

Article

Not peer-reviewed version

Deep Learning Object Detection During Pepper Harvesting: Comparing YOLO, Faster R-CNN, and SSD with Multiple Feature Extractors

Piero Vilcapoma , [Ingrid Nicole Vásconez](#) ^{*} , [Alvaro Javier Prado](#) , Viviana Moya , [Leonardo Guevara](#) , [Juan Pablo Vásconez](#) ^{*}

Posted Date: 28 May 2025

doi: 10.20944/preprints202505.2226.v1

Keywords: pepper; harvesting; object detection; convolutional neural networks



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Article

Deep Learning Object Detection During Pepper Harvesting: Comparing YOLO, Faster R-CNN, and SSD with Multiple Feature Extractors

Piero Vilcapoma ¹, Ingrid Nicole Vásconez ^{2,*}, Alvaro Javier Prado ³, Viviana Moya ⁴, Leonardo Guevara ⁵ and Juan Pablo Vásconez ^{1,6,*,†}

¹ Energy Transformation Center, Faculty of Engineering, Universidad Andres Bello, Santiago 7500971, Chile

² Centro de Biotecnología Vegetal, Facultad de Ciencias de la Vida, Universidad Andrés Bello, Santiago, Chile

³ Departamento de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Antofagasta 1249004, Chile

⁴ Dep. Automatización y Control Industrial, Escuela Politécnica Nacional, Quito, Ecuador

⁵ Lincoln Institute for Agri-Food Technology, University of Lincoln, Lincoln, LN2 2LG, UK

⁶ ANID - Millennium Nucleus in Data Science for Plant Resilience (PhytoLearning)

* Correspondence: ingrid.vasconez@sansano.usm.cl (I.N.V.); juan.vasconez@unab.cl (J.P.V.)

† Current address: Energy Transformation Center, Faculty of Engineering, Universidad Andres Bello, Santiago 7500971, Chile.

Abstract: Deep learning (DL) methods based on Convolutional Neural Networks (CNNs) have gained great relevance in agricultural applications in the last years. Localizing and categorizing multiple relevant objects in images during harvesting tasks can improve decision-making for different applications, such as semantic action recognition, robotic social navigation, among others. Approaches based on object detection, semantic segmentation, and instance segmentation are still being studied with great interest for these applications. However, several challenges still need to be faced, such as the selection of an adequate algorithm, the dataset distribution, and the application complexity, among others. In this work, we propose to compare the performance of three different image-based object detection CNNs based on a dataset obtained during pepper harvesting tasks. The detection methods are YOLOv7, Faster R-CNN, and Single Shot MultiBox Detector (SSD). For each detector, we tested different feature extraction models, including GoogLeNet, ResNet-18, and ShuffleNet. The detected object categories are humans, boxes, peppers, pepper groups, and pushcarts. We employed a total of 420 images obtained from a Chilean pepper farm during the training process. During testing, we achieved a mean average precision (mAP) value up to 99.6% for the YOLOv7 model with GoogLeNet, 91% for SSD with ShuffleNet, and 87% for Faster R-CNN with ResNet-18, respectively. These results demonstrate that it is possible to successfully implement CNNs for the detection of multiple objects during pepper harvesting processes, and that YOLOv7 works particularly well for this application.

Keywords: pepper; harvesting; object detection; convolutional neural networks

1. Introduction

Pepper (*Capsicum spp.*), especially sweet pepper (*Capsicum annuum* L.), is a crop of considerable economic and nutritional significance in Latin American countries, including Chile, Mexico, Peru, Ecuador, Colombia, and Brazil, as well as internationally in countries such as China, South Korea, Türkiye, Indonesia, and the United States [1,2]. Numerous pepper cultivars contribute to a global production of roughly 36.8 million tons in 2018 [1,2]. The cultivation of peppers necessitates precise detection and management due to their fragile structures, diverse growth patterns, and colors (red, green, yellow, and their combinations), as well as potential obstruction by leaves or stems in both open-field and greenhouse settings [3,4]. The challenges are especially evident in automated harvesting, where it is crucial to differentiate plant parts (e.g., fruits, stems, leaves) from backdrops of similar colors or overlapping canopies [5–7].

The growing development of technologies for agricultural applications based on artificial intelligence (AI), Machine Learning (ML), and Deep Learning (DL) has increased researchers' interest in providing solutions based on data for different agricultural applications in pepper farms in the last years [1,3,8]. For example, pepper seeding, irrigation, fertilization, harvesting, and pruning are key tasks that could benefit from automated monitoring [9,10]. Smart sensing technologies based on DL and ML have been proposed to fulfill various tasks such as the detection of diseases in leaves, peduncle detection, pepper detection, counting, tracking, among others [1,5,6,11–15]. In pepper harvesting applications, the most employed technologies that are used to obtain data for visual recognition tasks are red-green-blue (RGB) cameras, multispectral cameras, LiDAR Sensors, near-infrared (NIR) sensors, hyperspectral sensors, thermal cameras, and depth cameras. The application for those sensors combined with DL methods represents advanced solutions that facilitate decision-making in harvesting applications [12,16–22]. For example, RGB cameras mounted on mobile robots can be used for environment monitoring, including object detection, segmentation, and classification, pepper counting for yield prediction, human detection, and action recognition for performance evaluation, among others [7,23–31]. Multispectral and hyperspectral cameras can be used with DL algorithms to analyze pepper plant health (detecting diseases) while LiDAR and depth cameras can be used for phenotyping (creating 3D representations of pepper plants) but also for a robot to navigate safely and efficiently next to humans and other unexpected obstacles in a field [7,25–32]. Despite all the development efforts, implementing IA, ML and DL algorithms for visual recognition in pepper field applications is still a challenging problem. This is partly because of the unstructured nature of agricultural environments in terms of field layouts and environmental conditions. We can also add the immense variability among all the categories of objects that can be found within a field [6,33,34].

Contemporary agriculture applications in pepper fields can use vision-based ML and DL to create intelligent systems that directly improve harvesting process productivity by obtaining useful data from field operations [1,5,6,11–15,35,36]. In particular, object detection, object segmentation, and image classification problems have been studied for pepper field operations with high-performance results. The use of DL algorithms based on Convolutional Neural Networks (CNNs) has to date one of the state-of-the-art performances for those applications [37]. For example, CNNs such as AlexNet, GoogLeNet, Inception, ResNet, DarkNet, VGGNet, and MobileNet have been successfully used as feature extraction methods, able to classify pepper images [29,38,39]. Those networks can also work well as feature extraction stages for more elaborate architectures, such as object detection methods, such as You Only Look Once (YOLO), Faster Region Convolutional Neural Networks (Faster R-CNN), Single Shot Multibox Detector (SSD), RetinaNet, EfficientDet, CenterNet, and Detection Transformer (DETR). Finally, image segmentation approaches can also be used to obtain human and object information from agricultural fields, such as Mask R-CNN, U-Net, Fully Convolutional Network (FCN), DeepLab, SegNet, RefineNet, among others [5,37,40–44]. Although these approaches have been tested on images captured during pepper harvesting, there are still several considerations and challenges to be investigated, making this field of study continue along an extensive path. The reliability of the results obtained when evaluating DL algorithms heavily depends on the application and the dataset structure, size, and variability. A detailed analysis of variance, bias, underfitting, and overfitting is also required to evaluate if the developed AI models are correctly trained and calibrated, which is a feasible indicator of the algorithm's generalization capabilities [40,45–47].

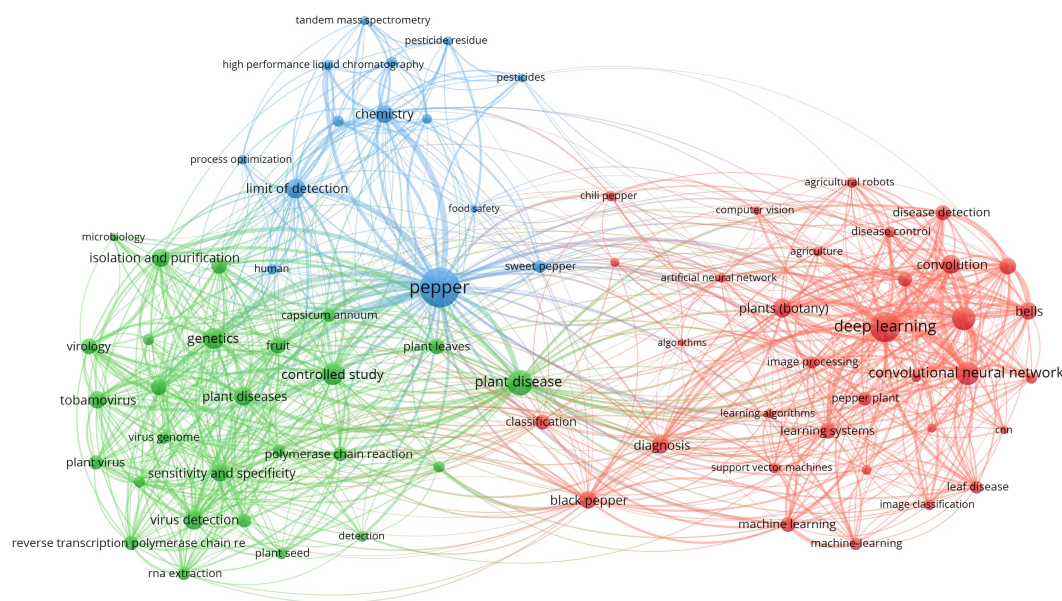


Figure 1. Conceptual map for a search in the Scopus database considering the article title criterion using the keywords "pepper" AND ("detection" OR "deep learning" OR "machine learning" OR "CNN" OR "YOLO" OR "SSD" OR "Faster R-CNN"). Literature from 2014 to May 2025.

In this context, this work addresses the problem of object detection during Pepper harvesting operations using different object detection methods based on CNNs, including YOLOv7, Faster R-CNN, and SSD. For each of the object detection algorithms, we tested three different CNN-based feature extraction models, which are GoogLeNet, ResNet-18, and ShuffleNet. Thus, we trained a total of nine different object detection models. We briefly explain this work outline as follows.

1.1. Outline

In Section 1.2, the state-of-the-art of the proposed work is presented. Section 1.3 summarizes the work related to human and object detection in agriculture applications. In Section 2, we present the proposed methodology for the detection of multiple objects during Pepper harvesting processes based on the Faster R-CNN, YOLOv7, and SSD algorithms in combination with the GoogLeNet, ResNet-18, and ShuffleNet networks as a feature extraction stage. Section 4 displays the precision-recall curves and several performance results achieved during validation and testing procedures. We also present a comparative analysis of bias, variance, overfitting, and underfitting for the proposed algorithms. Finally, we discuss the main findings in Section 4.6, as well as the conclusions in Section 5.

1.2. State-of-the-Art Review

In the literature, several studies have utilized AI algorithms for visual object detection in harvesting operations. A search in the Scopus database considering the article title criterion using the keywords "pepper" AND ("detection" OR "deep learning" OR "machine learning" OR "CNN" OR "YOLO" OR "SSD" OR "Faster R-CNN"). From these documents, Figure 1 shows an illustration that groups the most commonly used concepts and keywords as search results.

As can be observed in Figure 1, several studies related to pepper have implemented DL and ML algorithms for the detection of objects in harvesting processes in the last few years. The most popular topics in the published reports related to deep learning (in red) are CNNs, image processing, disease and virus control and detection, learning systems, image classification, computer vision, agricultural robots, among others. It is noticeable that the different research fields display a strong link between pepper and Machine Learning (ML), Deep Learning (DL), and computer vision (CV) fields. Each of the areas is briefly summarized as follows:

- Machine learning (ML) is a subfield of IA which comprises different algorithms or models capable of "learning" patterns from data to make predictions [48].
- Deep learning (DL) is a subdivision of ML that implements deep neural networks (DNNs) with multiple layers. When the data set of the study is large, it is suggested to implement DNNs since they are efficient in processing large volumes of data, being especially employed in computer vision problems by using convolutional neural networks (CNNs). In tasks related to image segmentation, detection, and classification, deep learning is superior to traditional learning [48–50].
- Computer vision (CV) allows technologies to have the ability to process and analyze, and interpret visual data. The topics with the most research coming from this field are filtering, segmentation, image classification, detection, and tracking, among others [50].

In addition, as can be seen in Figure 2, an analysis of the state-of-the-art corresponding to the publication history from 2014 to 2025 was carried out together with the total number of documents published for those dates. It is important to consider that, as far as we found in Scopus, the development of DL and ML in pepper farms has 7 publications in 2014, which increased to a total of 26 articles in 2020. Unlike the first years of research in this analysis, during 2024, a representative increase of up to 75 articles was observed regarding the implementation of ML and DL in harvesting processes. As of May 2025, 30 documents related to this topic have been published.

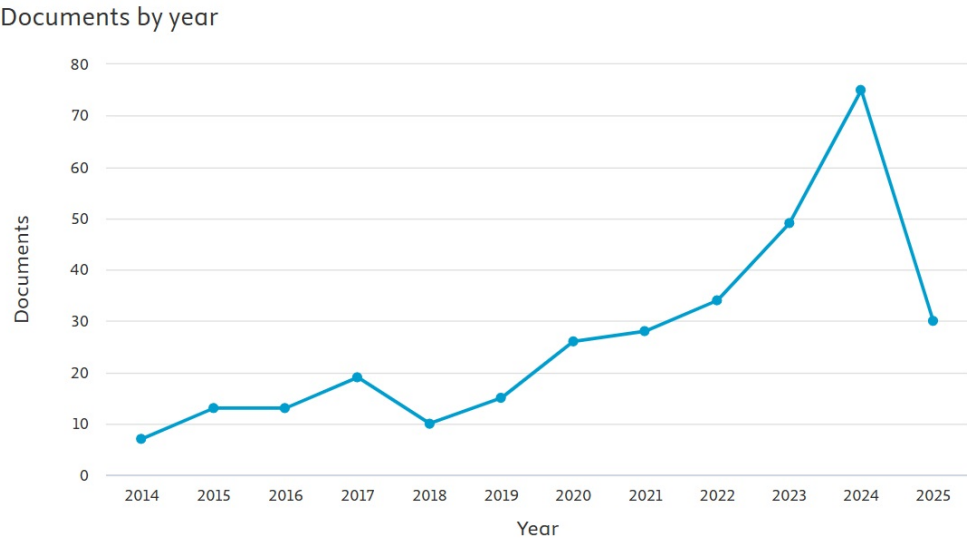


Figure 2. History of published documents based on the implementation of ML and DL in pepper farms from 2014 to May 2025.

Similarly, Figure 3 presents the analysis of the documents published by each country on artificial intelligence algorithms for harvesting processes. It can be observed that the countries that have the largest number of documents related to the subject are China, India, South Korea, United States, Turkey, Indonesia, Japan, and Italy. This list of countries shows how interest in studying ML and DL applications in pepper farms is on the rise, especially in developed countries. In the following subsection 1.3, we summarize several related works found in the literature that are related to ML and DL in pepper farms and possible applications in harvesting processes.

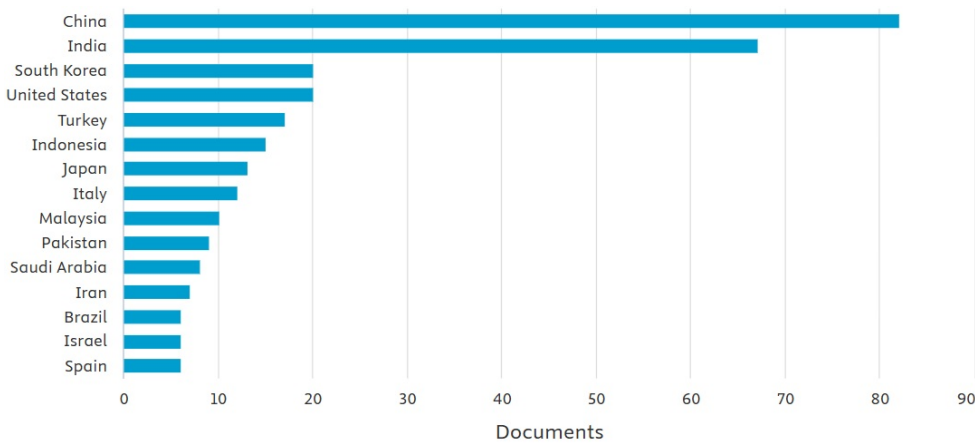


Figure 3. Published documents by country of origin related to ML and DL in pepper farms from 2014 to May 2025.

1.3. Related Work

In this subsection, we describe in detail some of the research works from the literature that implement ML and DL models applied in pepper farms. In the work developed by Halstead, M. et al. [5], the Faster R-CNN detection model based on multitasking learning in residual networks was used to perform the detection and classification of fruits (sweet peppers) in agricultural environments. To this end, three datasets were used as well as a Mask-RCNN framework (instance segmentation) combined with the incorporation of a parallel classification layer of the Faster R-CNN architecture to include both superclasses and subclasses (fruit and degree of maturity). This approach achieved an accuracy of 90% using the F1 score for the classification problem. In [1], López-Barrios, J.D. et al. propose a Mask R-CNN model with a ResNet-101 and feature pyramid network (FPN) backbone to improve machine vision for the detection of green sweet pepper fruits and peduncles in greenhouses. This is crucial for autonomous harvesting, as sweet peppers must be severed accurately at the stem to prevent damage to the fruit. The model produces binary masks by instance segmentation, facilitating accurate localization for 3D harvesting. Evaluated on 100 pictures, it attained a precision of 84.53% for 1148 fruits and 71.78% for 265 peduncles, yielding a mean average precision (mAP) of 72.64%. The mean detection duration was 1.18 seconds per image. The proposed method adeptly addresses occlusion and illumination fluctuations, surpassing previous 2D detection models for green sweet peppers. In [51], Fatchurrahman D. et al. used fluorescence imaging to recognize mechanical damage in 5 × 5 mm incisions in green bell peppers. The authors used traditional ML methods such as logistic regression, neural networks, random forests, k-nearest neighbors, and support vector machines to classify between damaged and undamaged peppers, achieving high performance accuracies above 86%. In [15], Viveros, L.D. presented an architecture for the detection and counting of sweet peppers at different stages of maturation. To this end, the YOLOv5 detection method was combined with the DeepSORT tracking method to detect and count the peppers. The proposed approach obtained high performance even with occlusion problems. The authors used a dataset of 1863 RGB images from a sweet pepper greenhouse, attained a confidence level of 97.3% for pepper detection and 85.7% in pepper counting. In [52], Ren, R. et al. trained a CNN-based model for classifying pepper quality. The authors import neuron weights by using transfer learning to train a VGG 16 CNN. The dataset was augmented using rotation, brightness, and contrast adjustments. The VGG 16 model attained a prediction accuracy of 98.14%, demonstrated enhanced performance, accelerated convergence, and improved stability when compared to ResNet50, MobileNet V2, and GoogLeNet. The work developed by Sa, I. et al. [6] used an approach combining 3D geometric features captured by an RGB-D camera and HSV color information to develop an automated detection algorithm. A total of 72 3D models of sweet peppers and their peduncles were used to train the algorithm using an SVM classifier with an RBF kernel. Thus, the algorithm was able to make peduncle detection of sweet pepper for an autonomous crop, with an accuracy value of up to 71% in the detection of peduncles in field-grown peppers. In the work described by Nan, Y. et al. [53], a pruning-oriented artificial intelligence algorithm NSGA-II, was

implemented using the YOLO model to improve the efficiency in detecting green peppers. For this purpose, the model was trained with 1100 images of green peppers collected under different lighting conditions. This algorithm achieved an accuracy percentage of 81.4%. This model was slightly lower than the original model but superior in terms of the number of parameters, model size, GFlops, and detection speed when compared to other pruning algorithms such as Slim. As demonstrated by Ilyas, T. et al. [35,54], disease analysis and pest segmentation in pepper plants have also been studied through a system based on deep neural networks. To this end, the authors developed a detection model through the EfficientNet-B3 to build the proposed Diffusion Analysis Network Approach (DIANA) model, which was trained with 5900 images captured by an RGB camera delivering accuracy results of up to 91.7% in the detection of anomalies and 70.78% in the estimation of severity with an inference time of 16.4 ms. In Sa, I. et al. [35], the authors used the Faster R-CNN object detection model together with the PASCAL VOC dataset (20 objects, 11,000 images, and 27,000 annotated objects) to perform plant disease analysis. The network was initialized with the previously trained ImageNet dataset, achieving accuracy results from 80.7% to 83.8% in the detection of sweet peppers. In Arad, B. et al. [25], the authors proposed a Flash-No-Flash (FNF) controlled illumination acquisition protocol to improve object detection in a sweet pepper harvesting robot. The database of this work was built from 468 total images, which were captured by an RGB camera mounted on a robotic manipulator to later be used to evaluate the performance of the SSD algorithm, which delivered representative results of accuracy with a value of 84.07% for the Flash set and 83.67% for the FNF set. Finally, in the work of Hespeler, S. C. et al. [55], a Deep CNN algorithm for robotic inspection and harvesting of chili peppers was proposed and validated. To do this, 122 RGB images and 122 thermal images of chili peppers with different levels of illumination and contrast were used with the Mask-RCNN and YOLOv3 algorithms. The YOLOv3 model achieved a training accuracy of 100% and 97% in the test set, displaying better results in the detection of chili peppers in real time with a computation time of 3.64 seconds per image.

As we described above, several researchers studied and developed different ML, DL, and CV algorithms for pepper farm scenarios, obtaining sundry results. Applications of classification, detection, segmentation, and tracking were the most representative. However, despite the existence of these studies and works, multi-class object detection during Pepper harvesting processes is still an open research topic. This is in part because of the high variability of the desired detected categories, as well as the complexity of field environments. This work is focused on finding DL-based object detection models and their corresponding feature extraction to achieve high-performance detection results during pepper harvesting. We present the main contributions of this work as follows.

1.4. Main Contributions

From the state-of-the-art and related work sections that we presented previously, we summarize the main contributions of the present work.

- We obtained an RGB image dataset during a pepper harvesting process in a Chilean farm based on a total of 210 RGB images. We manually labeled each image to identify objects of interest (human, box, pepper, pepper group, and pushcart). We also applied data augmentation techniques to increase our dataset distribution to 630 images.
- We used three different object detection algorithms, which are You Only Look Once v7 (YOLOv7), Faster Region-Convolutional Neural Network (Faster R-CNN), and Single Shot Multi-Box Detector (SSD). For each object detection algorithm, we used three different feature extraction CNNs, which are GoogLeNet, ResNet-18, and ShuffleNet. We trained, validated, and tested each of the proposed object detection algorithms.
- We compared the performance of each model using precision-recall (PR) curves and mean average precision metrics (mAP). Additionally, we analyzed bias, variance, overfitting, and underfitting of each of the trained models.

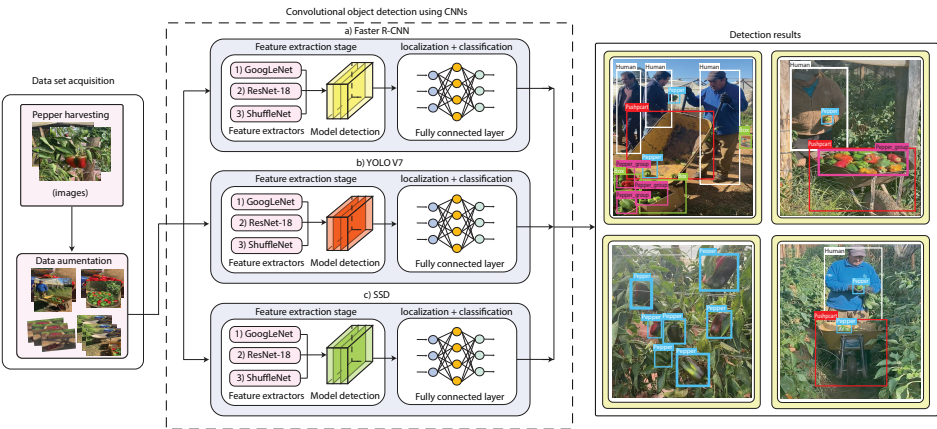


Figure 4. Proposed architecture for object detection during Pepper harvesting processes using Faster R-CNN, YOLOv7, and SSD with GoogLeNet, ResNet-18, and ShuffleNet feature extractor CNNs.

2. Materials and Methods

The proposed architecture is presented in Figure 4. As can be seen, the first stage presents the data acquisition process during pepper harvesting, in which we present our dataset information and distribution. In the second stage, we explain in detail the implementation of the proposed CNN-based object detection approaches as well as their feature extraction methods. Finally, we present the training, validation, and testing detection results, as well as a comparative analysis of the detection models. We also present a methodology workflow in Figure ??, which emphasizes the model calibration procedure, which was key to reaching high-performance results.

2.1. Data Acquisition

We perform the data acquisition procedure in a Chilean Farm in the Maule Region, located in central Chile. This is a geographically diverse area recognized for its rich agricultural landscapes and lies approximately at 35.46016°S latitude and 71.70916°W longitude. We acquired 210 RGB images from a Samsung A71 camera with an image resolution of 4624 × 2084 pixels. Next, the images were resized to 800 × 600 pixels to balance computational efficiency with sufficient visual detail for analysis. We manually labeled the dataset to obtain the ground truth for the proposed object detection approaches during pepper harvesting, which considers the following objects of interest: human, box, pepper, pepper group, and pushcart. We divided our dataset into training, validation, and test sets following the dataset distribution detailed in Table 1. The training set was used to train the proposed object detection algorithms, The validation set was used for calibration by identifying the best possible hyperparameter configuration to fit our dataset distribution. The final object detection performance was obtained from the best-found hyperparameter configuration for each model on the test set, which was key to analyzing the generalization capabilities of each detection model. This is also important to analyze the robustness of the model against overfitting or underfitting. We show an image mosaic of the proposed dataset in Figure 6.

We also perform three different random data augmentation techniques, which are: changes in brightness and saturation, image rotation, and a change in resolution scaling. The first represents a change in the brightness and saturation of 20% and 10%, respectively. The rotation randomly generates a rotation on each image in a range of 3 to 10 degrees clockwise. Finally, the resolution scaling technique changes the scales of the image resolution 800 × 600 to a new format of 600 × 600. This data augmentation procedure is illustrated in Figure 7, as well as the detailed Table 1.

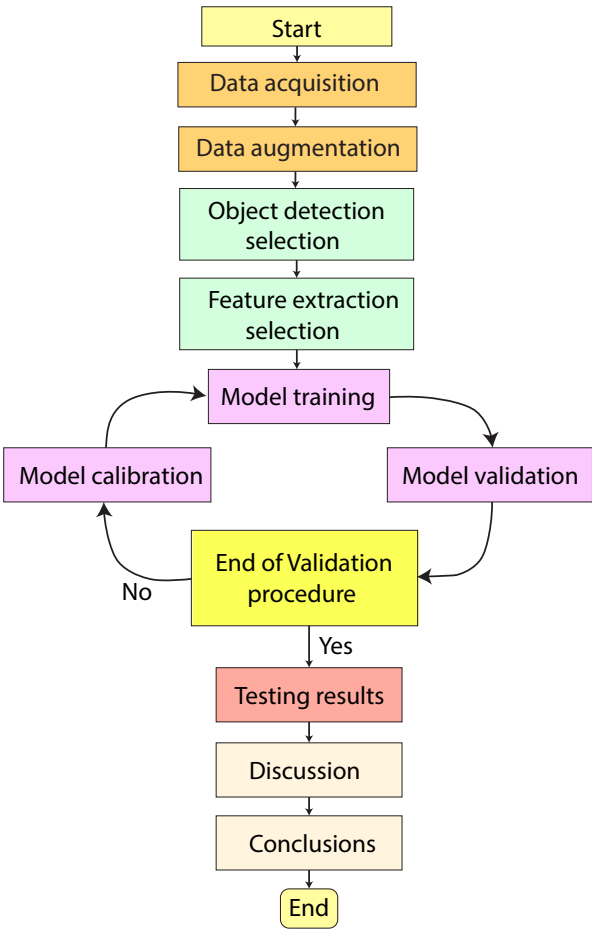


Figure 5. Proposed methodology workflow for object detection during Pepper harvesting processes.

Table 1. Distribution of RGB images in the dataset obtained during a pepper harvesting process.

Dataset	Normal	Data Augmentation	Total
Training	168	336	504
Validation	21	42	63
Test	21	42	63
Total	210	420	630



Figure 6. Proposed dataset mosaic of the harvesting process in a Chilean farm. a) Harvesting process images. b) Pepper images.

2.2. Object Detection Algorithms

In this work, we propose to use of Faster R-CNN, YOLOv7, and SSD object detection algorithms to detect objects of interest during pepper harvesting processes. Each object detector was tested with different CNN feature extraction methods, which were GoogLeNet, ResNet-18, and ShuffleNet. In total, nine object detectors were trained and tested, corresponding to the combination of three detection algorithms and three feature extractors. The Subsection 2.2.1 details how the object detection algorithms works, and Subsection 2.2.2 explains each of the proposed feature extraction methods.

2.2.1. Object Detection Methods

In this subsection, each of the proposed object detection methods will be described in detail as follows.

Faster Region-Based Convolutional Neural Network (Faster R-CNN)

The Faster R-CNN model is a CNN-based object detection method that works from three layers: the first layer is responsible for performing feature extraction, the second layer is used for region proposal networks (RPNs), and the last stage is responsible for performing object classifications [56–58]. In the feature extraction stage, the most relevant feature maps are identified and extracted from the images [58]. Generally, this detection model utilizes ResNet or VGG as feature extraction CNNs for object detection. Nevertheless, we employed GoogLeNet, ResNet-18, and ShuffleNet, since these CNNs demonstrated high performance in our dataset distribution for pepper harvesting. After feature extraction, the RPNs determine areas where the probability of identifying objects of interest is high. RPNs can, at the same time, identify and suggest multiple regions of multiple sizes and geometric configurations, which are known as anchor boxes. Finally, in the classification stage, the detection model has a fully connected layer followed by a softmax layer to classify the categories of interest (human, box, pepper, pepper group, and pushcart). The last fully connected layer utilizes RPN information to locate the coordinates corresponding to the bounding boxes of each category (further details can be found in [56,57]).

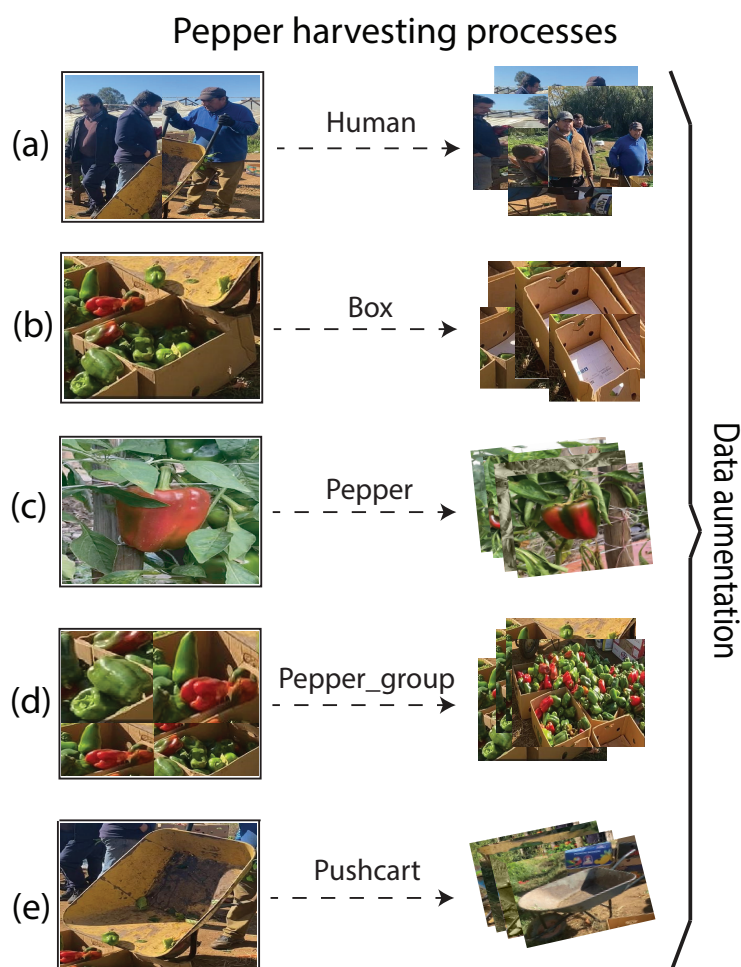


Figure 7. Categories in the pepper harvesting dataset, including data augmentation examples. (a) Human, (b) Box, (c) Pepper, (d) Pepper group, (e) Pushcart.

YOLOv7

The You Only Look Once version 7 (YOLOv7) represents a cutting-edge object detection technique that outperforms earlier YOLO versions (such as YOLOv4, YOLOv5, YOLOX, YOLOR) as well as other detection frameworks, including transformer-based models (e.g., Swin-L Cascade-Mask R-CNN) and convolutional models (e.g., ConvNeXt-XL), in terms of both speed and precision [26,59,59]. YOLOv7 utilizes a bag-of-freebies methodology to improve training accuracy without elevating inference costs, alongside a coarse-to-fine lead-guided label assignment with deep supervision to refine multi-layer output training [59]. The input image is segmented into a grid of cells of different sizes, with each cell tasked with recognizing objects inside its designated area [26,59,59]. YOLOv7 employs anchor boxes of different forms, scales, and sizes within each cell to accurately localize objects with various geometric configurations [26,59,59]. YOLOv7 uses compound scaling, modifying model depth and width for concatenation-based architectures to preserve efficiency. Typically, YOLOv7 uses a custom E-ELAN backbone as a feature extraction stage [60,61]. However, in this work, we employ GoogLeNet, ResNet-18, and ShuffleNet as customized backbones for feature extraction due to their high performance on the proposed pepper harvesting dataset distribution. These backbones encompass both low-level details (e.g., edges) and high-level semantic information (e.g., textures, forms) throughout the feature extraction process. The proposed backbones—GoogLeNet (concatenation-based), ResNet-18 (residual-based), and ShuffleNet (hybrid residual and concatenation-based)—eliminate identity connections in RepConv to preserve gradient diversity, hence optimizing training and inference performance. Next, the YOLOv7 neck utilizes a topology akin to a Path Aggregation Network (PANet), merging high-resolution intricate details with low-resolution semantic context. During training, intricate

convolutional modules (e.g., 3×3 , 1×1 , identity connections) are consolidated into a single efficient layer (RepConvN, excluding identity connections) at inference to minimize computing expense while maintaining accuracy [26,59,59]. Next, YOLOv7 uses anchor box prediction, which comprises a confidence score indicating the existence of an object and the precision of its class prediction. Non-maximum suppression (NMS) removes superfluous bounding boxes, preserving the best reliable prediction for each object to improve detection precision [26,59,59]. Predictions with confidence scores within a specified threshold are eliminated to enhance results. Utilizing an IoU-based loss function and trained on the proposed pepper harvesting process dataset, YOLOv7 generates bounding boxes annotated with object categories and confidence scores that reflect detection accuracy, employing its Extended Efficient Layer Aggregation Network (E-ELAN) and a coarse-to-fine label assignment for enhanced performance.

Single Shot MultiBox Detector (SSD)

SSD is an object detection method often selected for its low inference latency, which is achieved by ruling out the need to compile the RPN of other models, such as Faster R-CNN. The SSD algorithm implements a single-shot approach that manages to classify and locate objects simultaneously. Generally, this object detection algorithm uses the VGG16 network as a feature extractor. However, in this work, the SSD model will work with the GoogLeNet, Resnet-18, and ShuffleNet networks due to the high performance obtained in our pepper harvesting dataset distribution. For this model to be able to perform object detection at multiple scales, one of the stages present in the algorithm is used that combines the additional convolutional layers, which perform the production of predictions from feature maps with various resolutions [62]. This model, like the other models, uses a set of anchor boxes with different scales and aspect ratios for each cell of the feature map, allowing the model to detect objects independent of their sizes and geometric configurations. For each bounding box, the model predicts the category score and the displacements of the bounding boxes, with this, the technique can improve the location of the objects of interest. This model can handle object classification and location problems simultaneously by applying convolutional operations through anchor boxes on different feature layers [62–64].

2.2.2. Feature Extraction Methods

In this work, three different CNN-based feature extraction models were used for each of the object detection methods: Faster R-CNN, YOLOv7, and SSD. Particularly, we decided to use the GoogLeNet, ResNet-18, and ShuffleNet networks as feature extractors due to the high performance obtained when working with our proposed pepper harvesting dataset. The feature extraction methods used in this work will be explained in detail as follows.

GoogLeNet

GoogLeNet is a feature extractor CNN with 143 layers, 169 connections, and 7 million parameters for training [65]. Convolutional layers, grouping layers, and ReLU activation functions make up the entire 143-layer architecture of GoogLeNet. The input layer of GoogLeNet is $224 \times 224 \times 3$ and has 57 convolutional layers and 13 grouping layers. The rest of the layers will correspond to Relu layers, concatenation layers, Dropout layers, fully connected layers, normalization functions, and softmax for sorting. The structure of the GoogLeNet network begins with convolutional and max pooling layers that reduce the size of the spatial dimensions of the input RGB images, followed by a row of nine Inception modules, with maximal clustering layers between them to reduce the feature map [65]. GoogLeNet is a unique neural network due to the Inception modules, which use multiple convolutional filters of different sizes (1×1 , 3×3 , 5×5) in parallel within the same layer. This allows GoogLeNet to extract spatial patterns of different sizes from the RGB images. Each Inception module also has a maximum grouping layer of 3×3 [65,66]. GoogLeNet uses 1×1 convolutions as limiting layers for dimensionality reduction to prevent the number of factors from growing excessively and making calculation difficult. GoogLeNet's intermediate layers provide additional gradient paths and

act as regularizers to reduce overfitting and prevent gradient issues fading [65,67]. The resulting entities are processed through a fully connected layer and a softmax layer to get the classification results. While GoogLeNet has a robust structure capable of reaching high-performance accuracy for classification problems, it needs to be merged with the Faster R-CNN, YOLOv7, and SSD detection models to be able to solve object detection problem.

ResNet-18

ResNet-18 is a CNN distinguished for using 11.7 million parameters and incorporating residual layers to tackle the leakage gradient problem [52]. The architecture of ResNet-18 encompasses convolutional layers, a batch normalization layer, and 18 Relu activation function layers. The input resolution of this network is $224 \times 224 \times 3$, indicating that any image that does not adhere to this resolution must go through a resizing process. The further layers of the ResNet architecture correspond to the ReLU layers, batch normalization layers, fully connected layers, residual layers, and the softmax layers that allow classification to be performed. The initial phase of the network structure begins with a 7×7 convolutional layer with 64 filters (stride of 2), accompanied by batch normalization, a ReLU activation function, and a max pooling layer of 3×3 (stride of 2) [68,69]. Subsequently, the max-pooling layer allows the network to reduce the spatial dimensions of the input Pepper harvesting process images, thereby extracting the patterns present in the images. Once the patterns corresponding to the characteristics have been extracted, they are sent through a fully connected layer to a softmax layer to obtain the object's classification as a result. Additional details about ResNet-18 can be accessed at [69,70]. Despite the high-performance of the CNN ResNet-18 architecture, it is necessary to incorporate it with the proposed Faster R-CNN, YOLOv7, and SSD detection models for effective object detection.

ShuffleNet

ShuffleNet is a CNN that has 1.4 million parameters and 50 layers [71]. It combines group convolutions and channel shuffle operations to reduce computational complexity while maintaining performance as an efficient feature extractor. This allows ShuffleNet to perform two operations, the spot group convolutional and the shuffle channel, to maintain accuracy while reducing computational costs. However, this is highly dependent on the size and distribution of the dataset. The architecture of this network is composed of several normalization layers, Relu activation layers, and convolution layers. The image input resolution for ShuffleNet is $224 \times 224 \times 3$. The first part of this network starts with a 3×3 convolutional layer with 24 filters, followed by the batch normalization layer, the ReLU function, and the 3×3 maximum grouping layer [72,73]. Subsequently, like the other networks, the max-pooling layer allows the network to reduce the dimensions of each image of the input harvesting process, extracting the patterns present in the feature maps. Once this process is done, the process proceeds with a fully connected layer to drive the data to a softmax layer that derives the detection to some defined category in network training. More details about ShuffleNet can be found at [72,73]. This network achieves a high performance in the extraction and classification tasks. However, it is necessary to combine it with the Faster R-CNN, YOLOv7, and SSD detection models to perform object detection in the proposed pepper dataset.

3. Evaluation Metrics

To assess the performance of the Faster R-CNN, YOLOv7, and SSD models, it is necessary to test each object detection model by using evaluation metrics such as the Precision-Recall (PR) Curve and the Mean Average Precision (mAP). The following subsection will explain in detail the calculation performed to determine the PR curves and mAP.

3.1. Precision and Recall

The positive predictive value (PPV), or Precision, describes the proportion of meaningful segments of objects detected in the Pepper harvesting process images. On the other hand, sensitivity, also known as Recall, corresponds to the proportion of relevant cases that were successfully detected within

the pepper harvesting process images. The equation 1 is used to calculate the precision, while the estimation of the Recall requires the equation 2, respectively. In addition, Figure 8 illustrates the recall and accuracy metrics applied to evaluate object detection during the pepper harvesting process images.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

where TP denotes true positives, FP false positives, and FN false negatives.

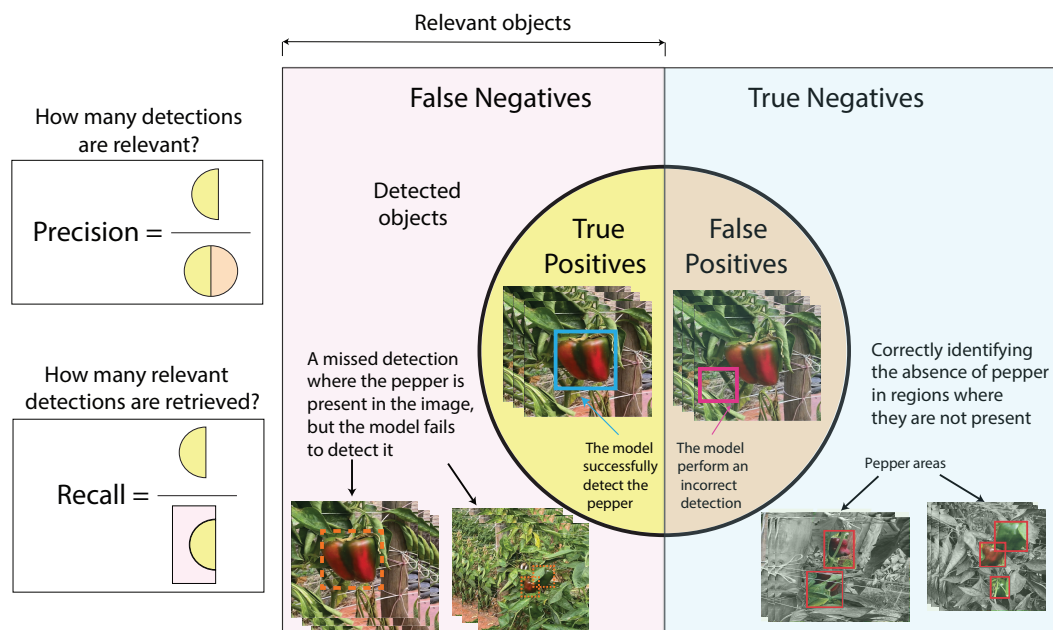


Figure 8. Precision and recall representation for object detection in the context of the pepper harvesting process.

3.2. Precision-Recall Curve

To represent the balance between precision and recall, we can use the precision-recall (PR) curve. This curve is specifically used to address object detection problems where the dataset is unbalanced, i.e., where the number of true positives (e.g., pushcarts in images of harvesting processes) is less than the number of true negatives (all areas where the pushcart tag is not present in the pictures). To generate the PR curve, it is required to change the decision threshold in the object detection method. The precision and recall variables are determined for each threshold, representing the precision on the vertical Y axis and the recall on the horizontal X axis. Figure 9 shows an example of a PR curve obtained in this work, which represents the object detection of the box category during pepper harvesting processes. It is important to take into account that the probability of detection, also named confidence value, is influenced by the confidence threshold used to establish the percentage of acceptance to consider a positive prediction.

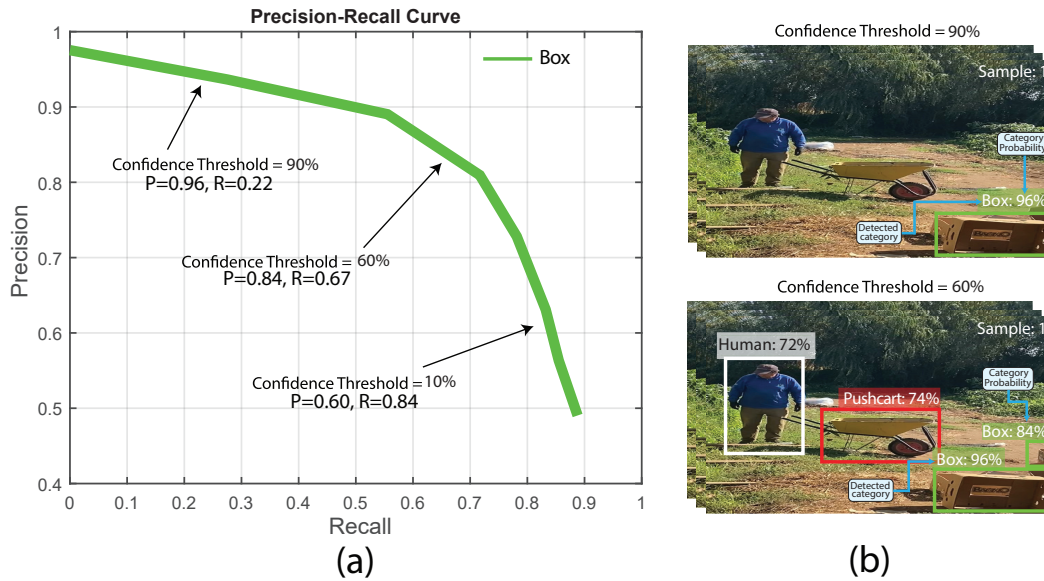


Figure 9. Precision-Recall (PR) Curve for object detection during Pepper harvesting processes context. (a) A sample of the Precision-Recall (PR) Curve for one category (Box). (b) Results for object detection at different threshold values.

3.3. Mean Average Precision (mAP)

One of the metrics employed to evaluate the PR performance of object detection models is the Mean Average Precision (mAP). This metric establishes the balance between precision and recall across different confidence threshold levels. To determine mAP, we first calculate the average precision (AP) for each category by integrating the area under the PR curve of that category. In this work, we calculated the mean of the AP values across all categories (human, box, pushcart, pepper, and pepper group) to obtain the mAP. Equation 3 represents the formula for calculating AP, while Equation 4 shows the mAP calculation formula as the mean AP across categories.

$$AP = \sum_n (R_n - R_{n-1}) P_n \quad (3)$$

where P_n and R_n are the precision and recall at the n -th threshold, respectively.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

where N is the number of classes, in this case, an object class in the Pepper harvesting process.

3.4. Inference Times

The inference time is known as the time it takes for an AI model to make predictions based on the input data. In object-detection-focused tasks, inference time is essential for evaluating the model's efficiency and performance in real-world scenarios. Faster inference times indicate that the models developed in this work can quickly extract information from images in the pepper harvesting process. This work compares the inference times of three models: Faster R-CNN, YOLOv7, and SSD, employing different CNN architectures for feature extraction, including GoogLeNet, ResNet-18, and ShuffleNet.

4. Results

This section presents the values obtained when evaluating the performance of the YOLOv7, Faster R-CNN, and SSD object detection models. The models were tested to assess their detection performance for human, pepper, pushcart, pepper group, and box label categories during the pepper harvesting processes. Each object detection model was trained and estimated using three CNN architectures for

feature extraction, which are GoogLeNet, ResNet-18, and ShuffleNet. The training, validation, and testing procedures were carried out for each object detection model with each feature extraction.

For the validation procedure, several hyper-parameter configurations were tested to train each of the proposed models. We established distinct values of epochs, learning rate, and optimization methods as displayed in Table 2. The maximum number of epochs was set at 22 and 45, while the learning rates used were 0.001, 0.005, and 0.0001. The training optimizers included ADAM (Adaptive Moment Estimation), SGDM (Stochastic Gradient Descent with Momentum), and RMSProp (Root Mean Square Propagation). The ADAM optimizer refine learning rates by using momentum and square gradients, while the SGDM optimizer utilizes momentum to accelerate convergence. In contrast, RMSProp adjusts learning rates based on the mean square of the gradients.

Table 2. Hyper-parameter configuration for training the detector methods.

Test	Hyper-parameter configuration		
	Learning Rate	Max Epochs	Optimizer
Test 1	0.001	22	adam
Test 2	0.001	22	sgdm
Test 3	0.001	22	rmsprop
Test 4	0.005	22	adam
Test 5	0.005	22	sgdm
Test 6	0.005	22	rmsprop
Test 7	0.0001	22	adam
Test 8	0.0001	22	sgdm
Test 9	0.0001	22	rmsprop
Test 10	0.001	22	adam
Test 11	0.001	45	sgdm
Test 12	0.001	45	rmsprop
Test 13	0.005	45	adam
Test 14	0.005	45	sgdm
Test 15	0.005	45	rmsprop
Test 16	0.0001	45	adam
Test 17	0.0001	45	sgdm
Test 18	0.0001	45	rmsprop

All proposed object detection models were trained using the configurations in Table 2. After training, the models were evaluated for bias, variance, underfitting, and overfitting based on the following criteria. High bias indicates that our models are overly simplistic with low complexity that do not fit our dataset distribution, resulting in significant across the training, validation, and test datasets (underfitting). High variance occurs when the model performs well on the training data but poorly on the validation or test data (overfitting). The following subsection analyzes the results based on these criteria.

4.1. *Faster R-CNN, YOLOv7, and SSD with GoogLeNet*

This section presents the results of the training, validation, and testing performed for Faster R-CNN, YOLOv7, and SSD with GoogLeNet as a feature extractor. These detailed results are presented in Table 3.

Table 3. Object detection results with GoogLeNet as feature extraction during Pepper harvesting processes results - mAP.

Test	Training			Validation			Testing		
	Faster R- CNN	Yolo v7	SSD	Faster R- CNN	Yolo v7	SSD	Faster R- CNN	Yolo v7	SSD
Test 1	83.2%	97.8%	51%	83%	93%	46%	81%	99%	48%
Test 2	55%	2%	3%	42%	1%	3%	51%	1%	2%
Test 3	70%	84%	4%	67%	71%	3%	66%	70%	1%
Test 4	3%	31%	2%	2%	30%	2%	2%	38%	1%
Test 5	80%	6.1%	3%	80%	5%	1%	77%	4%	2%
Test 6	8%	12%	4%	7%	11%	5%	4%	18%	1%
Test 7	82%	2%	28%	85%	1%	27%	81%	2%	25%
Test 8	8%	3%	4%	13%	2%	3%	8%	2%	2%
Test 9	81%	16%	58%	81%	18%	58%	79%	26%	54%
Test 10	73%	99%	80%	68%	99.8%	82%	72%	99.6%	80%
Test 11	73%	24.4%	4%	64%	16%	3%	69.9%	25%	3%
Test 12	83%	99%	2%	82%	99.8%	2%	79%	94.2%	1%
Test 13	7%	97%	8%	6%	97%	6%	5%	99.9%	4%
Test 14	87%	62%	7%	86%	74%	7%	83%	56%	6%
Test 15	4%	90%	4%	2%	77%	4%	3%	82%	3%
Test 16	88%	60%	57%	92%	29%	55%	85%	52%	52%
Test 17	19%	6%	4%	26%	5%	4%	21%	3%	5%
Test 18	89.8%	85.9%	76%	87%	80.2%	76%	85%	85.5%	74%

Figure 10 presents the most outstanding results for the proposed object detection algorithms using GoogLeNet. We evaluated training, validation, and testing results through PR curves. The Faster R-CNN achieved the best result for Test 18, while YOLOv7 accomplished the best on Test 10, and SSD for Test 10. The best results for this experiment set were obtained with YOLOv7 using GoogLeNet for test 10 (training: 99%, validation: 99.8%, and test: 99.6%).

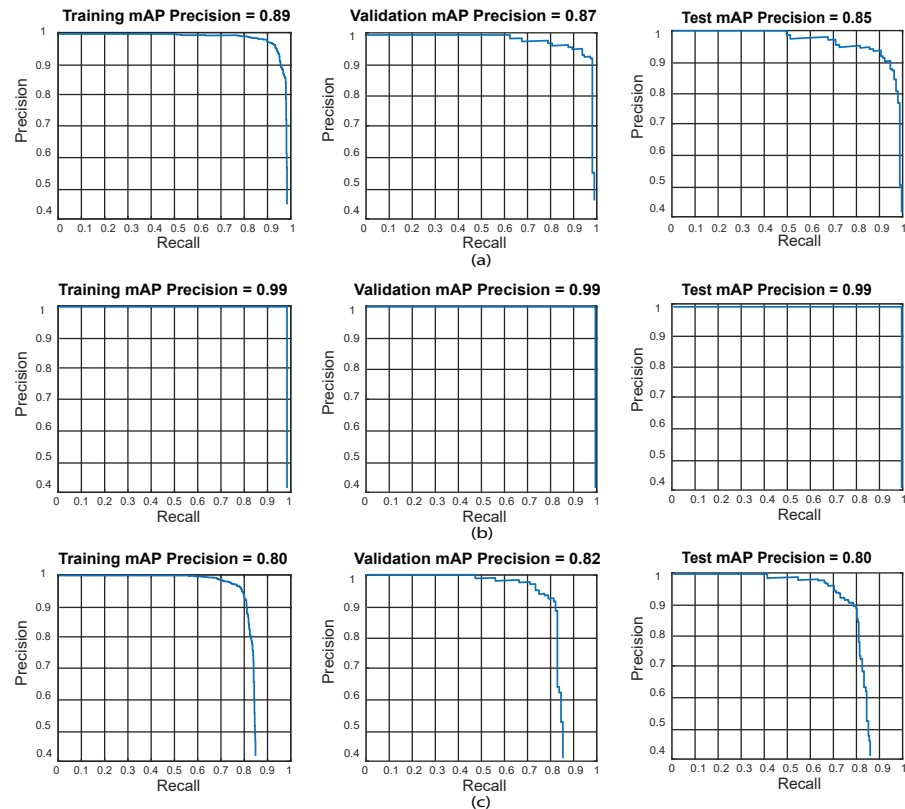


Figure 10. Best PR curves for the object detection algorithms using GoogLeNet. Training results (left), validation results (middle), and testing results (right). (a) Faster R-CNN - Test 18, (b) YOLOv7 - Test 10, (c) SSD - Test 10. Best results for this experiment were obtained with YOLOv7 for test 10 (training: 99%, validation: 99.8%, and test: 99.6%).

The images obtained when evaluating the trained models Faster R-CNN, YOLOv7, and SSD with GoogLeNet as a feature extractor in Pepper harvesting are presented in Figure 11.

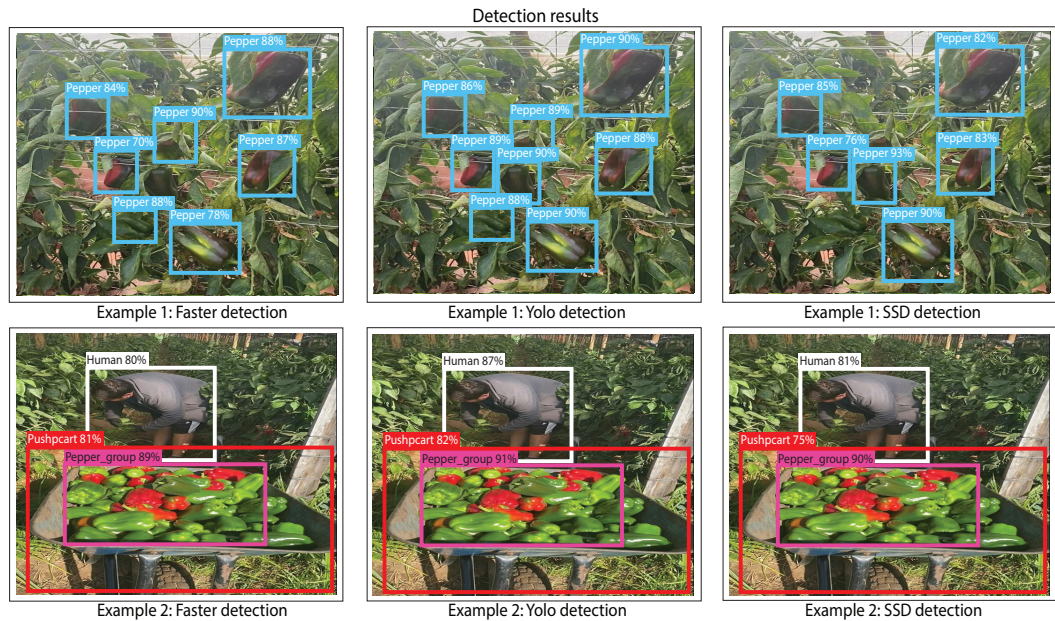


Figure 11. Object detection results during Pepper harvesting processes with GoogLeNet as a feature extractor for Faster R-CNN, YOLOv7, and SSD.

We summarize the main findings of the object detection methods using GoogLeNet as the feature extractor below.

- Table 3 summarizes that when employing GoogLeNet as a feature extractor, the Faster R-CNN, YOLOv7, and SSD models achieve very low accuracy results. For Faster R-CNN, Test 4 achieved the lowest results (training: 3%, validation: 2%, and test: 2%). Moreover, YOLOv7 exhibited low-performance results in test 2 (training: 2%, validation: 1% and tests: 1%). Finally, the SSD also displayed low-performance results in test 4 (training: 2%, validation: 2% and tests: 1%). These results reflect that the models trained under these configurations are experiencing underfitting issues. This means that the performance of the trained models does not fit with the proposed distribution of the pepper harvesting processes dataset, which also reflects a high bias.
- In test 16, the Faster R-CNN model achieves an accuracy of 88% in the training set, while the YOLOv7 and SSD models achieve values of 60% and 55%, respectively. This could be because the Faster R-CNN model is based on using an RPN (Region Proposal Network) stage, which, for this test, can be seamlessly adjusted to the distribution of the pepper harvesting processes dataset (low bias). In contrast, YOLOv7 and SSD employ a single-stage-based detection approach without utilizing an RPN stage, causing a misalignment with the distribution of the pepper harvesting processes dataset (underfitting). Compared to the Faster R-CNN results, the YOLOv7 and SSD models have low validation results, with 29% and 55%, respectively, and for testing, with 52% and 52%, respectively. The difference between the training accuracy values between the Faster R-CNN (88%), YOLOv7 (60%), and SSD (57%) models, suggests that for the hyper-parameter configuration used in test 16, the models with low results do not fully fit the dataset suggesting the presence of bias. Additionally, the slight decline in validation accuracy (YOLOv7: 29% and SSD: 55%) and tests for both models (YOLOv7: 52% and SSD: 52%) indicates a slight divergence, as the performance of the models exhibits a slight reduction in both validation and testing phases.
- The best results obtained when using the GoogLeNet as a feature extractor for the Faster R-CNN model were obtained in test 18 (training: 89.8%, validation: 87%, and tests: 85%). For YOLOv7 with GoogLeNet, the best result was obtained in test 10 (training: 99%, validation: 99.8%, and test: 99.6%). Ultimately, the SSD model using GoogLeNet achieved the best results in test 10 (training: 80%, validation: 82%, and tests: 80%). Based on the results presented in Table 3, the best model GoogLeNet as feature extraction in our dataset was for YOLOv7 using the hyper-parameter configuration of test 10. This model displays a minimal bias and variance, suggesting high generalization capabilities (low overfitting), since accuracy is hardly reduced in the training, validation, and testing phases.

4.2. *Faster R-CNN, YOLOv7, and SSD with ResNet-18*

This section presents the results of the training, validation, and testing performed for Faster R-CNN, YOLOv7, and SSD with ResNet-18 as a feature extractor. These detailed results are presented in Table 4.

Table 4. Object detection results with ResNet-18 as feature extraction during Pepper harvesting processes results - mAP.

Test	Training			Validation			Testing		
	Faster R- CNN	Yolo v7	SSD	Faster R- CNN	Yolo v7	SSD	Faster R- CNN	Yolo v7	SSD
Test 1	61.2%	95%	88%	62%	95%	91%	48%	94%	89%
Test 2	67%	6%	57%	64%	5%	58%	68%	4%	59%
Test 3	3%	93%	88%	2%	92%	91%	1%	96%	89%
Test 4	5%	24%	77%	5%	23%	80%	4%	34%	78%
Test 5	86%	2%	87.9%	83%	1%	83%	80%	4%	91%
Test 6	2%	57%	73%	2%	45%	70%	1%	63%	66%
Test 7	84%	13%	85%	81%	22%	87.3%	81%	7%	89%
Test 8	29%	2%	1%	23%	1%	6%	25%	2%	4%
Test 9	81%	10%	84%	82%	15%	87%	77%	7%	85%
Test 10	53%	99%	88%	42%	99%	91%	45%	99%	89%
Test 11	85%	6%	76.3%	84%	14%	81%	80%	2%	78%
Test 12	12%	99%	88%	14%	99%	90%	11%	99%	87%
Test 13	7%	96%	88%	7%	93%	90%	5%	95%	88%
Test 14	88.2%	86%	88%	88%	85%	94%	82%	85%	91%
Test 15	5%	95%	88.3%	3%	99%	89%	3%	87%	86%
Test 16	91%	32%	88%	89%	37%	92%	87%	31%	90%
Test 17	44%	3%	9%	40%	2%	15%	33.5%	3%	15%
Test 18	87%	49%	87%	86%	51%	92%	85%	41%	89%

Figure 13 presents the most outstanding results for the proposed object detection algorithms using ResNet-18. We evaluated training, validation, and testing results through PR curves. The Faster R-CNN obtained the best result for Test 16, YOLOv7 for Test 12, and SSD for Test 14, respectively. The best results for this experiment set were obtained with YOLOv7 using ResNet-18 for test 10 (training: 99%, validation: 99%, and test: 99%).

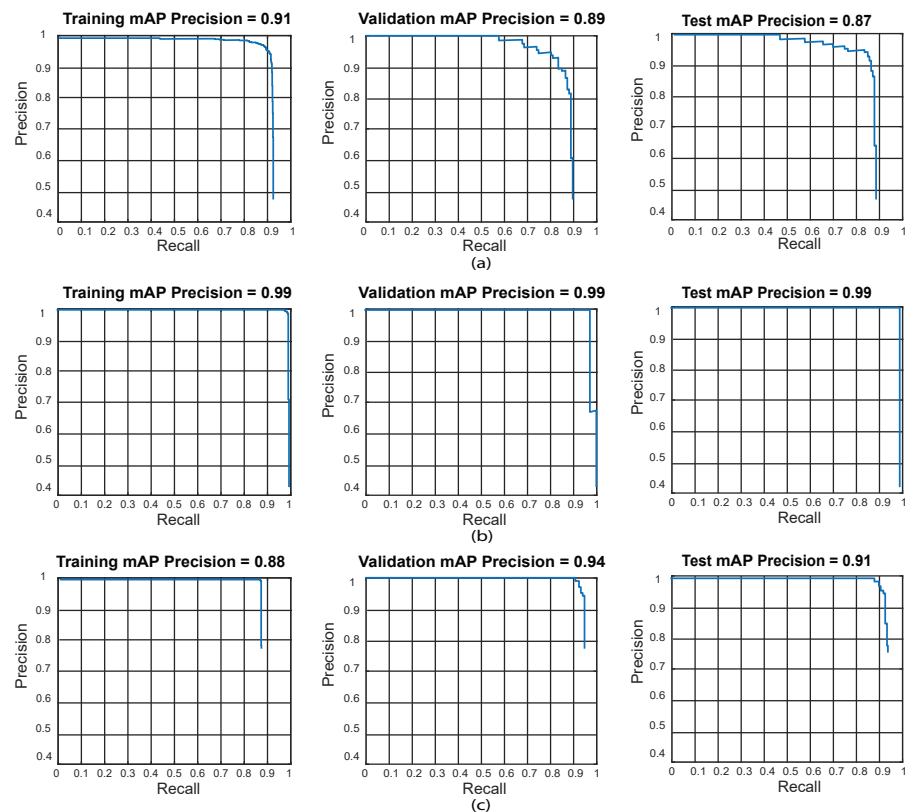


Figure 12. Best PR curves for the object detection algorithms using ResNet-18. Training results (left), validation results (middle), and testing results (right). (a) Test 16 - Faster R-CNN, (b) Test 12 - YOLOv7, (c) Test 14 - SSD. The greatest results for this experiment were achieved with YOLOv7 for test 12 (training:99%, validation:99%, and testing:99%).

The images obtained when evaluating the trained models Faster R-CNN, YOLOv7, and SSD with ResNet-18 as a feature extractor in Pepper harvesting are presented in Figure 11.

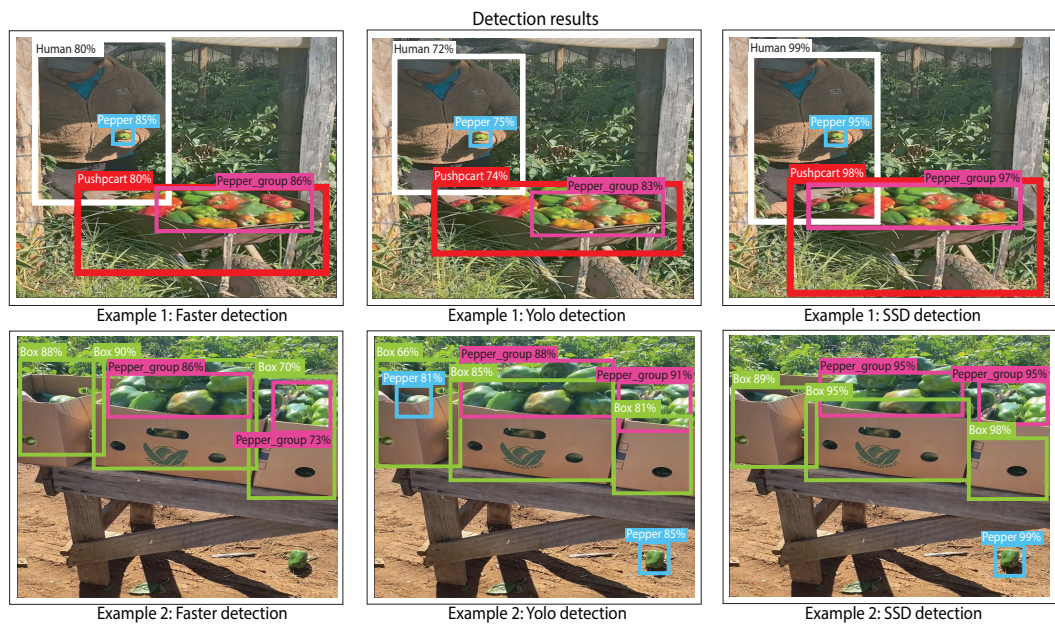


Figure 13. Object detection results during Pepper harvesting processes with ResNet-18 as a feature extractor for Faster R-CNN, YOLOv7, and SSD.

We outline the major findings regarding object detection results using ResNet-18 as a feature extractor as follows.

- As displayed in Table 4, there are tests where using ResNet-18 as a feature extractor, the Faster R-CNN, YOLOv7, and SSD models achieves very low accuracy results. For Faster R-CNN, the test with the lowest results is test 3 (training: 3%, validation: 2% and test: 1%). YOLOv7 also presents low-accuracy results in test 5 (training: 2%, validation: 1% and test: 4%). Finally, the SSD also shows low results in test 8 (training: 1%, validation: 6% and tests: 4%). These results reflect that the models trained under these configurations face underfitting issues, meaning their performance does not align with the proposed distribution of the Pepper harvesting processes dataset. This suggests that training the models with the configurations employed in each test will result in a high presence of bias.
- According to Table 4, the Faster R-CNN object detection model delivers the lowest precision results. The lowest results for this model are in test 6 (training: 2%, validation: 2% and test: 1%), while the best results are in test 16 (training: 91%, validation: 89% and test: 87%). In comparison with the previous result, the test 16 from the Faster R-CNN model, managed to fit seamlessly into the Pepper harvesting processes dataset. Unlike the rest of the algorithms, the Faster R-CNN model may deliver results with a low level of accuracy due to its use of an RPN (Region Proposal Network) stage, which tends to suffer from an insufficient adjustment for our dataset distribution.
- The best results obtained when using the ResNet-18 network as a feature extractor depend strongly on the hyper-parameter configuration. The Faster R-CNN model obtained its best result in test 16 (training: 91%, validation: 89% and tests: 87%). For YOLOv7, the best result is in test 12 (training: 99%, validation: 99%, and test: 99%). For the SSD model, the best result was achieved in test 14 (training: 88%, validation: 94% and tests: 91%). Based on the results displayed in Table 4, the greatest model using ResNet-18 for this dataset is YOLOv7 with the hyper-parameter configuration of test 12. This model disclose low bias and variance, demonstrating high generalization capabilities (low overfitting), as its accuracy is barely reduced in the training, validation, and testing phases.

4.3. *Faster R-CNN, YOLOv7, and SSD with ShuffleNet*

This section presents the training, validation, and testing results achieved by the Faster R-CNN, YOLOv7, and SSD models with ShuffleNet as a feature extractor. Detailed results are presented in Table 5.

Table 5. Object detection results with ShuffleNet as feature extraction during Pepper harvesting processes results - mAP.

Test	Training			Validation			Testing		
	Faster R- CNN	Yolo v7	SSD	Faster R- CNN	Yolo v7	SSD	Faster R- CNN	Yolo v7	SSD
Test 1	61%	97%	78.8%	64%	97.3%	83%	70%	99%	79%
Test 2	62%	41%	38%	63%	36%	41%	51%	29%	39.3%
Test 3	51%	99%	75%	47%	97%	76%	51%	97%	71%
Test 4	4%	98%	76%	4%	92%	81%	3%	97%	76%
Test 5	2%	85%	71%	1%	74%	74%	1%	90%	69%
Test 6	8%	82%	76%	7%	49%	79%	4%	58%	76%
Test 7	66.6%	90%	45%	57%	74%	49%	58%	86%	42%
Test 8	44%	4%	2%	43.5%	3%	3%	42%	4%	4%
Test 9	65%	92%	46%	66%	78%	47%	56%	93.3%	41%
Test 10	72%	96%	79%	65%	97%	80%	61%	96%	77%
Test 11	68%	80%	56%	66%	67.3%	53%	66%	84%	50%
Test 12	48%	99%	79%	51%	97%	82%	39%	99%	79%
Test 13	9.3%	95%	80%	8%	97%	82%	7%	96.5%	80%
Test 14	8%	97%	78%	7%	92.8%	81%	6%	95.6%	78%
Test 15	9%	82%	78%	10%	88%	79%	7%	83%	78%
Test 16	81%	98.8%	60%	77%	94%	60.2%	70%	96%	55%
Test 17	46%	3%	11%	37%	2%	14%	46%	3%	12.2%
Test 18	71%	99%	60%	71%	99%	58%	68%	99%	52%

Figure 14 present the most outstanding results for the proposed object detection algorithms using ShuffleNet. We evaluated training, validation, and testing results through PR curves. The Faster R-CNN obtained the best result for Test 16, YOLOv7 for Test 18, and SSD for Test 13, respectively. The greatest results achieved, using ShuffleNet, were obtained with YOLOv7 for test 18 (training: 99%, validation: 99%, and test: 99%).

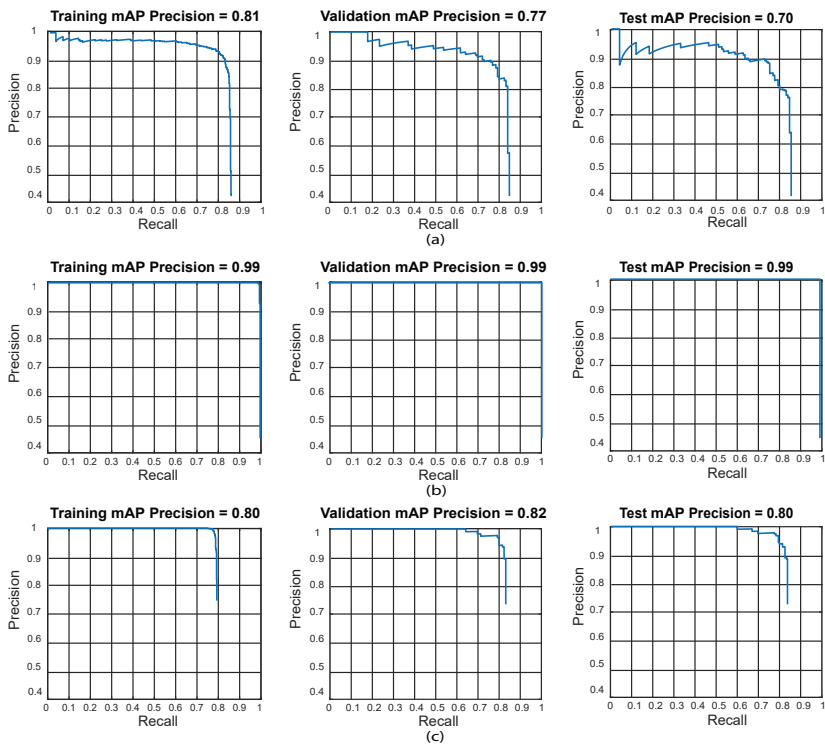


Figure 14. Best PR curves of object detection algorithms using ShuffleNet. Training results (left), validation results (middle), and testing results (right). (a) Test 16 - Faster R-CNN, (b) Test 18 - YOLOv7, (c) Test 13 - SSD. The greatest results achieved, were obtained with YOLOv7 for test 18 (training:99%, validation:99%, and testing:99%).

The images obtained when evaluating the trained models Faster R-CNN, YOLOv7, and SSD with ShuffleNet as a feature extractor in Pepper harvesting are presented in Figure 15.



Figure 15. Object detection during Pepper harvesting processes with ShuffleNet as a feature extractor for Faster R-CNN, YOLOv7, and SSD.

We summarize the main findings of the object detection results for ShuffleNet as feature extraction as follows.

- It can be observed in Table 5 that when using ShuffleNet as a feature extractor, there are tests where the Faster R-CNN, YOLOv7, and SSD models achieve very low accuracy results. In the Faster R-CNN case, the lowest results correspond to Test 5 (training: 2%, validation: 1% and test: 1%). YOLOv7 also presents low-accuracy results in test 17 (training: 3%, validation: 2% and tests: 3%). Also, the SSD model achieves low results in test 8 (training: 2%, validation: 3% and tests: 4%). These results reflect that the models trained under these configurations face underfitting issues, i.e., the performance of the trained models does not fit the proposed distribution of the Pepper harvesting processes dataset, reflecting high bias.
- According to the data from Table 5, the fastest R-CNN object detection with ShuffleNet as the feature extractor model is the one that offers low-accuracy results. The poorest results of Faster R-CNN come from test 6 (training: 2%, validation: 1% and test: 1%), while the best results correspond to test 16 (training: 81%, validation: 77% and test: 70%), which unlike the previous result, test 16 managed to fit well into the dataset of pepper harvesting processes. The faster R-CNN model might deliver results with a low level of accuracy due to its use of an RPN (Region Proposal Network) stage, which tends to suffer from insufficient adjustment for our distribution of datasets.
- The best detection results reached using the ShuffleNet network depends on the compatibility between the detection model and the hyper-parameter configuration. The Faster R-CNN model obtained its best result in test 16 (training: 81%, validation: 77% and tests: 70%). For YOLOv7, the best result is from test 18 (training: 99%, validation: 99%, and test: 99%). Additionally, for the SSD model, the best result was obtained in test 13 (training: 79%, validation: 82% and tests: 80%). Based on the results presented in Table 5, the best model for this dataset is YOLOv7 with the hyperparameter configuration of test 18. This model has a low-level bias and variance, demonstrating that the model has high generalization (low overfitting) as accuracy is barely reduced in the training, validation, and testing stages.

4.4. Comparison of Inference Times

In this subsection, we concisely summarize in Table 6 the inference times of each object detection models, Faster R-CNN, YOLOv7, and SSD, in combination with the feature extraction networks GoogLeNet, ResNet-18, and ShuffleNet. When analyzing the inference times, it is observed that the Faster R-CNN object detection model is the algorithm that achieved the highest inference times compared to the rest of the models. Nevertheless, YOLOv7 and SSD models have low inference times as they employ a single-stage detection approach that does not require an RPN phase. This difference between the YOLOv7, SSD, and Faster R-CNN models is relevant since the Faster R-CNN model when processing an RPN stage will mean that the algorithm will need more time to infer [56,74].

Table 6. Comparison of inference times in seconds between Faster R-CNN, YOLOv7, and SSD and their respective feature extraction methods.

Detection Model	Faster R-CNN	YOLOv7	SSD
GoogLeNet (s)	0.0714	0.0175	0.0330
ResNet-18 (s)	0.1477	0.0203	0.0258
ShuffleNet (s)	0.0795	0.0381	0.0400

4.5. Comparison Results with State-of-the-Art Works

Table 7 shows a comparative analysis between this study and the works described in the state-of-the-art. This analysis allows us to highlight the difference between works that previously implemented object detection algorithms for pepper harvesting processes-oriented tasks or similar. It is important to note that this paper presents significant contributions that were not previously dealt with in the literature. This work uses three different methods for object detection (Faster R-CNN, YOLOv7, and SSD), in conjunction with three different methods of feature extraction (GoogLeNet, ResNet-18, and ShuffleNet).

Table 7. Comparison results with state-of-the-art works.

Paper	Objective or Task	Data base	Algorithm	Feature Extractor	mAP
Hespeler et al [55]	Chili peppers detection via thermal imaging	112 images & 112 thermal images	Yolov3 & Mask R-CNN	Mask R-CNN	97%
Ilyas et al [54]	Deep learning for Pepper plant disease	6000 images	DCNN	DCNN	91.7%
Sert et al [12]	Detect pepper and potato leaves	544 images	Faster R-CNN	GoogLeNet	98%
Zhang et al [47]	Multi-class object detection for harvesting	675 images	Faster R-CNN	Alexnet, VGG16 y VGG19	82%
Halstead et al [5]	Fruit Detection in the Wild	522 images	Faster R-CNN & Mask R-CNN	Mask R-CNN	90%
Nan et al [53]	Green pepper detection for pruned	1100 images	Yolov5I & NSGA II	NSGA II	81.4%
Sa et al [6]	Peduncle Detection of Pepper for Harvesting	72 images	Supervised learning	Supervised learning	71%
Arad et al [25]	Lighting Target Detection for Sweet Pepper	156 scenes & 468 images	SSD & FNF	Flash No Flash	84.07%
Yu et al [46]	Visual Localization of the Picking Points for Planting	100 images	R-Yolo	MobileNet V1	84.35%
Sa et al [35]	Fruit Detection with DL	122 images	Faster R-CNN	VGG 16	83.8%
This Work	Multi-class object detection for Pepper harvesting	420 images	YOLOv7, Faster R-CNN & SSD	GoogLeNet, ResNet-18 & ShuffleNet	Yolo: 99% Faster : 87% SSD: 91%

4.6. Discussion

This section summarizes the key findings, limitations, and future work opportunities regarding our work in object-of-interest CNN-based detection.

- During the implementation of object detection algorithms for pepper harvesting processes, it was observed that the YOLOv7 detection model is the architecture that stands out both in precision values and its inference times. By implementing the GoogLeNet, ResNet-18, and ShuffleNet feature extractors, the YOLOv7 model achieves an average performance of 99% during testing for all the feature extractors. This makes YOLOv7 a model capable of fulfilling the object of interest detection problem during pepper harvesting effectively. The object detection methods based on SSD also displayed high-performance results up to 91% during testing. However, they do not reach the accuracy values of YOLOv7. On the other hand, the Faster R-CNN model demonstrated to have underfitting issues for several tests that reached performances below 10%, and some punctual cases with moderate results from 39% up to 70% during testing.
- Despite the varying number of parameters across the CNNs utilized for feature extraction, no association was observed between the parameter count and the accuracy achieved by the trained models. Although ResNet-18 has more parameters than the GoogLeNet and ShuffleNet feature extractors, the results do not reflect an increase in accuracy when it is used. YOLOv7 was the model that delivered the best results in terms of accuracy scores, regardless of the feature extractor used (YOLOv7 with GoogLeNet reached up to 99.6%, YOLOv7 with ResNet-18 and ShuffleNet up to 99%). According to the results of the model, no correlation influences the accuracy of the results obtained in the model for our proposed pepper process dataset distribution. In the case of increasing the size of the dataset, this behavior might change. For future work, we will try to increase the dataset of our work to analyze this behavior and compare it with even more feature extraction methods.
- We have shown that CNN-based object detection methods may effectively identify objects of interest in pepper harvesting processes. In our work, we employed Faster R-CNN, YOLOv7, and SSD to detect human, box, pepper, pepper group, and pushcart categories. All specified object detection techniques were trained with GoogLeNet, ResNet-18, and ShuffleNet as feature extraction stages. YOLOv7 employing GoogLeNet achieved the highest performance over all the experiments, with 99% during training, 99.8% during validation, and 99.6% during testing. These findings signify a promising solution for pepper harvesting applications, including human-object action recognition, automated monitoring, and potential training tools for robotic perception applications. Furthermore, we have demonstrated a substantial enhancement relative to our findings in other studies, where typically a singular object detection method is employed across various applications, including chili pepper detection, plant disease identification, potato leaf detection, fruit detection, green pepper detection, among others.
- Considering the inference times obtained, the detector developed from YOLOv7 shows great potential for applications oriented to the detection of objects during Pepper harvesting processes in real-time, with response times ranging from 0.0175 to 0.0381 seconds when using different networks as feature extractors. This positions the model within the desired threshold of less than 0.25 seconds (in real-time), which is critical for a decision-making application in harvest-oriented environments. The SSD model also achieves fast inference times between 0.0330 and 0.0400 seconds. This makes it a viable option for scenarios that require online decision-making. On the other hand, the Faster R-CNN algorithm achieves inference times between 0.0714 and 0.0795 seconds, representing the slowest inference times of this work. Overall, the model that stands out is YOLOv7, making it the most efficient model for real-time object detection in applications oriented to object of interest detection during Pepper harvesting processes.
- We have demonstrated that it is possible to use CNN-based object detection techniques for pepper harvesting processes. However, the path to reaching a feasible monitoring system able to detect all the objects of interest in different farms is still long because our proposed system

has several limitations. For example, the dataset comprised a dataset of 210 RGB images of pepper harvesting tasks from a Chilean farm, which might be insufficient for generalizing the findings to a different location. The efficacy of deep learning models based on CNNs might work better with more extensive datasets from different farms and countries. It is also important to mention that the human actions and movements during harvesting depend on each field worker's style and individual techniques, which may vary based on their experience, physical capability, crop distribution, field characteristics, among others. The structure and specific attributes of the proposed dataset limit our understanding of the full potential. For this reason, our future works will include the study of data from more regions and countries. Moreover, demographic data from these locations may exhibit subtle variations relevant to this application. In addition, the computational complexity and reliance on CNNs necessitate the utilization of computers with graphics cards or even control cards with parallel processing capabilities, such as the Jetson, for this implementation in the field, for example, in robotics.

- For future work, extensive testing and deep hyper-parameter calibration will be performed to implement additional feature extractors to improve accuracy and confidence in our AI-based object of interest detection system for pepper harvesting processes. Moreover, other feature extraction methods can also be investigated, such as the use of Xception, MobileNet, VGGnet, and more.

5. Conclusions

In this work, we proposed an artificial intelligence algorithm based on the Faster R-CNN, YOLOv7, and SSD as object detection models for object of interest detection in pepper harvesting processes. For each object detector, we tested different CNN architectures (GoogLeNet, ResNet-18, and ShuffleNet) as a feature extractor to carry out the detection problem. A data augmentation process was carried out to expand the data to build a dataset of greater robustness and increase data variability. We perform training, validation, and testing for different hyper-parameter configurations. We also analyzed bias, variance, overfitting, and underfitting based on the training, validation, and testing results for each model. The best results from our experiments were obtained by YOLOv7 using GoogLeNet (training: 99%, validation: 99.8%, and test: 99.6%). These results offer promising solutions for object-of-interest detection during pepper harvesting applications, which can range from precision monitoring to human action recognition applications. Additionally, we have demonstrated a notable improvement over previous studies, which typically focus on training a single object detection method or are focused on other tasks such as fruit or vegetable detection, disease recognition, and leave detection, among others. In future works, we will extend the dataset to improve the proposed method's generalization capabilities. Moreover, other feature extraction methods can be also investigated, such as the use of Xception, MobileNet, and VGGnet, as well as further complex algorithms such as Mask-RCNN or Vision Transformers (ViTs).

Funding: This work has been supported by ANID (National Research and Development Agency of Chile) under Fondecyt Iniciación 2025 Grant 11240105, Grant 11230962, and Fondecyt Postdoctorado N°3250059. This work was funded by ANID – Millennium Science Initiative Program - NCN2024_047.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: This work has been supported by ANID (National Research and Development Agency of Chile) under Fondecyt Iniciación 2025 Grant 11240105, Grant 11230962, and Fondecyt Postdoctorado N°3250059. This work was funded by ANID – Millennium Science Initiative Program - NCN2024_047.

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

References

1. López-Barrios, J.D.; Escobedo Cabello, J.A.; Gómez-Espinosa, A.; Montoya-Cavero, L.E. Green sweet pepper fruit and peduncle detection using mask R-CNN in greenhouses. *Applied Sciences* **2023**, *13*, 6296.
2. Food and Agriculture Organization of the United Nations. FAOSTAT: Crops and livestock products – Chillies and peppers, green, 2018. Accessed online at <http://www.fao.org/faostat/en/>.
3. Zhang, C.; Zhang, Y.; Liang, S.; Liu, P. Research on Key Algorithm for Sichuan Pepper Pruning Based on Improved Mask R-CNN. *Sustainability* **2024**, *16*, 3416.
4. Sert, E.; Özen, M.; Ayalp, H. Leveraging Convolutional Neural Networks for Disease Detection in Vegetables: A Comprehensive Review. *Applied Sciences* **2024**, *14*, 7842. <https://doi.org/10.3390/app14177842>.
5. Halstead, M.; Denman, S.; Fookes, C.; McCool, C. Fruit detection in the wild: The impact of varying conditions and cultivar. In Proceedings of the 2020 digital image computing: techniques and applications (DICTA). IEEE, 2020, pp. 1–8.
6. Sa, I.; Lehnert, C.; English, A.; McCool, C.; Dayoub, F.; Upcroft, B.; Perez, T. Peduncle detection of sweet pepper for autonomous crop harvesting—combined color and 3-D information. *IEEE Robotics and Automation Letters* **2017**, *2*, 765–772.
7. Gkioxari, G.; Girshick, R.; Dollár, P.; He, K. Detecting and recognizing human-object interactions. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8359–8367.
8. Wan, L.; Zhu, W.; Dai, Y.; Zhou, G.; Chen, G.; Jiang, Y.; Zhu, M.; He, M. Identification of Pepper Leaf Diseases Based on TPSAO-AMWNet. *Plants* **2024**, *13*, 1581.
9. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2674. <https://doi.org/10.3390/s18082674>.
10. Oliveira, L.F.P.; Moreira, A.P.; Silva, M.F. Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead. *Robotics* **2021**, *10*, 52. <https://doi.org/10.3390/robotics10020052>.
11. Zhao, Y.; Villota-Eraso, L.D.; Escobedo Cabello, J.A.; Gómez-Espinosa, A. Maturity Recognition and Fruit Counting for Sweet Peppers in Greenhouses Using Deep Learning Neural Networks. *Agriculture* **2024**, *14*, 331. <https://doi.org/10.3390/agriculture14030331>.
12. Sert, E. A deep learning based approach for the detection of diseases in pepper and potato leaves. *Anadolu Tarım Bilimleri Dergisi* **2021**, *36*, 167–178.
13. Onishi, Y.; Yoshida, T.; Kurita, H.; Fukao, T.; Arihara, H.; Iwai, A. An automated fruit harvesting robot by using deep learning. *Robomech Journal* **2019**, *6*, 1–8.
14. Kang, H.; Zhou, H.; Wang, X.; Chen, C. Real-time fruit recognition and grasping estimation for robotic apple harvesting. *Sensors* **2020**, *20*, 5670.
15. Viveros Escamilla, L.D.; Gómez-Espinosa, A.; Escobedo Cabello, J.A.; Cantoral-Ceballos, J.A. Maturity recognition and fruit counting for sweet peppers in greenhouses using deep learning neural networks. *Agriculture* **2024**, *14*, 331.
16. Lehnert, C.; McCool, C.; Sa, I.; Perez, T. A sweet pepper harvesting robot for protected cropping environments. *arXiv preprint arXiv:1810.11920* **2018**.
17. Polic, M.; Tabak, J.; Orsag, M. Pepper to fall: a perception method for sweet pepper robotic harvesting. *Intelligent Service Robotics* **2022**, *15*, 193–201.
18. Horng, G.J.; Liu, M.X.; Chen, C.C. The smart image recognition mechanism for crop harvesting system in intelligent agriculture. *IEEE Sensors Journal* **2019**, *20*, 2766–2781.
19. Sun, Y.; Zhang, X.; Zhao, X.; Xin, Q. Extracting building boundaries from high resolution optical images and LiDAR data by integrating the convolutional neural network and the active contour model. *Remote Sensing* **2018**, *10*, 1459.
20. Ma, N.; Wu, Y.; Bo, Y.; Yan, H. Chili pepper object detection method based on improved YOLOv8n. *Plants* **2024**, *13*, 2402.
21. Paul, A.; Machavaram, R. Greenhouse capsicum detection in thermal imaging: A comparative analysis of a single-shot and a novel zero-shot detector. *Next Research* **2024**, *1*, 100076.
22. Sánchez, M.T.; Pintado, C.; de la Haba, M.J.; Torres, I.; García, M.; Pérez-Marín, D. In situ ripening stages monitoring of Lamuyo pepper using a new-generation near-infrared spectroscopy sensor. *Journal of the Science of Food and Agriculture* **2020**, *100*, 1931–1939.
23. Guevara, L.; Khalid, M.; Hanheide, M.; Parsons, S. Probabilistic model-checking of collaborative robots: A human injury assessment in agricultural applications. *Computers and Electronics in Agriculture* **2024**, *222*, 108987.

24. Vasconez, J.P.; Kantor, G.A.; Cheein, F.A.A. Human–robot interaction in agriculture: A survey and current challenges. *Biosystems engineering* **2019**, *179*, 35–48.
25. Arad, B.; Kurtser, P.; Barnea, E.; Harel, B.; Edan, Y.; Ben-Shahar, O. Controlled lighting and illumination-independent target detection for real-time cost-efficient applications. the case study of sweet pepper robotic harvesting. *Sensors* **2019**, *19*, 1390.
26. Yue, X.; Li, H.; Song, Q.; Zeng, F.; Zheng, J.; Ding, Z.; Kang, G.; Cai, Y.; Lin, Y.; Xu, X.; et al. YOLOv7-GCA: A Lightweight and High-Performance Model for Pepper Disease Detection. *Agronomy* **2024**, *14*, 618.
27. Si, J.; Kim, S. CRASA: Chili Pepper Disease Diagnosis via Image Reconstruction Using Background Removal and Generative Adversarial Serial Autoencoder. *Sensors* **2024**, *24*, 6892.
28. Liu, X.; Yu, J.; Kurihara, T.; Wu, C.; Niu, Z.; Zhan, S. Pixelwise complex-valued neural network based on 1D FFT of hyperspectral data to improve green pepper segmentation in agriculture. *Applied Sciences* **2023**, *13*, 2697.
29. Shao, Y.; Ji, S.; Xuan, G.; Ren, Y.; Feng, W.; Jia, H.; Wang, Q.; He, S. Detection and analysis of chili pepper root rot by hyperspectral imaging technology. *Agronomy* **2024**, *14*, 226.
30. Ma, Y.; Zhang, S. YOLOv8-CBSE: An Enhanced Computer Vision Model for Detecting the Maturity of Chili Pepper in the Natural Environment. *Agronomy* **2025**, *15*, 537.
31. Du, P.; Chen, S.; Li, X.; Hu, W.; Lan, N.; Lei, X.; Xiang, Y. Green pepper fruits counting based on improved DeepSort and optimized YOLOv5s. *Frontiers in Plant Science* **2024**, *15*, 1417682.
32. Jang, Y.; Schafleitner, R.; Barchenger, D.W.; Lin, Y.p.; Lee, J. Evaluation of heat stress response in pepper (*Capsicum annuum* L.) seedlings under controlled environmental conditions using a high-throughput 3D multispectral phenotyping. *Scientia Horticulturae* **2025**, *345*, 114136.
33. Guevara, L.; Hanheide, M.; Parsons, S. Implementation of a human-aware robot navigation module for cooperative soft-fruit harvesting operations. *Journal of Field Robotics* **2024**, *41*, 2184–2214.
34. Guevara, L.; Wariyapperuma, P.; Arunachalam, H.; Vasconez, J.; Hanheide, M.; Sklar, E. Robot-assisted fruit harvesting: a real-world usability study. In Proceedings of the 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), 2024, pp. 2517–2523.
35. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. Deepfruits: A fruit detection system using deep neural networks. *sensors* **2016**, *16*, 1222.
36. Islam, S.; Samsuzzaman.; Reza, M.N.; Lee, K.H.; Ahmed, S.; Cho, Y.J.; Noh, D.H.; Chung, S.O. Image Processing and Support Vector Machine (SVM) for Classifying Environmental Stress Symptoms of Pepper Seedlings Grown in a Plant Factory. *Agronomy* **2024**, *14*, 2043.
37. Han, Y.; Ren, G.; Zhang, J.; Du, Y.; Bao, G.; Cheng, L.; Yan, H. DSW-YOLO-Based Green Pepper Detection Method Under Complex Environments. *Agronomy* **2025**, *15*, 981.
38. Abdallatif, R.F.; Murad, W.; Abu-Naser, S.S. Classification of Peppers Using Deep Learning **2025**.
39. Guerrero-Mendez, C.; Navarro-Solís, D.; Saucedo-Anaya, T.; Lopez-Betancur, D.; Silva-Acosta, L.; Robles-Guerrero, A.; Gómez-Jiménez, S. Evaluating CNN Models and Optimization Techniques for Quality Classification of Dried Chili Peppers (*Capsicum annuum* L.). *International Journal of Combinatorial Optimization Problems & Informatics* **2024**, *15*.
40. Vasconez, J.P.; Admoni, H.; Cheein, F.A. A methodology for semantic action recognition based on pose and human-object interaction in avocado harvesting processes. *Computers and Electronics in Agriculture* **2021**, *184*, 106057.
41. Cong, P.; Li, S.; Zhou, J.; Lv, K.; Feng, H. Research on instance segmentation algorithm of greenhouse sweet pepper detection based on improved mask RCNN. *Agronomy* **2023**, *13*, 196.
42. Wang, F.; Sun, Z.; Chen, Y.; Zheng, H.; Jiang, J. Xiaomila green pepper target detection method under complex environment based on improved YOLOv5s. *Agronomy* **2022**, *12*, 1477.
43. Chen, H.; Zhang, R.; Peng, J.; Peng, H.; Hu, W.; Wang, Y.; Jiang, P. YOLO-chili: An efficient lightweight network model for localization of pepper picking in complex environments. *Applied Sciences* **2024**, *14*, 5524.
44. Lu, J.; Xiang, J.; Liu, T.; Gao, Z.; Liao, M. Sichuan pepper recognition in complex environments: A comparison study of traditional segmentation versus deep learning methods. *Agriculture* **2022**, *12*, 1631.
45. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Computers and electronics in agriculture* **2019**, *163*, 104846.
46. Yu, Y.; Zhang, K.; Liu, H.; Yang, L.; Zhang, D. Real-time visual localization of the picking points for a ridge-planting strawberry harvesting robot. *IEEE Access* **2020**, *8*, 116556–116568.

47. Zhang, J.; Karkee, M.; Zhang, Q.; Zhang, X.; Yaqoob, M.; Fu, L.; Wang, S. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Computers and Electronics in Agriculture* **2020**, *173*, 105384.
48. Yamamoto, K.; Guo, W.; Yoshioka, Y.; Ninomiya, S. On plant detection of intact tomato fruits using image analysis and machine learning methods. *Sensors* **2014**, *14*, 12191–12206.
49. Divya, S.; Panda, S.; Hajra, S.; Jeyaraj, R.; Paul, A.; Park, S.H.; Kim, H.J.; Oh, T.H. Smart data processing for energy harvesting systems using artificial intelligence. *Nano Energy* **2023**, *106*, 108084.
50. Dhanya, V.; Subeesh, A.; Kushwaha, N.; Vishwakarma, D.K.; Kumar, T.N.; Ritika, G.; Singh, A. Deep learning based computer vision approaches for smart agricultural applications. *Artificial Intelligence in Agriculture* **2022**, *6*, 211–229.
51. Fatchurrahman, D.; Castillejo, N.; Hilaili, M.; Russo, L.; Fathi-Najafabadi, A.; Rahman, A. A Novel Damage Inspection Method Using Fluorescence Imaging Combined with Machine Learning Algorithms Applied to Green Bell Pepper. *Horticulturae* **2024**, *10*, 1336.
52. Ren, R.; Zhang, S.; Sun, H.; Gao, T. Research on pepper external quality detection based on transfer learning integrated with convolutional neural network. *Sensors* **2021**, *21*, 5305.
53. Nan, Y.; Zhang, H.; Zeng, Y.; Zheng, J.; Ge, Y. Faster and accurate green pepper detection using NSGA-II-based pruned YOLOv5l in the field environment. *Computers and Electronics in Agriculture* **2023**, *205*, 107563.
54. Ilyas, T.; Jin, H.; Siddique, M.I.; Lee, S.J.; Kim, H.; Chua, L. DIANA: A deep learning-based paprika plant disease and pest phenotyping system with disease severity analysis. *Frontiers in Plant Science* **2022**, *13*, 983625.
55. Hespeler, S.C.; Nemati, H.; Dehghan-Niri, E. Non-destructive thermal imaging for object detection via advanced deep learning for robotic inspection and harvesting of chili peppers. *Artificial Intelligence in Agriculture* **2021**, *5*, 102–117.
56. Vasconez, J.P.; Delpiano, J.; Vougioukas, S.; Cheein, F.A. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. *Computers and Electronics in Agriculture* **2020**, *173*, 105348.
57. Peng, C.; Zhao, K.; Lovell, B.C. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern recognition letters* **2020**, *140*, 109–115.
58. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks* **2020**, *168*, 107036.
59. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 7464–7475.
60. Hussain, M. Yolov1 to v8: Unveiling each variant—a comprehensive review of yolo. *IEEE access* **2024**, *12*, 42816–42833.
61. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications* **2023**, *82*, 9243–9275.
62. Kumar, A.; Srivastava, S. Object detection system based on convolution neural networks using single shot multi-box detector. *Procedia Computer Science* **2020**, *171*, 2610–2617.
63. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, 2016, pp. 21–37.
64. Zhou, S.; Qiu, J. Enhanced SSD with interactive multi-scale attention features for object detection. *Multimedia Tools and Applications* **2021**, *80*, 11539–11556.
65. Dhillon, A.; Verma, G.K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence* **2020**, *9*, 85–112.
66. Yuesheng, F.; Jian, S.; Fuxiang, X.; Yang, B.; Xiang, Z.; Peng, G.; Zhengtao, W.; Shengqiao, X. Circular fruit and vegetable classification based on optimized GoogLeNet. *IEEE Access* **2021**, *9*, 113599–113611.
67. Al-Huseiny, M.S.; Sajit, A.S. Transfer learning with GoogLeNet for detection of lung cancer. *Indonesian Journal of Electrical Engineering and computer science* **2021**, *22*, 1078–1086.
68. Sabanci, K.; Aslan, M.F.; Ropelewska, E.; Unlarsen, M.F. A convolutional neural network-based comparative study for pepper seed classification: Analysis of selected deep features with support vector machine. *Journal of Food Process Engineering* **2022**, *45*, e13955.
69. Ojo, M.O.; Zahid, A. Deep learning in controlled environment agriculture: A review of recent advancements, challenges and prospects. *Sensors* **2022**, *22*, 7965.

70. Kim, B.S.; Yeom, H.G.; Lee, J.H.; Shin, W.S.; Yun, J.P.; Jeong, S.H.; Kang, J.H.; Kim, S.W.; Kim, B.C. Deep learning-based prediction of paresthesia after third molar extraction: A preliminary study. *Diagnostics* **2021**, *11*, 1572.
71. Ji, W.; Pan, Y.; Xu, B.; Wang, J. A real-time apple targets detection method for picking robot based on ShufflenetV2-YOLOX. *Agriculture* **2022**, *12*, 856.
72. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.
73. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.
74. Vasconez, J.P.; Salvo, J.; Auat, F. Toward semantic action recognition for avocado harvesting process based on single shot multibox detector. In Proceedings of the 2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA). IEEE, 2018, pp. 1–6.

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.