Tomato Maturity Analysis: A Comparative Study of Detection and Instance Segmentation Using YOLOv8

Salma Ait Oussous, Rachid El Bouayadi, Driss Zejli, Aouatif Amine
Advanced Systems Engineering Laboratory (ISA)
National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco

***Abstract*—The accurate visual analysis of fruit maturity in complex agricultural scenes remains a fundamental challenge due to gradual appearance changes, object overlap, and partial occlusion. This study addresses tomato maturity analysis, formally defined as instance-level binary classification and spatial localization under varying degrees of visual density. While bounding-box-based object detection is widely used, it often lacks precision in dense clusters. We present a controlled experimental comparison between object detection and instance segmentation using a common YOLOv8-medium (YOLOv8m) backbone to isolate the effect of spatial representation. Experimental results demonstrate that instance segmentation achieves superior localization accuracy and boundary consistency, reaching a mask-based mAP@0.5:0.95 of 0.817. These findings suggest that pixel-level supervision effectively reduces localization ambiguity, providing a robust foundation for automated agricultural monitoring.**

***Keywords*—Tomato maturity detection; computer vision; object detection; instance segmentation; image analysis; Deep Learning**

I. Introduction

Understanding and interpreting visual information in complex scenes remains a fundamental challenge in computer vision and image analysis [1]. Many real-world visual environments are characterized by gradual appearance variations, dense object arrangements, and frequent occlusions, which complicate reliable object interpretation. These challenges are particularly pronounced in tasks that require fine-grained analysis of object properties rather than simple object presence or absence [2].

Fruit maturity analysis represents a representative visual computing problem within this context. In agricultural images, traditional image processing approaches for fruit analysis have relied on manual features and threshold-based decision rules derived from color or texture descriptors [3]. While effective in controlled settings, such approaches are sensitive to illumination changes and background complexity and often fail to generalize to visually diverse scenes [4]. Unlike problems involving distinct object categories, maturity assessment requires distinguishing between visually similar instances that differ only by small, continuous appearance variations. This makes maturity detection a demanding image analysis task, especially in unconstrained imaging conditions [5].

Formally, we define the tomato maturity analysis task as a fine-grained visual computing problem consisting of instance-level binary classification (ripe vs. unripe) combined with spatial localization under varying degrees of visual complexity. The task is constrained by assumptions of greenhouse-like environments where objects exhibit significant overlap and partial occlusion. The primary deployment scenario is the integration into automated agricultural monitoring or robotic harvesting systems, where precision in both maturity stage identification and spatial delineation is critical for operational success. In this study, our key contribution is a controlled experimental comparison between object detection and instance segmentation using the same YOLOv8-medium (YOLOv8m) backbone, dataset, and training protocol. This design enables us to isolate the effect of spatial representation (bounding-box vs. pixel-level) on visual analysis performance.

The introduction of Deep Learning (DL), particularly Convolutional Neural Networks (CNN), has significantly improved robustness and accuracy in visual recognition tasks, specifically in fruit detection and maturity classification performance [6]. Numerous studies have applied object detection frameworks to identify fruits and estimate their maturity stages by assigning class labels to detected instances [7] [8]. Single-stage detectors, including YOLO-based architectures, have been widely adopted due to their computational efficiency and strong performance in real-time visual analysis tasks [9]. In agricultural image analysis, YOLO-based models have been successfully used for tomato detection and ripeness estimation, demonstrating their effectiveness in localizing fruits under greenhouse and field conditions [10]. For example, YOLOv3 was used for tomato classification in greenhouse environments with promising accuracy, although model size posed challenges for deployment [11]. YOLOv5-based approaches have been employed to detect overlapping tomatoes and small fruits, often incorporating attention mechanisms to improve localization performance [12] [13]. Improvements to YOLOv7 have also been proposed to address challenges related to small object detection and fruit overlap in dense scenes [14]. Despite their success, object detection methods typically rely on bounding-box representations, which provide only an approximate description of object extent. In visually dense scenes, bounding boxes may include background regions or overlap with neighboring objects, leading to ambiguity in object boundaries [7]. This limitation becomes particularly evident in agricultural imagery, where fruits often grow in clusters and partially occlude one another, reducing the reliability of fine-grained visual interpretation.

Instance segmentation has emerged as an alternative object representation that assigns pixel-level labels to individual object instances [15]. By explicitly modeling object boundaries and surface extent, instance segmentation enables clearer separation of adjacent objects and more accurate delineation of object shape [16]. In visual computing research, segmentation-based approaches such as Mask R-CNN and DeepLab have demonstrated improved performance in scenarios involving

occlusion and complex object interactions, for example Mask R-CNN has been applied to tomato segmentation, demonstrating improved boundary localization compared to bounding-box-based detectors, particularly in congested scenes [17]. Semantic segmentation architectures such as DeepLabv3+ have also shown strong performance in handling irregular object shapes and occlusions in fruit recognition tasks [18]. Hybrid approaches combining region proposal mechanisms with segmentation networks have further improved separation of closely clustered fruits [19]. In addition, segmentation-based models have been successfully integrated into robotic harvesting systems, where precise spatial perception is critical for manipulation and interaction [20].

However, most existing studies focus on improving the performance of a specific detection or segmentation model rather than systematically examining the influence of object representation choice [21]. Comparisons between detection-based and segmentation-based approaches are often conducted under different experimental settings, datasets, or network configurations, making it difficult to isolate the effect of representation on visual analysis performance. As a result, the relative impact of bounding-box-based versus pixel-level representations for fruit maturity analysis in dense and occluded scenes remains insufficiently explored.

To address this gap, the present study conducts a systematic comparison of object detection and instance segmentation approaches within a common DL model. By training and evaluating both representations under identical conditions, the analysis focuses on how object representation affects localization accuracy, boundary definition, and robustness in visually complex scenes. Through quantitative evaluation and qualitative analysis, this study provides insight into the role of bounding-box-based and pixel-level representations in fine-grained image analysis and contributes to a clearer understanding of representation choices for maturity detection tasks in visual computing.

The remainder of this study is organized as follows: Section II describes the database and methodology, Section III presents the experimental results and discussion, Section IV concludes the study and outlines directions for future research.

II. Methodology

This section describes the methodology adopted to evaluate object detection and instance segmentation approaches for tomato maturity analysis. The objective is to conduct a controlled comparison of two visual representation strategies bounding-box-based detection and pixel-level instance segmentation under identical experimental conditions. The methodology includes the database description, annotation process, model configuration, training protocol, and evaluation metrics.

A. Database Description and Annotation

The experiments were conducted using a tomato image database obtained from Kaggle, consisting of images captured under greenhouse-like conditions. The database includes 177 images representing two tomato maturity stages: ripe and unripe. The images exhibit a variety of visual conditions, including changes in illumination, background complexity, fruit density, and varying levels of occlusion. In total, the dataset contains 209 annotated tomato instances (124 ripe and 85 unripe) across the validation set, with an average of approximately 5.8 instances per image. This density provides a representative basis for evaluating localization performance in both sparse and clustered scenarios. The dataset was split into training, validation, and testing subsets using an 80:10:10 ratio, ensuring that diverse imaging conditions were represented across all partitions.

Although the original images were sourced from Kaggle, all annotations used in this study were created manually by the authors to support object detection and instance segmentation tasks. Pixel-level annotations were generated using the VGG Image Annotator (VIA) tool. Each tomato instance was carefully labeled to ensure accurate spatial localization and shape representation, which is essential for reliable evaluation of both bounding-box-based detectors and segmentation models.

Pixel-level annotations were created to precisely delineate individual tomato instances. From these annotations, bounding boxes were derived to enable object detection experiments, while the original pixel-level masks were used for instance segmentation analysis. Due to variations in fruit size, maturity stage, viewpoint, and frequent occlusions, the annotation process required substantial manual effort to maintain consistency and accuracy across the database.

The annotated database provides reliable ground truth labels for supervised learning and quantitative performance evaluation. The database was split into training, validation, and testing subsets to ensure unbiased model assessment. Representative examples of original images and their corresponding manual annotations, which reflect the actual labels used during model training and evaluation, are shown in Fig. 1.

B. Object Representation Models and Training Configuration

To enable a fair comparison between object detection and instance segmentation for tomato maturity analysis, both approaches were implemented within a DL model based on the YOLOv8m architecture. Using a unified framework ensures that differences in performance can be attributed primarily to the choice of object representation rather than to variations in backbone design, feature extraction, or optimization strategy.

In the object detection configuration, the network predicts a set of bounding boxes associated with class labels corresponding to ripe and unripe tomato instances. Each prediction represents an object hypothesis defined by its bounding box coordinates, confidence score, and class probability. This representation provides a compact description of object location but does not explicitly model object boundaries or surface extent.

In the instance segmentation configuration, the detection formulation is extended by incorporating a segmentation head that generates a pixel-level mask for each detected instance. In this case, the network outputs both bounding boxes and corresponding instance masks, enabling explicit delineation of object boundaries and improved separation of adjacent objects. The masks are derived from high-resolution feature maps to preserve fine object details.

Both configurations share the same backbone network, feature extraction layers, and multi-scale feature aggregation com-

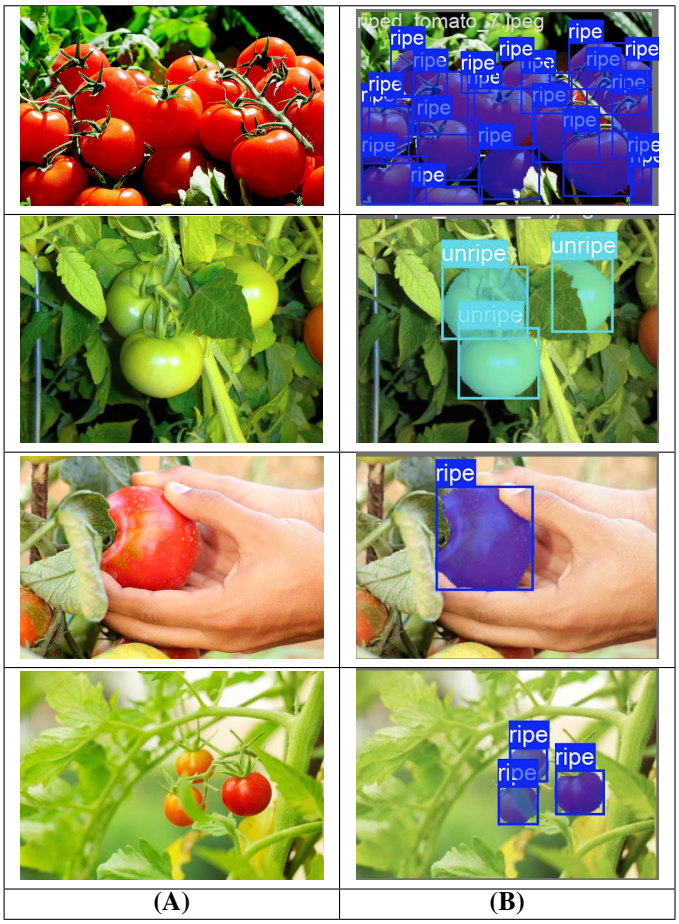

Fig. 1. Representative examples of original images and their corresponding manual annotations from the tomato maturity database. The left column (A) shows original images, while the right column (B) presents annotated images with bounding boxes, indicating individual tomato instances classified as ripe or unripe.

ponents. The only difference between the two approaches lies in the output representation: bounding-box-based localization for detection and pixel-level instance masks for segmentation.

Moreover, both models were trained using identical training configurations, This includes the same input image resolution, batch size, optimizer, learning rate schedule, and number of training epochs. By maintaining identical optimization settings, the experimental setup ensures that observed performance differences are attributable to the object representation strategy rather than to training-related variations. The database was split into training and validation subsets using the same partitioning strategy for both configurations. Model selection was based on validation performance using the evaluation metrics described in Section II-D.

C. Experimental Environment

All experiments were conducted using the Google Colab platform on a single NVIDIA Tesla T4 GPU. While the study relies on a single hardware configuration, the NVIDIA T4 provides a common benchmark for evaluating the computational efficiency of YOLO-based models. To assess the practical trade-offs for real-world agricultural deployment, we measured processing speed across preprocess, inference, and postprocess stages for both object detection and instance segmentation configurations.

For the object detection model (YOLOv8m-Det), the average processing speed per image was 0.2ms for preprocessing, 27.1ms for inference, and 1.7ms for postprocessing, totaling approximately 29.0ms per image. The instance segmentation model (YOLOv8m-Seg) exhibited a slight increase in computational overhead, with 2.8ms for preprocessing, 30.1ms for inference, and 5.0ms for postprocessing, totaling 37.9ms per image. This represents an approximate inference-time overhead of 30.7% compared to detection. Despite this increase, the segmentation model maintains a high frame rate suitable for real-time applications on edge devices.

All implementations were carried out in Python 3 using the Ultralytics YOLOv8 library. The YOLOv8m configuration was selected to provide a consistent and balanced computational setting for both object detection and instance segmentation tasks. Using the same model scale across experiments ensures that performance differences reflect representation choice rather than differences in model capacity or computational resources.

D. Evaluation Metrics

Model performance was evaluated using standard metrics commonly employed in object detection and instance segmentation tasks. These include Mean Average Precision (mAP) at threshold 50 (mAP@50), mAP from 50 to 95 (mAP@50:95), recall (R), and box precision [Box(P)].

The Mean Average Precision assesses the mean precision for all categories at a fixed Intersection over Union (IoU) of 0.5 (mAP50), and across a range from 0.5 to 0.95 (mAP50:95). It expresses the ability of the model to localize and label objects accurately. To ensure a fair comparison, we report mAP@0.5:0.95 computed using bounding-box IoU for the detection model and both bounding-box and mask-based IoU for the segmentation model. This dual evaluation isolates the benefit of pixel-level supervision (mask IoU) while maintaining a common denominator (box IoU) for benchmarking against traditional detectors. The mAP equation is given by Eq. (1), where the average precision for class i is denoted as AP_i and the number of classes as N [22].

$$mAP = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{1}$$

The Recall metric signify the capability of the model to find all the relevant objects. It is the number of objects detected accurately about the total ground truth instances. A high figure of recall indicates low numbers of missed detections, which is an important requirement for agricultural uses such as tracking the maturity of fruits. It is determined by Eq. (2), where *TP* is an outcome where the model accurately forecasts the positive class. An *FN* is a result where the model misclassifies the negative class [23].

$$\text{R} = \frac{TP}{TP + FN} \tag{2}$$

While, Box(P) refers to the ratio of the number of accurately predicted bounding boxes to the total number of predicted boxes. It determines the rate at which the model's detections are precise and not false positives. High precision implies that the model provides only the most confident and relevant detections. It is represented by Eq. (3), where the *FP* is an output where the model misclassifies the positive class [24].

$$\text{Box(P)} = \frac{TP}{TP + FP} \tag{3}$$

In addition to quantitative evaluation, qualitative analysis was conducted through visual inspection of detection outputs and segmentation masks. This analysis focuses on boundary definition, object separation, and behavior in scenes containing overlapping or partially occluded fruit instances.

III. Results and Discussion

This section presents and discusses the experimental results obtained from the comparative evaluation of object detection and instance segmentation approaches for tomato maturity analysis. The objective is to analyze how different object representations influence localization accuracy, boundary delineation, and confidence behavior in visually complex scenes. Quantitative metrics, Precision–Recall characteristics, and qualitative visual comparisons are jointly examined to provide a comprehensive assessment.

Before evaluating instance segmentation, object detection performance was examined in our previous work to establish a suitable baseline [25]. Multiple YOLOv8 detection variants, including nano, small, medium, large, and extra-large configurations, were evaluated under identical experimental conditions. Among these models, the YOLOv8 medium configuration achieved the best overall balance between detection accuracy and model capacity, with an mAP@0.5 of approximately 0.84. Increasing model size beyond this configuration resulted in only marginal performance gains, while smaller models exhibited reduced accuracy. These findings suggest that detection performance is not primarily limited by model capacity, motivating the present investigation into alternative object representations rather than further architectural scaling. We acknowledge that re-evaluating these variants under the exact segmentation comparison setup would further reduce selection bias; however, the medium variant's stable performance justifies its choice for isolating representation effects.

The quantitative performance comparison between the detection and segmentation approaches is summarized in Table I. The results reported represent outcomes from controlled training cycles. While these metrics are derived from a single stabilized training run, the consistency of performance gains across all classes and the stability of the validation loss curves suggest that these values reflect robust performance trends rather than stochastic variations of the training process. This level of descriptive consistency provides a reliable basis for our comparative analysis of spatial representations.

Class-level analysis further emphasizes the benefits of pixel-level representation. For ripe tomatoes, the segmentation model achieves an mAP@0.5:0.95 of 0.880 compared to 0.511 for detection, highlighting improved boundary alignment and instance separation in dense scenes. Similar trends are observed for unripe tomatoes, particularly in recall, where segmentation demonstrates a higher ability to detect relevant instances and reduce missed detections. Although the detection-based approach exhibits relatively high bounding box precision for unripe tomatoes, this does not translate into robust localization performance under stricter IoU thresholds.

TABLE I. Comparative Model Performance for Tomato Maturity Detection and Segmentation.

Model	Class	Box (P)	Recall (R)	Box mAP		Mask mAP	
				@0.5	@0.5:0.95	@0.5	@0.5:0.95
v8m-Seg	Unripe	0.873	0.886	0.912	0.784	0.901	0.761
v8m-Seg	Ripe	0.939	0.898	0.951	0.880	0.943	0.872
v8m-Seg	**All**	**0.906**	**0.892**	**0.931**	**0.832**	**0.922**	**0.817**
v8m-Det	Unripe	0.964	0.805	0.892	0.591	-	-
v8m-Det	Ripe	0.767	0.742	0.785	0.511	-	-
v8m-Det	All	0.865	0.773	0.839	0.551	-	-

A more detailed investigation into performance across different IoU thresholds reveals that the advantage of instance segmentation becomes increasingly pronounced as the overlap criteria become more stringent. For thresholds between 0.75 and 0.95, the recall for the detection model degrades rapidly, while the segmentation approach maintains significantly higher levels of detection stability. This behavior can be attributed to the inherent nature of bounding-box representations, which often capture background pixels or parts of neighboring fruits, thereby reducing the precise overlap with the ground truth. In contrast, pixel-level masks provide a finer delineation of the fruit boundaries, effectively reducing spatial ambiguity in crowded clusters and improving handling of overlapping fruit.

Furthermore, we observed that the localized geometric accuracy provided by segmentation translates into more reliable classification in cluttered regions. While object detection sometimes struggles with small or heavily occluded tomatoes—often misclassifying them or producing false negatives—the segmentation representation's ability to isolate object surface area leads to higher classification confidence.

An analysis across fruit scales reveals that the segmentation advantage is particularly pronounced for small and medium-sized tomatoes. In such cases, bounding boxes often capture a disproportionate amount of background information relative to the object surface, leading to lower IoU values. The segmentation masks, by contrast, maintain high geometric fidelity regardless of object scale. Similarly, while both models perform well in low-occlusion scenarios, the segmentation model shows a consistent performance advantage as occlusion increases beyond 30%. This suggests that the benefit of pixel-level representation is primarily driven by its ability to resolve visual ambiguities in heavily overlapping clusters, which are common in dense greenhouse arrangements.

Scenarios where detection might still offer an advantage are typically limited to very simple, low-density scenes where the coarse localization of a bounding box is sufficient and the lower computational latency (29.0ms vs. 37.9ms per image) might be favored for high-throughput sorting. However, for complex greenhouse conditions, the improved boundary delineations and reduced ambiguity of instance segmentation

provide a clear superiority for maturity analysis.

The Precision–Recall characteristics of the instance segmentation approach are illustrated in Fig. 2. The curves indicate that high precision is maintained over a wide range of recall values for both maturity classes, with class-specific average precision values of approximately 0.910 for ripe tomatoes and 0.943 for unripe tomatoes, and an overall mAP@0.5 close to 0.93. The smooth decline in precision at high recall levels suggests that errors predominantly occur when attempting to detect nearly all instances in highly cluttered scenes. Compared to detection-based models, the segmentation approach exhibits greater stability across confidence thresholds, indicating reduced sensitivity to parameter tuning.

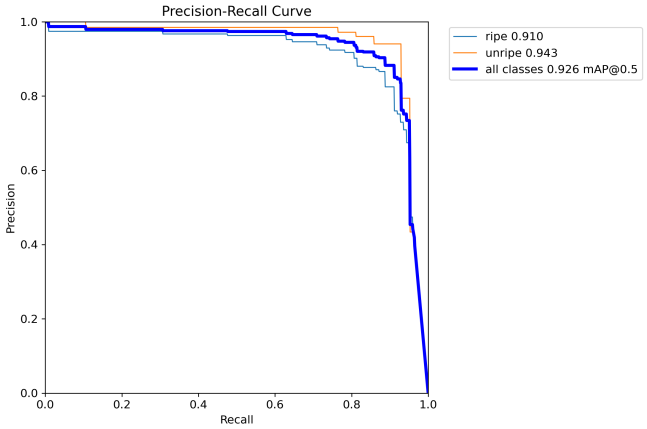

Fig. 2. Precision–Recall curve of YOLOv8m segmentation.

Compared to existing works, our results align with findings in [17], where Mask R-CNN demonstrated superior boundary localization for mature green tomatoes are compared to traditional detectors. However, our YOLOv8m-Seg approach offers a more efficient alternative with inference times around 37.9ms, significantly faster than typical Mask R-CNN implementations, while maintaining comparable spatial accuracy. Similarly, while SE-YOLOv3 [11] focuses on classification accuracy, our study highlights the geometric benefit of segmentation masks in clarifying object extent in dense greenhouse arrangements. The observed mAP@0.5:0.95 of 0.832 for box-based metrics in segmentation matches or exceeds several recent multi-modal fusion benchmarks [10], emphasizing that spatial representation choice is as critical as sensor integration.

A systematic failure mode analysis was conducted to understand the limitations of the proposed approach. We observed that segmentation boundaries occasionally exhibit "noise" or over-segmentation when tomatoes are situated against highly textured backgrounds or in extreme low-light conditions. In cases of severe occlusion (over 70% object surface hidden), both detection and segmentation models experience a reduction in recall; however, the segmentation model is more prone to producing fragmented masks. These boundaries may deviate from the true fruit contour when leaf shadows cast sharp edgeline gradients that the mask head misinterprets as object boundaries. Furthermore, although the results are based on a single training run, the stability of the validation loss and the consistency of mAP gains across classes suggest that these failure modes are systematic rather than random variabilities of the training process. Future work utilizing ensemble methods or multiple-run averaging could provide further confidence intervals for these performance trends.

Fig. 3 and Fig. 4 present a representative qualitative results comparing the YOLOv8m instance segmentation model and the YOLOv8m detection model. Each image pair illustrates the same input scene processed by both approaches.

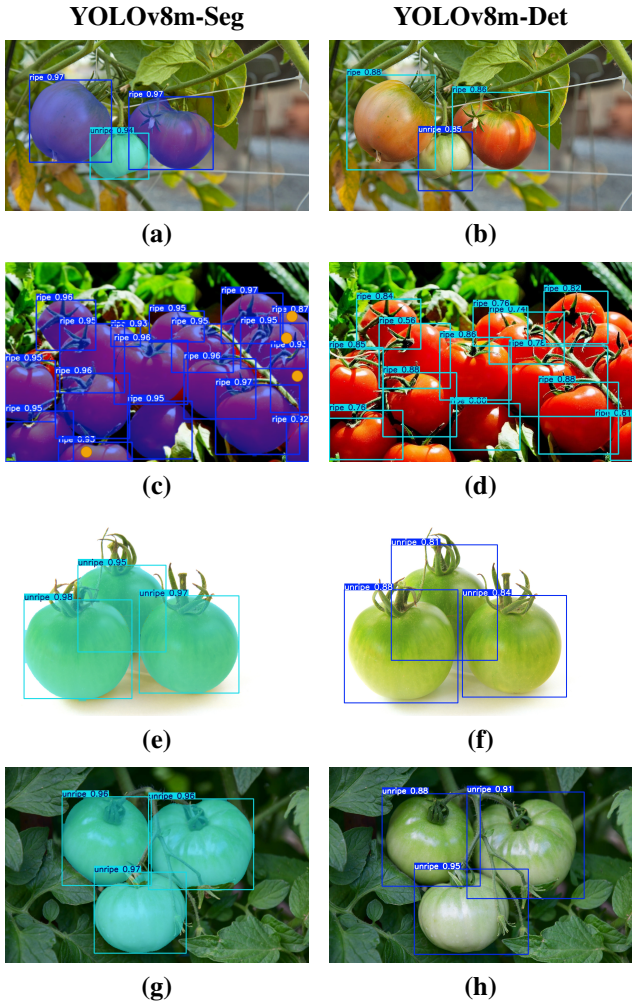

Fig. 3. Qualitative comparison between YOLOv8m instance segmentation (left) and YOLOv8m detection (right) for tomato ripeness stages (A).

In Fig. 3(a), the YOLOv8m-Seg model correctly detects two ripe tomatoes and one unripe tomato in a simple scene. It assigns high confidence values between 0.94 and 0.97 while producing pixel-level masks that closely follow object boundaries. The corresponding detection output in Fig. 3(b) identifies the same instances but assigns lower confidence values, typically between 0.85 and 0.88, and relies on bounding boxes that include surrounding background regions.

A more complex scenario is shown in Fig. 3(c), where a dense cluster of ripe tomatoes is present. The segmentation model successfully detects and segments nearly all visible instances, including partially occluded fruits, with confidence values mostly exceeding 0.95. In contrast, Fig. 3(d) shows the

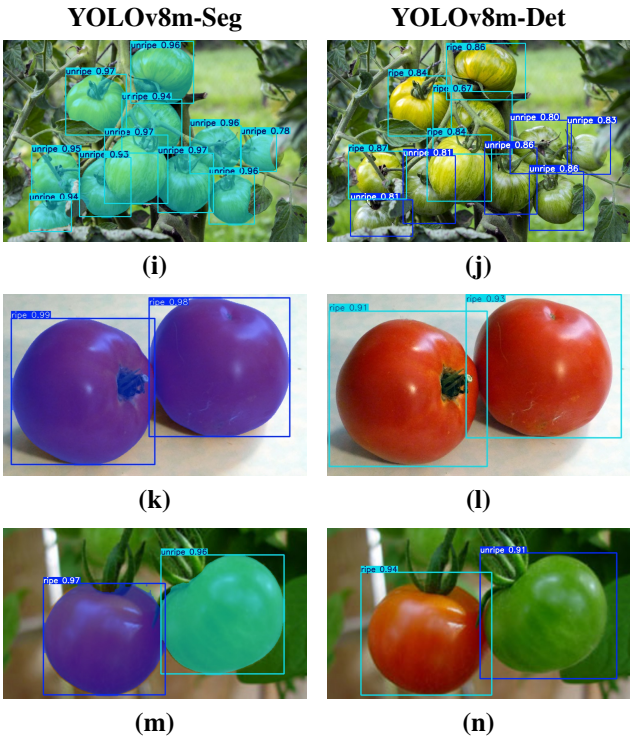

Fig. 4. Qualitative comparison between YOLOv8m instance segmentation (left) and YOLOv8m detection (right) for tomato ripeness stages (B).

detection-based output for the same scene, where several tomatoes are missed and confidence values vary widely, ranging from approximately 0.56 to 0.88, particularly in overlapping regions.

Another example dominated by visually similar unripe tomatoes are illustrated in Fig. 3(e) and Fig. 3(f). In Fig. 3(e), the segmentation model accurately delineates all instances and assigns confidence values between 0.95 and 0.98, maintaining consistent boundary definition despite low inter-class visual contrast. In Fig. 3(f), the detection-based model identifies most tomatoes but exhibits greater variability in confidence, typically between 0.81 and 0.88, and produces bounding boxes with limited separation between adjacent objects.

Fig. 3(g) and Fig. 3(h) present a moderately dense scene processed by both approaches. In Fig. 3(g), the segmentation model detects three unripe tomatoes with confidence values around 0.96–0.97 and clearly separates each instance using pixel-level masks. The detection output in Fig. 3(h) identifies the same instances but assigns lower confidence values, ranging from 0.88 to 0.95, and shows partial overlap between bounding boxes.

From Fig. 4, Fig. 4(i) and Fig. 4(j) present a visual representation dominated by unripe tomatoes with significant overlap and partial occlusions. In Fig. 4(i), the YOLOv8m-Seg model successfully identifies and segments nearly all visible unripe tomato instances. Confidence values are consistently high, typically ranging from 0.94 to 0.97, even for partially occluded fruits. The pixel-level masks enable clear separation of adjacent tomatoes despite the dense arrangement. In contrast, Fig. 4(j) shows the output of the YOLOv8m detection model for the same input image. While several tomatoes are detected, confidence values are lower and more variable, commonly between 0.80 and 0.87, and multiple bounding boxes overlap. Some instances are misclassified or weakly localized, reflecting increased ambiguity in crowded regions.

Fig. 4(k) and Fig. 4(l) illustrate a simpler visual configuration containing two ripe tomatoes placed against a uniform background. In Image (k), the segmentation model accurately delineates both fruits and assigns very high confidence values of approximately 0.98–0.99, indicating strong classification certainty and precise object extent modeling. In Fig. 4(l), the detection-based model also correctly identifies both ripe tomatoes; however, confidence values are slightly lower, around 0.91–0.93, and the bounding boxes include background regions, providing a coarser spatial description compared to the segmentation output.

Fig. 4(m) and Fig. 4(n) depict a mixed-maturity configuration with one ripe and one unripe tomato in close proximity. In Fig. 4(m), the YOLOv8m-Seg model correctly distinguishes between maturity stages, assigning a confidence of approximately 0.97 to the ripe tomato and 0.96 to the unripe tomato. The segmentation masks clearly separate the two instances, despite their spatial closeness.

In Fig. 4(n), the detection-based model also identifies both tomatoes; however, confidence values differ more noticeably, with the ripe tomato detected at around 0.94 and the unripe tomato at approximately 0.91. The bounding boxes partially overlap and provide less precise delineation between the two fruits.

Across all qualitative examples, the instance segmentation approach consistently demonstrates clearer object delineation and more stable confidence assignment as scene complexity increases. These qualitative observations complement the quantitative improvements reported in Table I and the Precision–Recall behavior shown in Fig. 2. Together, they reinforce the conclusion that object representation plays a critical role in tomato maturity analysis. Bounding-box-based detection provides efficient coarse localization but offers limited descriptive power in dense and ambiguous scenes. Instance segmentation addresses these limitations by explicitly modeling object boundaries at the pixel level, leading to improved localization accuracy, higher recall, and more reliable confidence behavior without increasing model capacity or architectural complexity.

IV. Conclusion

This study presented a systematic visual computing evaluation of object detection and instance segmentation approaches for tomato maturity analysis, with a particular focus on how object representation influences localization accuracy and boundary delineation in visually complex scenes. By conducting a controlled comparison within a common DL model and under identical experimental conditions, the analysis isolated the effect of bounding-box-based and pixel-level representations on performance. Our findings confirm that pixel-level representation provides a more reliable basis for fine-grained maturity analysis than coarse bounding-box localization, particularly as IoU thresholds increase. The observed geometric superiority of instance segmentation translates into

more consistent object separation and more stable confidence assignments in dense greenhouse scenes.

Despite these encouraging results, several limitations should be acknowledged. The evaluation was conducted on a localized dataset of 177 images and focused on two maturity classes (ripe and unripe) using a single YOLOv8m backbone. This scope may limit the immediate generalization of these absolute performance metrics to more diverse environmental conditions or complex multi-class maturity stages. Furthermore, the reliance on a single hardware configuration (NVIDIA T4) and a single training run per configuration provides a baseline but does not capture the full variance of hardware stability or training stochasticity.

Future research will aim to extend this analysis to larger and more diverse datasets, incorporating a broader range of tomato varieties and maturity stages to test generalizability across crop types such as peppers or apples. We propose to investigate deeper representation analyses, including panoptic modeling to handle background-foreground ambiguity and the integration of boundary-aware loss functions to further refine mask precision. Validating these conclusions on synthetic occlusion setups or multi-view datasets would also provide deeper insight into the robustness of pixel-level supervision. Finally, systemic failure mode analysis on an automated scale will be essential for refining these models for seamless deployment in autonomous robotic harvesting systems.

ACKNOWLEDGMENT

The authors would like to thank the Moroccan Ministry of Higher Education, Scientific Research and Innovation and the OCP Foundation who funded this work through the APRD research program.

REFERENCES

[1] M. Nixon and A. S. Aguado, *Feature Extraction and Image Processing for Computer Vision*. Oxford, U.K.: Academic Press, 2025.

[2] C. Che, H. Zheng, Z. Huang, W. Jiang, and B. Liu, "Intelligent Robotic Control System Based on Computer Vision Technology," *arXiv preprint*, arXiv:2404.01116, 2024.

[3] H. Zheng, L. Fu, and Q. Ye, "Flexible capped principal component analysis with applications in image recognition," *Information Sciences*, vol. 614, pp. 289–310, 2022.

[4] Y. Yang, Y. Han, S. Li, Y. Yang, M. Zhang, and H. Li, "Vision-based fruit recognition and positioning technology for harvesting robots," *Computers and Electronics in Agriculture*, vol. 213, p. 108258, 2023.

[5] G. C. Wakchaure, S. B. Nikam, K. R. Barge, S. Kumar, K. K. Meena, V. J. Nagalkar, J. D. Choudhari, V. P. Kad, and K. S. Reddy, "Maturity stages detection prototype device for classifying custard apple (Annona squamosa L.) fruit using image processing approach," *Smart Agricultural Technology*, vol. 7, p. 100394, 2024.

[6] H.-T. Vo, N. N. Thien, and K. C. Mui, "A deep transfer learning approach for accurate dragon fruit ripeness classification and visual explanation using Grad-CAM," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 11, 2023.

[7] P. Mittal, "A comprehensive survey of deep learning-based lightweight object detection models for edge devices," *Artificial Intelligence Review*, vol. 57, no. 9, p. 242, 2024.

[8] J. Ma, M. Li, W. Fan, and J. Liu, "State-of-the-art techniques for fruit maturity detection," *Agronomy*, vol. 14, no. 12, 2024.

[9] M. Amiriebrahimabadi, Z. Rouhi, and N. Mansouri, "A comprehensive survey of multi-level thresholding segmentation methods for image processing," *Archives of Computational Methods in Engineering*, vol. 31, no. 6, pp. 3647–3697, 2024.

[10] Y. Liu, C. Wei, S.-C. Yoon, X. Ni, W. Wang, Y. Liu, D. Wang, X. Wang, and X. Guo, "Development of multimodal fusion technology for tomato maturity assessment," *Sensors*, vol. 24, no. 8, p. 2467, 2024.

[11] F. Su, Y. Zhao, G. Wang, P. Liu, Y. Yan, and L. Zu, "Tomato maturity classification based on SE-YOLOv3-MobileNetV1 network under nature greenhouse environment," *Agronomy*, vol. 12, no. 7, p. 1638, 2022.

[12] S. N. Appe, G. Arulselvi, and G. N. Balaji, "CAM-YOLO: Tomato detection and classification based on improved YOLOv5 using combining attention mechanism," *PeerJ Computer Science*, vol. 9, p. e1463, 2023.

[13] H. Li, Z. Gu, D. He, X. Wang, J. Huang, Y. Mo, P. Li, Z. Huang, and F. Wu, "A lightweight improved YOLOv5s model and its deployment for detecting pitaya fruits in daytime and nighttime light-supplement environments," *Computers and Electronics in Agriculture*, vol. 220, p. 108914, 2024.

[14] Y. Bai, J. Yu, S. Yang, and J. Ning, "An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings," *Biosystems Engineering*, vol. 237, pp. 1–12, 2024.

[15] C. Charisis and D. Argyropoulos, "Deep learning-based instance segmentation architectures in agriculture: A review of the scopes and challenges," *Smart Agricultural Technology*, vol. 8, p. 100448, 2024.

[16] X. Gao, J. Ding, R. Zhang, and X. Xi, "YOLOv8n-CA: Improved YOLOv8n model for tomato fruit recognition at different stages of ripeness," *Agronomy*, vol. 15, no. 1, 2025.

[17] L. Zu, Y. Zhao, J. Liu, F. Su, Y. Zhang, and P. Liu, "Detection and segmentation of mature green tomatoes based on Mask R-CNN with automatic image acquisition approach," *Sensors*, vol. 21, no. 23, p. 7842, 2021.

[18] Z. Chen, D. Ting, R. Newbury, and C. Chen, "Semantic segmentation for partially occluded apple trees based on deep learning," *Computers and Electronics in Agriculture*, vol. 181, p. 105952, 2021.

[19] X. Liu, W. Gong, L. Shang, X. Li, and Z. Gong, "Remote sensing image target detection and recognition based on YOLOv5," *Remote Sensing*, vol. 15, no. 18, p. 4459, 2023.

[20] A. Lu, L. Ma, H. Cui, J. Liu, and Q. Ma, "Instance segmentation of lotus pods and stalks in unstructured planting environment based on improved YOLOv5," *Agriculture*, vol. 13, no. 8, p. 1568, 2023.

[21] C. M. Badgujar, A. Poulose, and H. Gan, "Agricultural object detection with You Only Look Once (YOLO) algorithm: A bibliometric and systematic literature review," *Computers and Electronics in Agriculture*, vol. 223, p. 109090, 2024.

[22] P. Li, J. Zheng, P. Li, H. Long, M. Li, and L. Gao, "Tomato maturity detection and counting model based on MHSA-YOLOv8," *Sensors*, vol. 23, no. 15, p. 6701, 2023.

[23] E. Tapia-Mendez, I. A. Cruz-Albarran, S. Tovar-Arriaga, and L. A. Morales-Hernandez, "Deep learning-based method for classification and ripeness assessment of fruits and vegetables," *Applied Sciences*, vol. 13, no. 22, p. 12504, 2023.

[24] R. Sapkota, D. Ahmed, and M. Karkee, "Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments," *Artificial Intelligence in Agriculture*, vol. 13, pp. 84–99, 2024.

[25] S. Ait Oussous, R. El Bouayadi, D. Zejli, and A. Amine, "Smart greenhouse tomato maturity detection based on YOLOv8 model," in *Proc. Int. Conf. on Computer and Communication Engineering*, 2025, pp. 115–126.