Edge-Driven Orchard Analytics: Automated Treetop Detection Using YOLOv11

Mihailo Ilić*†, Marko Kolarski*, Aleksandar Tomčić*†, Srđan Penjivrag*, Milenko Tošić*
* VizLore Labs Foundation, Novi Sad, Serbia
{firstname.lastname}@vizlore.com
† University of Novi Sad, Faculty of Sciences, Dept. of Mathematics and Informatics, Novi Sad, Serbia
{milic, atomic}@dmi.uns.ac.rs

***Abstract*—Smart farming represents the application of modern Information and Communication Technologies (ICT) in agriculture, and using these advances to maximize yield and optimize processes. One major agricultural challenge is that of fruit cracking, a peel disorder which occurs mainly in the pre-harvest stage and limits fruit quality and yield, and causes significant food waste. The Horizon project CrackSense aims to address this challenge by developing and upscaling smart sensing technologies, leading to efficient and high-resolution monitoring of agri-environmental conditions through the use of state-of-the-art machine learning (ML) models. This paper presents a localized edge-computing solution for automated treetop detection in citrus orchards. This is a crucial step in the overall CrackSense approach to food cracking risk assessment. We utilize the YOLOv11 (You Only Look Once) architecture, trained on an open-access dataset, to perform simultaneous object detection and instance segmentation of tree canopies. On-site deployment in remote environments with limited connectivity is achieved by implementing this detection system on specialized edge devices, capable of accommodating these models in the field. The technical pipeline integrates raw model inference with custom post-processing algorithms designed to refine canopy boundaries and extract precise treetop coordinates. This approach minimizes data transmission overhead and bandwidth costs by processing data at the source.**

***Index Terms*—Smart Farming, Edge Computing, Drones, Computer Vision, Machine Learning, Fruit Cracking**

I. Introduction

Farming challenges are one of the main drivers of edge-computing advancements. Contemporary smart farming solutions often include numerous different systems all working together towards a common goal, that being maximizing yield while optimizing other objectives as well, like fertilizer application and reduced pesticide use. The components of these systems are intertwined and deployed both in the edge and cloud, forming a closed loop. This loop starts in the field where data is collected using various sensing technologies, after which it is sent over the network to servers where it is processed, stored, and made available to farmers. Based on the information obtained from this data, farmers and Artificial Intelligence (AI) systems make decisions on actions to be conducted in the field to maximize yield.

These kinds of interconnected systems have proven to be powerful and reliable, helping farmers achieve high yields. However, an obstacle to implementing such systems is limited connectivity. Fields located in remote locations have trouble with reliable network connections, or in some cases may not even have internet access. Because of this, reliable edge-based systems still need to be developed to provide support for fields in remote locations. Part of the Horizon project CrackSense is to develop reliable edge models to support early signs of fruit cracking.

In this paper, we present our contributions to the CrackSense project[1], specifically in edge-computing for agriculture based on the You Only Look Once (YOLO) [1] architecture. The approach outlined in this paper allows for on-site treetop instance segmentation based on RGB images using specialized hardware on which these models are deployed. This type of setup brings powerful inference models to the field, eliminating the need for data transfer overhead in environments with limited connectivity.

The rest of this paper is structured as follows. Section II introduces the CrackSense project, its contribution to advancing smart farming, and technologies it uses to achieve this. A technical overview of the solution along with the object detection methodology is presented in Section III. The pilot site and results are given in Section IV, followed by concluding remarks in the last section.

II. Enabling Smart Farming Through CrackSense

The CrackSense project aims to address the problem of fruit cracking. The idea behind the project is to employ edge sensing to monitor the status of fruits and to guide farmers' actions to mitigate this problem. Fruit cracking mainly occurs in the pre-harvest stage and begins at the surface of the fruit. Extreme cases of cracking can cause the fruit to split, leading to decreased yield. This problem affects many fruits, and CrackSense focuses on citrus, table grapes, sweet cherries, and pomegranates to improve their agricultural production.

Edge-computing is a central topic in CrackSense, where it is used to enhance decision making responsiveness, improve autonomy and precision, and reduce the need for large quantities of sensory data sent to the cloud. By shifting computation to the edge, at the periphery of the network, data is processed as close as possible to its source, which in turn reduces network bandwidth and allows these systems to operate as independently as possible.

[1]https://cracksense.eu/ – The homepage of the CrackSense project.

In this paper we focus on edge data processing. Drones are deployed and fly over fruit orchards, collecting images during their flight. The collected data is then transferred to specialized edge controllers where it is processed. Treetops are segmented from the images using an algorithm that relies on a fine-tuned YOLO model.

A. Smart Farming & ICT

Smart farming represents the application of modern Information and Communication Technologies (ICT) in agriculture to maximize yield and optimize production processes. This paradigm shift, often referred to as Agriculture 4.0, is based on the seamless integration of sensing technologies and data analytics to address complex agricultural challenges, such as fruit cracking in citrus orchards. Central to this evolution is the concept of the Digital Twin (DT), which provides a virtual representation of physical assets to facilitate predictive modeling [2]. Recent studies underscore that DTs are most effective when they can assimilate data—including soil sensors and multispectral imaging based on Unmanned Aerial Vehicles (UAVs) to provide high resolution information on water stress and nutrient deficits [3].

The effectiveness of these systems is heavily dependent on the ability of disparate hardware and software components to communicate. Research indicates that interoperability serves as a fundamental enabler for DT based decision-making, allowing the integration of heterogeneous data sources into a unified analytical framework [4]. Furthermore, the development of interoperable agricultural digital twins enhanced by reinforcement learning intelligence allows more adaptive and autonomous orchard management [5]. By leveraging these ICT advances, projects like CrackSense can monitor agri-environmental conditions with high resolution, ensuring fruit quality and reducing pre-harvest waste.

B. Edge-Computing in Agriculture

Despite the capabilities of cloud-based platforms, their deployment in commercial orchards is frequently limited by the "connectivity gap" inherent in rural environments. Edge-computing addresses these constraints by shifting computational tasks from centralized servers to localized devices located on the network periphery. This architectural shift provides several critical advantages for treetop detection and orchard analytics [6].

- **Bandwidth Optimization –** Processing high-resolution imagery from UAVs or ground robots at the source minimizes data transmission overhead. By transmitting only the extracted treetop coordinates and canopy boundaries rather than raw image data, bandwidth requirements are reduced by up to 90% [7].
- **Reduced Latency –** Localized inference enables real time response times, which is essential for autonomous navigation and time sensitive disease detection.
- **Resilience in Remote Environments –** Edge devices allow continuous operation in areas with intermittent or non-existent internet connectivity, ensuring that the CrackSense approach remains robust regardless of infrastructure limitations [4].

By implementing the YOLOv11 architecture on specialized edge hardware, this study demonstrates a solution that maintains the high level intelligence of state of the art (SoTA) machine learning while adhering to the physical and connectivity constraints of the field. This localized processing is a vital component of the interoperable intelligence required for modern agricultural digital twins [5].

C. YOLO in Agriculture

You Only Look Once (YOLO) [1] is a successful and very well known family of deep learning models used for object detection and segmentation. These models process an image in a single pass and provide bounding boxes (or segmentation masks) for objects of interest located in the image. This process is very fast, especially when compared to other sliding window or region-proposal approaches.

These models have improved over the years, initially being used to detect large objects such as live stock and to output bounding boxes around them. Recent advancements like the YOLOv11 model allow for much more reliable and accurate detections, providing both segmentation masks and bounding boxes with pixel-perfect boundaries of objects of interest.

The YOLO family of models is highly reliable and has shown great performance in numerous domains. In agriculture it has also seen widespread use [8] for tasks such as maize detection [9], wheat ear detection [10], pest detection [11], monitoring cattle [12], and many more.

Related work explores the use of YOLO models on tree detection from RGB satellite images [13] and UAV images [14]. Similarly to this research, our approach uses the YOLO models as core object detection methods, and adjusts the pre- and postprocessing steps to optimize the approach for the specific object detection task. Our solution also offers a dual-stream approach, incorporating both quick single-image processing and a more comprehensive analysis of the entire orchard.

III. CrackSense Methodology & Implementation

A. Hardware Setup

Data collection was carried out by the CrackSense consortium partners using a DJI-Matrice 600 drone (DJI, Shenzhen, China). The images collected were utilized to provide cost-effective insights into tree health and its correlation with fruit cracking intensity in crops such as citrus, sweet cherries, grapes, and pomegranates. The DJI-Matrice 600 comes with an Altum Micasense multispectral sensor, a FLIR A655SC thermal camera, and a Zenmuse L2 LiDAR system. This multimodal approach enabled the simultaneous collection of different data sets necessary for a comprehensive assessment of the orchard condition.

Edge-computing is deployed as a core architectural component. The VizLore Foundation (VLF) edge controller unit has been optimized at both the hardware and software levels to facilitate high performance data acquisition and processing. Its hardware capabilities are comparable to the Raspberry Pi

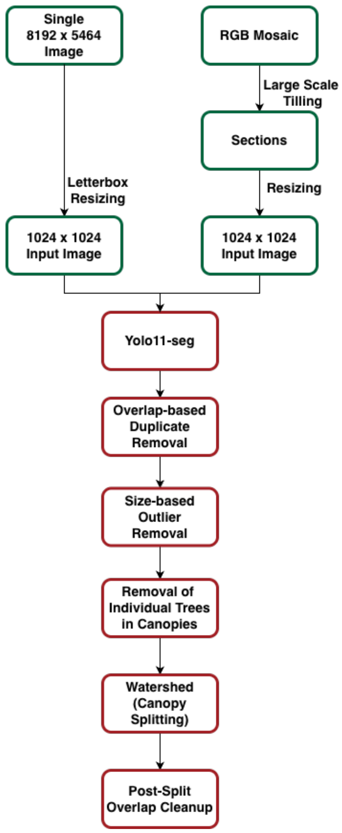


Fig. 1: The image input pipeline – Two input streams into the YOLO model are present: 1) Single image stream (left); 2) RGB Mosaic (right). Different preprocessing steps are taken based on the input source, but postprocessing steps after YOLO segmentation remain the same.

4 model B (Raspberry Pi Foundation, Cambridge, UK), which come equipped with a 1.8GHz 64-bit quad-core ARM Cortex-A72 CPU, available with 1GB, 2GB, 4GB, or 8GB LPDDR4-3200 RAM. By operating in real-time, the controller ensures minimal latency, allowing for immediate data analysis and decision-making at the network edge before transmission to central servers. The work in this paper focuses on processing only RGB images collected by the UAVs.

The controller offers multiple connectivity options to match different UAV requirements. It supports several data transfer methods, including secure WiFi, SFTP, LAN connectivity, and direct SD card insertion as a source for raw data.

B. Data Input Pipeline

The system is designed to handle high-resolution imagery provided by other consortium partners through two parallel input streams, as seen in Figure 1.

- **Single Image Processing –** Raw drone images captured at 8192×5464 px resolution are processed for rapid localized analysis. Letterbox resizing to a resolution of 1024×1024 px is the only preprocessing step in this case.
- **Full-Field Mosaic Processing –** Individual images are stitched together on the edge controller to form a single mosaic of the pilot site. The collected images are stitched together using OpenCV2 [15] based on their embedded GPS metadata. This results in one large image that cannot be processed directly, which is why it is split into sections, i.e. large scale tilling is applied. This results in 1024×1024 px tiles like in the first pipeline.

Both preprocessing pipelines feed into YOLO11n-seg.

C. Datasets

The YOLO11n-seg model with approximately 2.9M parameters was trained on the OAM-TCD dataset [16] containing high-resolution tree cover maps. This dataset contains 5072 2048×2048 px RGB images at $10\frac{cm}{px}$ resolution. All of these images were hand-labeled resulting in over $280k$ individual trees and $56k$ groups of trees. A particularly challenging problem with top-down aerial views of trees is that when they are close together, it is fairly difficult to discern where the boundaries of the trees are from only image data. Because of this, groups of trees are labeled as one instance of a "canopy" class, while individual trees are simply given the "tree" label.

UAV data was collected from a citrus field at a pilot site in Israel [17]. The data included LiDAR point clouds, RGB images, multispectral images, etc. The work presented in this paper focused only on the RGB image data collected during this time, which contain 734 raw images at 8192×5464 px resolution. The images were originally unlabeled, but a handful of them were manually labeled as part of the fine-tuning process of training the YOLO model.

D. Model Training

For our specific use case we employed YOLO11n-seg, the nano variant of the model used for instance segmentation. The training was done in two linked phases. The first phase involved base training of the model on the open-source OAM-TCD dataset [16], while the second phase involved fine-tuning the model trained in the first phase on the pilot site dataset [17] with manually created labels (18 images).

In the initial training phase, each image of dimensions 2048×2048 px was cut into four equal crops, 1024×1024 px each. This made it possible to follow the recommended scale of $10\frac{cm}{px}$. Two classes were used as prediction targets: (1) individual trees and (2) groups of trees labeled canopy.

In the first phase, the model was trained to learn basic tree features and successfully recognize them under various conditions. Training was set to 100 epochs with a batch size of 4 and strong augmentation, which included: contrast limited adaptive histogram equalization (CLAHE), lighting variance, Gaussian noise, gamma correction, hue shift, saturation adjustment, rotations, mosaic, mixup, etc. The optimizer used was AdamW with a learning rate of 10^{-3}, while the input image size was 1024×1024 px.

In the second phase, the model was fine-tuned on the locally labeled dataset with the goal of increasing performance under the specific conditions the model would encounter in a real-world environment (such as the specific orchard, flight height, and lighting). During fine-tuning, the first 10 layers of the trained model were frozen to preserve the generalized knowledge about trees acquired so far. This achieved adaptation to these specific conditions without forgetting previous knowledge. Of a total of 18 personally labeled images, two were manually selected for qualitative validation. The same strong augmentation as in the first phase was used to allow the model to extract as much information as possible from the small training set and to prevent the degradation of generalization. It was trained for 60 epochs using the AdamW optimizer with a reduced learning rate of 10^{-4}.

E. Post-Processing

The raw output of the YOLO model produced instance segmentation masks. Each detected instance has a corresponding confidence score. Some instance detections may be incorrect, which is usually correlated with a low confidence score for that particular instance. Consequently, a post-processing algorithm needs to be applied to the raw outputs to remove irregularities. All the post-processing steps can be seen in Figure 1. The process can be divided into three parts: **initial filtering and merging**, the **watershed algorithm**, and **final error filtering**.

- **Canopy Union-Find Merge –** For detections of the Canopy class, we often encountered masks that span across multiple orchard rows, resulting in overlapping detections where discarding one would actually reduce the total mask coverage. As a solution, we introduced a Union-Find Merge algorithm, which resolves this by merging all connected detections into one large entity, resulting in larger, separated sections ready for next steps. Connected detections are those with partly overlapping borders, or borders which touch at some point.
- **Non-Maximum Suppression (NMS) –** All overlapping duplicate tree detections are deleted if they exceed a specified overlap threshold, retaining only those with a higher predicted confidence score. If a tree detection overlaps with a canopy detection, it is absorbed into the canopy.
- **Statistical Tree Noise Elimination –** A dynamic minimum area threshold was applied based on the mean value of all other Tree class detections, where any detection smaller than this calculated threshold is removed as noise.
- **Separation of Canopy Class into Multiple Tree Detections via Watershed Algorithm –** To correctly separate Canopy detections into individual Tree detections, a Marker-controlled Watershed algorithm was implemented using the OpenCV2 library [15] specifically for speed and execution on resource-constrained controllers like the VLF edge controller. A Euclidean distance transform is applied to binary masks to generate a topographic map, where local maxima (peaks) serve as points from which boundaries between merged canopies are reconstructed via Voronoi propagation.
- **Post-Watershed Tree Duplicate Filtering –** As a last step, NMS is applied again, ensuring the correctness of the results even after the watershed algorithm has been executed.

IV. Implementation & Discussion

A. Pilot Site

The citrus dataset collected as part of the CrackSense project and used in this study was collected from the Kfar Habad pilot site over 11 aerial imaging campaigns.

All flight campaigns were conducted at an altitude of 50 meters above the ground. Spatial resolution was $5.5\frac{cm}{px}$ for multispectral data and $7.5\frac{cm}{px}$ for thermal imagery. LiDAR achieved a density of $800 - 900\frac{points}{m^2}$. To ensure optimal lighting, missions were scheduled between 11:00 and 13:00 local time, within two and a half hours of solar noon. The flight paths were meticulously planned with an 80% forward and side overlap.

During flight, the drone's camera captured a dataset of 734 raw images at 8192×5464 px resolution. The imagery was written directly to the drone's SD card as the primary storage medium. This was a reliable capture process under bandwidth limited field conditions.

Edge preprocessing was performed to prepare the images for YOLOv11 machine-learning detection. A special service on the edge unit was developed to optimize the raw images for efficient computation and runs YOLOv11 inference locally. The flow stores the processed data and ML results at the end along with the original dataset.

B. Qualitative Results

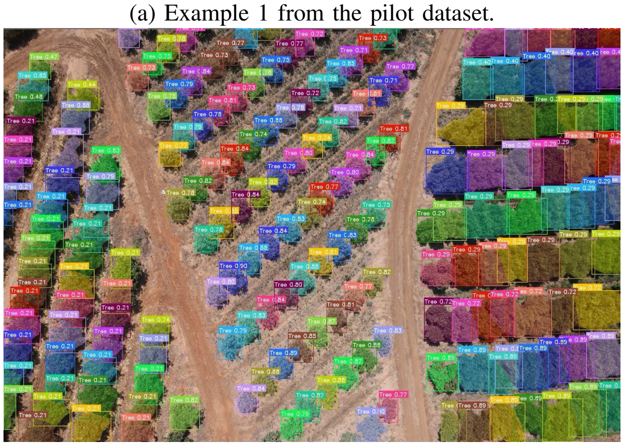
The image processing pipeline shows good results on both the raw images collected by the drones, and on the mosaic image constructed by stitching together the original drone RGB images.

Figure 2 shows three examples of raw drone images, which were fed through the image processing pipeline. The output contains segmented treetops, all colored differently to make them easily distinguishable. As seen in the figure, the overlap of treetops is quite high in some cases, which makes the segmentation problem fairly difficult. However, the model performs well, being able to correctly segment the trees in most cases, with only a few exceptions.

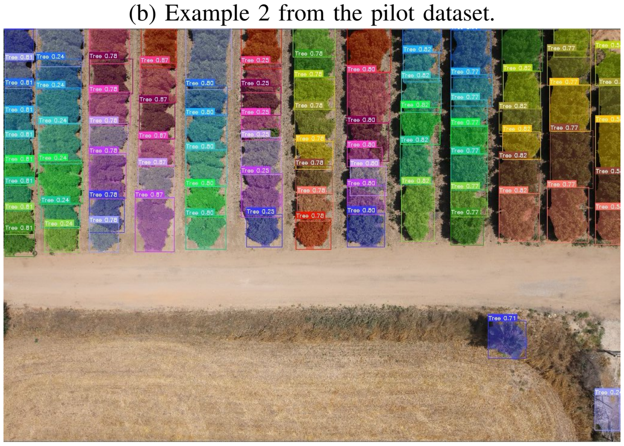
In order to obtain the total number of trees in the entire orchard, the images are stitched together on the edge controller to form a single mosaic of the orchard. This mosaic is then split into tiles according to the pipeline in Figure 1, and processed tile-by-tile by the edge unit. The performance on the mosaic tiles can be seen in Figure 3. The white segments in the images represent either missing parts of the orchard as part of the stitching process, or the edges of the orchard which belongs to the pilot.

The system outputs reliable results even though it utilizes the nano version of the YOLOv11 model, and runs with limited resources at the edge. Since it runs autonomously at

(a) Example 1 from the pilot dataset.

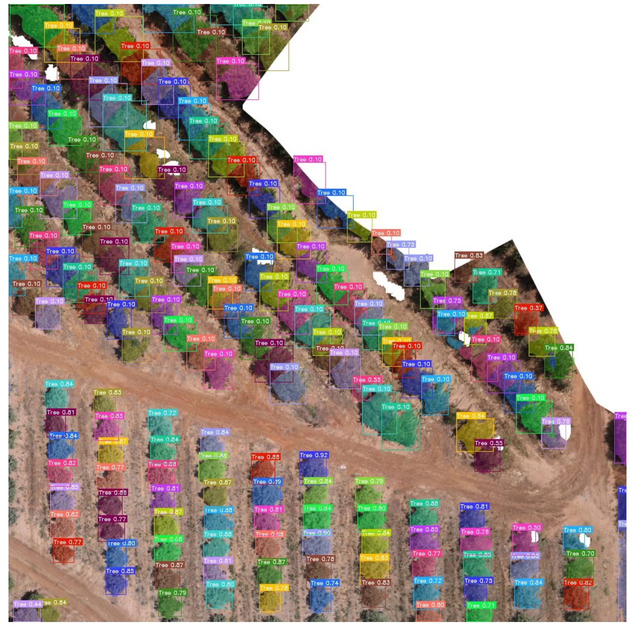


(b) Example 2 from the pilot dataset.



(c) Example 3 from the pilot dataset.

Fig. 2: Treetop instance segmentation output examples from the Israeli pilot site on single image examples.



(a) Mosaic tile example 1.

(b) Mosaic tile example 2.

Fig. 3: Instance segmentation examples from the full-field mosaic. The white sections in the images are parts of the mosaic which extend beyond the border of the pilot's orchard.

the periphery of the network, it is crucial for time-sensitive monitoring.

V. Conclusion

This paper demonstrates a localized edge-computing solution for treetop segmentation based on the YOLOv11 architecture for use in citrus orchards. The developed system is a vital step in the CrackSense methodology for assessing fruit cracking risk, thus helping reduce pre-harvest waste and ensure food quality.

The implementation outlined in this work successfully overcomes the connectivity gap found in rural areas. By processing at the edge, reliance on stable network connections is reduced, and smart farming solutions are therefore more robust and autonomous. These kinds of advancements are crucial for modern autonomous orchard management.

Future work includes extending the specialized edge hardware to accommodate software aimed at utilizing other types of data collected by CrackSense UAVs, such as LiDAR and multispectral imaging.

ACKNOWLEDGMENT

This research was supported by the CrackSense project. The CrackSense project has received funding from the European Union's Horizon research and innovation programme under grant agreement No. 101086300.

REFERENCES

[1] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[2] M. Awais, X. Wang, S. Hussain, F. Aziz, and M. Q. Mahmood, "Advancing precision agriculture through digital twins and smart farming technologies: a review," *AgriEngineering*, vol. 7, no. 5, p. 137, 2025.

[3] D. Popescu, L. Ichim, and F. Stoican, "Orchard monitoring based on unmanned aerial vehicles and image processing by artificial neural networks: a systematic review," *Frontiers in Plant Science*, vol. 14, p. 1237695, 2023.

[4] M. Ilić, M. Tošić, and A. Tomčić, "Interoperability as an enabler for digital twin based decision making in smart agriculture," in *2025 MIPRO 48th ICT and Electronics Convention*. IEEE, 2025, pp. 1875–1880.

[5] M. Kallenberg, H. Baja, M. Ilić, A. Tomčić, M. Tošić, and I. N. Athanasiadis, "Interoperable agricultural digital twins with reinforcement learning intelligence," *Smart Agricultural Technology*, p. 101412, 2025.

[6] R. Gong, H. Zhang, G. Li, and J. He, "Edge computing-enabled smart agriculture: technical architectures, practical evolution, and bottleneck breakthroughs," *Sensors*, vol. 25, no. 17, p. 5302, 2025.

[7] A. H. Shnain, Z. Abed, and E. Annapoorna, "Edge computing for real-time climate data analysis in smart farming," in *SHS Web of Conferences*, vol. 216. EDP Sciences, 2025, p. 01022.

[8] M. A. R. Alif and M. Hussain, "Yolov1 to yolov10: A comprehensive review of YOLO variants and their application in the agricultural domain," *CoRR*, vol. abs/2406.10139, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.10139

[9] A. J. Moshayedi, Z. Wang, M. Sharifdoust, A. S. Khoojine, W. Zhang, A. Kolahdooz, and J. Hu, "Smart farming solutions: A user-friendly GUI for maize tassel estimation using YOLO with dynamic and fixed labelling, featuring video support," *IEEE Access*, vol. 13, pp. 57 809–57 833, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3554984

[10] Ž. Grbović, M. Panić, D. Stefanović, S. Brdar, V. Crnojević, and T. Lončar-Turukalo, "Transfer learning models for wheat ear detection on multi-source dataset," *Scientific Reports*, vol. 15, no. 1, p. 44740, 2025.

[11] A. Hakim, A. K. Srivastava, A. Hamza, M. Owais, M. Habib-ur Rahman, S. Qadri, M. A. Qayyum, F. Z. Ahmad Khan, M. T. Mahmood, and T. Gaiser, "Yolo-pest: an optimized yolov8x for detection of small insect pests using smart traps," *Scientific Reports*, vol. 15, no. 1, p. 14029, 2025.

[12] K. Jeong, D.-R. Kim, J.-H. Ryu, H.-W. Kim, J. Cho, E. Lee, and J.-H. Jeong, "A monitoring system for cattle behavior detection using yolo-v8 in iot environments," in *2024 IEEE International Conference on Consumer Electronics (ICCE)*, 2024, pp. 1–4.

[13] I. Abbas and R. Damaševičius, "Tree detection in rgb satellite imagery using yolo-based deep learning models." *Computers, Materials & Continua*, vol. 85, no. 1, 2025.

[14] Y. Chen, H. Xu, X. Zhang, P. Gao, Z. Xu, and X. Huang, "An object detection method for bayberry trees based on an improved yolo algorithm," *International journal of digital earth*, vol. 16, no. 1, pp. 781–805, 2023.

[15] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[16] J. Veitch-Michaelis, A. Cottam, D. Schweizer, E. N. Broadbent, D. Dao, C. Zhang, A. M. A. Zambrano, and S. Max, "OAM-TCD: A globally diverse dataset of high-resolution tree cover maps," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/58efdd77196fa8159062afa0408245da-Abstract-Datasets_and_Benchmarks_Track.html

[17] M. Cohen Laor, A. Sadka, and T. Paz-Kagan, "D3.4: Upscaling sensing tools i," CrackSense Project, Horizon Europe Grant Agreement No. 101086300, Project Deliverable, January 2025, work Package 3. Contributors: I. Ginzberg, U. Hochberg, A. Lichter, G. Nikolakopoulou, L. Julhia, J. Quero Garcia.