Vision-Guided Agricultural Robot for Crop Detection Using Edge Computing on Embedded Hardware

Luis Prieto-López[0009−0003−8492−6678], Iván Pascual, Mencía George, Francisco J. Rodríguez-Lera[0000−0002−8400−7079], and Lidia Sánchez-González[0000−0002−0760−1170]

Robotics Group, I4 Institute, Department of Mechanical, Computer Science and Aerospace Engineering, Universidad de León, León, 24007, Spain
lpril,lidia.sanchez@unileon.es
http://robotica.unileon.es

Abstract. Agricultural automation is increasingly important for addressing global food security challenges and labor shortages in the farming sector. This paper presents a vision-guided robotic system for real-time fruit and vegetable detection deployed on embedded hardware. The system utilizes LeYOLO-Nano, a lightweight object detection model optimized for edge devices, running on an NVIDIA Jetson platform. We trained and evaluated the model on a multi-class dataset containing 31 different categories of fruits and vegetables. The LeYOLO-Nano architecture was chosen for its minimal computational requirements while maintaining detection accuracy, making it suitable for real-time inference on resource-constrained embedded systems. Our experimental results demonstrate the feasibility of deploying advanced computer vision models on embedded platforms for agricultural applications, enabling autonomous crop detection and harvesting assistance in controlled indoor conditions, representing a first step toward real-world field deployment. The proposed system represents a step toward affordable and efficient agricultural robotics that can operate in real farming environments.

Keywords: Agricultural robotics · Object detection · Embedded systems · Computer vision · YOLO · Edge computing

1 Introduction

Modern agriculture faces significant challenges including labor shortages, increasing production demands, and the need for sustainable farming practices. Automation and robotics have emerged as key technologies to address these challenges, with computer vision playing a central role in enabling autonomous agricultural operations. The ability to accurately detect and localize crops in real-time is fundamental for tasks such as selective harvesting, yield estimation, crop health monitoring, and precision agriculture applications.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40

Traditional object detection systems for agricultural robotics often rely on powerful GPU workstations or cloud-based processing, which limits their practical deployment in field conditions due to power consumption, cost, and connectivity requirements. Edge computing offers a compelling alternative by enabling on-device inference with low latency and reduced dependency on external infrastructure. However, deploying state-of-the-art deep learning models on embedded platforms presents significant challenges due to limited computational resources, memory constraints, and power budgets. Recent advances in efficient neural network architectures have made it possible to achieve real-time object detection on embedded devices without substantial loss in accuracy. YOLO (You Only Look Once) family models have proven particularly effective for agricultural applications due to their balance between speed and precision. However, standard YOLO variants remain computationally intensive for resource-constrained platforms, necessitating the development of lightweight alternatives optimized for edge deployment.

This work addresses the challenge of real-time crop detection on embedded hardware by implementing a vision-guided robotic system based on LeYOLO-Nano, a recently proposed lightweight object detection framework. Our contributions include: (1) the application of LeYOLO-Nano to a comprehensive multi-class fruit and vegetable detection task comprising 31 categories, (2) a detailed analysis of the model's performance on embedded hardware, specifically the NVIDIA Jetson platform, (3) the integration of the vision system into a mobile robotic platform for agricultural environments, and (4) experimental validation in controlled indoor conditions demonstrating real-time detection capabilities suitable for practical farming operations.

The remainder of this paper is organized as follows: Section 2 reviews related work in agricultural robotics and embedded computer vision. Section 3 describes our methodology, including the dataset, model architecture, and training configuration. Section 4 details the robot integration process. Section 5 presents experimental results and discussion. Finally, Section 6 concludes the paper and outlines future research directions.

2 Related works

The application of computer vision and deep learning to agricultural robotics has experienced significant growth in recent years, driven by advances in efficient neural network architectures and the availability of powerful embedded computing platforms.

The different YOLO model variants have demonstrated high efficiency in object detection tasks, with multiple configurations and architectural improvements introduced over time to enhance performance. A comprehensive comparison and deeper analysis of these variants can be found in [10]. Recent YOLO variants have been specifically optimized for real-time object detection on resource-constrained edge platforms. EdgeYOLO [6] enables low-complexity inference on devices such as the NVIDIA Jetson AGX Xavier while maintaining real-time performance

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40

(FPS $\geq$ 30). Similarly, optimized YOLOX and YOLOv12 implementations have demonstrated effective real-time operation on Jetson AGX Orin in agricultural automation and robotic harvesting scenarios. Beyond general-purpose models, several works focus on crop-specific optimization for edge deployment, such as DS-YOLO [11], a lightweight YOLOv8n-based framework for strawberry detection in complex orchard environments, and comparative studies on potato harvesting systems [5] analyzing the trade-offs between detection accuracy and computational efficiency. More recent approaches, including YOLOv11-Litchi [9] and PGLD-YOLO [7], further demonstrate how tailored architectural modifications and lightweight backbones can preserve real-time performance in edge-based agricultural perception systems.

The deployment of object detection models on embedded platforms requires significant architectural modifications to reduce computational complexity without compromising accuracy. LeYOLO [4] represents a notable contribution in this direction, introducing a new embedded architecture specifically designed for object detection on resource-constrained devices. The LeYOLO-Nano variant, employed in this work, achieves substantial reductions in both parameter count and computational requirements through strategic use of inverted bottleneck blocks and optimized feature fusion strategies.

Beyond detection models, complete robotic systems for agricultural applications require integrated solutions for navigation, mapping, and behavior control. VAULT [3] provides a mobile mapping system for ROS 2-based autonomous robots, addressing the broader infrastructure needed for agricultural robotics deployment. Similarly, hybrid cognitive architectures [2] enable robots to generate, control, plan, and monitor behaviors in interactive agricultural scenarios.

3 Materials and Methods

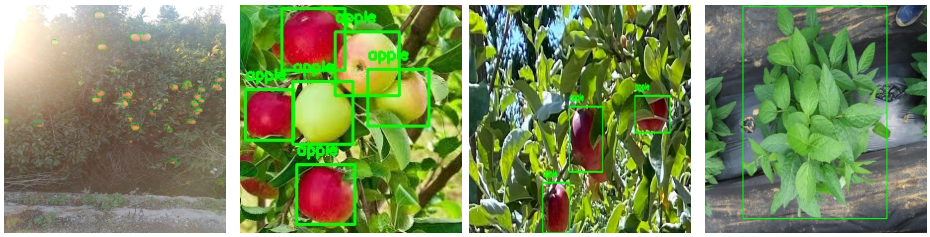
3.1 Dataset

The dataset used in this work is a multi-class fruit object detection dataset composed of 31 different categories, including a wide variety of fruits and vegetables. All datasets were collected from publicly available online platforms such as Roboflow and Kaggle, then analyzed to eliminate duplicates and merged to create a larger and more comprehensive dataset. The classes considered are: apple, orange, tomato, peach, quince, pomegranate, pear, lemon, rice, plum, dragon fruit, bell pepper, durian, aubergine, radish, cherry, apricot, courgette, bean, carrot, mango, strawberry, salad, banana, fig, grape, durian, pineapple, watermelon, lime and lychee. The dataset contains a total of 22,157 images, which were divided into three subsets: 17,731 for training, 2,797 for validation and 1,629 images for testing. This split was designed to provide a sufficient amount of data for learning while enabling reliable evaluation of the model's performance and generalization capability.

All images in the dataset were annotated using bounding boxes following the YOLO annotation format, where each object is represented by its class identifier

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40

and normalized bounding box coordinates ($x_{center}, y_{center}, width, height$). This dataset provides a diverse and representative collection of fruits and constitutes a solid basis for training and evaluating a real-time object detection system deployed on an embedded platform. Figure 1 shows representative samples from the dataset, illustrating the diversity of fruit and vegetable categories and the variability in image conditions. The dataset is publicly available at [8].



(a) (b) (c) (d)

Fig. 1: The dataset includes images captured under different sunlight conditions at various times of the day (a); it contains fruits of various colors (b) and fruits that present occlusions (c); and it also includes images from different stages of plant growth (d).

Data augmentation The original dataset did not include any form of data augmentation. To address this limitation, the augmentation process was implemented using a Python-based script that operates directly on datasets formatted for YOLO. For each image in the training set, several augmented versions were generated applying a subset of predefined transformations while preserving label consistency. The corresponding bounding box annotations were automatically adapted when required to ensure annotation accuracy. The applied augmentation techniques include horizontal flipping, small random rotations, brightness adjustment, contrast modification, saturation variation, additive noise injection, and Gaussian blur.

3.2 Vision model

The vision model used in this work is based on LeYOLO [4], a lightweight object detection framework specifically designed for real-time performance on edge and embedded devices. In particular, the LeYOLO-Nano variant was selected due to its highly compact architecture and suitability for low-power embedded hardware such as the NVIDIA Jetson platform. LeYOLO-Nano introduces architectural optimizations that reduce both the number of parameters and the computational complexity compared to standard YOLO models. The overall architecture of the LeYOLO-Nano model, including the backbone, neck, and detection head, is shown in Fig. 2.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40

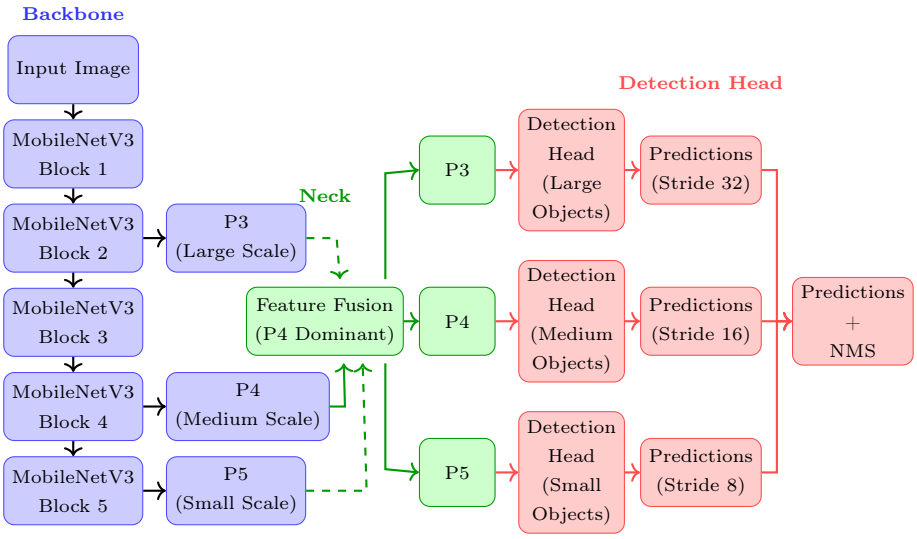


Fig. 2: Architecture of the LeYOLO-Nano model, illustrating the backbone, neck, and detection head [4]. The dominant semantic level (P4) is shown with solid lines

Backbone The backbone of LeYOLO [4] is designed to achieve an efficient balance between computational cost and representational capacity, making it suitable for real-time object detection on resource-constrained platforms. It is built upon inverted bottleneck blocks, a structure widely adopted in efficient convolutional neural networks such as the MobileNet family. Each inverted bottleneck block follows a lightweight design composed of a depthwise convolution for spatial feature extraction and pointwise convolutions for channel projection. In contrast to standard inverted bottlenecks, LeYOLO introduces architectural optimizations by selectively omitting the initial pointwise expansion convolution in stages where the input and expanded channel dimensions coincide. This reduces redundant computation, particularly in early layers where feature maps have high spatial resolution. To further control model complexity, the channel expansion ratio is limited to a maximum of six, ensuring efficient information flow while preventing excessive parameter growth. Through progressive downsampling, the backbone generates multi-scale feature representations at different semantic levels (P3, P4 and P5), which are subsequently forwarded to the detection neck for multi-scale feature aggregation. The SiLU activation function is consistently used throughout the backbone.

Neck The neck of the network, referred to as LeNeck, is responsible for aggregating multi-scale semantic information from the backbone. Unlike traditional Feature Pyramid Networks (FPN) or PANet structures, which introduce significant computational overhead, LeNeck adopts a single dominant semantic level

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40

(P4) as the primary fusion point for features coming from both lower (P3) and higher (P5) levels. This design choice reduces repeated computations at high-resolution feature maps while maintaining sufficient spatial detail for accurate object localization. Computation at the P3 and P5 levels is performed only once, and feature fusion is achieved using efficient concatenation and depthwise convolutions. Similar to the backbone, costly pointwise convolutions are removed at the P3 level to further reduce computation.

Head The detection head follows a multi-scale design, generating predictions at three different feature resolutions (P3, P4 and P5), which enables the detection of objects of varying sizes. Each feature map is processed by an independent detection branch responsible for bounding box regression, objectness estimation and class prediction.

Loss function LeYOLO employs a loss function consistent with the standard YOLO single-stage object detection framework. The overall loss is composed of three main components: bounding box regression loss, objectness confidence loss, and classification loss. Bounding box regression is performed using center-based coordinates (X_{center}, Y_{center}, width, height) which enables efficient optimization of object position and scale. The regression loss penalizes discrepancies between predicted and ground-truth bounding boxes, encouraging accurate spatial localization of detected objects. The objectness confidence loss measures the likelihood that a predicted bounding box contains an object, allowing the network to suppress background regions and focus on relevant detections. The classification loss is applied to the predicted class probabilities and encourages correct class assignment for each detected object.

3.3 Model configuration

The model configuration used in this work corresponds to the LeYOLO-Nano variant, selected for its suitability for real-time inference on embedded hardware. The network was trained using an input image resolution of 640 x 640 pixels, which provides balance between the dataset annotations, class definitions and model output dimensions.

4 Robot integration

The proposed perception system was integrated and deployed on an NVIDIA Jetson embedded platform. This type of hardware is commonly found in mobile robots, such as Go2. The system is implemented within the ROS 2 framework, using a YOLO-ROS [1] interface to integrate the trained detection model into the robotic software architecture. Camera images published as ROS 2 topics are processed directly on the Jetson device, allowing the model to run fully onboard and perform real-time object detection without external computation.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40

5 Experimental results

5.1 Experimental setup

To analyze the impact of data augmentation and dataset balancing strategies, different dataset configurations were evaluated: (i) **No augmentation**, using only the original images; (ii) **Online data augmentation**, applied dynamically during training using the augmentation pipeline provided by the YOLO framework; (iii) **Offline data augmentation**, where additional samples were generated prior to training using a custom augmentation script.

The system was deployed on an NVIDIA Jetson AGX Thor with an Intel RealSense D435 camera directly connected to the device, capturing RGB frames that were processed on-board in real time during model execution. All experiments were conducted in a controlled indoor laboratory environment. The system was implemented within the ROS 2 framework and integrated using YOLO ROS for real-time object detection.

5.2 Results

This section reports the quantitative results obtained after training the models with the different dataset configurations, evaluated on a common validation set. Table 1 summarizes these results. The trained model was exported to ONNX format and deployed within the experimental setup described previously. Inference was executed directly on the NVIDIA Jetson AGX Thor using the Intel RealSense D435 camera connected to the device. The system demonstrated stable real-time performance during operation in a controlled laboratory environment. An average throughput of approximately 15 FPS was achieved under live camera conditions. This rate was intentionally constrained by the ROS 2 node configuration to ensure deterministic and stable behavior of the perception pipeline.

Table 1: Training results

Strategy	**Precision**	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**
No augmentation	0.64	0.643	0.642	0.404
Online augmentation	0.573	0.519	0.561	0.354
Offline augmentation	0.605	0.588	0.589	0.357
Extended dataset with augmentation	**0.697**	**0.657**	**0.688**	**0.436**

6 Conclusions

This work presents a complete vision-based robotic perception system for real-time fruit detection on embedded hardware. The approach combines a curated

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40

and augmented multi-class dataset with the lightweight LeYOLO-Nano architecture, enabling efficient deployment on resource-constrained platforms. The system was integrated into a ROS 2-based robotic framework and executed onboard an NVIDIA Jetson AGX Thor device, achieving stable real-time performance in a controlled laboratory environment. These results demonstrate the feasibility of embedding deep learning-based perception directly into autonomous agricultural robots without relying on external computation. However, further validation in diverse outdoor farming scenarios is required to assess deployment readiness. Future work will focus on integrating the perception module with robotic manipulation and navigation systems to enable fully autonomous harvesting and precision agriculture tasks under real field conditions.

Acknowledgments. Grant PID2024-161761OB-C21 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. González, M.: yolo_ros. https://github.com/mgonzs13/yolo_ros (2023)
2. González, M., Rodríguez, F., Fernández, C., Matellán, V.: A hybrid cognitive architecture to generate, control, plan, and monitor behaviors for interactive autonomous robots. International Journal of Social Robotics (2025)
3. González, M., Rodríguez, F., Matellán, V.: VAULT: A mobile mapping system for ROS 2-based autonomous robots (2025), https://arxiv.org/abs/2506.09583
4. Hollard, L., Mohimont, L., Steffenel, L.A., Gaveau, N.: LeYOLO, New Embedded Architecture for Object Detection. Proceedings of the Conference on Robots and Vision (May 2025). https://doi.org/10.21428/d82e957c.aed2cb06
5. Kim, J., Kim, G., Yoshitoshi, R., Tokuda, K.: Real-Time Object Detection for Edge Computing-Based Agricultural Automation: A Case Study Comparing the YOLOX and YOLOv12 Architectures and Their Performance in Potato Harvesting Systems. Sensors **25**(15) (2025). https://doi.org/10.3390/s25154586
6. Liu, S., Zha, J., Sun, J., Li, Z., Wang, G.: EdgeYOLO: An edge-real-time object detector (2023), https://arxiv.org/abs/2302.07483
7. Lu, J., Zhao, Y., Yu, M.: PGLD-YOLO: a lightweight algorithm for pomegranate fruit localisation and recognition. PeerJ Computer Science **11**, e3307 (Oct 2025). https://doi.org/10.7717/peerj-cs.3307
8. Pascual-Cisneros, I.: Multi-class fruit detection dataset for real-time embedded vision systems (2026). https://doi.org/10.5281/zenodo.18618629
9. Peng, H., Xie, H., Li, W., Liu, H., Li, X.: YOLOv11-Litchi: Efficient Litchi Fruit Detection Based on UAV-Captured Agricultural Imagery. arXiv preprint arXiv:2510.10141 (2025), https://arxiv.org/pdf/2510.10141
10. Sapkota, R., Karkee, M.: Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 object detectors for computer vision and pattern recognition (2025), https://arxiv.org/abs/2510.09653
11. Teng, H., Sun, F., Wu, H., Lv, D., Lv, Q., Feng, F., Yang, S., Li, X.: DS-YOLO: A lightweight strawberry fruit detection algorithm. Agronomy **15**(9) (2025). https://doi.org/10.3390/agronomy15092226

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_40