Transactions on Energy Systems and Engineering Applications

Article

Deep learning model for recognizing fresh and rotten fruits in industrial processes

Carlos Arias[1], Camilo Baldovino[1], José Gómez[1] Brian Restrepo[1], and Sergio Sánchez[2,*]

[1] Departamento de Ingeniería Electrónica, Universidad de Sucre, Sincelejo, Colombia.
[2] Departamento de Ingeniería Electrónica, Corporación Universitaria Antonio José de Sucre, Sincelejo, Colombia.
[*] docente_investigador9@uajs.edu.co

Received: 06 December 2024; Accepted: 01 July 2025; Published: 04 July 2025

Abstract: The detection of fruit condition is essential to ensure quality control in industrial processes. Currently, this task is often performed manually, which is inefficient and time-consuming for operators. Therefore, it is crucial to implement emerging technologies that reduce human effort, costs, and production time while enabling more effective defect detection in fruits. In this context, this work presents the implementation of an artificial intelligence model based on computer vision to identify the condition of fruits. Various models were compared, including YOLOv8, YOLOv11, Detectron2, and Fast R-CNN, trained on a dataset that classifies fruits into two categories: ripe and rotten. The models were evaluated in terms of accuracy, speed, and robustness under different lighting and background conditions to select the most suitable for real-time applications. The results showed that YOLOv8 achieved the best generalization, reaching a mAP@50 of 83.8% and an accuracy of 77.3%.

 https://doi.org/10.32397/tesea.vol6.n1.811

1. Introduction

Agriculture is a key sector for global economic development, providing essential products for food and health, as well as vital inputs for industrial goods production. However, this sector faces the challenge of overcoming traditional methods of production and commercialization. The lack of innovation and modernization in processes has led to stagnation in productivity efficiency and a decline in the quality of agricultural inputs, directly impacting the sector's competitiveness and the participation of new generations in its development [1, 2].

In this context, investment in science, technology, and innovation is crucial to transform agricultural processes into viable solutions for the industry. These strategies aim not only to improve the competitiveness of the sector but also to ensure that advancements align with the sustainability and quality demands of global markets [3–6].

How to cite this article: Arias, Carlos; Baldovino, Camilo; Gómez, José; Restrepo, Brian; Sánchez, Sergio. Deep learning model for recognizing fresh and rotten fruits in industrial processes. *Transactions on Energy Systems and Engineering Applications*, 6(1): 811, 2025. DOI:10.32397/tesea.vol6.n1.811

Food waste is a global challenge that requires innovative solutions, with artificial intelligence (AI) emerging as a promising tool. In fruits and vegetables, ripening involves evident physical changes, such as variations in color, texture, and size, as well as less perceptible chemical, biochemical, and microbiological transformations. Characteristics such as soluble solids, acidity, and sugar content are important but require destructive laboratory tests, complicating their analysis with AI. Additionally, physiological factors like respiration and ethylene production, along with microbiological influences from bacteria, viruses, and fungi, affect the ripening process. These changes, genetically programmed, occur progressively and involve variables such as tissue softening, pigmentation, and volatile compound production [7]. The incorporation of AI in this context has the potential to revolutionize quality control and reduce waste, although its full implementation still depends on technological advancements that enable more accessible and effective analysis [8].

Fruits, valued for their high nutritional content of proteins and vitamins, can harbor harmful microorganisms due to inadequate handling and inspection processes, affecting food safety. Many companies still distribute fruits unfit for consumption, highlighting the need for technological solutions. Advances in machine learning (ML) and AI have driven automation in agricultural tasks, emphasizing the importance of computer vision systems for fruit detection and classification. These technologies optimize industrial processes by overcoming the limitations of manual inspection, such as fatigue and human error, ensuring more precise and efficient selection in high-demand environments [9].

In the industrial sector, AI-based computer vision has revolutionized tasks such as product classification, defect detection, and quality control. These technologies provide efficient, fast, and sustainable solutions, not only improving quality standards but also reducing costs and losses throughout the production chain. In agroindustry, their application enables the determination of fruit ripeness and detection of spoilage with high precision, optimizing classification and minimizing waste [10].

Computer vision emulates human visual perception by processing digital images, but with significantly enhanced capabilities, such as continuous operation and reduced errors. For example, tools like the Segment Anything Model (SAM), developed by Meta AI, precisely segment objects in images with minimal resources, strengthening the automation of tasks like fruit classification. Additionally, the use of high-resolution cameras, such as FLIR and Basler, ensures the capture of high-quality images, essential for training and operating AI models [11].

Cloud training platforms like Google Cloud AI Platform and Amazon SageMaker enable large-scale model training with greater efficiency. Google Cloud AI Platform, also known as Vertex AI, offers advanced tools for training models without the need for coding, while Amazon SageMaker provides an integrated environment for creating, training, and deploying machine learning models, supporting customization and the use of pre-trained models [12][13][14][15].

Within this framework, the incorporation of a deep learning model for recognizing ripeness and spoilage in tropical fruits promises to transform agro-food production processes. This model combines the capabilities of computer vision with the analytical power of deep neural networks, ensuring high-quality products and solidifying a competitive and sustainable industrial approach.

2. Related Work

Traditional fruit sorting processes have been carried out manually, with workers visually inspecting each product to determine its quality, ripeness, or the presence of defects. Although widely used, this method has significant limitations, such as subjectivity, operator fatigue, and inefficiency when handling large production volumes [16]. In contrast, automated techniques based on AI and computer vision have revolutionized these tasks, enabling faster, more accurate, and consistent sorting processes [17].

These technologies process digital images to identify specific fruit characteristics, such as color, texture, and size, ensuring high-quality standards and significantly reducing human errors. Transitioning to automated methods not only increases productivity but also optimizes resources and enhances competitiveness in the agri-food industry [18].

In recent years, advances in artificial intelligence (AI) in the field of computer vision have transformed traditional techniques for object classification and detection. Advanced processes such as segmentation and object detection using neural networks with state-of-the-art architectures, including ResNet [19], SSD [20], MobileNet [21], SegNet [22], Grounding DINO [23], and various versions of YOLO [24], have proven essential for tackling complex fruit identification and classification tasks in industrial environments, optimizing efficiency and accuracy.

Furthermore, the development of transformer-based architectures has represented a significant breakthrough, improving precision and performance compared to traditional methods. Multimodal models, such as CLIP [25] and Vision-Language Models (VLM) [26], have opened new possibilities by integrating multiple data sources. These multimodal technologies allow for deeper and more detailed data understanding, optimizing detection and classification even in complex scenarios, such as partially obscured fruits or those with similar characteristics between varieties.

The food industry has recently begun adopting advanced AI models to improve precision in determining the ripeness and conservation quality of fruits. By leveraging advanced image processing techniques and deep learning algorithms, these approaches automate classification processes, achieving more accurate and efficient results. Below, some key studies in this field are analyzed.

Sukkasem et al. (2023) [27] presented research on fruit classification using transfer learning and image processing, demonstrating how this approach enhances precision compared to other deep learning methods. In their study, the MobileNetV2 model achieved 99% accuracy, surpassing the original model by 3% and architectures like AlexNet and VGG16 by 10%. This research highlights the potential of transfer learning to further optimize performance without requiring extensive domain-specific data.

Similarly, Gulzar et al. (2023) [28] demonstrated in their Sustainability publication that MobileNetV2, utilizing transfer learning, achieved a 99% success rate in fruit image classification, significantly outperforming models like VGG16 and ResNet. The study also emphasizes that fine-tuning the classification layer to match project-specific requirements is crucial for achieving robust results with pretrained networks.

Regarding the detection of fruit ripeness and decay, Wang (2020) and Zhao (2021) [29] highlighted the effectiveness of models such as YOLO and Faster R-CNN due to their high precision in fruit identification. However, these models face challenges with partially obscured or very small fruits. Neural networks like ResNet and AlexNet improve precision in automated agricultural applications and are crucial for real-time detection, especially when integrated into autonomous harvesting robots.

A noteworthy study in this area was conducted by Garcés et al. (2023) [30], who proposed a portable system for detecting and classifying apples during harvest using convolutional neural networks. This system integrates two main methods: apple type detection for counting using the SSD-MobileNet model, and pixel-level quality segmentation implemented with an FCN-ResNet18 network. This approach stands out for its adaptability to hardware with limited capabilities, facilitating real-world implementation in harvesting processes.

In automated phenotypic analysis [31], researchers have demonstrated the effective use of AI technologies to enhance efficiency and precision in the phenotypic analysis of melons. This study utilized a combination of deep learning models, including DANet for semantic segmentation, RTMDet for object detection, RTMPose for keypoint detection, and MobileSAM (Segment Anything Model) for mobile-friendly segmentation. The results showed a high correlation between algorithm-predicted values and manually measured values, validating the proposed approach's feasibility and precision.

Implementing an efficient AI model for the rapid and accurate detection of spoiled fruits is essential to optimize this process and leverage the potential of emerging technologies. Neural networks have demonstrated superiority in image processing and fruit decay detection. However, selecting the ideal architecture depends on various factors, including the specific task, dataset characteristics, and available computational resources. Each architecture offers particular advantages in specialized scenarios, emphasizing the importance of selecting the appropriate model to maximize performance.

3. Methods and Materials

3.1. Dataset

The dataset used in this study consists of 3133 images of tropical fruits, including oranges, guavas, apples, mangoes, and lemons, with a resolution of 640x640 pixels. These images were classified into two main categories: ripe fruits and spoiled fruits. The dataset distribution included 700 images of oranges, 600 of guavas, 600 of apples, 600 of mangoes, and 633 of lemons, ensuring a wide range of visual characteristics for training and evaluating the models. The images were sourced from Kaggle databases, encompassing various lighting conditions and environments.

To enhance the model's generalization ability and reduce the risk of overfitting, a systematic data augmentation process was implemented using the Roboflow platform. This process included a set of controlled transformations, such as:

- Rotation of ±15°, to simulate common angular variations during image capture.
- Horizontal and vertical translation up to 10% of the image size, enabling the model to adapt to objects appearing in different positions within the frame.
- Brightness and contrast adjustment within a ±20% range, improving the model's robustness under variable lighting conditions.
- Shear transformations up to ±10°, to account for typical distortions found in real-world environments.

These transformations were applied randomly but in a balanced manner, generating multiple augmented versions of each original image. As a result, the dataset was significantly expanded, not only in size but also in contextual and visual diversity. This enrichment of the training data enabled the model to learn more generalized patterns, ultimately improving its performance in real-world scenarios with varying conditions.

3.2. Architecture

For this study, four advanced convolutional neural network architectures specialized in object detection were selected: YOLOv8, YOLOv11, Detectron2, and Fast R-CNN. Each of these architectures offers specific advantages in terms of speed, accuracy, and the ability to detect objects of varying sizes, making them standout options for fruit classification and segmentation tasks.

3.2.1. YOLOv8

YOLOv8 (You Only Look Once version 8) is a state-of-the-art object detection architecture developed by Ultralytics, widely recognized for its high speed and accuracy in real-time computer vision tasks. Unlike its predecessors, YOLOv8 features a completely redesigned architecture with enhancements to the backbone, neck, and detection head, enabling superior overall performance. The YOLOv8 architecture is primarily composed of three stages:

The YOLOv8 architecture is primarily composed of three stages:

- ***Backbone:*** Responsible for extracting hierarchical features from the input image. YOLOv8 uses a more efficient design than earlier versions, incorporating C2f blocks (an improved variant of C3), which enhance feature reuse without significantly increasing computational complexity.
- ***Neck:*** Utilizes a PAN (Path Aggregation Network) structure to merge multiscale features, improving the model's ability to detect objects of various sizes. This stage enhances spatial context understanding and refines intermediate representations.
- ***Head:*** The output layer generates the final predictions, including bounding box coordinates, class labels, and confidence scores. YOLOv8 features a lighter and more accurate detection head that supports detection, segmentation, and classification tasks.

YOLOv8 comes in multiple size variants (n, s, m, l, x), allowing users to scale the model according to available computational resources. This makes it particularly suitable for edge computing applications, drones, smart cameras, mobile devices, and embedded systems.

In addition, YOLOv8 introduces a more intuitive training pipeline, improved annotation management, and direct compatibility with formats such as COCO and YOLO, facilitating seamless integration into various computer vision workflows. Thanks to these features, YOLOv8 is currently one of the most efficient and versatile solutions for real-time object detection tasks.

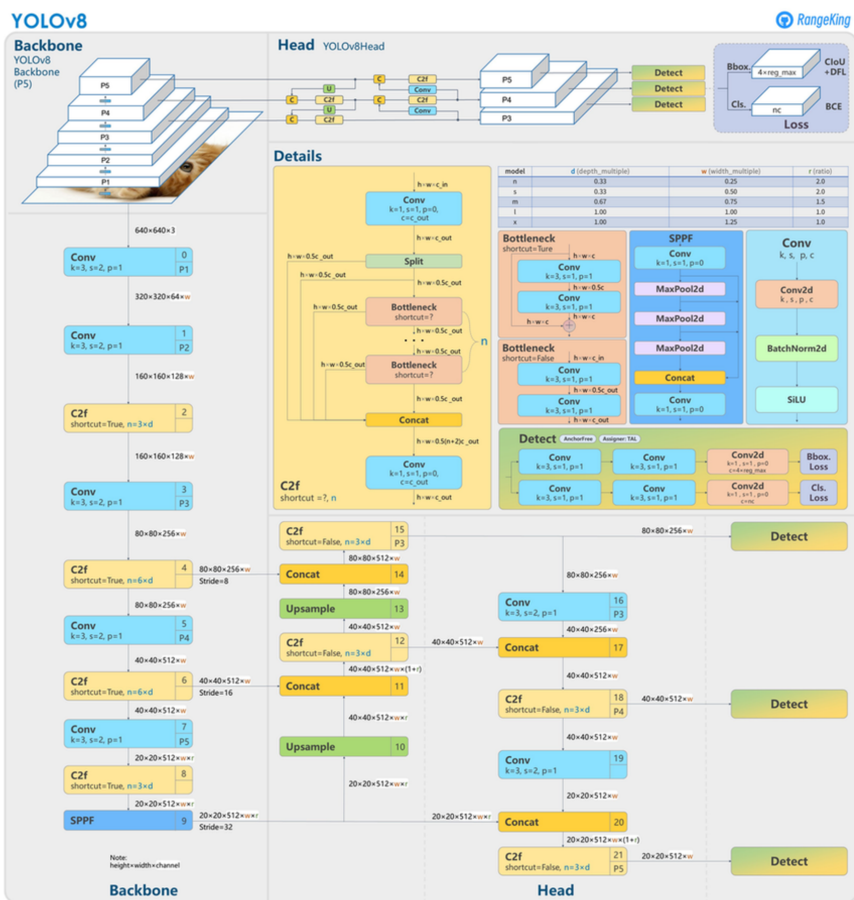


Figure 1. YOLOV8 model architecture [32].

3.2.2. YOLOv11

YOLOv11 is an advanced evolution in the YOLO (You Only Look Once) family, designed to push the boundaries of real-time object detection, especially in scenarios requiring high precision. It significantly improves upon previous versions by incorporating enhanced attention mechanisms, multi-scale feature fusion, and better contextual reasoning. These advancements enable YOLOv11 to deliver superior performance in detecting small objects and maintaining accuracy in environments with challenging lighting conditions and complex or cluttered backgrounds.

The architecture of YOLOv11 is composed of three main stages:

- ***Backbone:*** This stage is responsible for initial feature extraction from input images. YOLOv11 utilizes a hybrid backbone based on CSPNet and Transformer-based modules, allowing it to capture both local and global features. The backbone consists of approximately 40–60 layers, depending on the model variant (e.g., YOLOv11-s, -m, -l, or -x).
- ***Neck:*** The neck includes enhanced feature aggregation structures such as PANet++ and BiFPN (Bidirectional Feature Pyramid Network), which improve multi-scale feature representation. This is crucial for detecting objects of varying sizes particularly small and occluded targets without sacrificing speed.
- ***Head:*** YOLOv11 features a decoupled head for classification and localization tasks. This separation allows for more precise bounding box regression and class confidence estimation. It also incorporates adaptive anchor-free mechanisms to better localize small and irregularly shaped objects.

Key characteristics of YOLOv11 include:

1. Advanced small object detection using refined spatial pyramid pooling and attention modules.
2. High robustness under suboptimal lighting and noisy backgrounds.
3. Optimized performance trade-off, maintaining real-time capability with only a slight reduction in speed compared to YOLOv8.
4. Flexible deployment with different model sizes adapted for resource-constrained environments or high-end GPU setups.

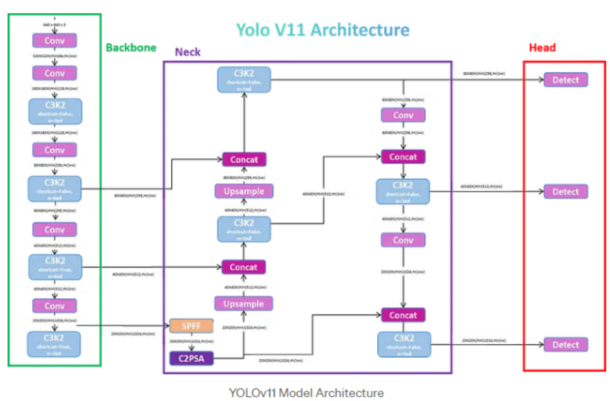


Figure 2. YOLOV11 model architecture [33].

3.2.3. Detectron2

Developed by Facebook AI Research (FAIR), Detectron2 is one of the most powerful and flexible open-source frameworks for computer vision tasks such as object detection, instance segmentation, semantic segmentation, and keypoint detection. Built on PyTorch, Detectron2 is a complete redesign of the original Detectron, offering a more efficient, modular, and scalable architecture.

Detectron2 uses deep convolutional neural networks (CNNs) as backbones typically ResNet-50, ResNet-101, or ResNeXt-101 with configurations that can include over 100 convolutional layers depending on the model. Its architecture is structured into three main stages:

- ***Backbone:*** This stage is responsible for extracting rich visual features from input images using residual networks (e.g., ResNet), which allow training of very deep architectures due to their skip connections. Backbones often contain between 50 and 101+ layers.
- ***Neck / Feature Pyramid Network (FPN):*** The FPN enhances feature extraction across multiple scales, making it highly effective at detecting both large and small objects. It improves the semantic richness of features at all levels of the hierarchy, which is critical for accurate segmentation.
- ***Head:*** This stage includes task-specific heads that process the features to generate predictions:

1. Region Proposal Network (RPN) for generating candidate object regions.
2. Box Head for object classification and bounding box regression.
3. Mask Head for instance segmentation, as in Mask R-CNN.
4. Keypoint Head for human pose estimation and keypoint detection.

Key Features:

1. High segmentation precision, thanks to the use of architectures like Mask R-CNN.
2. Multi-task support, allowing simultaneous object detection, segmentation, and keypoint estimation.
3. Highly modular and customizable, ideal for research and complex production systems.
4. Supports multi-GPU training and near real-time inference, although it is generally slower than lighter models like YOLOv8.
5. Best suited for controlled environments where segmentation accuracy is more important than inference speed, such as quality control in agriculture, medical image analysis, or industrial inspection.

Despite being more computationally intensive, Detectron2 offers exceptional performance in detailed object understanding. Its capability to generate pixel-level masks and perform multi-object tracking with high precision makes it an excellent choice for tasks where spatial accuracy and flexibility are paramount.

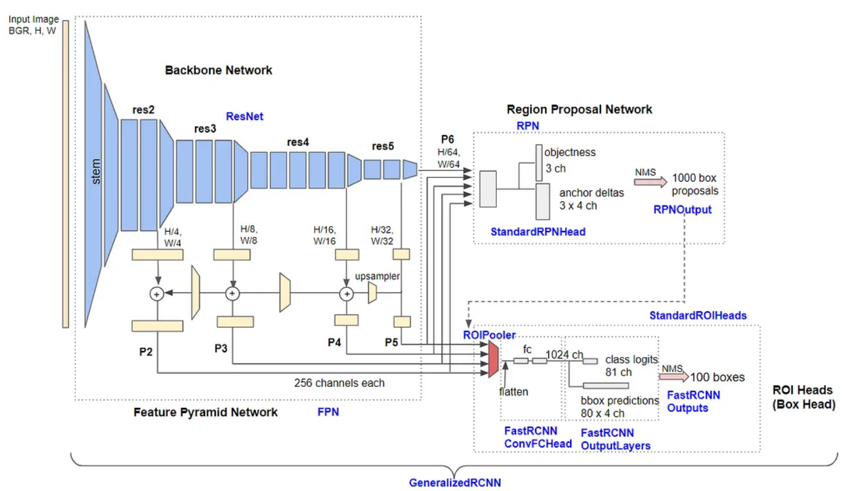


Figure 3. DETECTRON2 model architecture [34].

3.2.4. Fast R-CNN

Fast R-CNN is an advanced object detection architecture developed by Ross Girshick, known for significantly improving detection accuracy over earlier approaches like R-CNN and SPPnet. Although slower than newer models such as YOLO, Fast R-CNN excels at detecting objects of various sizes with high precision, making it especially suitable for scenarios where detection quality is more important than real-time processing such as high-resolution fruit classification or detailed scientific analysis.

The architecture of Fast R-CNN consists of three main stages:

- ***Feature Extraction:*** A deep convolutional neural network (e.g., VGG16 or ResNet50) is used as a backbone, comprising 16 to over 100 layers, depending on the chosen model. It processes the input image to produce a detailed feature map.
- ***Region of Interest (RoI) Pooling:*** Instead of cropping and resizing image regions individually like in R-CNN, Fast R-CNN applies RoI pooling directly to the shared feature map, transforming variable-sized regions into fixed-size representations. This increases both speed and accuracy.
- ***Classification and Bounding Box Regression:*** Each RoI is passed through fully connected layers for object classification and bounding box refinement, allowing precise localization and label assignment.

Key Features:

1. More efficient than R-CNN and SPPnet, allowing end-to-end training with shared computations.
2. High detection accuracy, even in high-resolution or visually complex images.
3. Reduced memory and computational redundancy, due to single-pass feature extraction.
4. Best suited for applications where detection accuracy outweighs speed, such as quality control, scientific classification, and detailed visual documentation.

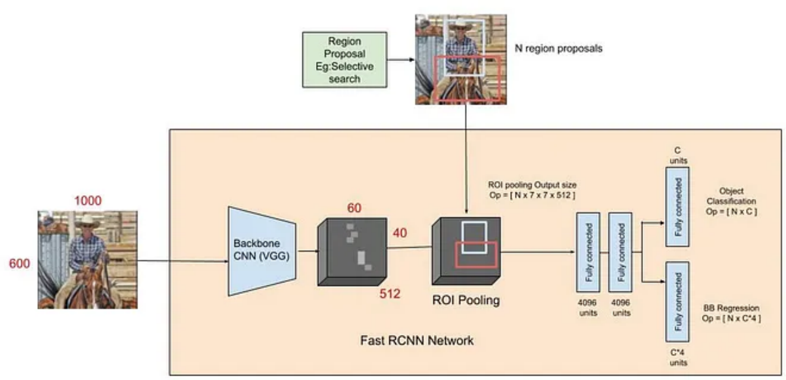


Figure 4. FAST R-CNN model architecture [35].

3.3. Experiment setup

In this study, the YOLOv8, YOLOv11, Detectron2 and Fast R-CNN models were selected due to their outstanding performance in object detection tasks. These models have proven to be key in similar studies, excelling in detecting objects of different sizes and shapes, making them ideal for addressing the specific challenges of this work.

The dataset used was divided into three parts: 85.56% intended for training, 10.85% for validation and 5.58% reserved for testing. The training was carried out using specific configurations designed to optimize the performance of each model and ensure reliable results in the detection process, as can be seen below.

Table 1. Training Configurations for AI Models

Configuration	YOLOv8	YOLOv11	Detectron2	Fast R-CNN
Optimizer	Adam	Adam	Adam	SGD
Learning Rate	0.001	0.001	0.0025	0.01
Batch Size	16	16	128	16
Number of Iterations	200	200	10000	200
Hardware	Nvidia Tesla T4	Nvidia Tesla T4	Nvidia Tesla T4	Nvidia Tesla T4
Additional Configuration	Patience = 15	Patience = 15	Patience = 15	Patience = 15

Table 1 details the configurations used for training each of the models. These settings were carefully selected to optimize the performance of each model according to their specific characteristics. The Adam optimizer was chosen for YOLOv8, YOLOv11, and Detectron2 due to its ability to adapt quickly and converge efficiently, while Fast R-CNN uses SGD, which typically performs better in traditional networks and allows more direct control over the learning rate.

The learning rate values were adjusted to balance convergence speed and training stability, being slightly higher for Detectron2 because of its more complex architecture. Batch size was determined based on computational capacity and the need for statistical stability; notably, Detectron2 employed a considerably larger batch size to improve gradient estimation.

The number of iterations reflects the complexity and time required for each model to reach optimal performance, with Detectron2 needing significantly more iterations due to its greater depth and detailed segmentation tasks. The use of Nvidia Tesla T4 hardware provides the necessary acceleration to efficiently train all models under these configurations.

Finally, the Patience parameter was set to 15 to enable early stopping of training if no improvement is observed, thereby preventing overfitting and optimizing resource use. The following figure shows a block diagram of the proposed system.

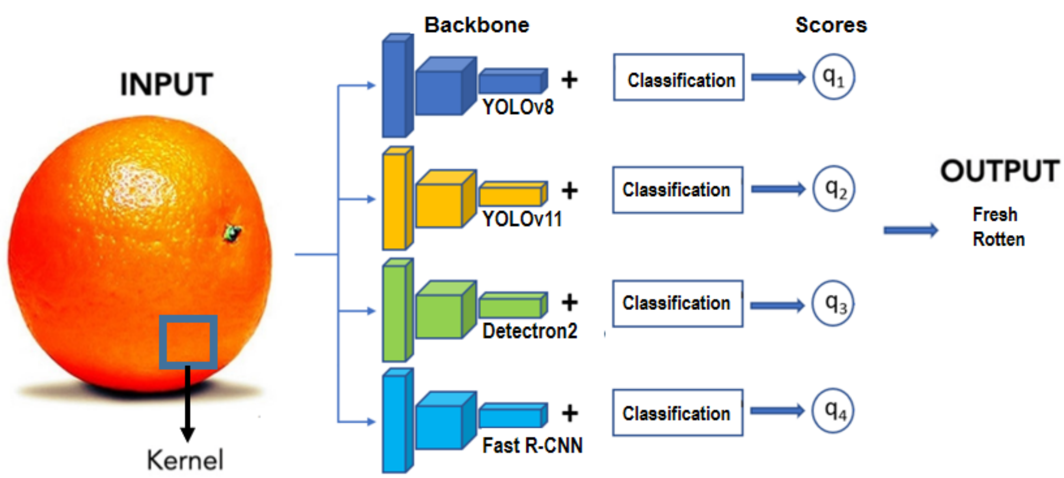


Figure 5. Block diagram of the proposed system.

4. Results and discussion

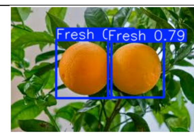
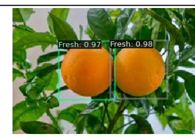
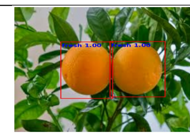
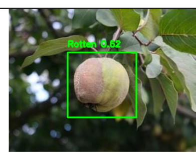
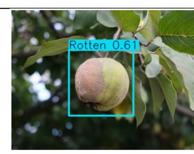
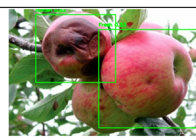
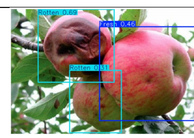
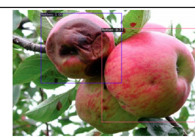
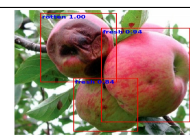
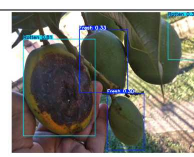
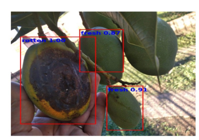
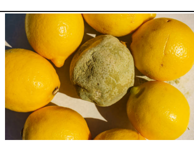
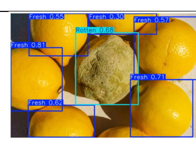
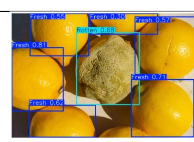
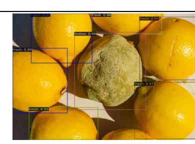
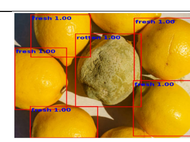
For this research, the supervised unimodal models YOLOv8, YOLOv11, Detectron2, and Fast R-CNN were used. These models were pre-trained using the Google Colab tool with a 15 GB NVIDIA Tesla T4

GPU. The images used presented a variety of environmental and lighting conditions, adding a level of realism and complexity to the detection task. For the implementation, we used the TensorFlow framework, the Python programming language, and the NumPy and Pandas libraries. After running the proposed model with different architectures, the following results were obtained:

Table 2. Comparison of Object Detection Models.

Model	Architecture	Precision	Recall	mAP50	mAP50-95
YOLOv8	CSPDarknet53	77.3	74.7	83.8	60.0
YOLOv11	CSPDarknet53x	73.1	73.7	81.9	58.7
Detectron2	ResNet-50-FPN	80.1	64.9	78.8	55.0
Fast R-CNN	ViT	63.9	81.8	76.9	52.8

Table 3. Comparison of AI Model Predictions.

Image	YOLOv8	YOLOv11	Detectron2	Fast R-CNN
				
				
				
				
				

In Table 2 and 3 The evaluation of object detection models YOLOv8, YOLOv11, Detectron2, and Fast R-CNN revealed notable differences in their performance across key metrics, such as precision, recall, mAP@50, and mAP@50-95. These results highlight the strengths and limitations of each model in the context of fruit classification tasks, offering insights into their suitability for real-world industrial applications.

Detectron2 achieved the highest precision (80.125%), indicating its superior ability to minimize false positives. This attribute is critical for ensuring high-confidence detections, especially in applications where misclassification of defective fruits could compromise quality control. However, its recall (64.9%) was significantly lower than other models, suggesting it missed a substantial number of actual detections, which limits its overall robustness in identifying all target objects. Conversely, Fast R-CNN demonstrated the highest recall (81.85%), outperforming all other models in capturing most of the target objects. However, this came at the cost of lower precision (63.97%), which could lead to an increased rate of false positives, a drawback in scenarios demanding high accuracy.

YOLOv8 demonstrated a strong balance between precision (77.3%) and recall (74.7%), achieving the highest mAP@50 (83.8%). This indicates its excellent generalization ability and accuracy across various conditions, making it the most suitable model for detecting fruit defects in real-time industrial applications. Furthermore, its mAP@50-95 score (60.097) surpassed those of YOLOv11, Detectron2, and Fast R-CNN, emphasizing its robustness in detecting objects of varying sizes and complexities.

YOLOv11, while slightly lagging behind YOLOv8 in precision and recall, still performed competitively with an mAP@50 of 81.9 and an mAP@50-95 of 58.7. This positions YOLOv11 as a viable alternative for scenarios where computational constraints are less stringent, as its architecture (CSPDarknet53x) offers improved feature extraction over standard CSPDarknet53.

Detectron2 and Fast R-CNN, despite their strengths in specific metrics, struggled to achieve competitive mAP scores compared to YOLOv8 and YOLOv11. Detectron2's ResNet-50-FPN architecture may be less suited for detecting fine-grained defects in fruits, while Fast R-CNN's transformer-based ViT architecture might face challenges with the dataset's specific characteristics, such as variable lighting and background conditions.

The superior performance of YOLOv8 in terms of both precision and mAP metrics makes it the optimal choice for deployment in real-time industrial systems. Its ability to maintain a strong balance between precision and recall ensures both reliability and efficiency in defect detection, reducing manual inspection efforts and minimizing post-harvest losses. While Detectron2 and Fast R-CNN could be valuable for specific tasks emphasizing precision or recall individually, their overall lower mAP scores limit their practical applicability in high-throughput environments.

The study also underscores the importance of selecting model architectures tailored to the specific requirements of the application. While YOLOv8's CSPDarknet53 demonstrated robustness and adaptability, alternative architectures like Detectron2's ResNet-50-FPN may require additional optimization or specialized preprocessing to achieve comparable results in agricultural settings.

Future research should focus on integrating advanced preprocessing techniques and domain-specific augmentation strategies to enhance the performance of models like Detectron2 and Fast R-CNN. Additionally, evaluating these models on larger and more diverse datasets could provide deeper insights into their scalability and generalization capabilities. Furthermore, exploring pretrained multimodal models and architectures leveraging transfer learning strategies would be crucial to optimizing model generalization.

5. Conclusions

In conclusion, the advanced convolutional neural network models evaluated in this study, such as YOLOv8, YOLOv11, Detectron2, and Fast R-CNN, demonstrated remarkable performance in fruit detection tasks, each with its own strengths and limitations. The results highlight the importance of choosing the appropriate architecture based on the system's specific needs, such as accuracy, real-time detection capability, or detailed object segmentation. In particular, YOLOv8 excelled with its balance between precision and speed, making it the most suitable option for real-time industrial applications, where efficiency is required without sacrificing accuracy.

However, models like Detectron2 and Fast R-CNN also offered advantages in scenarios where precise segmentation or detection of complex objects is critical. Although these models showed promising results, their performance in terms of mAP and precision did not match that of YOLOv8 in high-performance industrial settings. The study suggests that future research could focus on optimizing these models using advanced preprocessing techniques and domain-specific data augmentation, which could improve their generalization ability and applicability in more diverse environments.

Funding: This research did not receive any external funding.

Author contributions: Conceptualization, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo; Methodology, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo; Software, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo; Validation, Sergio Sánchez; Investigation, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo; Resources, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo; Data Curation, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo; Writing – Original Draft Preparation, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo; Writing – Review & Editing, Sergio Sánchez, Carlos Arias, José Gómez, Camilo Baldovino, and Brian Restrepo.

Disclosure statement: The authors declare no conflict of interest.

References

[1] Guillermo Aznarán Castillo. *La competitividad global agrícola*. UNMSM, 2003.

[2] Andres Vergara Narvaez and Marcos Alvarez Herrera. Análisis de la competitividad agrícola del departamento de sucre. *Tesis de pregrado, Universidad Tecnológica de Bolívar*, 2023.

[3] Francisco Javier Maza Ávila, Juan Carlos Vergara Schmalbach, Gustavo Adolfo Herrera Seba, Anny del Mar Agámez Arias, and Walter José Mejía Valeta. Potencialidad de la capacidad agrícola de la zona de desarrollo económico y social-zodes montes de maría del departamento de bolívar-colombia. *Desarrollo Regional y Competitividad*, 13, 2012.

[4] Ivan D Mardini G, Christian G Quintero M, César A Viloria N, Winston S Percybrooks B, Heydy S Robles N, and Karen Villalba R. A deep-learning-based grading system (asag) for reading comprehension assessment by using aphorisms as open-answer-questions. *Education and Information Technologies*, 29(4):4565–4590, 2024.

[5] María Valle, Jairo A Cardona, Cesar Viloria-Nunez, et al. Predicting atmospheric dispersion of industrial chemicals using machine learning approaches. *IEEE Access*, 2025.

[6] Bethsy Guerrero, César Viloria-Núñez, Miguel Ángel Jimeno Paba, et al. Enhancing disaster management through multi-objective water wave optimization for medical supplies storage and distribution. *Transactions on Energy Systems and Engineering Applications*, 5(2):1–12, 2024.

[7] Mukhriddin Mukhiddinov, Azamjon Muminov, and Jinsoo Cho. Improved classification approach for fruits and vegetables freshness based on deep learning. *Sensors*, 22(21):8192, 2022.

[8] Aafreen Kazi and Siba Prasada Panda. Determining the freshness of fruits in the food industry by image classification using transfer learning. *Multimedia Tools and Applications*, 81(6):7611–7624, 2022.

[9] Archana Chougule, Apurva Pawar, Rahul Kamble, Juned Mujawar, and Akshay Bhide. Recognizing fresh and rotten fruits using deep learning techniques. In *Data Engineering and Intelligent Computing: Proceedings of ICICC 2020*, pages 205–212. Springer, 2021.

[10] Sovon Chakraborty, FM Javed Mehedi Shamrat, Md Masum Billah, Md Al Jubair, Md Alauddin, and Rumesh Ranjan. Implementation of deep learning methods to identify rotten fruits. In *2021 5th international conference on trends in electronics and informatics (ICOEI)*, pages 1207–1212. IEEE, 2021.

[11] Abha Singh, Gayatri Vaidya, Vishal Jagota, Daniel Amoako Darko, Ravindra Kumar Agarwal, Sandip Debnath, and Erich Potrich. Recent advancement in postharvest loss mitigation and quality management of fruits and vegetables using machine learning frameworks. *Journal of Food Quality*, 2022(1):6447282, 2022.

[12] Google Cloud. Ai and machine learning products. https://cloud.google.com/products/ai, 2023. Accessed: 2024-12-01.

[13] Google Cloud. Introduction to vertex ai. https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform?hl=es-419, 2023. Accessed: 2024-12-01.

[14] Amazon Web Services. ¿qué es amazon sagemaker? https://docs.aws.amazon.com/es_es/sagemaker/latest/dg/whatis.html, 2023. Accessed: 2024-12-01.

[15] Amazon Web Services. Amazon sagemaker. https://aws.amazon.com/es/sagemaker/, 2023. Accessed: 2024-12-01.

[16] Yuhang Fu, Minh Nguyen, and Wei Qi Yan. Grading methods for fruit freshness based on deep learning. *SN Computer Science*, 3(4):264, 2022.

[17] Feng Xiao, Haibin Wang, Yaoxiang Li, Ying Cao, Xiaomeng Lv, and Guangfei Xu. Object detection and recognition techniques based on digital image processing and traditional machine learning for fruit and vegetable harvesting robots: an overview and review. *Agronomy*, 13(3):639, 2023.

[18] Youness Bahaddou, Lahcen Tamym, and Lyes Benyoucef. Ensuring fruits and vegetables freshness in sustainable agricultural supply chain networks: A deep learning approach. In *IFIP International Conference on Advances in Production Management Systems*, pages 364–378. Springer, 2024.

[19] Hanwen Kang and Chao Chen. Fruit detection, segmentation and 3d visualisation of environments in apple orchards. *Computers and Electronics in Agriculture*, 171:105302, 2020.

[20] Diwakar Agarwal and Anuja Bhargava. On-tree fruit detection system using darknet-19 based ssd network. *Journal of Food Measurement and Characterization*, 18(8):7067–7076, 2024.

[21] Xiuli Li, Yi Qin, Fujie Wang, Feng Guo, and John TW Yeow. Pitaya detection in orchards using the mobilenet-yolo model. In *2020 39th Chinese Control Conference (CCC)*, pages 6274–6278. IEEE, 2020.

[22] Gang Wu, Bin Li, Qibing Zhu, Min Huang, and Ya Guo. Using color and 3d geometry features to segment fruit point cloud and improve fruit recognition accuracy. *Computers and electronics in agriculture*, 174:105475, 2020.

[23] Connor C Mullins, Travis J Esau, Qamar U Zaman, Chloe L Toombs, and Patrick J Hennessy. Leveraging zero-shot detection mechanisms to accelerate image annotation for machine learning in wild blueberry (vaccinium angustifolium ait.). *Agronomy*, 14(12):2830, 2024.

[24] Hamzeh Mirhaji, Mohsen Soleymani, Abbas Asakereh, and Saman Abdanan Mehdizadeh. Fruit detection and load estimation of an orange orchard using the yolo models through simple approaches in different imaging and illumination conditions. *Computers and Electronics in Agriculture*, 191:106533, 2021.

[25] Unseok Lee, Md Parvez Islam, Nobuo Kochi, Kenichi Tokuda, Yuka Nakano, Hiroki Naito, Yasushi Kawasaki, Tomohiko Ota, Tomomi Sugiyama, and Dong-Hyuk Ahn. An automated, clip-type, small internet of things camera-based tomato flower and fruit monitoring and harvest prediction system. *Sensors*, 22(7):2456, 2022.

[26] Hongyan Zhu, Shuai Qin, Min Su, Chengzhi Lin, Anjie Li, and Junfeng Gao. Harnessing large vision and language models in agriculture: A review. *arXiv preprint arXiv:2407.19679*, 2024.

[27] Santi Sukkasem, Watchareewan Jitsakul, and Phayung Meesad. Fruit classification with deep transfer learning using image processing. In *2023 7th International Conference on Information Technology (InCIT)*, pages 464–469. IEEE, 2023.

[28] Yonis Gulzar. Fruit image classification model based on mobilenetv2 with deep transfer learning technique. *Sustainability*, 15(3):1906, 2023.

[29] Xiaohong Kou, Yuan Feng, Shuai Yuan, Xiaoyang Zhao, Caie Wu, Chao Wang, and Zhaohui Xue. Different regulatory mechanisms of plant hormones in the ripening of climacteric and non-climacteric fruits: a review. *Plant Molecular Biology*, pages 1–21, 2021.

[30] Andrés Alejandro Garcés Cadena, Oswaldo Aníbal Menéndez Granizo, Edgar Patricio Córdova, and Alvaro Javier Prado Romo. Clasificación de calidad de manzana para monitoreo de cosechabilidad utilizando visión por computador y algoritmos de aprendizaje profundo. *Ingeniare. Revista chilena de ingeniería*, 31:0–0, 2023.

[31] et al. Zhang. Automatic plant phenotyping analysis of melon (cucumis melo l.) germplasm resources using deep learning methods and computer vision. *Plant Methods*, 20(1):1–15, 2024.

[32] VK Venkatkumar. Yolov8 architecture & cow counter with region based dragging using yolov8. https://medium.com/@VK_Venkatkumar/yolov8-architecture-cow-counter-with-region-based-dragging-using-yolov8-e75b3ac71ed8, 2023. Accessed: 2024-12-01.

[33] Nikhil Rao. Yolov11 explained: Next-level object detection with enhanced speed and accuracy. https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71, 2023. Accessed: 2024-12-01.

[34] Hiroto Schwert. Digging into detectron 2. https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd, 2020. Accessed: 2024-12-01.

[35] Towards Data Science. Fast r-cnn for object detection: A technical summary. https://towardsdatascience.com/fast-r-cnn-for-object-detection-a-technical-summary-a0ff94faa022, 2020. Accessed: 2024-12-01.