DETERMINING TOMATO STEM ANGLES USING A YOLO-BASED APPROACH

Miroslav HOLÝ[1], Vladimír CVIKLOVIČ[1], Vladimír MADOLA[1], Martin OLEJÁR[1], Stanislav PAULOVIČ[1], Ladislav TÓTH[1]

[1]Institute of Electrical Engineering, Automation, Informatics and Physics, Slovak University of Agriculture in Nitra, Tr. A. Hlinku 2, SK-94976 Nitra, Slovakia

Abstract
This study aims to develop a computationally efficient method for estimating tomato plant stem angles and identifying cutting points by applying a geometric model to outputs from a YOLOv8-seg instance segmentation model. Using RGB images of two distinct tomato varieties (Cherry and Merlice) as input to the YOLOv8-seg model, the geometric model assigns instances of side stems to main stems, analyzes junction regions, extracts stem orientations, and estimates cutting points. Evaluation against manually annotated ground truth data shows that the proposed method achieves a mean angle error of 13.27° and cutting point position error of 6.99 pixels. These results demonstrate the method's potential for automated harvesting systems in greenhouse tomato production. The computational efficiency of the approach makes it suitable for deployment on resource-constrained robotic platforms.

Key words: *precision agriculture; plant morphology; geometric model; computer vision; instance segmentation.*

INTRODUCTION

Precise measurement of stem angles and estimation of cutting points are essential for robotic pruning, plant phenotyping, and automated harvesting *(Rong et al., 2022)*. Traditional manual approaches remain labour-intensive and inconsistent, while current image-based methods either lack accuracy around junctions and branching regions or rely on computationally expensive deep learning architectures. Geometric region-growing and distance-field segmentation algorithms, such as DFSP and octree-based methods, offer strong stem-leaf segmentation performance but still face missegmentation at junctions and near branches *(Wang et al., 2023)*. Robotic vision systems like RoTSE perform skeletonization and graph-based extraction of branch traits reported branch-angle MSE up to 31° in controlled environments *(Tabb & Medeiros, 2017)*. Similarly, a vision-based robotic system for sweet pepper harvesting achieved a mean absolute error *(MAE)* of 24.5° in estimating stem angles *(Barth et al., 2019)*. Another study evaluating a sweet pepper harvesting robot reported an MAE of 54° in estimating stem angles and a positioning error of 31 mm for the Sardinero variety, while the Gialte variety showed improved performance with an MAE of 25° in estimating stem angles and a positioning error of 34 mm *(Arad et al., 2020)*. The aim of this study is to develop a computationally efficient method for estimating plant stem angles and identifying cutting points by applying a geometric model to outputs from a YOLOv8-seg segmentation model.

MATERIALS AND METHODS

Plant image data acquisition

The images were collected using RGB-D camera *(Intel RealSense D456)* with 1280×720 pixel resolution. To allow for a wide range of adaptability in environment, two distinct varieties of tomatoes *(Cherry and Merlice)* were captured. Since the larger tomatoes of the Merlice variety tend to overlap and obscure the stem in side-view images, we performed two types of image acquisition – top and side views – to ensure sufficient visibility of the stem when collecting data. Samples of such views are presented in Fig. 1. A total of 111 images were selected. For the Cherry variety, 18 side-view images and 1 top-view image were selected, while for the Merlice variety, 62 side-view images and 30 top-view images were collected.

Preprocessing of segmented instances of YOLOv8-seg model

Each image is first processed using the YOLOv8-seg model to retrieve segmentation instances. As illustrated in Fig. 2, the geometric model then applies a sequence of operations to prepare the data for

stem angle analysis. During preprocessing, adjacent masks are merged – an optional step, but helpful when using smaller YOLOv8-seg models that may split a single stem into multiple instances. During first step all masks are separated into class-specific groups based on their YOLO classification. The masks within individual classes are then dilated using a structuring element *(*3×3 *square kernel)* a total of 10 times. If the dilated mask and any other binary mask within same class share common pixels, they are considered overlapping and grouped together. Final merged masks are obtained by merging all masks within the individual groups. As they are merged into a single continuous mask, a polygonal representation is extracted.

(a) ***(b)***

Fig. 1 Tomato varieties in different views: **a** Merlice variety – top view; **b** Cherry variety – side view

Geometric model processing pipeline for angle and cutting point estimation

Processing pipeline consists of assignment of side stems to main stems, junction vicinity analysis, main and side stem direction extraction, and estimation of cutting point. During assignment of side stems to main stems, each side stem is paired with its nearest main stem using a combination of tip point analysis and polygon boundary distances, with assignments only made if the distance falls below a specified threshold. This ensures that the side stem is paired with an instance that is directly *(or almost directly)* connected at its end. To target only the relevant region for further analyses, junction vicinity analysis determines the local region where stems meet by calculating an adaptive radius based on stem sizes as defined in equation *(1)*

$$r = \alpha \times min\big(size(M_{main}), size(M_{side})\big) + \beta \qquad (1)$$

where r denotes the adaptive vicinity radius, α and β are multiplier and additive base parameters, M denotes the mask *(main or side)* and $size$ of the mask is defined as the square root of the sum of its pixels and min function ensures the smaller mask size is used for the calculation.

For determining the junction center, either mask intersection or closest-point analysis is used. If the masks intersect, the mean of the intersection region is taken. Otherwise, the closest pair of boundary points is identified, and the midpoint between them is used as the junction center. Main stem direction extraction analyses edge patterns within the adaptive vicinity radius r of the junction center. This is done using Canny edge detection followed by PCA to estimate the primary growth direction, which is parallel to the main stem boundary. Side stem direction extraction focuses on the immediate contact region around the junction center. The direction is calculated from the junction center to the centroid of side stem pixels located within a contact radius r_c as defined in equation *(2)*

$$\overrightarrow{d_{side}} = \frac{\overrightarrow{C_{contact}} - \overrightarrow{C_{junction}}}{\left\|\overrightarrow{C_{contact}} - \overrightarrow{C_{junction}}\right\|} \qquad (2)$$

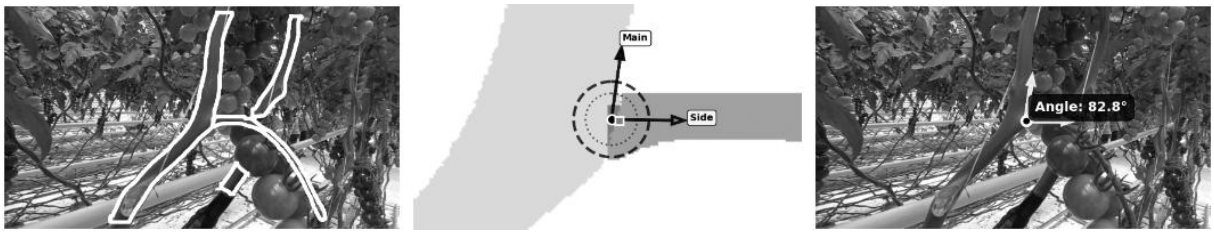
where $\overrightarrow{d_{side}}$ is the normalized side stem direction vector, $\overrightarrow{C_{contact}}$ is the centroid of side stem pixels within contact radius r_c *(which is a parameter)* and $\overrightarrow{C_{junction}}$ is the junction center point position.

Finally, the cutting point is estimated by applying an offset from the junction center toward the side stem. This offset accounts for the expected geometric error and the width of the cutting tool, as defined in equation *(3)*

$$\overrightarrow{P_{cut}} = \overrightarrow{C_{junction}} + d_{bias} \times \overrightarrow{d_{side}} \qquad (3)$$

ISBN 978-80-213-3541-7

where $\overrightarrow{P_{cut}}$ is the cutting point position, $\overrightarrow{C_{junction}}$ is the junction center point position, d_{bias} is bias distance and can be calculated as $b_{bias} \times \left\|\overrightarrow{C_{contact}} - \overrightarrow{C_{junction}}\right\|$ *(where b_{bias} is a parameter)*, and $\overrightarrow{d_{side}}$ is the side stem direction vector from equation *(2)*.



(a) *(b)* *(c)*

Fig. 2 Stem angle analysis pipeline: **a** Instance segmentation results on the original image; **b** Geometric model results with main stem direction vector *(black arrow)*, side stem direction vector *(grey arrow)*, junction center *(black circle)*, optimal cutting point *(grey square)*, adaptive vicinity radius *(dashed circle)* and contact radius *(dotted circle)*; **c** Resulting vectors overlaid on the original image

Hyperparameter search

To determine optimal values for all parameters *(α, β, r_c and b_{bias})*, a comprehensive hyperparameter search was conducted across a 4-dimensional parameter space. This exhaustive search strategy involved systematically testing multiple combinations of parameter values to identify the configuration that yielded the best performance. Given that optimal hyperparameter values are highly dependent on the characteristics of the input data and the specific plant species, careful tuning and exploration are necessary for each new dataset *(Bergstra & Bengio, 2012)*.

Performance evaluation

To evaluate the performance of the geometric model, each pair of connected main stem and side stem was assessed against manually annotated ground truth data. The primary evaluation metrics used were mean angle error and position error. An angle error of up to 25° is generally considered acceptable for successful fruit harvesting *(Barth et al., 2019)*. In terms of position error, lower values are preferred, as excessive residual stem left on the plant can increase susceptibility to diseases and negatively impact plant health. All image analysis, geometric modelling, and performance evaluation were performed using Python 3.12.3 on an AMD Ryzen 7 7800X3D CPU.

RESULTS AND DISCUSSION

The hyperparameter search results *(Tab. 1)* show clear performance differences across view types and varieties, each requiring different optimal hyperparameters. This indicates that collecting image metadata such as view type and variety can enhance model performance by reducing both angle and position error. While view type is readily available during image acquisition, identifying variety would require incorporating variety classification into the processing pipeline.

Tab. 1 Hyperparameter search results by variety and view

Variety	View	Optimal hyperparameters				Mean angle error	Mean position error	Samples
		α	β	r_c	b_{bias}	°	px	
Cherry	Top	0.2	10	12	0.50	0.39	2.59	1
	Side	0.1	5	8	0.01	9.55	5.31	36
Merlice	Top	0.1	5	8	0.50	14.88	6.73	60
	Side	0.1	5	10	0.50	13.50	8.06	98
Combined	Top	0.1	5	8	0.50	14.67	6.67	61
	Side	0.1	5	8	0.50	12.64	7.14	134
	Both	0.1	5	8	0.50	13.27	6.99	195

Performance, measured by mean angle error and mean position error, was found to be within acceptable ranges. Notably, angle errors were below the 25° threshold considered tolerable for automated harvesting tasks *(Barth et al., 2019)*, and position errors remained low enough to limit residual stem length, helping minimize disease risk. Timing analysis showed that the geometric model runs in 0.07 seconds per image on average, with 0.06 seconds spent on preprocessing. This step could be avoided by using a larger YOLOv8-seg model, as preprocessing primarily serves to correct split masks from smaller models.

CONCLUSIONS

The developed method successfully combines deep learning segmentation with geometric analysis to automate tomato stem angle and cutting point estimation. Experimental results show a mean angle error of 13.27° and a mean positional error of 6.99 pixels, with an average processing time of 0.07 seconds per image, confirming the method's computational efficiency suitable for real-time robotic applications. These results indicate that the proposed approach can provide the precision required for automated harvesting systems in greenhouse tomato production. The ability to accurately identify cutting points and stem orientations addresses a critical bottleneck in robotic harvesting, where incorrect cuts can damage plants and reduce yield quality. The successful validation across two tomato varieties *(Cherry and Merlice)* demonstrates the method's robustness, though further refinement is needed to optimize performance with top-view imaging configurations. Future work should focus on expanding the dataset to cover diverse lighting conditions and edge cases, and on developing specialized algorithms for top-view camera perspectives to support more robust robotic harvesting systems.

ACKNOWLEDGMENT

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

REFERENCES

1. Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström, T., Hemming, J., Kurtser, P., Ringdahl, O., Tielen, T., & van Tuijl, B. *(2020)*. Development of a sweet pepper harvesting robot. Journal of Field Robotics, 37*(6)*, 1027–1039.
2. Barth, R., Hemming, J., & Van Henten, E. J. *(2019)*. Angle estimation between plant parts for grasp optimisation in harvest robots. *Biosystems Engineering, 183*, 26–46.
3. Bergstra, J., & Bengio, Y. *(2012)*. Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13, 281–305.
4. Rong, J., Dai, G., & Wang, P. *(2022)*. A peduncle detection method of tomato for autonomous harvesting. Complex & Intelligent Systems, 8*(4)*, 2955–2969.
5. Tabb, A., & Medeiros, H. *(2017)*. A robotic vision system to measure tree traits. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems *(IROS)*, 6005–6012. IEEE.
6. Wang, D., Song, Z., Miao, T., Zhu, C., Yang, X., Yang, T., Zhou, Y., Den, H., & Xu, T. *(2023)*. DFSP: A fast and automatic distance field-based stem-leaf segmentation pipeline for point cloud of maize shoot. Frontiers in Plant Science, 14, 1109314.

Corresponding author:
Ing. Miroslav Holý, Institute of Electrical Engineering, Automation, Informatics and Physics, Faculty of Engineering, Slovak University of Agriculture in Nitra, Tr. A. Hlinku 2, 949 76 Nitra, Slovak Republic, phone: +421 37 641 4723, e-mail: xholy@uniag.sk