camera framework for effective green pepper segmentation. IEEE Access. 2021;9:90142–52.

27. Cong P, Li S, Zhou J, Lv K, Feng H. Research on instance segmentation algorithm of greenhouse sweet pepper detection based on improved Mask R-CNN. Agronomy. 2023;13(1):196.

28. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2017:2961–9.

29. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst. 2015;28:1137–49.

30. Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, et al. Microsoft COCO: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, Proceedings, Part V 13. 2014:740–55.

31. Padilla R, Netto SL, Da Silva EAB. A survey on performance metrics for object-detection algorithms. In: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). 2020:237–42.

32. Chollet F. Deep Learning with Python. 2nd ed. Simon and Schuster; 2021.

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