Acta Technologica Agriculturae 3
Nitra, Slovaca Universitas Agriculturae Nitriae, 2025, pp. 131–140

TOMATO STEM RECOGNITION BASED ON YOLOX

Miroslav HOLÝ[1], Vladimír CVIKLOVIČ*[1], Ladislav TÓTH[1], Lukáš VACHO[1], Martin OLEJÁR[1], Patrik KÓSA[2], Juraj BALÁŽI[2], Stanislav PAULOVIČ[1]

[1]Slovak University of Agriculture in Nitra, Faculty of Engineering, Institute of Electrical Engineering, Automation, Informatics and Physics, Nitra, Slovakia, xholy@uniag.sk (M. H.), ladislav.toth1@uniag.sk (L. T.), lukas.vacho@uniag.sk (L. V.), martin.olejar@uniag.sk (M. O.), stanislav.paulovic@uniag.sk (S. P.),
[2]Slovak University of Agriculture in Nitra, Faculty of Engineering, Institute of Design and Engineering Technologies, Nitra, Slovakia, patrik.kosa@uniag.sk (P. K.), juraj.balazi@uniag.sk (J. B.)

*correspondence: vladimir.cviklovic@uniag.sk

In automated tomato harvesting, it is essential to distinguish between the main stem of the plant and the stem with the fruit, which was the aim of this research. We hypothesised that instance segmentation can recognise the main stem and the stem with the fruit in tomato images, even with unbalanced data, with a precision better than 0.7 and an inference speed better than 3 fps. The tomato variety Merlice was used. From the measurement, 399 images were extracted concerning three conditions. The training, testing, and validating sets were randomly divided in an 80 : 10 : 10 ratio, with 10% of the training set allocated as the validation set for cross-validation during model training. We propose to use the YOLO-v8 model for recognition. With image size 800 × 800 pixels and model YOLO-v8m-seg, precision was 0.736, recall 0.738, mAP50 0.712 and mAP50-95 0.301. Inference speed was 6.34 fps. When using the YOLO-v8l-seg model, precision achieved was 0.761, recall 0.673, mAP50 0.689 and mAP50-95 0.308. Interference speed was 3.92 fps. Using integrated GPU, the inference can reach the speeds of up to 10 fps. Experimental results demonstrate usable overall recognition performance for unbalanced samples. The findings of this work provide a technical foundation for developing tomato harvesting robots.

Keywords: computer vision; instance segmentation; deep learning; image processing; stem detection; plant phenotyping

Fully automated harvesting of tomatoes in greenhouses has many positive effects. At present, when labour is a problem in the agro-industry, it provides an alternative with the potential for high economic efficiency (Arad et al., 2020; McAllister et al., 2019). Vision systems are an integral part of an agricultural robot. Their task is to recognise and locate plant organs. Plant stem recognition is of great importance with high application value in automatic fertilisation, spraying (Wang et al., 2012), and harvesting fruit (Ma et al., 2021; Sun et al., 2021).

To control the robotic arm, it is essential to know information about the position of fruits and their maturity. Tomato bushes are harvested by cutting off the bush of plants at junction with the main stem. Many works have been published (Xiao et al., 2023; Li et al., 2021; Liu et al., 2024) using the machine vision technology to detect the ripeness status.

In 2015, the YOLO (You Only Look Once) real-time object detection system was published and has rapidly grown in the number of iterations with the latest version YOLO-v11 in 2024 (Ali and Zhang, 2024). YOLO achieves high detection accuracy and inference time with a single-stage detector (Vijayakumar and Vairavasundaram, 2024). Many applications readily adopt the versions of YOLO due to their high inference speed (Ju and Cai, 2023; Solimani et al., 2024; Song et al., 2024; Yue et al., 2023). Also, for these reasons, individual plant stems identification is currently partially handled using YOLO models (He et al., 2022; Li et al., 2025; Lyu et al., 2023; Li et al., 2023; Wang et al., 2023).

Li et al. (2025) addressed the detection of ripe tomato fruits and stems in their work. They solved the problem of poor recognition due to occlusion, overlapping, sample imbalance, and large differences in the extent of fruit stems. The loss function in the YOLOX model was replaced by the GIoU loss function due to improved accuracy. In the experiment, the mAP of the YOLOX-SE-GIoU model reached 92.17%. The incorporation of the SE module improved the accuracy of unbalanced sample recognition. However, the results are not applicable in practice because it requires further research to optimise the model complexity and the efficiency of algorithms to ensure integration with robotic harvesting systems. Xiang et al. (2022) compared DEM (Deep Embedding Model) and the cascaded neural network with Mask R-CNN (Regions with Convolutional Neural Networks) in recognising tomato plant stems. The average precision, recall, and F1-score for the three models – DEM, Mask R-CNN, and the cascaded neural network combining DEM and Mask R-CNN – were as follows:

- DEM: 33.0% precision, 59.0% recall, and 42.3% F1-score;
- Mask R-CNN: 66.2% precision, 50.0% recall, and 57.0% F1-score;
- Cascaded DEM + Mask R-CNN: 60.2% precision, 50.5% recall, and 55.2% F1-score.

These results showed that the recall rate of DEM was better than that of Mask R-CNN, and the average precision rate of Mask R-CNN was better than that of DEM. DEM performs well in the detection of thin and long objects, even if objects have similar colours to their backgrounds. The cascaded neural network has better performance than that of DEM after combining the advantages of DEM and Mask R-CNN, but its performance is still poorer than that of Mask R-CNN due to its less pixel-level recall rate (Xiang et al., 2022). Wang et al. (2024) used a point cloud model of tomato plants at different growth days with a proposed method of stem and leaf segmentation with the measurement of phenotypic parameters. The average accuracy, average recall, and average F1 score of stem and leaf segmentation were 0.88, 0.80, and 0.84, respectively, compared with the manual segmentation results. Compared with the segmentation methods based on the skeleton, the normal differential difference method, the regional growth method, and the segmentation method based on concavity, the stem and leaf segmentation was successful (Wang et al., 2024).

Most of the above studies focused on the detection of objects without differentiating the main stem from the side stem with fruit. To solve these problems, this experiment proposes the use of the YOLO-v8m-seg and YOLO-v8l-seg models, which have the advantage of detecting objects with low computational power requirements. YOLO-v8m-seg represents a medium size neural network with 78.9 GFLOPs, and YOLO-v8l-seg is a large size neural network with 165.2 GFLOPs (Ultralytics, 2024).

Based on previous research, we propose the following hypothesis. Instance segmentation can recognise the main stem and stem with the fruit in tomato images, even with unbalanced data. We expect the model to achieve a precision better than 0.7 and an inference speed better than 3 fps. This achieves the ability of the system to operate on platforms with lower computational power in real time.

Material and methods

The tomato image dataset used in this experiment was collected from September to December 2024 at Oremus Farm, a tomato grower, Podhájska, Slovakia. The tomato variety Merlice was used. The image data was collected manually by a human, using Xiaomi Redmi Note 9 Pro with a resolution of 3472 × 4640. The standard resolution of a smartphone camera was used; however, lower resolutions down to 800 × 800 pixels can also be utilised without compromising the model's performance. To ensure the diversity of fruit stem identification in complex environments and the authenticity of the planting scene, different time periods (8:00 am to 6:00 pm) and different shooting angles, different light, different degrees of shading, and different distances from plants were used. A total of 10,000 images were created. The image samples for tomato plants in different scenes are shown in Fig. 1.

Fig. 1 Selected image samples of the tomato image dataset
(a) Image of tomato plants in the morning; (b) Image of tomato plants with large shadows; (c) Image of tomato plants under direct sunlight; (d) Image of tomato plants in the middle of the day; (e) Image of tomato plants in the evening; (f) Image of tomato plants in the evening with shadows

The following rules have been set for images:

1. the image must contain at least one main stem plus secondary stem pair;
2. the point where the main and secondary stems are separated must not be overlapped;
3. the image has met the quality characteristics – it is not blurred or otherwise distorted.

From the measurement, 905 images satisfying condition 3 were extracted, while 399 images satisfied conditions 1 and 2 (i.e. 319 : 40 : 40 images). The training, testing, and validating sets were randomly divided in an 80 : 10 : 10 ratio, with 10% of the training set allocated as the validation set for cross-validation during model training. The 80 : 10 : 10 split is a commonly used standard that ensures sufficient data for training the model (Nazarkar et al., 2023). The training set consisted of 319 images, the test set contained 40 images, and the validation set contained 40 images.

The experiment is complemented by five-fold cross-validation to reduce the influence of random data distribution on the results.

Given the dataset size, we applied data augmentation techniques, specifically changes in brightness, saturation, image scaling, flipping (left to right), and mosaic (combining four images into one). These techniques help to improve model generalisation and enhance robustness to varying image conditions (Xu, 2023).

In the early stages of the research, we distinguished 3 classes – main stem, side stem with fruit and tomato. Experiments on a small number of images showed several problems with the tomato class. It created a disproportion to the other classes and impaired the detection of the other classes. Therefore, we decided to focus exclusively on the main stem and side stem with fruit classes.

Due to strict image selection rules, the dataset was relatively balanced, i.e. 885 instances of main stem and 699 instances of side stem with fruit. This ensured that both classes had sufficient representation for effective model training and reduced the risk of bias toward the dominant class. We manually annotated the dataset composed of 399 images (Fig. 2). We used Semantic Segmentation with Polygons to accurately label relevant objects. The method was chosen due to the need for the most accurate pixel-level segmentation. Instance segmentation is a promising deep learning technique that builds upon object detection to perform the pixel-level segmentation of individual instances, effectively tackling pattern recognition challenges (Charisis and Argyropoulos, 2024).

YOLO is a single-stage object detection network designed for high-speed, real-time detection. YOLOv8 (Jocher et al., 2023) improves on its predecessor with several innovative features and performance enhancements. In the backbone, YOLOv8 replaces the C3 and SPP modules from YOLOv5 (Jocher, 2020) with the C2f and SPPF modules, respectively. The C2f module optimises convolution and feature fusion, reducing computational layers and improving processing speed, while the SPPF module simplifies multi-scale pooling with a single pooling layer, enhancing feature extraction. YOLOv8's head structure (Lin et al., 2017) combines top-down semantic information with bottom-up detail, further improving feature fusion (Wang et al., 2025). For main and side stems with fruit detection, YOLO-v8m-seg and YOLO-v8l-seg are ideal due to their multi-task inference capability and instance segmentation within a single model. These models were selected due to their demonstrated advantages in balancing detection accuracy, computational efficiency, and integrated instance segmentation capabilities, making them particularly well-suited for complex tasks such as main and side stem detection with associated fruit instances.

The architecture of the YOLOv8 model consists of three main components: the backbone for feature extraction, the neck for feature fusion, and a decoupled head, as shown in Fig. 3.

For the backbone, a modified convolutional neural network CSPDarknet53 is used. The neck utilises C2f blocks. There are three detection heads, each focusing on a different object size. Predictions are aggregated to obtain the final output.

Conv block performs convolution operations followed by batch normalisation and SiLU activation function. C2f block enhances the model's ability to retain important gradient details. It consists of 2 Conv blocks and DarknetBottleneck blocks. The Spatial Pyramid Pooling Fast (SPPF) module converts varying-size feature maps into feature vectors with fixed size.

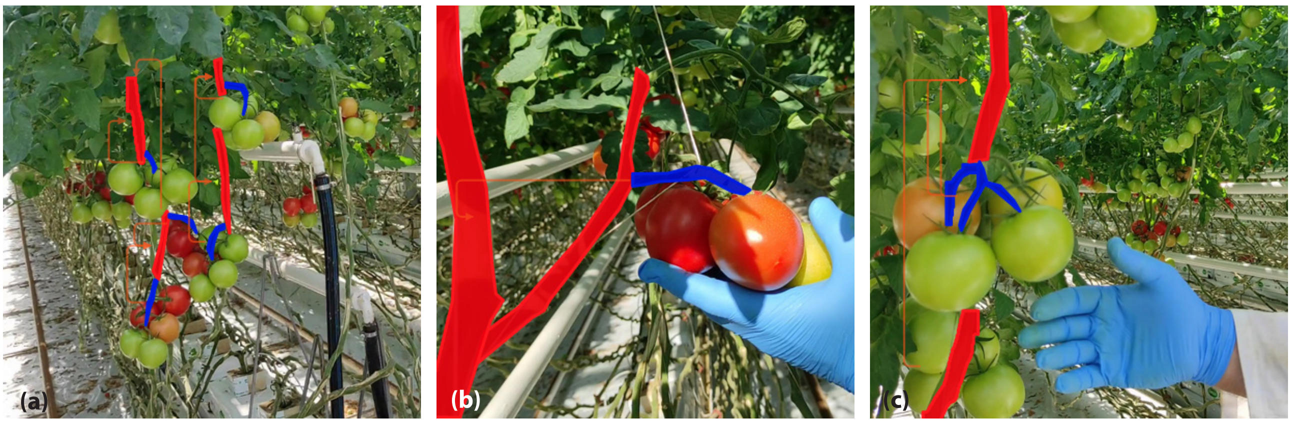

Fig. 2 Selected image samples with labels
(red – main stem; blue – side stem with fruit) of the tomato image dataset: (a) A long shot with multiple stem instances; (b) A close-up shot; (c) A side stem with a more complex branching pattern

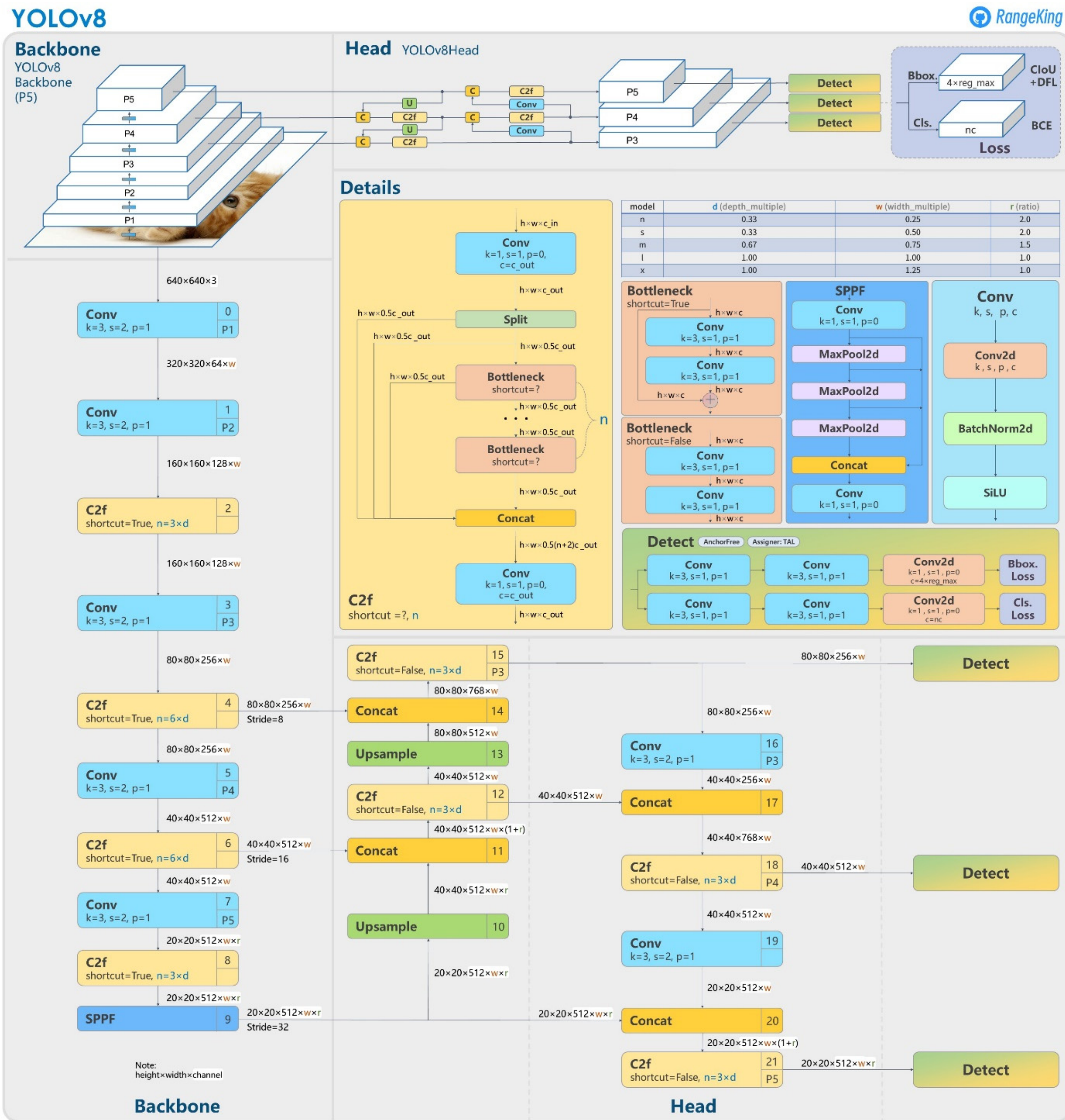

Fig. 3 Detailed illustration of YOLOv8 model architecture
Source: github.com, 2023

The model comes in several variants: *n, s, m, l* and *xl*. As the model size increases, so does its computational demand. All variants are available pre-trained on the COCO (Common Objects in Context) dataset, offering significant benefits, especially for small datasets. With a pre-trained model, the network has already learned to extract useful low-level features, such as edges and textures, as well as high-level features like object parts. This knowledge is transferred, and during training, weights are fine-tuned to our specific problem. The main modification is in the final layer to match the class count.

In Conv block, a SiLU activation function (Eq. 2) is used. This stands for Sigmoid Linear Unit, also known as Swish. SiLU allows for smooth gradients, which avoids issues with vanishing gradients that pose an obstacle to training deep neural networks (Yang et al., 2024). This issue typically arises when using the traditional sigmoid activation function (see Eq. 1).

$$sigmoid\,(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$silu\,(x) = sigmoid\,(x) = x\frac{1}{1+e^{-x}} \tag{2}$$

where: x – logit real number from R (R is the set of real numbers)

The loss function used is Binary Cross Entropy, which is calculated pixel-wise in the case of segmentation task. The loss is normalised by object area to prevent large objects from dominating the loss. Single loss is defined as:

$$\mathrm{l}(y, p) = -\left[y \cdot \log\,(p) + (1 - y) \cdot \log\,(1 - p)\right] \tag{3}$$

where: y – ground truth label (0 or 1); p – the predicted probability of model. This is calculated as $\sigma(z)$ (sigmoid function), z – raw logit real number from R

To address class imbalance, weighted loss can be used instead. The total loss is calculated by averaging the sum of losses over several instances. The comparison of activation functions is shown in Fig. 4.

The training and evaluation were conducted on different hardware configurations to assess both training efficiency and real-world deployment feasibility. Training and fine-tuning were performed on a machine with a Windows 10 operating system, AMD Ryzen 7 7800X3D CPU, and NVIDIA RTX 4090 GPU. For testing and validation, Intel NUC BOXNUC8i7HNK2 with Windows 10, Intel Core i7-8705G CPU, and Radeon RX Vega M GPU were used, which represents an edge-computing environment.

Results and discussion

To identify the optimal parameters for model training, we performed hyperparameter tuning, focusing on those outlined in the Material and Methods section, along with learning rate and weight decay. Each training round involved running a YOLO-v8l-seg model for 30 epochs with an image size of 800 px, a batch size of 8, and the AdamW optimiser. The best results were achieved at iteration 208, where the model reached its highest mAP, prompting an early stop after 270 iterations.

The final training parameters were 1000 training epochs, image size 800 px, batch size 8, and hyperparameters obtained during hyperparameter tuning. To prevent overfitting, an early stopping criterion was set for 150 epochs based on prior experimentation. The training concluded at epoch 413, meaning the best model was observed at epoch

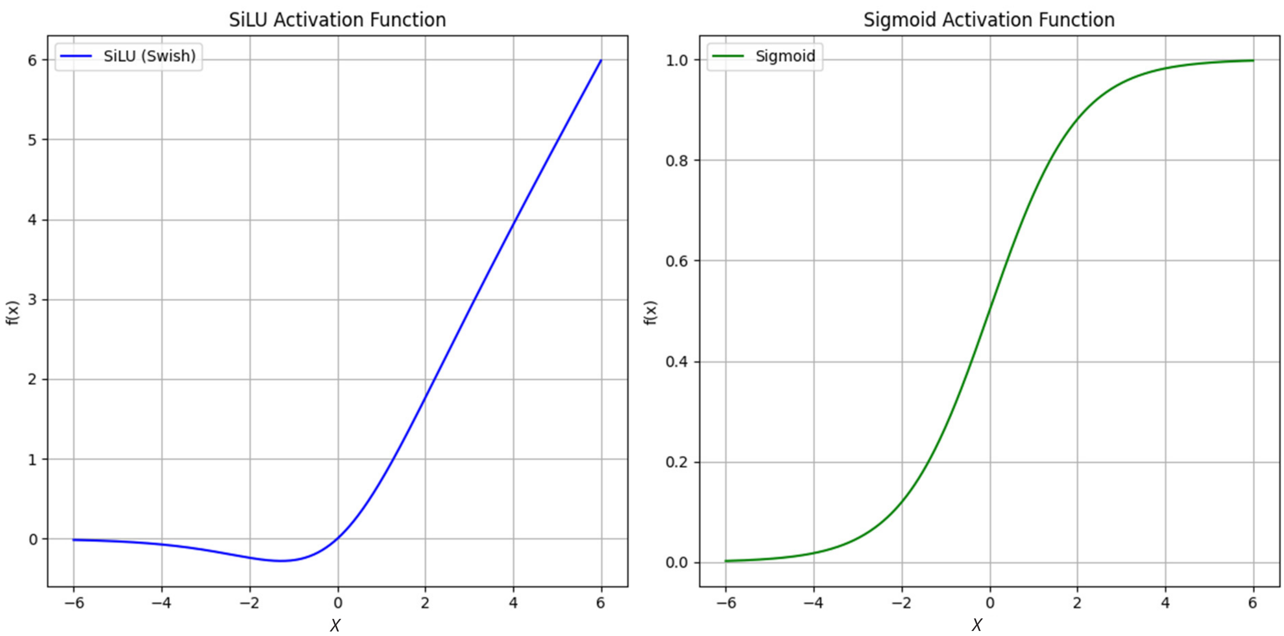

Fig. 4 Comparison of activation functions

263. The same model was used for subsequent validation. The sample outputs of this model are shown in Fig 5.

During validation, the best model was validated against an unseen test dataset. We considered two models with two image sizes (640 and 800 px) to balance model accuracy and computational efficiency. The larger model achieved a precision of 0.761, recall 0.673, mAP50 0.689, and mAP50-95 0.308. As seen in the confusion matrix shown in Fig. 6, a notable success was no confusion of the main stem and side stem. The confusion between the two classes and background suggests that an advanced preprocessing technique dealing with image background might improve the results. Upon manual inspection, it was found that there were numerous instances where the model correctly labelled the main stems; however, these were not part of the labelled dataset as they were isolated (without side stems with fruit). The smaller model achieved a precision of 0.736, recall 0.738, mAP50 0.712, and mAP50-95 0.301. Compared to the larger model, there are improvements in recall and mAP50; however, mAP50-95 and precision are worse. As seen in Fig. 7a, b, the model has a better performance on the main stem class.

Another part of the experiment was 5-fold cross-validation. This resulted in a mean value of mAP50-95 0.284 (mAP50 0.647) with a standard deviation of std = 0.0287 (0.0394). These results indicate robust model performance

Fig. 5 Examples of successful instance segmentation
The “SideStem” label represents the actual class “side stem with fruit” for brevity. (a) Image of tomato plants in the morning; (b) Image of tomato plants with large shadows; (c) Image of tomato plants under direct sunlight; (d) Image of tomato plants in the middle of the day; (e) Image of tomato plants in the evening; (f) Image of tomato plants in the evening with shadows

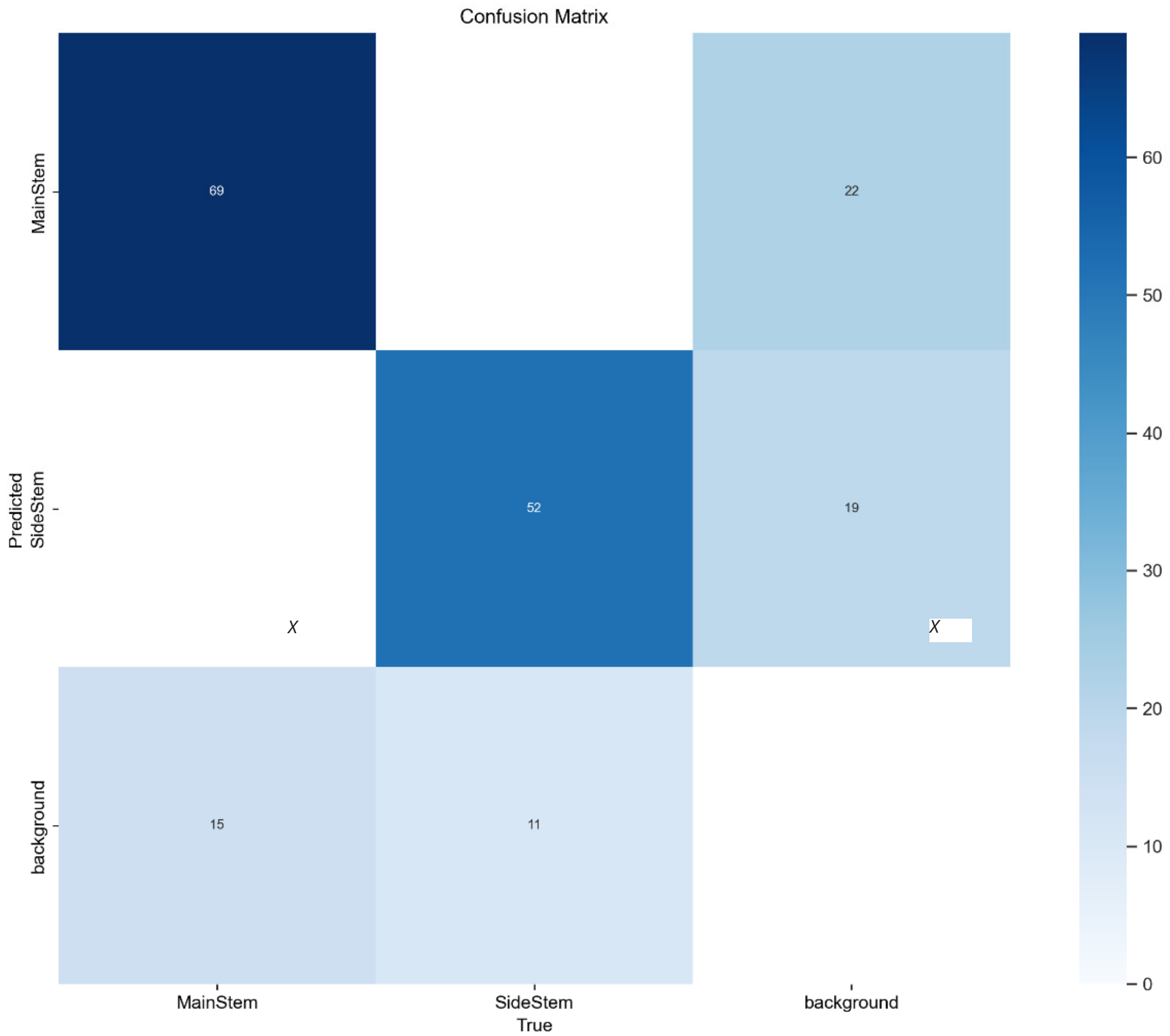

Fig. 6 Confusion matrix from the validation on the test dataset using the YOLOv8l-seg model with image size 800
The "SideStem" label represents the actual class "side stems with fruit" for brevity

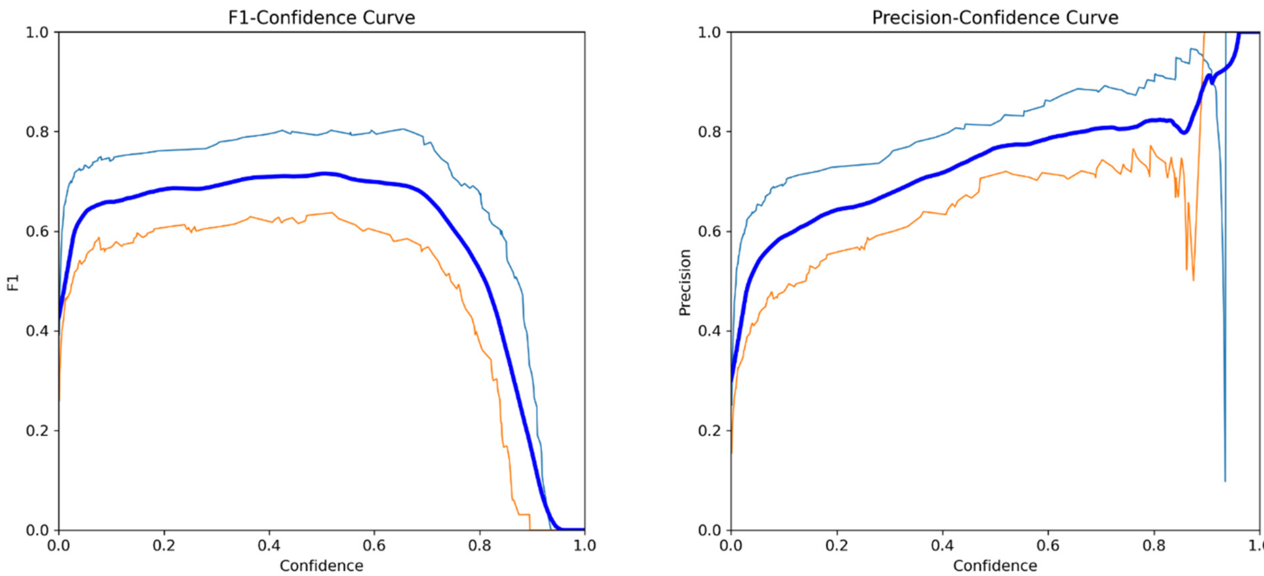

Fig. 7a Detailed illustration of the validation on the test dataset using the YOLO-v8l-seg model with image size 800
blue – main stem; orange – side stems with fruit; dark blue – two classes average

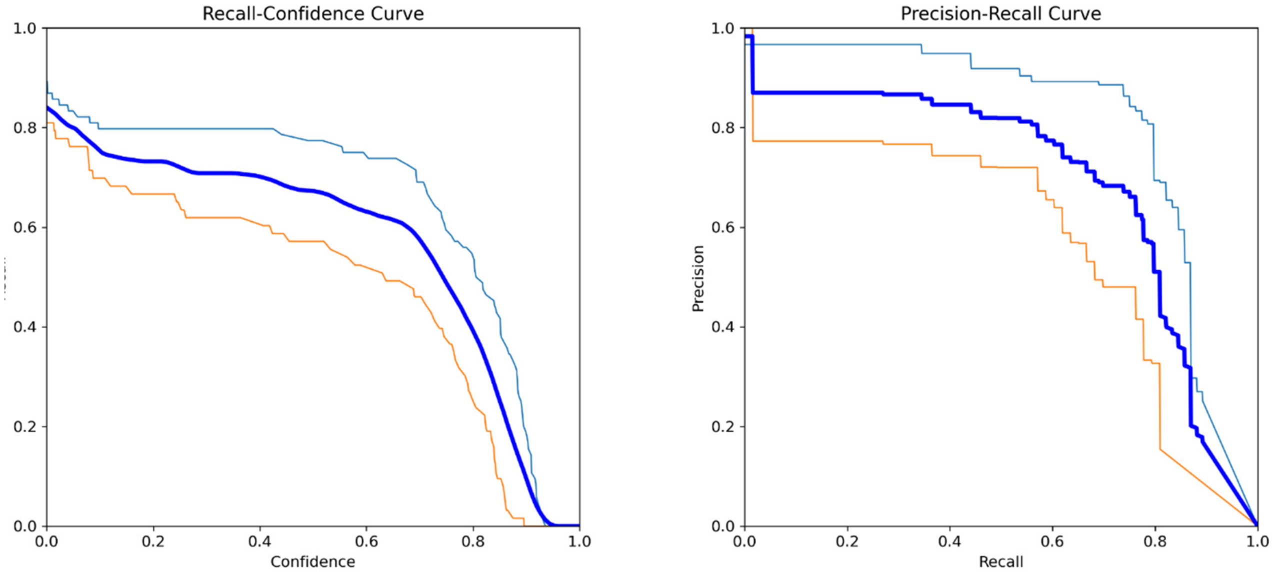

Fig. 7b Detailed illustration of the validation on the test dataset using the YOLO-v8l-seg model with image size 800
blue – main stem; orange – side stems with fruit; dark blue – two classes average

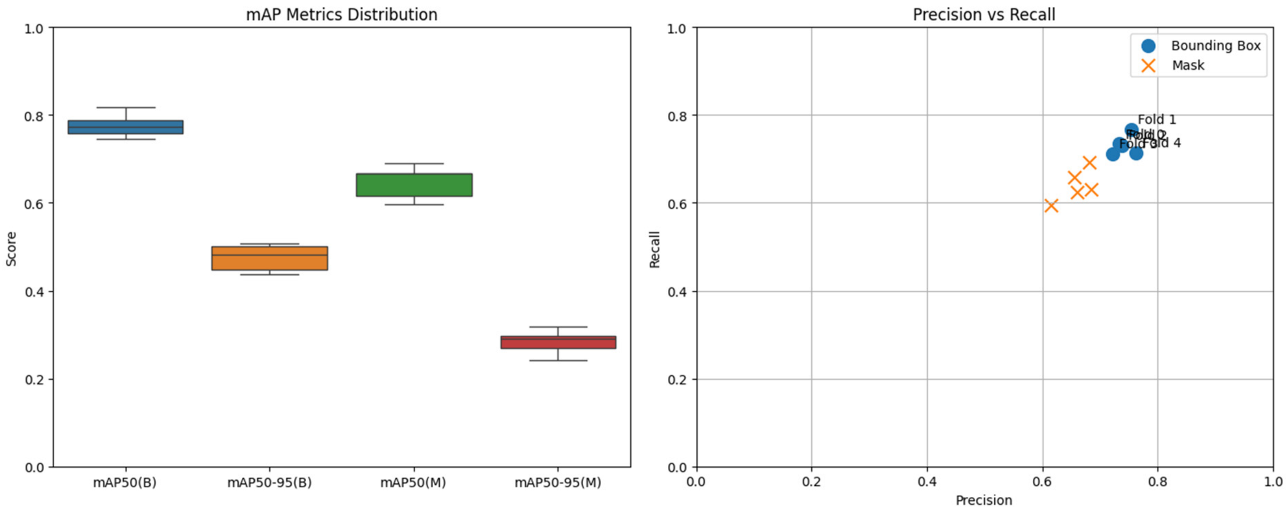

Fig. 8 K-fold validation results with five folds
blue – main stem; orange – side stems with fruit; dark blue – two classes average

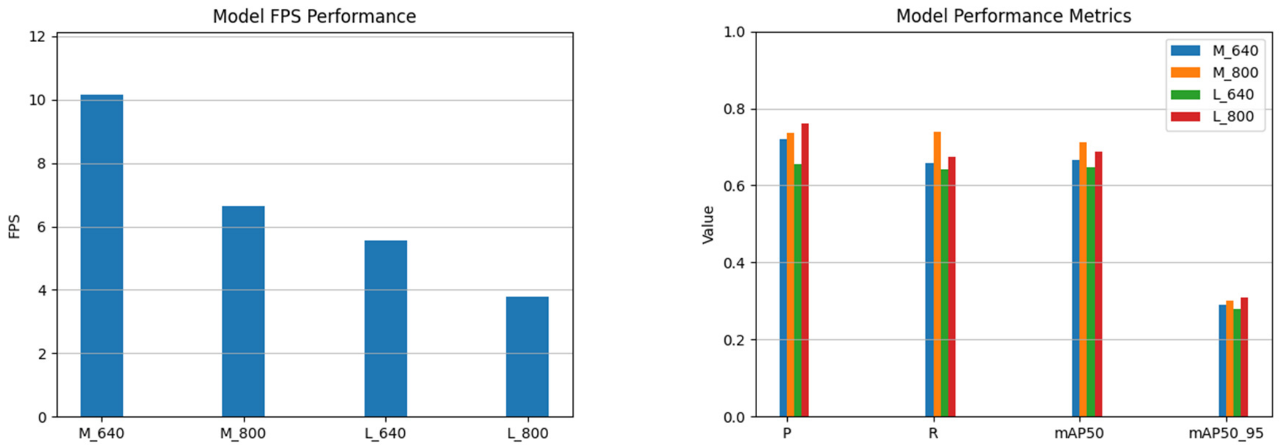

Fig. 9 Results across all the model variants on Intel NUC using the integrated GPU
(a) Measured in frames per second; (b) P-precision, R-recall

with small variability between individual samples. It confirms that the dataset splits were not biased towards achieving artificially high performance. All five folds are visualised in Fig 8.

Regarding real-time performance, CPU inference is on average 651 ms per image using openvino int8 and 1035 ms per image using native PyTorch models for the larger model. With the smaller model being approximately twice as fast, this is still not a satisfactory result. The training experiments were conducted using Python version 3.13.1 and PyTorch version 2.6.0. For inference testing on an Intel NUC device, ONNX version 1.19 and OpenVINO toolkit version 2025.1.0 were employed.

However, using the integrated GPU instead, the inference can reach the speeds of up to 10 fps, depending on the model, which is significantly faster than CPU-based processing. A comparison is shown in Fig. 9.

Conclusion

The use of the YOLO-v8m-seg and YOLO-v8l-seg models effectively addresses the challenge of differentiating the main stem from the side stems with fruit. By leveraging these models, the system can segment the instances of both classes efficiently while requiring low computational power, making it suitable for real-time operation on platforms with lower processing capabilities. With an image size of 800×800 pixels on Intel NUC, we achieved the following results:

- The YOLO-v8m-seg model achieved precision, recall, mAP50 and mAP50-95 of 0.736, 0.738, 0.712, and 0.301 respectively. The inference speed is 6.34 fps.
- The YOLO-v8l-seg model achieved precision, recall, mAP50 and mAP50-95 of 0.761, 0.673, 0.689, and 0.308 respectively. The inference speed is 3.92 fps.

The stated hypothesis was confirmed, i.e., it is possible to recognise the main and secondary stem of tomato from image information with a precision better than 0.7 and an inference speed better than 3 fps. In future work, research will be focused on dataset expansion and preprocessing enhancements. Adding more images can further refine segmentation quality, and the trained model can assist in annotating new data. Implementing background removal techniques could mitigate misclassifications.

Experimental results demonstrate usable overall recognition performance for unbalanced samples. The findings of this work provide a technical foundation for developing tomato harvesting robots. Low computing power requirements, usable accuracy and processing speed on low-power platforms create conditions for real-world greenhouse applications.

Acknowledgements

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

References

ALI, M. L. – ZHANG, Z. 2024. The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. In Computers, vol. 13, no. 12, article no. 336. DOI: https://doi.org/10.3390/computers13120336

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. In Journal of Field Robotics, vol. 37, no. 6, pp. 1027–1039. DOI: https://doi.org/10.1002/rob.21937

GITHUB.COM. 2023. Brief summary of YOLOv8 model structure. Available at: https://github.com/ultralytics/ultralytics/issues/189

HE, B. – ZHANG, Y. – GONG, J. – FU, G. – ZHAO, Y. – WU, R. 2022. Fast recognition of tomato fruit in greenhouse at night based on improved YOLO v5. In Transactions of the Chinese Society for Agricultural Machinery, vol. 53, no. 5, pp. 201–208. DOI: https://doi.org/10.6041/j.issn.1000-1298.2022.05.020

CHARISIS, C. – ARGYROPOULOS, D. 2024. Deep learning-based instance segmentation architectures in agriculture: A review of the scopes and challenges. In Smart Agricultural Technology, vol. 8, article no. 100448. DOI: https://doi.org/10.1016/j.atech.2024.100448

JOCHER, G. – CHAURASIA, A. – QIU, J. 2023. YOLO by Ultralytics. Available at: https://github.com/ultralytics/ultralytics

JOCHER, G. – CHAURASIA, A. – STOKEN, A. – BOROVEC, J. – NANOCODE012 – KWON, Y. – MICHAEL, K. – TAOXIE – FANG, J. – IMYHXY – LORNA – ZENG, Y. – WONG, C. – V, A. – MONTES,

D. – WANG, Z. – FATI, C. – NADAR, J. – LAUGHING – UNGLVKITDE – SONCK, V. – TKIANAI – YXNONG – SKALSKI, P. – HOGAN, A. – NAIR, D. – STROBEL, M. – JAIN, M. 2020. Ultralytics/YOLOv5: v7.0 – YOLOv5 SOTA Realtime Instance Segmentation. Available at: https://doi.org/10.5281/zenodo.3908559

JU, R.-Y. – CAI, W. 2023. Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. In Scientific Reports, vol. 13, article no. 20077. DOI: https://doi.org/10.1038/s41598-023-47460-7

LI, R. – JI, Z. – HU, S. – HUANG, X. – YANG, J. – LI, W. 2023. Tomato maturity recognition model based on improved YOLOv5 in greenhouse. In Agronomy, vol. 13, no. 2, article no. 603. DOI: https://doi.org/10.3390/agronomy13020603

LIN, T.-Y. – DOLLÁR, P. – GIRSHICK, R. – HE, K. – HARIHARAN, B. – BELONGIE, S. 2017 Feature pyramid networks for object detection. arXiv:1612.03144. DOI: https://doi.org/10.48550/arXiv.1612.03144

LIU, S. – XUE, J. – ZHANG, T. – LV, P. – QIN, H. – ZHAO, T. 2024. Research progress and prospect of key technologies of fruit target recognition for robotic fruit picking. In Frontiers in Plant Science, vol. 15, article no. 1423338. DOI: https://doi.org/10.3389/fpls.2024.1423338

LI, T. – SUN, M. – DING, X. – LI, Y. – ZHANG, G. – SHI, G. – LI, W. 2021. Tomato recognition method at the ripening stage based on YOLO v4 and HSV. In Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), vol. 37, no. 21, pp. 183–190. DOI: https://doi.org/10.11975/j.issn.1002-6819.2021.21.021

LI, Y. – LI, J. – LUO, L. – WANG, L. – ZHI, Q. 2025. Tomato ripeness and stem recognition based on improved YOLOX. In Scientific Reports, vol. 15, article no. 1924. DOI: https://doi.org/10.1038/s41598-024-84869-0

LYU, Z. – ZHANG, F. – WEI, X. – HUANG, Y. – LI, J. – ZHANG, Z. 2023. Synergistic recognition of tomato flowers and fruits in greenhouse using combination enhancement of YOLOX-ViT. In Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), vol. 39, no. 4, pp. 124–134. DOI: https://doi.org/10.11975/j.issn.1002-6819.202211246

MA, B. – DU, J. – WANG, L. – JIANG, H. – ZHOU, M. 2021. Automatic branch detection of jujube trees based on 3D reconstruction for dormant pruning using the deep learning-based method. In Computers and Electronics in Agriculture, vol. 190, article no. 106484. DOI: https://doi.org/10.1016/j.compag.2021.106484

McALLISTER, W. – OSIPYCHEV, D. – DAVIS, A. – CHOWDHARY, G. 2019. Agbots: Weeding a field with a team of autonomous robots. In Computers and Electronics in Agriculture, vol. 163, article no. 104827. DOI: https://doi.org/10.1016/j.compag.2019.05.036

NAZARKAR, A. – KUCHULAKANTI, H. – PAIDIMARRY, C. S. – KULKARNI, S. 2023. Impact of various data splitting ratios on the performance of machine learning models in the classification of lung cancer. In Proceedings of the Second International Conference on Emerging Trends in Engineering (ICETE 2023), Advances in Engineering Research, vol. 223, pp. 96–104. DOI: https://doi.org/10.2991/978-94-6463-252-1_12

SOLIMANI, F. – CARDELLICCHIO, A. – DIMAURO, G. – PETROZZA, A. – SUMMERER, S. – CELLINI, F. – RENÒ, V. 2024. Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity. In Computers and Electronics in Agriculture, vol. 218, article no. 108728. DOI: https://doi.org/10.1016/j.compag.2024.108728

SONG, G. – WANG, J. – MA, R. – SHI, Y. – WANG, Y. 2024. Study on the fusion of improved YOLOv8 and depth camera for bunch tomato stem picking point recognition and localization. In Frontiers in Plant Science, vol. 15, article no. 1447855. DOI: https://doi.org/10.3389/fpls.2024.1447855

SUN, Q. – CHAI, X. – ZENG, Z. – ZHOU, G. – SUN, T. 2021. Multi-level feature fusion for fruit bearing branch keypoint detection. In Computers and Electronics in Agriculture, vol. 191, article no. 106479. DOI: https://doi.org/10.1016/j.compag.2021.106479

ULTRALYTICS. 2024. YOLOv8. Explore Ultralytics YOLOv8. Supported tasks and modes. Available at: https://docs.ultralytics.com/models/yolov8/#supported-tasks-and-modes

VIJAYAKUMAR, A. – VAIRAVASUNDARAM, S. 2024. YOLO-based object detection models: A review and its applications. In Multimedia Tools and Applications, vol. 83, pp. 83535–83574. DOI: https://doi.org/10.1007/s11042-024-18872-y

WANG, C.-Y. – BOCHKOVSKIY, A. – LIAO, H.-Y. M. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7464–7475. DOI: https://doi.org/10.1109/CVPR52729.2023.00721

WANG, R. – ZHENG, X. – CHEN, Y. 2025. A vision model for automated frozen tuna processing. In Scientific Reports, vol. 15, article no. 3216. DOI: https://doi.org/10.1038/s41598-025-87339-3

WANG, Y. – LIU, Q. – YANG, J. – REN, G. – WANG, W. – ZHANG, W. – LI, F. 2024. A method for tomato plant stem and leaf segmentation and phenotypic extraction based on skeleton extraction and supervoxel clustering. In Agronomy, vol. 14, no. 1, article no. 198. DOI: https://doi.org/10.3390/agronomy14010198

WANG, X. – HAN, X. – MAO, H. 2012. Vision-based detection of tomato main stem in greenhouse with red rope. In Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), vol. 28, no. 21, pp. 135–141.

YANG, Y. – CHEN, J. – SUN, L. – ZHOU, Z. – HUANG, Z. – WU, B. 2024. Unsupervised domain-adaptive SAR ship detection based on cross-domain feature interaction and data contribution balance. In Remote Sensing, vol. 16, no. 2, article no. 420. DOI: https://doi.org/10.3390/rs16020420

YUE, X. – QI, K. – NA, X. – ZHANG, Y. – LIU, Y. – LIU, C. 2023. Improved YOLOv8-Seg network for instance segmentation of healthy and diseased tomato plants in the growth stage. In Agriculture, vol. 13, no. 8, article no. 1643. DOI: https://doi.org/10.3390/agriculture13081643

XIAO, F. – WANG, H. – LI, Y. – CAO, Y. – LV, X. – XU, G. 2023. Object detection and recognition techniques based on digital image processing and traditional machine learning for fruit and vegetable harvesting robots: An overview and review. In Agronomy, vol. 13, no. 3, article no. 639. DOI: https://doi.org/10.3390/agronomy13030639

XIANG, R. – ZHANG, M. – ZHANG, J. 2022. Recognition for stems of tomato plants at night based on a hybrid joint neural network. In Agriculture, vol. 12, no. 6, article no. 743. DOI: https://doi.org/10.3390/agriculture12060743

XU, M. – YOON, S. – FUENTES, A. – PARK, D. S. 2023. A comprehensive survey of image augmentation techniques for deep learning. In Pattern Recognition, vol. 137, article no. 109347. DOI: https://doi.org/10.1016/j.patcog.2023.109347