Proceedings of the 5th Faculty of Industrial Technology International Congress Bandung, Indonesia, September 17-18, 2025
ISSN 2962 – 1798

Real-Time Plant Stem Segmentation on Smartphones Using YOLO for Precision Agriculture

Miroslav Holý [a, 1], *Vladimír* Cviklovič [a], and *Martin* Olejár [a], and *Stanislav* Paulovič [a]

[a] Slovak University of Agriculture in Nitra, Faculty of Engineering, 94976 Nitra, Slovakia

Abstract. A crucial aspect of precision agriculture is plant phenotyping, as it provides insights into plant development, status and productivity. In this study, we present an approach for plant stem segmentation optimized for deployment on smartphones to support cost-effective precision agriculture practices. Using the YOLO (You Only Look Once) object detection framework, we trained a custom lightweight model capable of segmenting main stems under variable lighting and plant conditions. The model was evaluated on a dataset collected using a smartphone in a greenhouse, where it achieved 0.727 mAP50 and 20 FPS on Google Pixel 7a smartphone. Therefore, real-time stem segmentation can be effectively integrated into agricultural practices as a practical tool for data-driven crop management.

1 Introduction

Tomato plants hold a significant position in global agriculture due to their high economic value [1]. Data-driven crop management is a continuous pursuit, and plant phenotyping—the quantitative measurement of plant traits—plays a crucial role in this [2]. Traditional methods of manual phenotyping are slow, often inefficient, and can even cause damage to the plants' physical structure. This highlights the need for automated, non-destructive techniques for accurately assessing plant characteristics. However, analysis is often costly when relying on high-performance hardware or cloud computing resources [3]. Therefore, shifting toward in-field devices offers a more cost-effective and scalable solution. The potential for cost-effective and timely monitoring of plant development, status and productivity through edge computing applications is substantial. In this context, a less computationally demanding model is deployed directly on the device. However, a common limitation of handheld devices such as smartphones is slow measurement speed [4], which can lead to labor-intensive data collection. Therefore, the exploration and evaluation of more efficient models is necessary.

Among the various aspects of plant phenotyping, the segmentation of plant stems is a fundamental step. Identifying and isolating the stem allows for further extraction of key morphological parameters such as stem diameter, height, and internode length, which are vital indicators of plant growth and health [2]. Furthermore, collected data can be used for other tasks like distinguishing crops and weeds [5, 6]. Accurately and efficiently

[1] Corresponding author: xholy@uniag.sk

segmenting plant stems presents several challenges. Tomato plants exhibit complex structures with overlapping leaves and intricate branching patterns. The visual similarity in color between stems and leaves in tomato plants further complicates the task of stem recognition in images. Additionally, variations in lighting conditions and the presence of complex backgrounds can significantly impact the performance of segmentation algorithms. Therefore, robust and reliable methods for plant stem segmentation are essential, but they are often computationally demanding.

2 Background

Loyani and Machuve [7] developed a deep learning-based mobile application specifically designed to segment damage caused by Tuta Absoluta pest on tomato plants. Their CNN segmentation model was successfully deployed on a smartphone via the application TutaSegmenter and it could detect and segment infected areas on tomato leaves with a minimum confidence of 0.7 in only 5 seconds. While the model achieved high performance (f1-score 0.91), its inference speed leaves room for improvement.

Widiyanto et al. [8] created a real-time processing framework using a Mask Region-Convolutional Network (R-CNN) with ResNet101 backbone to monitor tomato growth. Although their work primarily focused on fruit development rather than stem segmentation, the method achieved high accuracy with 0.97 using the Dice Coefficient and 0.94 using the Jaccard Coefficient. However, the authors did not specify the hardware used to meet real-time requirement, nor did they report the inference speed, therefore deployment feasibility is questionable.

Zhang et al. [9] presented a stem and leaf segmentation method for tomato seedlings that is based on an improved ResNet architecture aimed at reducing model complexity and parameter count. By integrating bottleneck modules and downsampling techniques, their lightweight model achieved 95.11% accuracy (3.26% improvement over traditional ResNet18) with 25% reduction in inference speed (4.02 seconds compared to 5.37 seconds) [9]. The model also extracted key phenotypic parameters including plant height, stem diameter, leaf area, and leaf inclination angle, with high correlation coefficients (R^2 values of 0.941, 0.752, 0.945, and 0.943 respectively). The study demonstrates feasibility of parameter detection with deep learning models from image data. However, inference speed remains a bottleneck.

Clearly, while researchers have made significant contributions in applying deep learning to tomato phenotyping, opportunities for advancement persist. In particular, optimizing the trade-off between model complexity and inference time is essential for real-world deployments. Although smaller models generally come with a reduction in precision, this trade-off may be acceptable—especially in field conditions, where slight variations in model precision might not critically affect decision-making.

3 Materials and methods

In our experiment, we began by capturing a dataset of 400 high-resolution images (with resolution 1920×1080 px, total size ~400 MB) of Cherry tomato plant variety using a standard smartphone camera under natural lighting conditions in the greenhouse. For manual segmentation of plant stems the images were then imported into Label Studio where we used polygon segmentation. The annotation process was initially carried out manually for the first 50 images to establish a foundational training set. To enhance labeling efficiency and model performance, we adopted an active learning strategy. After the initial manual annotations, we trained a preliminary segmentation model and used its predictions on subsequent unlabeled images to assist the annotators. The model-generated polygons were reviewed and corrected, effectively accelerating the labeling process while continuously refining the training data. This iterative loop of model training and correction allowed us to gradually improve prediction quality with minimal human effort as the dataset grew.

Once a sufficient volume of labeled data had been prepared, we proceeded with hyperparameter tuning to optimize model performance. A total of 24 hyperparameters—encompassing learning rate, momentum, batch size, data augmentation settings, confidence thresholds, and more—were systematically explored over 270 optimization iterations, where during each iteration a model was trained for 30 epochs. This tuning was performed using genetic algorithm to identify a configuration that maximized fitness function. The fitness function is a weighted sum of metrics—precision, recall, and mAP50 (mean average precision at 50% IoU)—with a weight distribution of 0.05, 0.05, and 0.9, respectively, placing the highest emphasis on maximizing overall detection quality as measured by mAP50.

For model training, we decided to use Single Stage Detectors (SSDs). SSDs are deep learning-based architectures that enable real-time performance for object detection and instance segmentation tasks. One widely praised architecture is YOLO, which divides an image into a grid and predicts bounding boxes and class probabilities simultaneously.

From YOLO framework we selected the YOLO11 architecture and its smallest segmentation model YOLO11n-seg that is particularly well-suited for deployment on resource-constrained devices such as smartphones. The images were resized to 640×640 pixels, and the dataset was split into training, validation, and test subsets using an 80/10/10 ratio. The batch size was set to 8, number of epochs was 800 with early stopping (when model performance does not improve) after 150 epochs to prevent overfitting.

All experiments were conducted on a machine with Windows 10 operating system, AMD Ryzen 7800X3D cpu and NVIDIA RTX 4090 gpu. For testing on smartphone, Google Pixel 7a with Google Tensor G2 cpu, TPU (Janeiro) npu and Arm Mali-G710 MP7 gpu was used. As for software, Python version 3.13.1 and PyTorch version 2.6.0 were used.

4 Results and Discussion

Hyperparameter tuning identified the best configuration at iteration 208, where the model achieved its highest fitness score of 0.71. During the training process, early stopping was triggered at epoch 422, indicating that the model had converged. The optimal weights were obtained at epoch 272 and were based on validation performance. Evaluation on the test split yielded a precision of 0.821, a recall of 0.667, and a mAP50 of 0.727. These results are promising, especially considering that the model was trained on a relatively small dataset. This demonstrates the model's ability to generalize effectively, even under constrained data conditions.

Figure 1 illustrates the training performance across all 422 epochs, showing separate curves for test-set precision, recall, mAP50, and the training segmentation loss. The segmentation loss shows a steady decline throughout the training process, reflecting consistent learning, while the evaluation metrics exhibit moderate fluctuations, typical for smaller datasets. The performance plateau observed after epoch 272 aligns with the early stopping criteria and supports its selection as the best-performing checkpoint.

To assess the model's practicality for real-world use, it was deployed on a smartphone using the Ultralytics HUB mobile app. Inference speed tests revealed an average processing time of approximately 50 milliseconds per frame, equating to a real-time throughput of 20 frames per second (fps). These results confirm that real-time inference is feasible on resource-constrained devices.

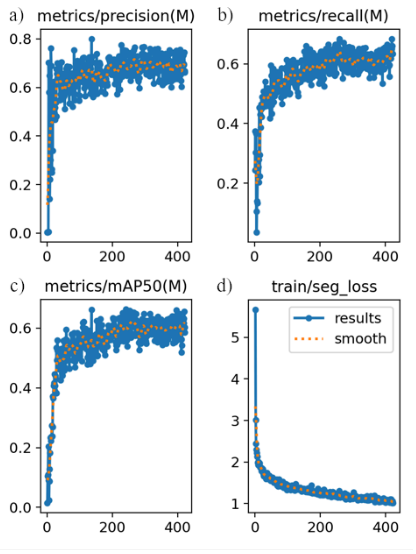

Fig. 1. Training performance and evaluation metrics of the segmentation model across 422 epochs. (a) precision on test set; (b) recall on test set; (c) mAP50 on test set; (d) segmentation loss on train set.

5 Conclusion

The results demonstrate that YOLO-based models provide an efficient solution for real-time stem detection in greenhouse environments. The ability to perform inference directly on a smartphone further highlights the practicality of deploying such models in resource-constrained settings, making the technology accessible to small-scale farmers and agricultural researchers alike. With further integration into agricultural practices, this approach has the potential to become a cost-effective tool for supporting data-driven crop management.

These findings support our initial hypothesis: that a stem segmentation model can be effectively trained and deployed on mobile devices without requiring access to cloud services, high-end computational infrastructure and large datasets. The successful deployment and real-time segmentation with satisfactory performance of ~20 frames per second (precision of 0.821, recall of 0.667, and mAP50 of 0.727) under variable greenhouse conditions demonstrate the practical potential of integrating such solutions into everyday agricultural workflows for precision agriculture. Moreover, the methodology used can serve as a foundation for future studies with similar objectives.

Future studies should focus on the development of lightweight (our model has size of 6.9 MB), fast (12.8 GFLOPs, ~20 FPS on Pixel 7a), and robust segmentation models that maintain strong performance while remaining feasible for edge computing deployment. Additionally, the absence of standardized benchmarks for deployment platforms (e.g., smartphones) and the limited reporting of inference times or floating-point operation counts (FLOPs) underscore the need for estabilishing benchmarking suites for real-time models in agricultural edge computing devices. This would accelerate research focused not only on accuracy, but also runtime efficiency, scalability, and device compatibility.

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-23-0615.

References

1. J. Chitwood-Brown, G.E. Vallad, T.G. Lee, S.F. Hutton, *Breeding for Resistance to Fusarium Wilt of Tomato: A Review*, in Genes, **12** (2021)
2. Y. Wang, Q. Liu, J. Yang, G. Ren, W. Wang, W. Zhang, F. Li, *A Method for Tomato Plant Stem and Leaf Segmentation and Phenotypic Extraction Based on Skeleton Extraction and Supervoxel Clustering*, in Agronomy, **14** (2024)
3. T. Mizik, *How can precision farming work on a small scale? A systematic literature review*, in Precision Agric, **24** (2023)
4. Q. Zhang,, R. Luan, M. Wang, J. Zhang, F. Yu, Y. Ping, L. Qiu, *Research Progress of Spectral Imaging Techniques in Plant Phenotype Studies*, in Plants, **13** (2024)
5. F. Daichang, X. Lihong, L. Dawei, X. Longjiao, *Automatic detection and segmentation of stems of potted tomato plant using Kinect*, in Sixth International Conference on Digital Image Processing (ICDIP 2014), **9159** (2014)
6. F. Zhang, A. Hassanzadeh, J. Kikkert, S.J. Pethybridge, J. v. Aardt, *White Mold and Weed Detection in Snap Beans Using UAS-Based Lidar*, in IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia (2022)
7. L. Loyani, D. Machuve, *A Deep Learning-based Mobile Application for Segmenting Tuta Absoluta's Damage on Tomato Plants*, in Eng. Technol. Appl. Sci. Res., **11** (2021)
8. S. Widiyanto, D.P. Nugroho, A. Daryanto, M. Yunus, D.T. Wardani, *Monitoring the Growth of Tomatoes in Real Time with Deep Learning-based Image Segmentation*, in International Journal of Advanced Computer Science and Applications(IJACSA), **12** (2021)
9. L. Zhang, X. Li, Z. Yang, B. Yang, S. Yu, S. Zhao, Z. Huang, X. Zhang, H. Yang, Y. Lin, H. Yu, M. Yang, *Tomato Seedling Stem and Leaf Segmentation Method Based on an Improved ResNet Architecture*, in Front. Plant Sci., **16** (to be published)