

## Article

# Application of AI in Date Fruit Detection—Performance Analysis of YOLO and Faster R-CNN Models

Seweryn Lipiński \* , Szymon Sadkowski and Paweł Chwietczuk 

Faculty of Technical Sciences, University of Warmia and Mazury in Olsztyn, 10-036 Olsztyn, Poland;  
szymon.sadkowski@student.uwm.edu.pl (S.S.); pawel.chwietczuk@uwm.edu.pl (P.C.)

\* Correspondence: seweryn.lipinski@uwm.edu.pl

**Abstract:** Presented study evaluates and compares two deep learning models, i.e., YOLOv8n and Faster R-CNN, for automated detection of date fruits in natural orchard environments. Both models were trained and tested using a publicly available annotated dataset. YOLO, a single-stage detector, achieved a mAP@0.5 of 0.942 with a training time of approximately 2 h. It demonstrated strong generalization, especially in simpler conditions, and is well-suited for real-time applications due to its speed and lower computational requirements. Faster R-CNN, a two-stage detector using a ResNet-50 backbone, reached comparable accuracy (mAP@0.5 = 0.94) with slightly higher precision and recall. However, its training required significantly more time (approximately 19 h) and resources. Deep learning metrics analysis confirmed both models performed reliably, with YOLO favoring inference speed and Faster R-CNN offering improved robustness under occlusion and variable lighting. Practical recommendations are provided for model selection based on application needs—YOLO for mobile or field robotics and Faster R-CNN for high-accuracy offline tasks. Additional conclusions highlight the benefits of GPU acceleration and high-resolution inputs. The study contributes to the growing body of research on AI deployment in precision agriculture and provides insights into the development of intelligent harvesting and crop monitoring systems.

**Keywords:** precision agriculture; smart farming; date fruit detection; computer vision; image recognition; deep learning; convolutional neural networks; YOLO; Faster R-CNN



Academic Editor: Xiaoqiang Hua

Received: 23 May 2025

Revised: 10 June 2025

Accepted: 13 June 2025

Published: 13 June 2025

**Citation:** Lipiński, S.; Sadkowski, S.; Chwietczuk, P. Application of AI in Date Fruit Detection—Performance Analysis of YOLO and Faster R-CNN Models. *Computation* **2025**, *13*, 149. <https://doi.org/10.3390/computation13060149>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The date palm (*Phoenix dactylifera* L.) is one of the most important fruit crops cultivated in arid and semi-arid regions, particularly across the Middle East, North Africa, and parts of South Asia [1,2]. Its fruits are not only of cultural and religious significance but also serve as a vital source of nutrition, income, and food security for millions of people worldwide [2–4]. Rich in carbohydrates, dietary fiber, essential minerals, phenolic compounds, and antioxidants, date fruits are increasingly used in health-promoting food products and functional ingredients [5–7]. Recent research has further emphasized their potential in developing value-added products such as energy bars, nutraceuticals, dietary supplements [8,9], and even nutricosmetics [10].

As global demand for premium-quality dates rises, especially within the functional food industry and export markets, modern agriculture faces growing challenges in improving the efficiency, precision, and sustainability of date production systems [6,7]. Accurate and automated fruit detection supports the principles of sustainable agriculture by reducing labor dependency, improving yield monitoring, and minimizing waste through targeted harvesting,

while traditional methods of fruit detection, yield estimation, and quality assessment remain time-consuming, labor-intensive, and prone to human error [11–13]. Moreover, manual harvesting of date fruits is not only costly but also poses safety challenges due to the considerable height of palm trees, further emphasizing the need for automated solutions. These limitations have accelerated interest in digital agriculture technologies, including machine vision and artificial intelligence (AI).

Among AI-based solutions, deep learning, particularly convolutional neural networks (CNNs), has demonstrated strong potential for automating visual tasks in agriculture [14,15]. CNN models have been successfully applied to object detection, fruit counting, ripeness estimation, and disease identification in a variety of crops [11,13,16,17]. However, detecting date fruits in natural orchard conditions presents specific challenges, including variable lighting, partial occlusion by palm leaves or protective covers, and visual similarity across cultivars and ripening stages [18–20].

Despite the growing interest in AI applications in agriculture, no studies have systematically compared the performance of single-stage and two-stage object detectors specifically for date fruit detection under realistic field conditions.

To address these challenges and knowledge gaps, this study evaluates the performance of two state-of-the-art object detection architectures, You Only Look Once (YOLO) and Faster Region-Based Convolutional Neural Network (Faster R-CNN), in detecting date fruits in diverse real-world scenarios. While both models have demonstrated robust results in general-purpose object detection tasks [21–24], their performance in complex agricultural settings, particularly in orchards with tall trees and dense foliage, remains underexplored.

The primary objective of this work is to perform a comparative analysis of YOLO and Faster R-CNN in terms of detection accuracy, inference speed, and robustness. Using a diverse, high-resolution dataset of annotated date palm images, the study aims to provide practical insights for researchers and practitioners developing AI-based tools for automated harvesting, yield estimation, and intelligent orchard management in date palm cultivation.

## 2. Materials and Methods

### 2.1. Data Source and Characteristics

For the purpose of this paper, we utilized employed publicly available dataset developed by Zarouit et al. [20], accessible via [25], which contains 9092 annotated images of date fruit clusters captured under natural lighting conditions between June and September 2022. The images were acquired using two devices: a Samsung Ultra smartphone ( $2006 \times 4128$  pixels) and a Canon digital camera ( $4288 \times 2848$  pixels). Data collection was conducted in two Moroccan date palm orchards located in the Errachidia region and in Tismoumine, Tinghir.

The dataset comprises four widely cultivated date palm varieties, i.e., *Majhoul*, *Boufagous*, *Kholt*, and *Bouisthami*, captured across four distinct ripeness stages: Immature, Khalal (mature but unripe), Rutab (ripe and soft), and Tamar (fully mature). A total of 8092 images were collected, with the following distribution by variety: *Boufagous*—4325; *Majhoul*—2645; *Kholt*—1762; and *Bouisthami*—360. By ripeness stage, the dataset includes 1452 images of Immature dates, 1687 of Khalal, 3715 of Rutab, and 2238 of Tamar.

A detailed breakdown by variety and ripeness stage is as follows:

- *Boufagous*: Immature—913, Khalal—563, Rutab—1675, Tamar—1174;
- *Majhoul*: Immature—304, Khalal—573, Rutab—1345, Tamar—423;
- *Bouisthami*: Immature—95, Khalal—137, Rutab—46, Tamar—82;
- *Kholt*: Immature—140, Khalal—414, Rutab—649, Tamar—559.

The dataset features significant variability in viewing angles, distances, lighting conditions, and occlusion sources, including shadows, protective harvesting bags, and overlap-

ping palm leaves. This diversity enhances its suitability for benchmarking object detection algorithms in realistic and challenging agricultural environments.

Annotations were created using the LabelImg 1.8.6 tool and follow the YOLO format, specifying the object class along with bounding box coordinates and dimensions. Since the annotations were provided by the dataset authors, we did not perform additional labeling or consistency verification. Inter-annotator agreement metrics were not reported in the original publication [20].

No additional data augmentation (like e.g., rotation, cropping, or color perturbation) was applied; all images were only resized to the target input resolution, and the effect of resolution was then investigated in subsequent experiments.

Representative images from the dataset will be included in later sections of the paper to illustrate specific detection scenarios; they are not presented here to avoid redundancy.

## 2.2. Hardware Specification

The experiments and model training were conducted on a desktop computer with the following hardware configuration:

- Processor: AMD Ryzen 7 3800X with 8 cores and a base clock speed of 3.9 GHz;
- Graphics Card: NVIDIA GeForce RTX 2060;
- RAM: 16 GB;
- Storage: WDC WD10EZEX—00WN4A0 (1 TB HDD).

This hardware setup provided sufficient computational resources for training and evaluating both the YOLO and Faster R-CNN models within reasonable timeframes.

## 2.3. Development Environment and Libraries

The following libraries and frameworks were employed throughout the project:

- PyTorch 2.6—used for implementing and training deep learning models, including Faster R-CNN;
- YOLOv8n—utilized for object detection tasks (this lightweight variant of the YOLOv8 architecture is specifically designed for efficient inference on devices with limited computational resources; it is well-suited for real-time applications in environments such as mobile, IoT, or GPU-less systems, where speed and efficiency are prioritized over maximum accuracy);
- scikit-learn—applied for model evaluation;
- pycocotools—used for handling COCO-format datasets and evaluating Faster R-CNN performance;
- Matplotlib 3.10.0 and Pandas 2.2.3—employed for data visualization and analysis;
- Pillow—used for image preprocessing and manipulation.

To enable GPU acceleration, CUDA and cuDNN were installed and configured. These components are essential for allowing deep learning frameworks such as PyTorch to utilize NVIDIA GPU hardware effectively.

All code was written in Python 3.12, a widely adopted language in machine learning and deep learning due to its extensive ecosystem and compatibility with the aforementioned libraries. Development and experimentation were conducted within the Jupyter Notebook 7.3.0 environment.

## 2.4. Preparation of Training, Validation, and Test Data

A critical component in the development and optimization of any deep learning model is the careful preparation and organization of the datasets used for training, validation, and evaluation. Two widely used approaches to dataset partitioning are commonly employed in machine learning workflows [26,27].

The first approach divides the dataset into two subsets: a training set, used to fit the model parameters, and a validation set, used to monitor the model's performance during training and guide hyperparameter tuning.

The second, more comprehensive strategy introduces an additional test set, which is reserved exclusively for final evaluation on previously unseen data. This ensures an unbiased assessment of the model's generalization capabilities. The use of a separate test set is particularly important in preventing overfitting to the validation set and in obtaining a realistic estimate of model performance in real-world deployment scenarios.

In the present study, this three-way split approach was adopted: both models were trained on the training set, validated throughout the training process using the validation set, and ultimately evaluated based on their performance on the independent test set.

The dataset was partitioned as follows: 60% for training (5456 images), 20% for validation (1818 images), and 20% for testing (1818 images).

### 2.5. Used Metrics

A fixed confidence threshold of 0.25 was applied uniformly across all models during inference to ensure standardized and fair comparisons. No additional tuning of this parameter was performed.

Precision and recall are fundamental metrics in object detection. Precision quantifies the proportion of correctly identified objects among all detections, reflecting the model's ability to minimize false positives. Recall, in contrast, measures the proportion of actual objects that were correctly detected, indicating the model's sensitivity to identifying all relevant instances. Together, these metrics provide insight into the balance between detection completeness and reliability.

Mean Average Precision at IoU 0.5 (mAP@0.5) is a widely adopted benchmark for object detection performance. It represents the average precision across all object classes, calculated at a fixed Intersection over Union (IoU) threshold of 0.5. A detection is considered correct if the predicted bounding box overlaps with the ground truth by at least 50%. This metric captures both localization and classification accuracy, with higher values indicating superior overall performance.

To provide a more comprehensive assessment, mAP@0.5:0.95 is also reported. This metric averages mAP values across multiple IoU thresholds (from 0.5 to 0.95 in 0.05 increments), offering a stricter evaluation of localization precision. A high mAP@0.5:0.95 score suggests that the model not only detects objects effectively but also places bounding boxes with high spatial accuracy.

These metrics were employed to evaluate and compare the performance of the models.

## 3. AI Models Description

In the context of automated date fruit detection, key requirements include both detection speed and precision. The system must be capable of identifying fruits under varying lighting conditions (depending on the time of day, example given in Figure 1), as well as in complex visibility scenarios [28–30].

A particularly critical challenge in our case is the model's ability to handle cases where fruits are partially occluded by palm leaves (example given in Figure 2) [31,32]. These factors make the choice of appropriate AI model architecture essential in order to achieve reliable results.

As previously noted, this study centers on the implementation and comparison of two deep learning architectures—YOLO and Faster R-CNN. The following subsections provide a brief overview of each model.



**Figure 1.** Example of date fruits under varying lighting conditions.



**Figure 2.** Example of partially occluded date fruits.

### 3.1. YOLO

YOLO is a single-stage object detection model that reframes the detection task as a regression problem. It divides the input image into a grid, and for each grid cell, it predicts bounding boxes and class probabilities in one forward pass. This unified approach enables YOLO to perform object detection with speed, making it suitable for real-time applications such as e.g., robotic harvesting.

The version used in this study is YOLOv8n, which builds upon previous versions by incorporating several architectural improvements, including anchor-free detection, decoupled heads, and improved backbone networks. YOLOv8 also supports flexible input dimensions and is optimized for performance across a wide range of edge devices [33,34].

Key advantages of YOLO include:

- high inference speed;
- efficient use of computational resources;
- good performance on large and well-separated objects.

However, YOLO may exhibit limitations when dealing with:

- small, densely clustered objects;
- heavily occluded objects;
- fine-grained distinctions between similar classes.

Despite these challenges, YOLO provides a strong baseline for fast and efficient detection, which is critical in agricultural applications requiring real-time decision-making.

### 3.2. Faster R-CNN

Faster R-CNN is a two-stage object detection framework known for its high detection accuracy and robustness in complex visual environments. The model first uses a convolutional backbone to extract feature maps from the input image. These maps are passed on to a Region Proposal Network, which generates candidate object regions (region proposals). Each proposal is then refined and classified using a region-of-interest pooling layer and a second-stage classifier. This architecture allows Faster R-CNN to achieve state-of-the-art results on many benchmark datasets, particularly in scenarios requiring high localization precision. It is well suited for detecting partially occluded or overlapping objects—conditions often encountered in agricultural settings like date palm orchards [35,36].

The main advantages of Faster R-CNN include:

- high detection accuracy, especially on small and occluded objects;
- flexibility in backbone selection and hyperparameter tuning;
- high performance on complex datasets.

The primary limitation of Faster R-CNN is its relatively slow inference speed, which may restrict its use in time-critical applications. However, in cases where precision is prioritized over speed, such as fruit quality assessment or yield estimation, Faster R-CNN offers clear benefits.

### 3.3. Justification for Comparative Study

The decision to conduct a comparative study between YOLO and Faster R-CNN stems from the need to balance two critical performance factors in automated date fruit detection: speed and accuracy. Agricultural environments, particularly date palm orchards, present unique challenges such as variable lighting, partial occlusions from palm leaves, and densely clustered fruits [36,37]. These conditions demand a robust detection system capable of maintaining high performance across diverse scenarios.

By comparing these two architectures under the same dataset and environmental conditions, this study aims to:

- Identify the strengths and limitations of each model in the context of date fruit detection;
- Determine the most suitable model for specific agricultural use cases (e.g., real-time harvesting vs. post-harvest analysis);
- Provide insights into how model selection impacts overall system performance in real-world orchard environments.

## 4. Results

### 4.1. YOLO

At first, the YOLOv8n model was loaded and trained, with the following key training parameters:

- epochs—20 full passes through the training dataset;
- batch size—16 images per iteration;
- optimizer—AdamW, with a learning rate of 0.002 and momentum of 0.9;
- resolution—input images were resized to  $256 \times 256$  pixels.

It should be stated here that to enhance reproducibility and practical applicability; we used the default hyperparameter settings provided by the official implementations of YOLOv8 and Faster R-CNN. No additional optimization was performed. This approach reflects a realistic use case where models are deployed with minimal manual intervention.

After 20 training epochs, the model achieved a precision of 91.5% and a recall of 79.4%.

Representative detection results are shown in Figure 3, illustrating the model's effectiveness in identifying date fruits. Encouraged by the consistent performance gains and strong generalization capabilities, evidenced by increasing precision and steadily improving recall, a decision was made to extend the training process. To this end, the model weights saved after 20 epochs were reloaded, and training was continued for an additional 10 epochs to further enhance performance.



**Figure 3.** First detection results on test images—five exemplary test images of date palm trees with detected date fruits highlighted using blue bounding boxes. Each detection is labeled with a confidence score, indicating the model's certainty in identifying the object.

Given the consistent progress and strong generalization capabilities, reflected in increasing precision and a steady rise in sensitivity, a decision was made to continue training the model. This process involved loading the weights saved after 20 epochs and extending the training by an additional 20 epochs, carried out in two stages (10 + 10 epochs). Exemplary results from this extended training phase are presented in Figure 4.

A comparison of Figures 3–5 reveals a progressive increase in the model's confidence when identifying date fruits.

Performance metrics improved accordingly, with precision and recall reaching 91.5% and 79.4% after 30 epochs, and further increasing to 93.1% and 79.7% after 40 epochs, respectively. To explore the model's learning dynamics further, training was extended. However, analysis of training set metrics after 50 epochs revealed a decline in precision from 0.931 to 0.901, despite a continued increase in recall from 0.797 to 0.812. This trade-off

suggests the onset of overfitting, where the model begins to memorize training data at the expense of generalization [38,39].

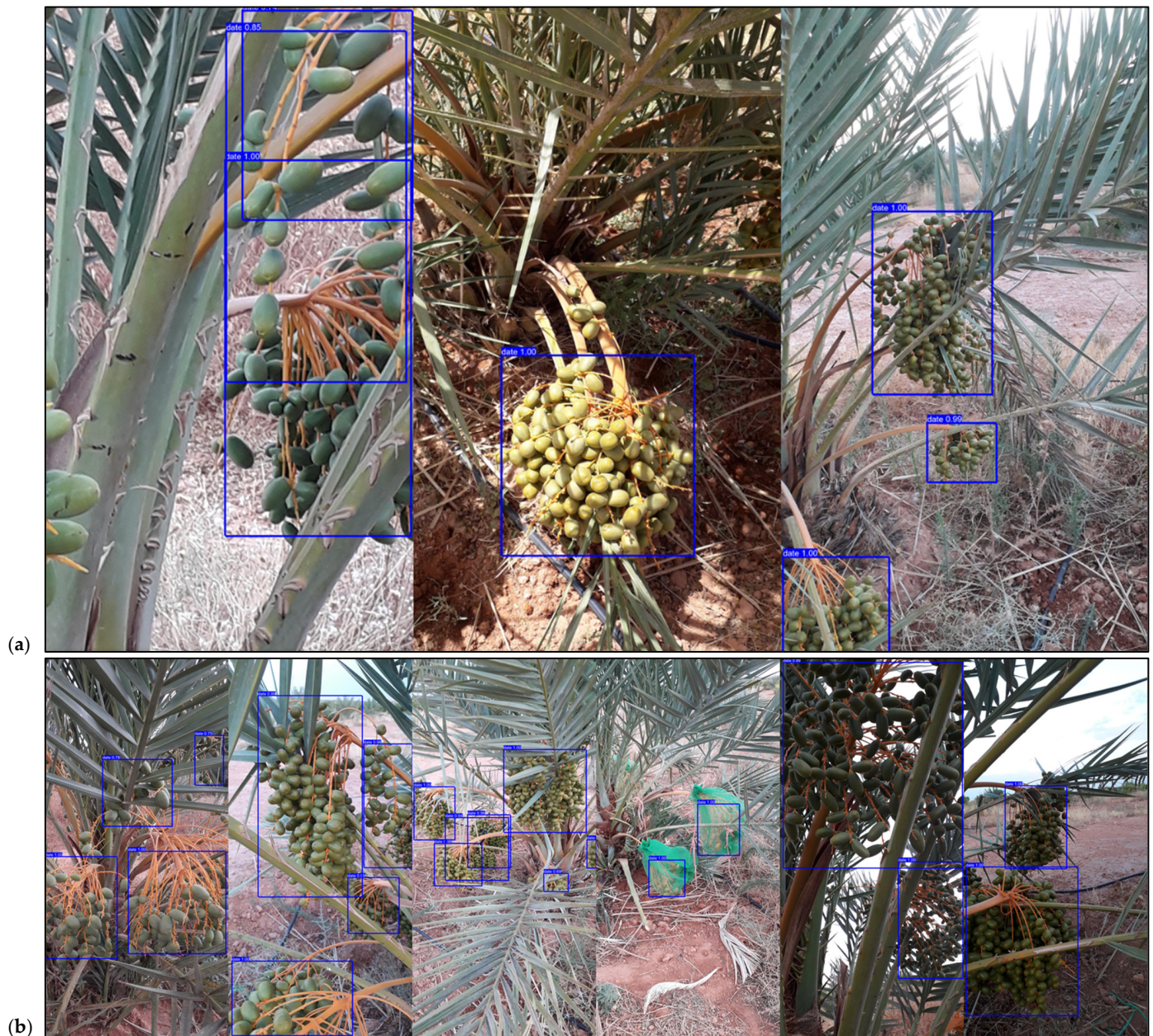


**Figure 4.** Exemplary detection results after 30 (a) and 40 (b) epochs.

Consequently, it was concluded that additional training beyond this point would likely yield diminishing returns or even degrade certain aspects of performance, particularly precision.

Subsequent experiments investigated the impact of input image resolution. The model was retrained using input sizes of  $416 \times 416$ ,  $512 \times 512$ , and  $640 \times 640$  pixels. The best results were obtained with  $512 \times 512$  and  $640 \times 640$  resolutions, achieving precision scores of 92.0% and 91.8%, and a recall of 88.4% in both cases. Notably, higher-resolution inputs ( $640 \times 640$ ) reached optimal performance after only 10 epochs, whereas lower-resolution inputs ( $416 \times 416$ ) required 20 epochs.

Final evaluation yielded a precision of 92%, recall of 88.4%, mAP@0.5 of 0.893, and mAP@0.5:0.95 of 0.581—indicating strong detection accuracy and reliable localization across varying IoU thresholds. These results confirm the YOLOv8 model's suitability for practical agricultural applications, particularly after resolution optimization.



**Figure 5.** Exemplary detection results using the Faster R-CNN model: (a) easy cases, where date fruits are close to the camera and unobstructed; (b) challenging cases, where date fruits are distant, partially occluded, placed inside bags, or captured under varying lighting conditions.

#### 4.2. Faster R-CNN

This subsection presents the training process of the Faster R-CNN model, which differs from YOLO in terms of computational complexity and implementation. The primary objective was to compare the performance of both models under similar conditions. While Faster R-CNN is well-regarded for its accuracy in object detection tasks, it requires more advanced configuration and significantly greater computational resources. This was reflected in the training duration, which totaled approximately 19 h, i.e., substantially longer than the 2 h required for YOLO.

The model was implemented using a pretrained ResNet-50 backbone—a deep convolutional neural network known for its 50-layer architecture and strong performance in image classification tasks. The use of residual connections in ResNet-50 helps mitigate the vanishing gradient problem, enabling more effective training of deep networks [39]. Lever-

aging pretrained weights from large-scale datasets such as ImageNet provided a strong initialization, which accelerated convergence and improved generalization, particularly important given the limited size of our training dataset [40,41]. All layers of the backbone were fine-tuned during training, as no layer freezing was applied. This full fine-tuning strategy was adopted to allow the model to better adapt to the specific characteristics of the dataset used.

The training configuration included a learning rate of 0.005, momentum of 0.9, and weight decay of 0.0005. A learning rate scheduler reduced the learning rate by a factor of 10 every three epochs ( $\gamma = 0.1$ ). The model was trained for 10 epochs, with accuracy metrics stabilizing around epochs 6–7.

Faster R-CNN employed a unified loss function that jointly optimized object localization and classification. Unlike YOLO, which separates these components, this integrated approach enabled the model to achieve high detection quality early in training—reaching a mAP@0.5 of 0.9219 in the first epoch and stabilizing above 0.940 in subsequent epochs. Final evaluation yielded a precision of 90.1%, recall of 81.2%, mAP@0.5 of 0.893, and mAP@0.5:0.95 of 0.581, confirming the model’s strong performance in both classification and localization tasks.

Exemplary results are shown in Figure 5, where the model consistently identified date fruits with high confidence. In most cases, detection confidence reached 100%, particularly when fruits were fully visible and lighting conditions were favorable. Slightly lower confidence scores (above 80%) were observed in cases of partial occlusion or suboptimal lighting. These results underscore the model’s robustness and reliability in realistic orchard conditions.

## 5. Discussion

Two object detection models, i.e., YOLOv8n and Faster R-CNN, were trained and evaluated for the task of date detection. Both models were assessed based on training and validation performance, as well as testing on a separate dataset.

The YOLOv8 model completed training in approximately 2 h. It utilized three distinct loss functions (Box Loss, Cls Loss, and DFL Loss), corresponding to bounding box regression, classification, and distribution-focused localization. The model demonstrated consistent improvement across epochs, particularly in mAP@0.5, precision, and recall, ultimately achieving a mAP@0.5 of 0.942, with final precision and recall values of 0.917 and 0.883, respectively. Validation results confirmed the model’s effectiveness, with decreasing loss values indicating successful convergence.

In contrast, the Faster R-CNN model required approximately 19 h of training. It employed a unified loss function that jointly optimized classification and localization. Using a pretrained ResNet-50 backbone, the model achieved high detection quality early in training, with mAP@0.5 reaching 0.9219 in the first epoch and stabilizing above 0.940. Final precision and recall were 0.901 and 0.812, respectively.

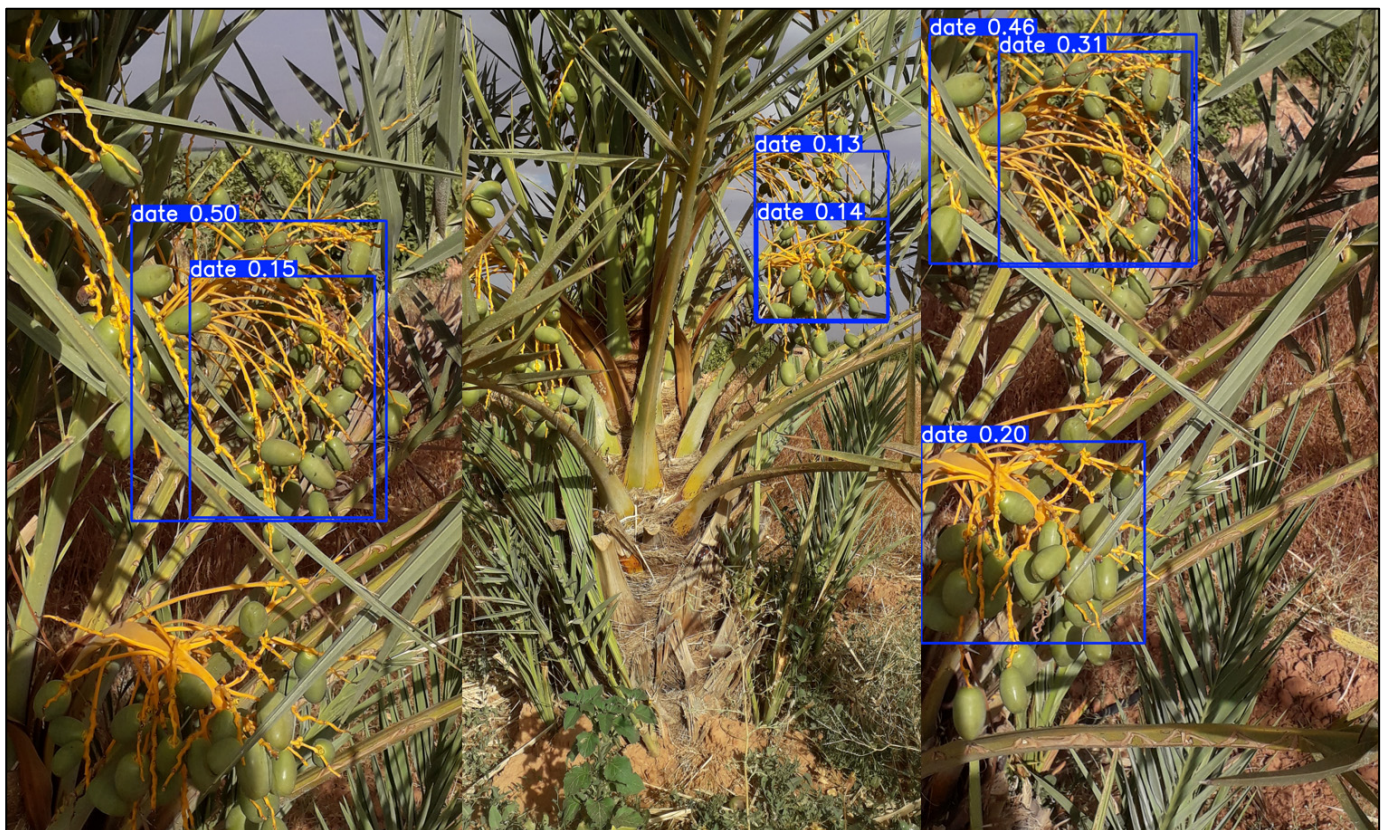
While Faster R-CNN slightly outperformed YOLOv8 in terms of mAP@0.5 and early detection confidence, YOLOv8 demonstrated superior training efficiency and flexibility. This makes YOLOv8 particularly suitable for scenarios with limited computational resources or strict time constraints, as further confirmed by the inference time analysis below.

On the experimental setup used, the average inference time per image (input size  $640 \times 640$  px) was approximately 20 ms for YOLOv8, corresponding to a throughput of around 50 FPS, confirming its suitability for real-time object detection applications. In contrast, Faster R-CNN required about 320 ms per image, achieving roughly 3 FPS, which significantly limits its applicability in real-time scenarios.

A key architectural distinction lies in the loss function design: YOLOv8's modular loss components allow for more granular optimization, whereas Faster R-CNN's unified loss may contribute to greater training stability.

In test set evaluations, Faster R-CNN demonstrated high confidence in detecting date fruits under favorable conditions. YOLOv8 also performed well overall, though it showed slightly reduced confidence in more challenging scenarios. Specifically, the model exhibited lower detection accuracy in cases involving small object clusters and partial occlusion i.e., situations that were common in the dataset

Figure 6 presents representative examples of missed detections and bounding box imprecision in such cases, highlighting a limitation of the YOLOv8 model in handling dense or visually ambiguous scenes.

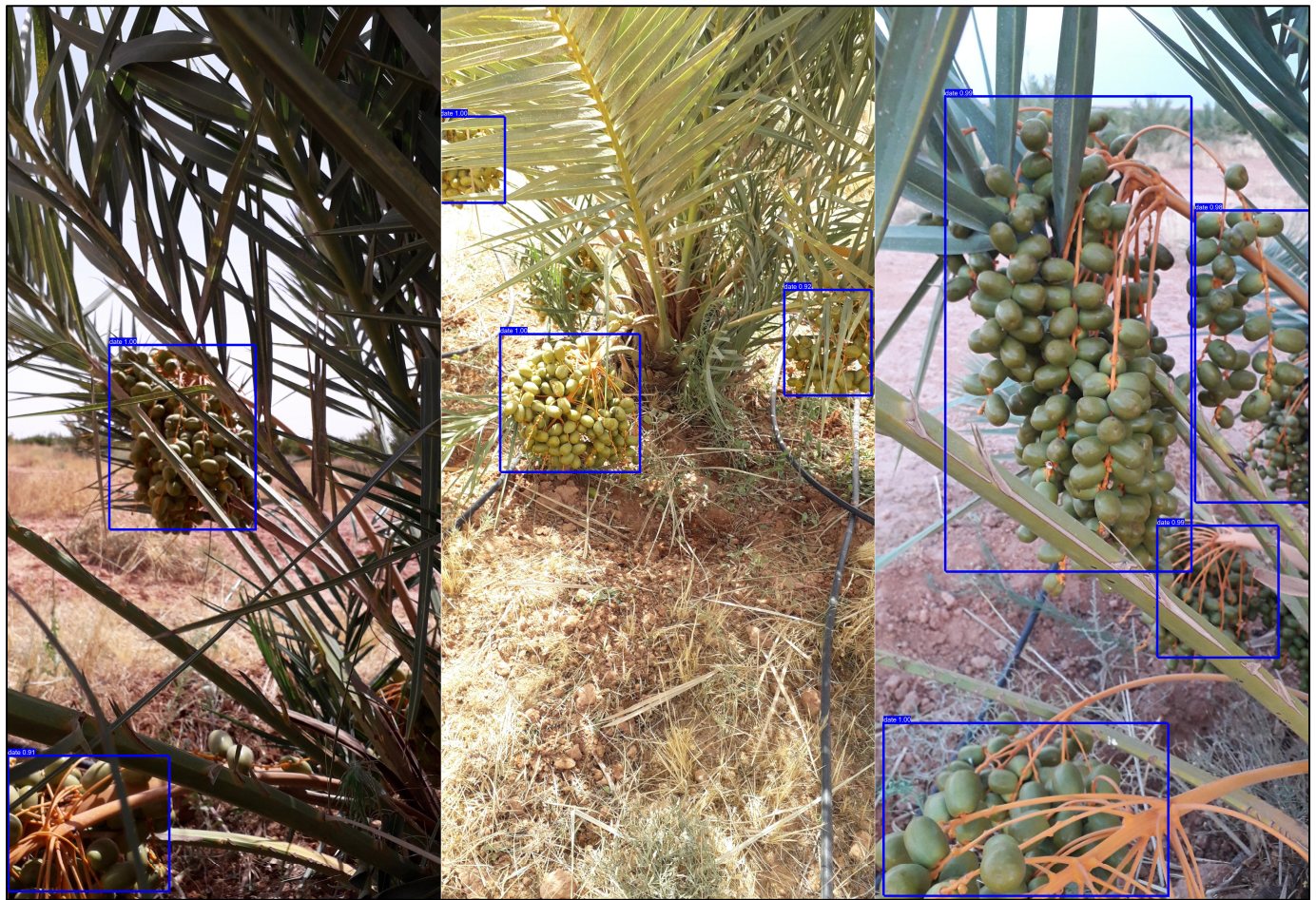


**Figure 6.** Examples of missed detections and bounding box inaccuracies produced by YOLOv8n in scenes with dense object clusters or partial occlusions. These cases highlight the model's limitations in handling visually ambiguous or complex spatial arrangements.

Figure 7, in contrast, shows the detection results obtained using the Faster R-CNN model, which demonstrate significantly improved accuracy and consistency in identifying date fruits under similar occlusion conditions.

These findings align with previous research [23,28,35,36,42], confirming the effectiveness and specificity of both models used. Ultimately, the choice between YOLOv8 and Faster R-CNN should be guided by specific application requirements—whether prioritizing precision or computational efficiency.

A detailed comparison of both models is presented in Table 1—it provides a concise overview of key characteristics such as inference speed, accuracy under various conditions, computational requirements, and recommended use cases, supporting informed decision-making for practical deployment.



**Figure 7.** Examples of bounding boxes produced by Faster R-CNN in scenes with dense object clusters or partial occlusions. These results demonstrate the model’s robustness in handling visually complex and ambiguous environments.

**Table 1.** Comparative Summary of YOLO and Faster R-CNN for date fruit detection.

| Criterion                             | YOLO                                                                                       | Faster R-CNN                                                            |
|---------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Inference speed                       | High—well-suited for near real-time detection in date palm orchards                        | Lower—less suited for time-sensitive applications                       |
| Accuracy in simple scenes             | Good performance under clear visibility and limited occlusion                              | High accuracy even in well-structured scenes                            |
| Performance in challenging conditions | May face limitations with small or partially occluded fruits (e.g., hidden by palm leaves) | More robust in detecting fruits under occlusion or varied lighting      |
| Computational requirements            | Relatively low—can be deployed on mobile or edge devices                                   | High—requires more processing power and memory                          |
| Ease of training                      | Easier to implement and optimize; faster training times                                    | More complex training process with additional tuning steps              |
| Suitability for field use             | Appropriate for real-time monitoring in palm groves and mobile robotics                    | Better suited for offline analysis and quality control tasks            |
| Recommended use cases                 | Rapid fruit localization in open-field monitoring or autonomous systems                    | Detailed detection and assessment in research or post-harvest scenarios |

### *Study Limitations*

One limitation of this study is the use of the YOLOv8 model, which, despite its efficiency and widespread adoption, is not the most recent version in the YOLO family. Newer models such as YOLOv9, YOLOv10, and YOLOv11 introduce architectural improvements and may offer enhanced performance in certain detection tasks. At the time of model selection, YOLOv8 was chosen for its stability, comprehensive documentation, and computational efficiency, which aligned well with the requirements of our application. Nonetheless, future work should consider evaluating these newer models to assess potential gains in accuracy and generalization.

Another limitation is the evaluation of only one configuration per object detection model family. Specifically, we used YOLOv8n and Faster R-CNN with a ResNet-50 backbone. While these variants offer a practical balance between accuracy and efficiency, they do not represent the full range of available architectures. Larger YOLOv8 models (e.g., YOLOv8m, YOLOv8l, YOLOv8x) or Faster R-CNN variants with alternative backbones could yield different results. Future studies should explore multiple configurations to enable more robust comparisons and generalizations.

Additionally, training was conducted using a single random seed, which may affect the robustness of the results due to the stochastic nature of deep learning. Incorporating multiple seeds or cross-validation in future work would improve the reliability and reproducibility of the findings.

While we report standard performance metrics and discuss trends in model convergence, no formal statistical analyses, such as confidence intervals or significance testing, were performed. As a result, the observed differences between models should be interpreted with caution. This limitation arises from the use of single-run evaluations, which may not fully capture variability due to random initialization or data shuffling.

Finally, since we used a publicly available annotated dataset [20,25], we had no control over the annotation process. The absence of reported inter-annotator agreement limits our ability to assess label consistency, which may affect the reliability of model evaluation.

## **6. Summary and Conclusions**

Our paper evaluated two distinct neural network-based approaches for date fruit detection: YOLOv8 and Faster R-CNN. Both models were trained, tested, and validated using a dedicated dataset collected under natural orchard conditions to assess their detection capabilities.

### *6.1. Advantages and Limitations of the Proposed Approaches*

YOLOv8 and Faster R-CNN differ significantly in architecture and detection strategy. YOLOv8 is a single-stage detector that divides an image into a grid and predicts bounding boxes and class probabilities in real time. Its main advantage lies in speed and efficiency, making it ideal for low-latency applications such as field monitoring. However, it may struggle with small, overlapping, or partially occluded objects and is sensitive to loss function tuning.

Faster R-CNN, in contrast, employs a two-stage pipeline consisting of a Region Proposal Network followed by classification and bounding box regression. This architecture enables higher precision, especially under challenging conditions like low lighting or occlusion. However, it is computationally intensive and less suitable for real-time or edge-based applications.

### *6.2. Potential Practical Applications*

YOLOv8's speed and low resource requirements make it well-suited for real-time agricultural monitoring, including mobile applications and autonomous field robots. Faster R-CNN,

despite its higher computational cost, is more appropriate for precision-critical tasks such as post-harvest quality assessment or monitoring under variable environmental conditions.

### 6.3. Future Development Directions

Future work may focus on:

- Expanding the dataset to include e.g., various date cultivars and ripening stages to enhance model robustness;
- Introducing multi-class detection (e.g., by fruit type or maturity level) for more detailed classification;
- Increasing input image resolution to improve detection accuracy (while carefully managing loss function behavior—particularly in YOLOv8).

### 6.4. Additional Observations

- GPU acceleration (e.g., using an RTX 2060) significantly reduced training time—by approximately an order of magnitude compared to CPU-only training;
- Higher input image resolution positively impacted model performance, improving detection accuracy and reliability, but only to a certain extent.

### 6.5. Summary

In conclusion, both YOLOv8 and Faster R-CNN offer distinct advantages and can be effectively applied in agricultural scenarios depending on specific project requirements. YOLOv8 excels in speed and deployment flexibility, while Faster R-CNN provides superior accuracy in complex conditions. Further improvements through parameter tuning, dataset augmentation, and class expansion could enhance their practical value in smart farming applications.

This comparative analysis not only supports informed model selection for date fruit detection but also contributes to broader research on the application of deep learning in precision agriculture, particularly growing body of research on AI-driven precision agriculture.

**Author Contributions:** Conceptualization, S.L.; methodology, S.L.; software, S.S.; validation, S.L., S.S. and P.C.; formal analysis, S.L.; investigation, S.S.; resources, S.L. and S.S.; data curation, S.L., S.S. and P.C.; writing—original draft preparation, S.L., S.S. and P.C.; writing—review and editing, S.L.; visualization, S.L., S.S. and P.C.; supervision, S.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available in Zenodo at <https://zenodo.org/records/8315235> (accessed on 20 May 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Chao, C.T.; Krueger, R.R. The date palm (*Phoenix dactylifera* L.): Overview of biology, uses, and cultivation. *HortScience* **2007**, *42*, 1077–1082. [CrossRef]
2. Zaid, A. *Date Palm Cultivation*, 2nd ed.; Food and Agricultural Organization of the United Nations: Rome, Italy, 2024.
3. Al-Karmadi, A.; Okoh, A.I. An overview of date (*Phoenix dactylifera*) fruits as an important global food resource. *Foods* **2024**, *13*, 1024. [CrossRef]
4. Soomro, A.H.; Marri, A.; Shaikh, N. Date palm (*Phoenix dactylifera*): A review of economic potential, industrial valorization, nutritional and health significance. In *Neglected Plant Foods of South Asia: Exploring and Valorizing Nature to Feed Hunger*; Tariq, I., Akhtar, S., Lazarte, C.E., Eds.; Springer: Cham, Switzerland, 2023; pp. 319–350.
5. Hamad, I.; Abdelgawad, H.; Al Jaouni, S.; Zinta, G.; Asard, H.; Hassan, S.; Hegab, M.; Hagagy, N.; Selim, S. Metabolic analysis of various date palm fruit (*Phoenix dactylifera* L.) cultivars from Saudi Arabia to assess their nutritional quality. *Molecules* **2015**, *20*, 13620–13641. [CrossRef]

6. Muñoz-Tebar, N.; Viuda-Martos, M.; Lorenzo, J.M.; Fernandez-Lopez, J.; Perez-Alvarez, J.A. Strategies for the valorization of date fruit and its co-products: A new ingredient in the development of value-added foods. *Foods* **2023**, *12*, 1456. [CrossRef]
7. Rambabu, K.; Bharath, G.; Hai, A.; Banat, F.; Hasan, S.W.; Taher, H.; Mohd Zaid, H.F. Nutritional quality and physico-chemical characteristics of selected date fruit varieties of the United Arab Emirates. *Processes* **2020**, *8*, 256. [CrossRef]
8. Barakat, H.; Alfheaid, H.A. Date palm fruit (*Phoenix dactylifera*) and its promising potential in developing functional energy bars: Review of chemical, nutritional, functional, and sensory attributes. *Nutrients* **2023**, *15*, 2134. [CrossRef]
9. Fernández-López, J.; Viuda-Martos, M.; Sayas-Barberá, E.; Navarro-Rodríguez de Vera, C.; Pérez-Álvarez, J.Á. Biological, nutritive, functional and healthy potential of date palm fruit (*Phoenix dactylifera* L.): Current research and future prospects. *Agronomy* **2022**, *12*, 876. [CrossRef]
10. Alharbi, K.L.; Raman, J.; Shin, H.J. Date fruit and seed in nutricosmetics. *Cosmetics* **2021**, *8*, 59. [CrossRef]
11. Anjali; Jena, A.; Bamola, A.; Mishra, S.; Jain, I.; Pathak, N.; Sharma, N.; Joshi, N.; Pandey, R.; Kaparwal, S.; et al. State-of-the-art non-destructive approaches for maturity index determination in fruits and vegetables: Principles, applications, and future directions. *Food Prod. Process. Nutr.* **2024**, *6*, 56. [CrossRef]
12. Mohammed, M.; Munir, M.; Aljabr, A. Prediction of date fruit quality attributes during cold storage based on their electrical properties using artificial neural networks models. *Foods* **2022**, *11*, 1666. [CrossRef]
13. Mohyuddin, G.; Khan, M.A.; Haseeb, A.; Mahpara, S.; Waseem, M.; Saleh, A.M. Evaluation of Machine Learning approaches for precision farming in Smart Agriculture System—A comprehensive Review. *IEEE Access* **2024**, *12*, 60155–60184. [CrossRef]
14. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 1–74. [CrossRef] [PubMed]
15. Taye, M.M. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation* **2023**, *11*, 52. [CrossRef]
16. Chu, X.; Miao, P.; Zhang, K.; Wei, H.; Fu, H.; Liu, H.; Jiang, H.; Ma, Z. Green Banana maturity classification and quality evaluation using hyperspectral imaging. *Agriculture* **2022**, *12*, 530. [CrossRef]
17. Gupta, S.; Tripathi, A.K. Fruit and vegetable disease detection and classification: Recent trends, challenges, and future opportunities. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108260. [CrossRef]
18. Santelices, I.R.; Cano, S.; Moreira, F.; Fritz, Á.P. Artificial Vision Systems for Fruit Inspection and Classification: Systematic Literature Review. *Sensors* **2025**, *25*, 1524. [CrossRef]
19. Xiao, F.; Wang, H.; Li, Y.; Cao, Y.; Lv, X.; Xu, G. 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* **2023**, *13*, 639. [CrossRef]
20. Zarouit, Y.; Zekkouri, H.; Ouhda, M.; Aksasse, B. Date fruit detection dataset for automatic harvesting. *Data Br.* **2024**, *52*, 109876. [CrossRef]
21. Bilous, N.; Malko, V.; Frohme, M.; Nechyporenko, A. Comparison of CNN-Based Architectures for Detection of Different Object Classes. *AI* **2024**, *5*, 2300–2320. [CrossRef]
22. Jiang, Q.; Jia, M.; Bi, L.; Zhuang, Z.; Gao, K. Development of a core feature identification application based on the Faster R-CNN algorithm. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105200. [CrossRef]
23. Shobaki, W.A.; Milanova, M.A. Comparative Study of YOLO, SSD, Faster R-CNN, and More for Optimized Eye-Gaze Writing. *Sci* **2025**, *7*, 47. [CrossRef]
24. Tulbure, A.A.; Tulbure, A.A.; Dulf, E.H. A review on modern defect detection models using DCNNs—Deep convolutional neural networks. *J. Adv. Res.* **2022**, *35*, 33–48. [CrossRef] [PubMed]
25. Date Fruit Detection Dataset for Computer Vision-Based Automatic Harvesting. Available online: <https://doi.org/10.5281/zenodo.8315235> (accessed on 20 May 2025). [CrossRef]
26. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*; Springer: New York, NY, USA, 2013.
27. Shobha, G.; Rangaswamy, S. Machine learning. *Handb. Stat.* **2018**, *38*, 197–228.
28. Mirhaji, H.; Soleymani, M.; Asakereh, A.; Mehdizadeh, S.A. Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions. *Comp. Electron. Agric.* **2021**, *191*, 106533. [CrossRef]
29. Sengupta, S.; Lee, W.S. Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions. *Biosyst. Eng.* **2014**, *117*, 51–61. [CrossRef]
30. Wang, H.; Zhang, G.; Cao, H.; Hu, K.; Wang, Q.; Deng, Y.; Gao, J.; Tang, Y. Geometry-Aware 3D Point Cloud Learning for Precise Cutting-Point Detection in Unstructured Field Environments. *J. Field Robot.* **2025**. [CrossRef]
31. Shiu, Y.S.; Lee, R.Y.; Chang, Y.C. Pineapples' detection and segmentation based on faster and mask r-cnn in uav imagery. *Remote Sens.* **2023**, *15*, 814. [CrossRef]

32. Tang, Y.; Qiu, J.; Zhang, Y.; Wu, D.; Cao, Y.; Zhao, K.; Zhu, L. Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: A review. *Precis. Agric.* **2023**, *24*, 1183–1219. [\[CrossRef\]](#)
33. Hussain, M. Yolov1 to v8: Unveiling each variant—a comprehensive review of yolo. *IEEE Access* **2024**, *12*, 42816–42833. [\[CrossRef\]](#)
34. Kang, S.; Hu, Z.; Liu, L.; Zhang, K.; Cao, Z. Object Detection YOLO Algorithms and Their Industrial Applications: Overview and Comparative Analysis. *Electronics* **2025**, *14*, 1104. [\[CrossRef\]](#)
35. Ezzeddini, L.; Ktari, J.; Frikha, T.; Alsharabi, N.; Alayba, A.; Alzahrani, A.J.; Jadi, A.; Alkholidi, A.; Hamam, H. Analysis of the performance of Faster R-CNN and YOLOv8 in detecting fishing vessels and fishes in real time. *PeerJ Comput. Sci.* **2024**, *10*, e2033. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2020**, *168*, 107036. [\[CrossRef\]](#)
37. Noutfia, Y.; Ropelewska, E. What can artificial intelligence approaches bring to an improved and efficient harvesting and postharvest handling of date fruit (*Phoenix dactylifera* L.)? A review. *Postharvest Biol. Technol.* **2024**, *213*, 112926. [\[CrossRef\]](#)
38. Fan, C.L.; Chung, Y.J. Design and optimization of CNN architecture to identify the types of damage imagery. *Mathematics* **2022**, *10*, 3483. [\[CrossRef\]](#)
39. Krohn, J.; Beyleveld, G.; Bassens, A. *Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence*; Addison-Wesley Professional: Boston, MA, USA, 2019.
40. Koonce, B. ResNet 50. In *Convolutional Neural Networks with Swift for Tensorflow. Image Recognition and Dataset Categorization*; Koonce, B., Ed.; APress: New York, NY, USA, 2021; pp. 63–72.
41. Zhang, X.; Li, H.; Sun, S.; Zhang, W.; Shi, F.; Zhang, R.; Liu, Q. Classification and identification of apple leaf diseases and insect pests based on improved ResNet-50 model. *Horticulturae* **2023**, *9*, 1046. [\[CrossRef\]](#)
42. Wang, H.; Xu, X.; Liu, Y.; Lu, D.; Liang, B.; Tang, Y. Real-time defect detection for metal components: A fusion of enhanced canny–devernay and YOLOv6 algorithms. *Appl. Sci.* **2023**, *13*, 6898. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.