applied sciences

MDPI

Article

Coffee-Leaf Diseases and Pests Detection Based on YOLO Models

Jonatan Fragoso [1], Clécio Silva [1], Thuanne Paixão [1], Ana Beatriz Alvarez [1,*], Olacir Castro Júnior [1], Ruben Florez [2], Facundo Palomino-Quispe [2], Lucas Graciolli Savian [3] and Paulo André Trazzi [3]

[1] PAVIC Laboratory, University of Acre (UFAC), Rio Branco 69915-900, Brazil; jonatan.fragoso@sou.ufac.br (J.F.); clecio.rodri00@gmail.com (C.S.); thuannepaixao@gmail.com (T.P.); olacir.junior@ufac.br (O.C.J.)
[2] LIECAR Laboratory, Universidad Nacional de San Antonio Abad del Cusco (UNSAAC), Cuzco 08003, Peru; ruben.florez@unsaac.edu.pe (R.F.); facundo.palomino@unsaac.edu.pe (F.P.-Q.)
[3] Phytopathology Laboratory, University of Acre (UFAC), Rio Branco 69915-900, Brazil; lucassavian17@gmail.com (L.G.S.); paulo.trazzi@ufac.br (P.A.T.)
* Correspondence: ana.alvarez@ufac.br

Abstract: Coffee cultivation is vital to the global economy, but faces significant challenges with diseases such as rust, miner, phoma, and cercospora, which impact production and sustainable crop management. In this scenario, deep learning techniques have shown promise for the early identification of these diseases, enabling more efficient monitoring. This paper proposes an approach for detecting diseases and pests on coffee leaves using an efficient single-shot object-detection algorithm. The experiments were conducted using the YOLOv8, YOLOv9, YOLOv10 and YOLOv11 versions, including their variations. The BRACOL dataset, annotated by an expert, was used in the experiments to guarantee the quality of the annotations and the reliability of the trained models. The evaluation of the models included quantitative and qualitative analyses, considering the mAP, F1-Score, and recall metrics. In the analyses, YOLOv8s stands out as the most effective, with a mAP of 54.5%, an inference time of 11.4 ms and the best qualitative predictions, making it ideal for real-time applications.

Keywords: You Only Look Once; object detection; diseases and pests detection; coffee leaves; precision agriculture

check for updates

Academic Editor: Andrea Prati

Received: 21 March 2025
Revised: 17 April 2025
Accepted: 30 April 2025
Published: 1 May 2025

Citation: Fragoso, J.; Silva, C.; Paixão, T.; Alvarez, A.B.; Júnior, O.C.; Florez, R.; Palomino-Quispe, F.; Savian, L.G.; Trazzi, P.A. Coffee-Leaf Diseases and Pests Detection Based on YOLO Models. *Appl. Sci.* **2025**, *15*, 5040. https://doi.org/10.3390/app15095040

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

Coffee, one of the most widely consumed beverages in the world, is grown mainly in tropical regions and is an important source of income for millions of farmers [1]. In Brazil, the cultivation of coffee, especially the Arabica variety, plays a vital role in the economy, with the country being one of the largest global producers and exporters [2]. However, coffee production faces several challenges, one of the most significant being the diseases and pests that affect the plants, compromising the quality and quantity of production [3]. Among the most common diseases are rust, miner, phoma, and cercospora, which, if not detected and controlled early, can cause irreparable damage to plantations [4].

The early detection of these diseases and pests is essential for implementing effective control and management strategies, reducing the economic and environmental impact [5]. Traditionally, the identification of these conditions is carried out by phytopathology specialists, through visual inspections and laboratory samples, which can be time consuming, cost a lot of money, and be limited by the specialist's ability to reach [3,6]. In view of this, faster and more accurate alternatives are emerging, such as the application of deep learning techniques, which have proven to be highly effective in image analysis, providing an automated and high-performance way to detect diseases and pests in real time [7–9].

The use of CNNs (convolutional neural networks) in computer-vision tasks has advanced considerably in recent years, enabling the accurate detection of complex patterns in images and the diagnosis of plant diseases [10]. Models such as the YOLO (You Only Look Once) [11] family are especially popular due to their efficiency and accuracy in image object detection. These models are able to identify and classify diseases and pests on leaves or plants with a high accuracy rate, even in variable lighting conditions, and capture angles, as proposed in [12].

YOLO is a neural network architecture that processes the entire image at once, optimizing detection to be carried out in real time. Other methods, such as the R-CNN [13], which extracts proposed regions for analysis, or the SSD (Single-Shot MultiBox Detector) [14], which uses multiple feature maps for detection at different scales, can also achieve real-time detection. However, R-CNN tends to be slower due to its region-selection process, while SSD can be less accurate on small objects due to its multi-scale approach.

From its first version, YOLO has evolved rapidly, with continuous improvements in subsequent versions, becoming more robust, faster, and more accurate. YOLO version 8 [15], for example, introduced significant improvements in architecture and training, optimizing performance in various detection tasks. Version 9 [16] improved the ability to adapt to different types of images and considerably reduced inference time. Version 10 of YOLO [17] made even more significant advances, with the introduction of new regularization techniques and better performance in complex environments, making it better suited to dealing with images of plants and agricultural diseases. Version 11 [18], the most recent, brought significant refinements in detection accuracy and the ability to classify multiple objects in a single image, which is crucial for the simultaneous identification of different diseases and pests on coffee leaves. This improvement was made possible by innovations in training techniques and deep learning algorithms, allowing YOLO 11 to become one of the most advanced networks for real-time detection [11].

These innovations have allowed YOLO to be successfully applied in a variety of areas, including precision agriculture, where the automated detection of diseases and pests can mean a faster and more effective response to the threats that compromise agricultural production [12,19,20]. The combination of high precision and speed of detection makes YOLO an ideal tool for helping farmers to identify problems in crops early on, improving crop management and reducing the excessive use of pesticides and other inputs.

Some studies using YOLO in agriculture have demonstrated its effectiveness in detecting diseases and pests in various crops. For example, Cheng et al. (2024) [12] applied YOLO to detect rice diseases, managing to quickly identify fungal and bacterial infections, which allowed for early and more targeted intervention. Abid et al. (2024) [20] used YOLO version 10 in tomato plantations in Bangladesh, achieving high accuracy rates in detecting pests such as whitefly and spider mite, with a significant impact on reducing pesticide use. On the other hand, Liu et al. (2023) [19] implemented YOLO in pineapple plantations, identifying not only diseases, but also nutritional deficiencies that could affect plant growth, demonstrating the versatility of the network in agriculture. Therefore, the use of neural networks such as YOLO in the detection of diseases and pests not only improves the efficiency of the process, but also contributes to more sustainable agricultural practices by enabling more precise and conscious control of the inputs applied to crops. With the continuous improvement of these technologies, it is hoped that, in the future, YOLO will be even more integrated into automated monitoring systems, benefiting farmers and researchers in the fight against the diseases and pests that affect global agricultural production.

Among the public datasets for disease detection and classification in Arabica coffee leaves are JMuBEN, JMuBEN2, and BRACOL (Brazilian Arabica Coffee Leaf). The JMuBEN [21] dataset has clippings of coffee leaves containing a single disease per

clipping and is commonly used for the classification task. This dataset is composed of three compressed files containing images of affected coffee leaves: the first file contains 7682 images of leaves infected with cercospora, the second has 8337 images of leaves with rust, and the third includes 6572 images of leaves affected by phoma. JMuBEN2 [21] consists of two additional compressed files. The first contains 16,979 images of leaves with the miner pest, while the second brings together 18,985 images of healthy leaves. In total, the JMuBEN and JMuBEN2 datasets add up to 58,555 images of coffee leaves, distributed among five classes, including cercospora, rust, phoma, miner, and healthy. In addition to the images, the datasets include detailed annotations about the condition of the leaves and the names of the diseases. Due to the lack of images with whole leaves and their respective markings, the JMuBEN and JMuBEN2 datasets are not used for the detection task. On the other hand, the BRACOL [22] dataset originally proposed for the segmentation and classification tasks has 1747 images of whole leaves and disease clippings. Due to its name structure, which indicates the location of the disease clippings on the leaves, it was possible to create precise annotations, which facilitate its adaptation for object-detection models.

This paper provides the first systematic evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 architectures for the detection of diseases and pests in Arabica coffee leaves using a unified experimental protocol. The research found in the literature uses previous versions of YOLO applied to disease detection in other crops, making this paper uniquely relevant to both the coffee domain and more advanced object-detection architectures. The research explores the advanced capabilities of YOLO models, which stand out for their high detection speed and accuracy, making them fundamental for efficient plant monitoring. To ensure a high-quality database, the BRACOL dataset was revised and enhanced through an expert-guided annotation process, which included the correction of inconsistent labels and the addition of previously unmarked diseased regions. This meticulous refinement resulted in a more robust dataset, directly improving the reliability and generalization of the trained models. Furthermore, in addition to conventional accuracy metrics such as mAP, precision, recall, and F1-Score, the models were evaluated in terms of computational efficiency, including inference time and parameter count.

The main contributions of the paper are:

- The enhancement of the BRACOL dataset with markings reviewed by experts, ensuring higher quality data for training and evaluating the models. The dataset will be made available to the scientific community.
- A comparative analysis of the performance of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 in miner, rust, phoma, and cercospora disease detection in coffee leaves.
- The identification of the most efficient model considering performance metrics (mAP, precision, recall, and F1-Score) and the shortest inference time in disease prediction.

This paper is organized as follows, Section 2 deals with the Literature Review, where the main related works are described. Section 3 details the materials and methods used. The experimental results and discussion are presented in Section 4, and Section 5 contains the conclusions of the research.

2. Literature Review

Several studies have explored the application of CNNs to image analysis in plants, highlighting the efficiency and precision of these techniques compared to traditional methods. In the area of coffee-leaf disease detection, approaches based on computer vision have shown promising results. Models such as YOLO have stood out for their ability to perform detections in real time, and are widely used in tasks that require quick and accurate identification of complex patterns. Recent research has investigated the use of different versions of YOLO, taking advantage of the improvements in efficiency and robustness introduced

throughout the updates. For example, versions such as YOLOv5 and YOLOv8 have been employed to identify pests and perform post-harvest production control [23,24], while more recent studies explore YOLOv10 and YOLOv11 in complex agricultural scenarios [25,26].

This section presents an analysis of the main studies related to the detection of crop diseases using neural networks and computer vision, with an emphasis on the contributions that use the YOLO architecture and its variations. The techniques used, the results achieved, and the limitations observed will be discussed, with the aim of contextualizing and substantiating the approach proposed in this paper.

In the study by Cheng et al. (2024) [12], an improved model based on YOLOv7-Tiny was proposed for identifying rice–leaf diseases, addressing specific detection challenges in agricultural environments. To mitigate interference from complex backgrounds and improve identification accuracy, the model incorporated the CBAM (Convolutional Block Attention Module) attention module into the backbone network, as well as using the RepGhost module to deal with the irregularity of the regions affected by the diseases. To separate the classification and localization tasks, a lightweight head based on the YOLOX decoupled head was introduced. The experimental results showed that the model outperformed other architectures, including YOLOv3-Tiny and YOLOv5-S, achieving a mAP@0.5 of 0.922 and an inference time of 26.4 ms per image, making it suitable for use in embedded devices.

Similarly, the research by Surya and Santhi (2024) [25] explored the application of YOLOv10 for the advanced detection of rice–leaf diseases, with the aim of improving crop health management and minimizing production losses. The model was developed to overcome the limitations of traditional methods, such as SVM (Support Vector Machine), KNN (K-Nearest Neighbors) and CNNs, which, although effective in some tasks, present difficulties in real-time application and scalability for field scenarios. With its optimized architecture, YOLOv10 demonstrated high precision and speed in identifying leaf diseases, allowing for rapid diagnosis and early intervention. The proposed methodology contributes significantly to smart agriculture, enabling the more efficient monitoring of plantations and reducing the need for time-consuming manual inspections. The results indicated that the model not only outperformed conventional techniques, but also showed promise for implementation in automated systems, promoting more sustainable and efficient agricultural practices.

In the paper by Soeb et al. (2023) [27], the authors introduce an artificial intelligence-based solution to quickly and accurately diagnose and identify diseases in tea leaves. The approach uses the YOLOv7 model, known to be one of the fastest and most efficient object-detection networks, trained with a dataset of 4000 images of tea leaves collected from four prominent gardens in Bangladesh. This dataset was carefully annotated and augmented with data augmentation techniques to overcome the limitation of the number of samples. The experimental results demonstrate the superiority of YOLOv7 compared to other architectures, with metrics such as detection accuracy of 97.3% and mAP of 98.2%, among others. In addition to reducing manual effort and subjectivity in identification, the system has the potential to minimize economic losses in tea production, which faces challenges such as climate change and an increase in diseases.

In the paper by Liu et al. (2022) [19], an approach was proposed for detecting and locating pineapple fruit in natural environments, using binocular stereo vision and an improved model of YOLOv3. The improved model incorporates DenseNet into the Darknet-53 backbone network, optimizing the 13 $\times$ 13 and 26 $\times$ 26 resource layers, as well as integrating SPP-net (Spatial Pyramid Pooling) into the 52 $\times$ 52 dimension-detection module to improve information representation capacity. Pineapple detection is based on images captured by a binocular camera, which acquires left and right images, where the left image is

processed by the improved YOLOv3 model to identify the position of the fruit. The stereo-matching algorithm then calculates the parallax and stereo matching to determine the three-dimensional coordinates of the pineapples, based on the triangulation principle. Comparative tests showed that the proposed model outperformed other approaches, such as Faster-RCNN and Mobilenet-SSD, in terms of F1-Score and average precision (AP), standing out for its detection efficiency, even under occlusion conditions.

In the research conducted by Javierto et al. (2021) [28], a model based on YOLOv3 combined with MobileNetv2 was developed to detect diseases in Robusta coffee leaves, one of the main agricultural crops in the Philippines. This model, aimed at identifying biotic agents present in the leaves, seeks to help farmers make the correct diagnosis and appropriate treatment, reducing significant losses caused by misdiagnosis. The integration of YOLOv3 with MobileNetv2 allows image processing to work well even on devices with low-performance graphics units. With an accuracy of 90% in detecting diseases, the system proved to be effective, but only under conditions of sufficient lighting and uniformity in the background of the images used for training.

Other studies also aim to achieve sustainable pest management through the use of deep learning. However, they use the disease classification technique instead of real-time detection. This is the case of the paper by Albuquerque et al. (2024) [29], the ShuffleNet architecture was applied to the classification of diseases in coffee leaves, using data from the JMuBEN and JMuBEN2 sets. The methodology adopted included the use of convolutional neural networks and traditional computer-vision feature-extraction approaches. ShuffleNet, designed for mobile devices with limited resources, proved to be efficient, achieving good results even with a smaller volume of data and fewer parameters, compared to other architectures such as MobileNetV2 and VGG-16. External validation used other datasets to corroborate the robustness of the model, highlighting its superior performance in terms of computational cost and accuracy.

Similarly, the paper by Nawaz et al. (2023) [30] developed the CoffeeNet model, a deep learning-based approach for classifying coffee-leaf diseases. The research addresses common challenges in identifying these diseases, such as variations in lighting, differences in leaf coloration and the similarity between healthy and infected areas. To overcome these limitations, the authors proposed an improved version of the CenterNet architecture, incorporating a spatial- and channel-attention strategy based on the ResNet-50 model. This approach allowed for the extraction of more discriminative features from the images, improving the model's accuracy in classifying different types of infections. The experiments were conducted using a dataset of Arabica coffee leaves captured in realistic environmental conditions. The results obtained demonstrated the effectiveness of CoffeeNet, which achieved an accuracy of 98.54% and a mAP of 97%, standing out as a robust solution for the automated diagnosis of diseases in coffee plants.

Although existing studies show a wide application of deep learning in the agricultural context, the use of versions of YOLO has stood out, especially in plant disease and pest-detection tasks. As evidenced in the research by Javierto et al. (2021) [28], YOLO is effective in quickly and accurately detecting diseases in coffee leaves, making it crucial for monitoring plants in real time. In addition, YOLOv7 and YOLOv10, used in other agricultural research, such as the studies by Soeb et al. (2023) [27] and Surya and Santhi (2024) [25], have proved useful for controlling pests and diseases in various agricultural crops. However, the different approaches, including classification techniques and more specialized methods, point to the need for continuous improvements in the accuracy and robustness of these models, especially when dealing with variable lighting conditions and the diversity of diseases that affect plants.

Considering that no studies have been found in the literature using recent versions of the YOLO architectures to perform disease and pest detection on coffee leaves, this work focuses on the use of recent versions of YOLO, including YOLOv8, YOLOv9, YOLOv10, and YOLOv11. In order to analyze the performance of each model, experiments are carried out to assess the impact of recent updates to these architectures on the accuracy and speed of detection of specific coffee diseases. In addition, the expert review enhances the BRACOL dataset, strengthening data quality and contributing to the reliability of the trained models. In this way, this study not only expands the application of YOLO to precision agriculture in coffee cultivation, but also offers a detailed analysis of the architecture's advances in the specific application.

3. Materials and Methods

The YOLOv8, YOLOv9, YOLOv10, and YOLOv11 models were trained using the four diseases from the original dataset (miner, rust, phoma and cercospora), the markings labeled as healthy were disregarded. This section presents the materials and methods used, including a description of the dataset, the configuration and adaptation of the models, the evaluation metrics used and details of the training process.

3.1. Dataset

BRACOL, developed by Esgario et al. [22], consists of 1747 images of Arabica coffee leaves, of which 274 correspond to healthy leaves, while the rest contain one or more diseases. In addition, a subset was generated by cutting out symptomatic regions from the original images, so that each image contained only a single disease. This process resulted in a total of 1899 images of isolated diseases. The disease classes present in BRACOL include rust, miner, phoma, and cercospora, covering the main diseases affecting coffee cultivation. Some examples of images from the dataset can be seen in Figure 1.



Figure 1. Examples of images contained in the dataset, illustrating different conditions of coffee leaves, including the presence of diseases and pests.

In order to use the dataset, the images were processed so that they could be applied to object-detection models. Based on the names of the image files, information on the location of the disease clippings was extracted and annotations containing the coordinates of the affected regions on each original leaf were generated. These annotations were created using Roboflow (https://www.roboflow.com/, accessed on 2 March 2025) and served as ground truth for the object-detection task.

3.2. Diseases and Pests in Coffee Leaves

Based on [31], the rust disease, caused by the fungus *Hemileia Vastatrix*, manifests itself as yellowish spots that develop into an orange color, covering the underside of the leaves.

As the disease progresses, the infected leaves fall prematurely, reducing the plant's ability to carry out photosynthesis and compromising its fruit production.

Leaf miner, on the other hand, is caused by the larvae of the *Leucoptera Coffeella* insect, which dig galleries in the leaves, forming whitish, dry trails. Severe infestations result in early defoliation, directly impacting the productivity and quality of the coffee beans.

Another relevant disease is phoma, caused by the fungus *Phoma Costarricensis* , which is characterized by dark, circular lesions on leaves, branches, and fruit. These spots can grow and join together, leading to tissue necrosis and causing intense defoliation, especially in conditions of high humidity.

Cercospora, caused by the fungus *Cercospora Coffeicola*, presents circular brown lesions with light edges and a dark center, giving it an eye-like appearance. In addition to the leaves, the disease also affects the fruit, causing it to fall prematurely, and is especially aggravated in humid environments and in plants with nutritional deficiencies.

Figure 2 illustrates the different types of diseases analyzed in this study, highlighting the affected regions on the coffee leaves and the corresponding classification.

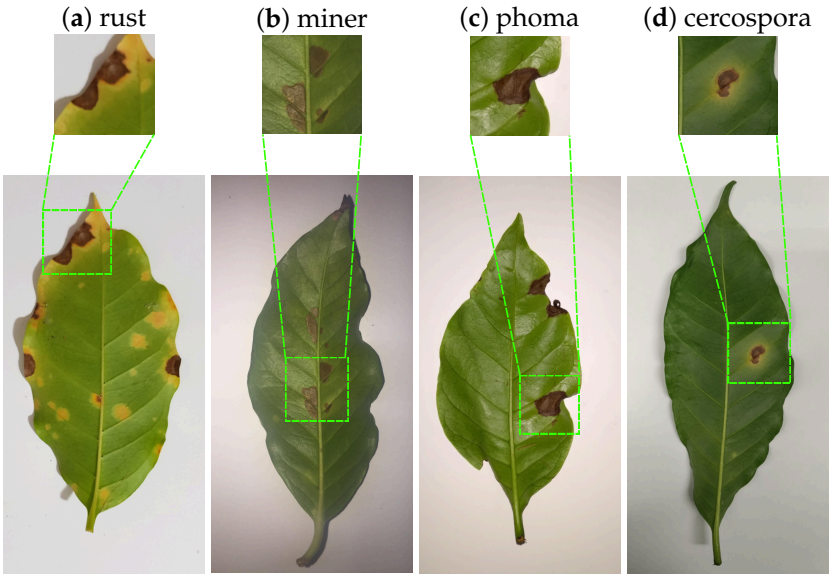


Figure 2. Examples of the main coffee-leaf diseases: (**a**) rust; (**b**) miner; (**c**) phoma; (**d**) cercospora.

3.3. Dataset Enhancement

The BRACOL dataset was enhanced to maximize data quality and optimize model training. The Roboflow platform was also used for this. To ensure that all the markings were accurate, the original BRACOL annotations were reviewed and improved with the help of a plant pathology expert. This process included re-evaluating the existing markings, correcting any inaccuracies, and adjusting the bounding boxxes of the annotated areas. New markings were added to previously unidentified areas, increasing the representativeness of the dataset. Approximately 900 of the 1747 images were updated during this process, resulting in a substantial increase in total annotations from 1899 to 8226. This refinement aimed to provide a more comprehensive and reliable dataset for training object-detection models. Figure 3 shows two samples to exemplify the improvement of the dataset; in sample 1 of the original dataset only one diseased region was marked. When revised, four more diseased regions were marked. In sample 2, only two diseased regions were initially marked on the sheet, and after reviewing the markings, all the visible diseases were correctly marked. It can be seen that one disease was initially classified inaccurately.

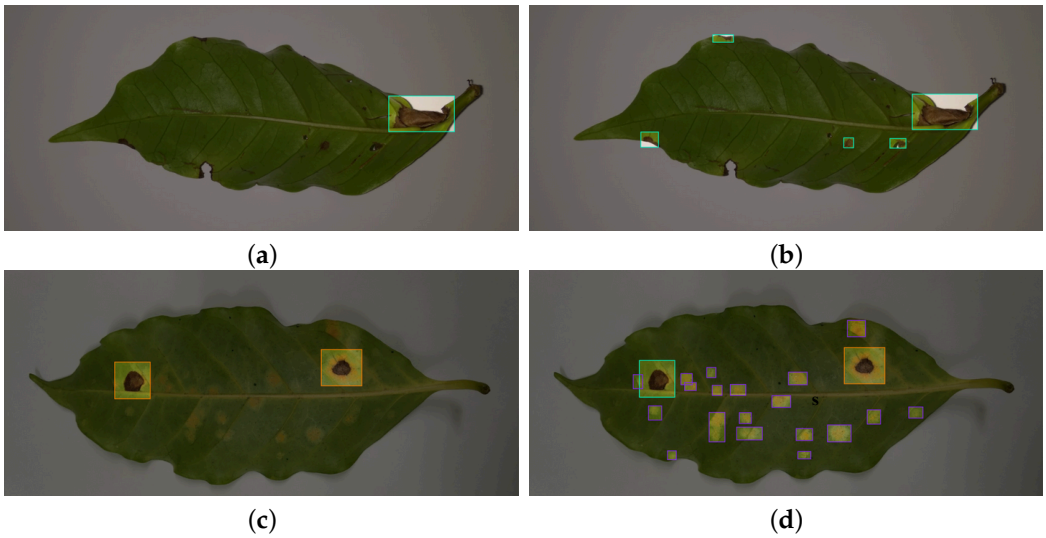


Figure 3. Samples of the original BRACOL and enhanced BRACOL. (**a**) Sample 1 in the original BRACOL dataset. (**b**) Sample 1 in the enhanced BRACOL dataset. (**c**) Sample 2 in the original BRACOL dataset. (**d**) Sample 2 in the enhanced BRACOL dataset.

In the enhanced version of the BRACOL dataset, a significant variation can be observed in the number of annotations across different disease classes. As shown in Table 1, rust and phoma were the most frequently annotated diseases, with 6013 and 1671 markings, respectively, while miner and cercospora had fewer instances, with 341 and 201 annotations. This class imbalance reflects the natural distribution of diseases observed in the field during the data-collection process and expert analysis. On the other hand, the evaluation metrics used, such as mAP and F1-Score, are calculated by class and averaged, which mitigates the impact of the dominant classes on the overall results.

Table 1. The distribution of disease markers in the BRACOL dataset.

Classes	Original Markings	Enhanced Markings
miner	540	341
rust	621	6013
phoma	464	1671
cercospora	274	201
Total	1899	8226

Table 1 shows the number of disease markings for the original BRACOL dataset and the improved BRACOL dataset. It can be seen that the total number of disease annotations has increased, reflecting an improvement in the representativeness of the diseases, ensuring greater accuracy in the ground truth and a more robust database for training and evaluating the models.

For the identification and localization of areas affected by diseases, the annotations of healthy regions do not represent areas of interest for the detection of diseases in coffee leaves and are therefore not being considered.

To enable the improved dataset to be used in future research into the detection of diseases and pests on coffee leaves, the dataset has been made publicly available through the Kaggle platform (https://www.kaggle.com/datasets/jonatanfragoso/bracol-for-yolov8-detection.), accessed on 2 March 2025.

3.4. Train, Validation and Test Split

To ensure an appropriate balance between the classes at all stages of the process, the improved BRACOL dataset was divided up in a stratified manner, ensuring that each subset preserved the original proportion of classes. In this way, distortions in learning caused by the predominance of a specific class were avoided, allowing the model to train, validate, and be tested with a representative distribution of leaf diseases. Based on [32], the training subset was made up of 70% of the images, ensuring a diverse and broad base so that the model could learn to identify the characteristics of the different classes. The validation subset, containing 20% of the images, was used to monitor the model's performance during training, helping to prevent problems such as overfitting when adjusting the hyperparameters. Finally, the test subset, with the remaining 10%, was reserved exclusively for the final evaluation of the model's performance, ensuring that the results reflect the actual ability to generalize to new data.

3.5. YOLO Family

The YOLO family of architectures [33,34] has revolutionized object detection by introducing a unified and efficient approach. Unlike previous methods such as R-CNN (Region-Based Convolutional Neural Networks) and its variants, which divided the problem into several stages, YOLO models process the image in a single way, allowing for real-time detection with high precision. The architecture is made up of three main components: the backbone, the neck, and the head. Based on [35], the general flow of the YOLO architecture is illustrated in Figure 4.

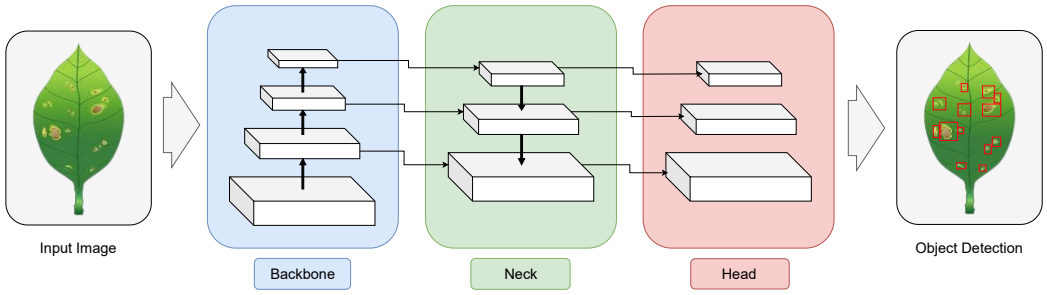


Figure 4. YOLO architecture, highlighting the main components: backbone, neck, and head.

The backbone is responsible for extracting visual characteristics from the image, such as textures and patterns. It acts as the basis of the network, processing the raw image and generating spatial representations that will be refined in the following stages. In versions of YOLO, the backbone is often implemented with architectures such as Darknet or CSPDarknet, which are efficient at extracting features. The neck acts as a bridge between the backbone and the head, and is crucial for combining information at different scales. Techniques such as the Feature Pyramid Network (FPN) or Path Aggregation Network (PANet) are often used to integrate features from multiple depth levels, allowing the YOLO architecture to have an enhanced ability to detect objects of varying sizes and resolutions. This ensures that both large and small objects are dealt with effectively [33].

Next, head generates multiple bounding boxes, assigns confidence scores and class probabilities for each detection. After generating these predictions, non-maximum suppression (NMS) is applied as a post-processing step to compare the boxes based on their overlap (using the Intersection over Union index - IoU) and keep only those with the highest confidence scores, discarding the rest. This results in a cleaner and more accurate final detection, avoiding duplications that could harm the system's efficiency. The integration of the NMS into YOLO's processing flow allows the model to maintain its accuracy even in

complex scenarios with multiple objects and overlapping [33]. The functioning between the head and the NMS is illustrated in Figure 5.

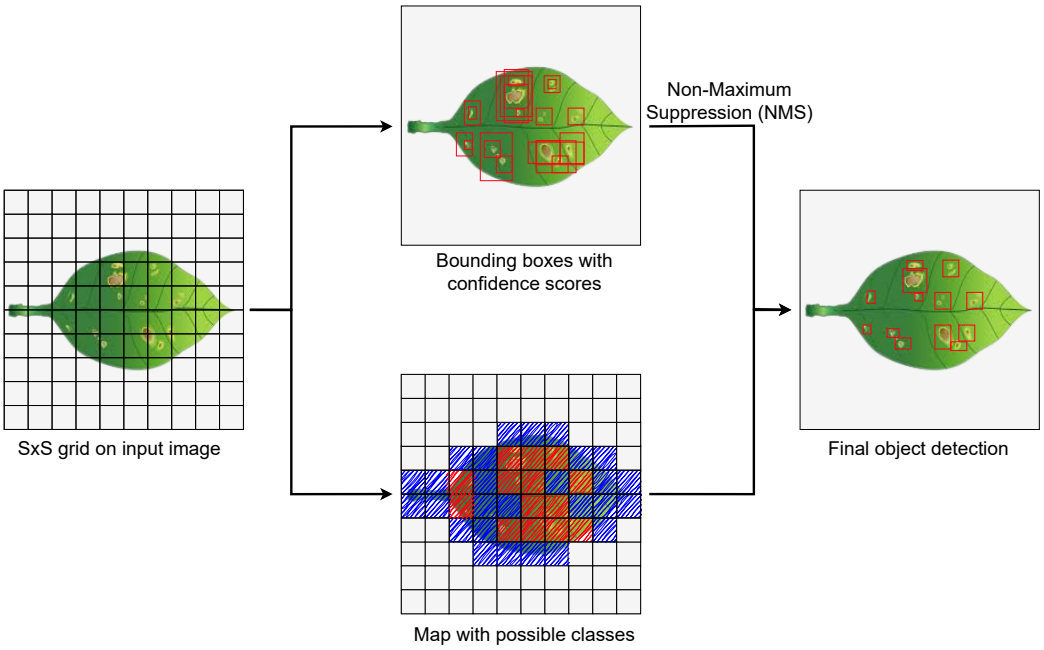


Figure 5. Operation of head and NMS in YOLO.

The operation of head is based on dividing the image into a grid, where each grid cell is responsible for predicting the presence of objects and their characteristics, such as position, size and class. This mechanism is represented in Figure 5, which highlights the process of subdividing the image and assigning detection responsibilities to the cells. After the bounding boxes, confidence scores and class probabilities have been generated by the head, the NMS is applied as a post-processing step to eliminate redundant or overlapping detections [35].

3.5.1. YOLOv8

YOLOv8 [15] was developed with the aim of reducing the number of parameters while maintaining or even improving detection accuracy compared to previous versions. Its main improvements include:

- The YOLOv8 Backbone uses CSPDarknet53+, an improved version of CSPDarknet, which incorporates additional convolutional layers and optimized CSP blocks. These improvements provide more efficient feature extraction and increase detection accuracy.
- YOLOv8 neck combines FPN (Feature Pyramid Network) with PANet (Path Aggregation Network), introducing SE (Squeeze-and-Excitation) blocks to improve the model's focus on essential features. The SE blocks apply a channel-attention mechanism, dynamically recalibrating the feature maps to highlight the most relevant information.
- The YOLOv8 detection head benefits from the refined feature maps generated by the enhanced backbone and neck, resulting in superior accuracy, especially for the detection of small and partially occluded objects.

YOLOv8 offers five variants, including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. Each variant adjusts the depth and width of the net to balance speed and precision, allowing for greater flexibility of use according to the needs of the target.

3.5.2. YOLOv9

The YOLOv9 [16] architecture is based on the advances of YOLOv8, with significant advances in the detection of objects in real time, standing out for its efficiency, precision, and adaptability. This model features innovations that optimize performance and the retention of essential information in deep neural networks.

One of the main innovations introduced by YOLOv9 is Programmable Gradient Information (PGI). This technique was developed to mitigate the loss of information in the deeper layers of the network, a recurring problem in neural architectures. PGI preserves crucial data throughout the feedforward process, allowing for the generation of reliable gradients that improve model updating. As a result, performance in detection tasks is considerably improved, even in complex scenarios.

YOLOv9 is available in five variants, including YOLOv9t, YOLOv9s, YOLOv9m, YOLOv9c, and YOLOv9e, offering different configurations to balance speed and accuracy, following the same philosophy as previous YOLO versions. Its main improvements include:

- The YOLOv9 backbone uses CSPResNeXt50, a new backbone architecture that combines CSP (Cross-Stage Partial) links with ResNeXt blocks. This integration allows for a more efficient representation of features, while reducing the complexity of the model.
- The YOLOv9 neck features a modified PANet (Path Aggregation Network), incorporating DCN (Deformable Convolutional Networks). These networks dynamically adjust their receptive fields according to the geometric variations of the objects, improving the detection of varied shapes and sizes.
- The YOLOv9 head maintains the multi-scale detection approach present in previous versions, but adds new refinement steps. This improves the accuracy of bounding boxes and class predictions, making detection more reliable.

3.5.3. YOLOv10

The YOLOv10 [17] architecture represents a significant evolution in real-time object detection. One of the main innovations of YOLOv10 is the introduction of consistent dual assignments, combining "one-to-many" and "one-to-one" strategies in training. This allows the model to learn efficiently without relying on NMS (non-maximum suppression), reducing latency and ensuring more accurate predictions during inference. The approach promotes rich supervision by aligning the training and inference processes in a holistic way. Its main improvements include:

- Backbone YOLOv10 introduces CSPNet++, a new backbone architecture that combines CSP (Cross-Stage Partial) with EfficientNet-B3. This hybrid architecture takes advantage of the compound scaling efficiency of EfficientNet and the gradient flow improvements of CSP, leading to a highly efficient feature-extraction process.
- With no neck, YOLOv10 adopts PAN (Path Aggregation Network), which improves the fusion of features at multiple scales, enabling better detection of objects in different sizes and conditions. This approach improves spatial representation and facilitates the efficient propagation of information within the network.
- The detection head maintains the traditional multi-scale approach of YOLO models, but optimizes the allocation of computational resources to make inference faster and more accurate. With these structural improvements, YOLOv10 is able to achieve superior performance without compromising speed, making it an ideal solution for real-time applications.

The YOLOv10 is available in different variants, including YOLOv10n, YOLOv10s, YOLOv10m, YOLOv10b, YOLOv10l, and YOLOv10x, providing options for balancing speed and accuracy according to the needs of the application. Its improved design al-

lows it to adapt efficiently to different scenarios, from detecting small objects in complex environments to high-performance applications with a lower computational load.

3.5.4. YOLOv11

YOLOv11 [35] redefines what is possible in terms of accuracy, speed, and efficiency. Building on the advances of previous versions of YOLO, YOLOv11 brings significant improvements in both architecture and training methods, making it a highly versatile choice for a wide range of computer-vision tasks.

One of the main features of YOLOv11 is improved feature extraction. The improved backbone and neck architecture offers superior feature-extraction capabilities, resulting in more accurate object detection and better performance in complex tasks. This improved architecture ensures that YOLOv11 is able to handle a wide variety of visual challenges with greater efficiency.

YOLOv11 offers state-of-the-art object-detection capabilities through a number of architectural improvements. Building on its predecessors, YOLOv11 integrates transform-based modules and advanced feature-extraction techniques, providing greater accuracy and efficiency in defect detection. Its main improvements include:

- YOLOv11 adopts a new backbone called CSP-DenseNet, which combines the dense connections strategy of DenseNet with CSP connections. This hybrid design improves feature reuse and gradient flow while reducing computational overhead, making the backbone highly effective for detecting intricate and overlapping features in extrusion defects.
- The YOLOv11 neck incorporates a transform-based Feature Pyramid Network (T-FPN), which uses self-attention mechanisms to improve feature aggregation at multiple scales. This architecture improves the model's ability to detect objects of varying sizes and shapes, especially in cluttered or noisy environments. In addition, the neck has DCNs, which dynamically adapt the receptive field to different defect geometries.
- The model head has an anchorless mechanism, combined with a lightweight dynamic prediction module, eliminating dependence on predefined anchor boxes. This improves detection efficiency and flexibility, while reducing computational complexity. The head is optimized for multi-scale prediction, ensuring the accurate regression and classification of bounding boxes, especially for subtle and complex defects such as ropes and spars.

3.6. Evaluation Metrics

Metrics estimate the performance of machine learning models by comparing the result with the desired objective. Based on [36], metrics commonly used in object-detection tasks such as precision, mean average precision, recall, and F1-Score were used to evaluate the YOLO models.

3.6.1. Precision (P)

It quantifies the proportion of true positives in relation to the total number of instances classified as positive by the model. Precision is calculated using Equation (1).

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{1}$$

where TP (True Positives) refer to instances correctly identified as positive, while FP (False Positives) represent instances incorrectly identified as positive.

3.6.2. Average Precision (AP)

It calculates the area under the precision–recall curve, resulting in a single value that reflects the model's performance in terms of precision and recall. Average precision is calculated using Equation (2).

$$\text{AP} = \int_0^1 P(R) \tag{2}$$

where $P(R)$ represents precision as a function of recall R.

3.6.3. Intersection over Union (IoU)

Measures the overlap between the box predicted by the model and the ground-truth box. It is calculated as the ratio between the area of intersection $|A \cap B|$ of the two boxes and the area of union $|A \cup B|$ between them. IoU values range from 0 to 1, where 1 indicates a perfect match between the prediction and the ground truth. The Intersection over Union is calculated using Equation (3).

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

3.6.4. Mean Average Precision (mAP)

Calculated as the average PA for all classes, considering different IoU thresholds. The mAP reflects the overall performance of the model in terms of precision and recall for all classes in the dataset. The mAP calculation is given by Equation (4).

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4}$$

where N represents the total number of classes and AP_i is the average precision of class i.

3.6.5. Recall

Also known as sensitivity or detection rate, it measures the ability of a model to correctly identify all the positive instances of a class. In other words, recall indicates the proportion of truly positive examples that have been correctly identified by the model in relation to the total number of positive examples present in the data. Recall is described mathematically by Equation (5).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

where FN is the number of False Negatives (instances incorrectly identified as negative).

3.6.6. F1-Score

A metric that combines precision and recall into a single measure, it is especially useful when there is an imbalance between classes. It is calculated as the harmonic mean between precision and recall, according to Equation (6).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

This metric is fundamental for assessing the model's ability to make correct predictions without overly favoring precision or recall, providing a balance between both measures.

4. Experimental Results and Discussion

4.1. Initial Considerations

Initially, to assess the improvements achieved with the improved BRACOL dataset, tests were carried out with the YOLOv8s model and the quantitative and qualitative results were compared with those using the original BRACOL dataset.

To evaluate the performance of the models, exhaustive experiments were carried out with the OLOv8, YOLOv9, YOLOv10, and YOLOv11 architectures. Versions v8, v10, and v11 were tested in four variations: nano (n), small (s), medium (m), and large (l). YOLO v9 was trained with specific variations: tiny (t), small (s), medium (m), and custom (c).

Based on [15–17,35], Table 2 lists the details of the models used in the analysis.

Table 2. Variations in the YOLO models and their respective number of parameters (in millions).

Model	Variations	Params (M)
YOLOv8	YOLOv8n	3.2
	YOLOv8s	11.2
	YOLOv8m	25.9
	YOLOv8l	43.7
YOLOv9	YOLOv9t	2.0
	YOLOv9s	7.2
	YOLOv9m	20.1
	YOLOv9c	25.5
YOLOv10	YOLOv10n	2.3
	YOLOv10s	7.2
	YOLOv10m	15.4
	YOLOv10l	24.4
YOLOv11	YOLO11n	2.6
	YOLO11s	9.4
	YOLO11m	20.1
	YOLO11l	25.3

Although the YOLO family models have optimized architectures for real-time detection, it is important to note that the number of parameters directly influences computational complexity and the feasibility of using them in devices with limited resources. Larger models, such as the large variants (YOLOv8l, YOLOv9c, YOLOv10l, and YOLOv11l), have more than 24 million parameters, requiring more processing power and memory, which can make them difficult to implement in embedded environments, such as drones or remote field stations. On the other hand, more compact variants, such as the nano and tiny versions, have lighter architectures, with less than 4 million parameters, favoring real-time applications that require low power consumption and fast response.

The results of the detection task are analyzed quantitatively and qualitatively, considering quantitative metrics such as mAP, F1-Score, and recall, and a qualitative analysis of four selected samples. These metrics allowed for a detailed assessment of the accuracy and efficiency of the models in diseases and pests detection on coffee leaves, enabling a comparison between different versions and variations.

4.2. Hardware

A laptop with an Intel Core i7-12700H processor, 64 GB of RAM, and an NVIDIA GeForce RTX 3050 laptop graphics card was used. This hardware, although limited compared to high-performance desktop configurations, offered sufficient capacity to efficiently train the nano (n), small (s), medium (m) and large (l) variant models.

4.3. Hyperparameters for Training

In order to maintain consistency in all experiments, allowing for fair performance comparisons between models, the hyperparameters used followed the values recommended by the Ultralytics framework [37]. For training, hyperparameters with default values were considered with a total of 300 epochs and early stopping was applied with a patience of 20 epochs to interrupt training if there was no improvement in the validation metric. Table 3 lists the hyperparameters used to train the models.

Table 3. Hyperparameters used to train the localization models.

Hyperparameter	Value
Input size	$640 \times 640 \times 3$
lr0	0.01
lrf	0.01
Optimizer	Adam
Weight Decay	0.0005
Batch Size	16
Epochs	300
Patience	20

4.4. Behavior with the BRACOL Enhanced Dataset

Initially, the models were tested using the enhanced BRACOL dataset to assess the impact of the dataset on disease detection in coffee leaves, and the results are compared with those achieved with the original BRACOL dataset. Table 4 shows the quantitative results obtained with the YOLOv8s model for the mAP, recall, and F1-Score metrics.

Table 4. The results of the YOLO model on the BRACOL original and BRACOL enhanced datasets.

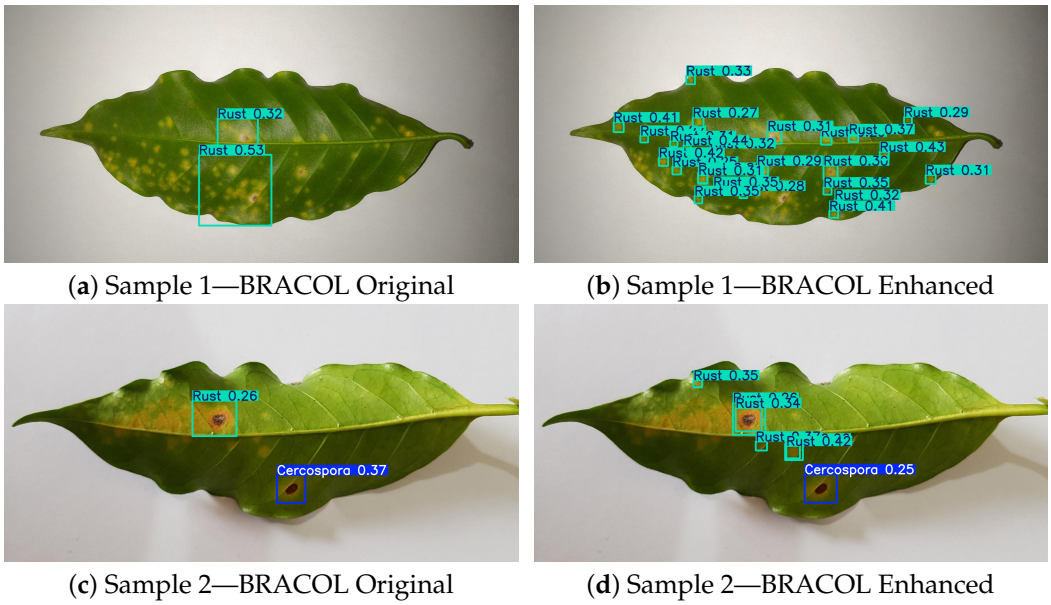
Model	BRACOL Original			BRACOL Improved		
	mAP (%) ↑	Recall (%) ↑	F1-Score (%) ↑	mAP (%) ↑	Recall (%) ↑	F1-Score (%) ↑
YOLOv8s	69.1	94	63	54.5	93	54

Note: ↑ indicates that higher values are better.

In order to more comprehensively evaluate the behavior of the models with the enhanced dataset, two samples were chosen from the test set for a qualitative analysis. Figure 6 shows the detection results in the samples using the model trained with the original dataset and with the enhanced dataset. In the predicted images, there are differences in the model's detections. In Figure 6a, the model detected only two larger regions. On the other hand, in Figure 6b, the model detected smaller areas and a greater number of affected regions. In Figure 6c, the model correctly detected two regions. In Figure 6d, in addition to detecting the two regions found in Figure 6c, the model was also able to correctly detect the other diseased regions.

Quantitative results show a significant difference in the metrics achieved with the datasets. The improved dataset increased the difficulty of the task because it has more marked regions and the model needs to detect them correctly. Specifically, higher mAP and F1-Score values achieved with the original dataset are justified, because originally some affected regions were not marked and the model was not penalized for not detecting them. In the improved dataset, any omission error is accounted for, thus considerably reducing the F1-Score. The mAP also tends to decrease in the results with the enhanced dataset because the model may not be detecting all the new labels correctly. From the qualitative results, although some regions are more difficult to detect, resulting in lower-confidence

detections, the model trained with the enhanced dataset has a superior ability to detect affected areas, ensuring that fewer diseased regions go unnoticed.



(**a**) Sample 1—BRACOL Original (**b**) Sample 1—BRACOL Enhanced

(**c**) Sample 2—BRACOL Original (**d**) Sample 2—BRACOL Enhanced

Figure 6. Results of disease and pest detection in coffee leaves using YOLOv8s.

4.5. Quantitative Results

Table 5 shows the quantitative results obtained by the different YOLO models trained. Each row in the table represents a specific model and its performance in terms of the mAP, recall, and F1-Score metrics.

Table 5. The evaluation results of the trained models.

Model	mAP (%) ↑	Recall (%) ↑	F1-Score (%) ↑
YOLOv8n	53.6	92	52
YOLOv8s	**54.5**	<u>93</u>	**54**
YOLOv8m	53.2	92	49
YOLOv8l	51.7	90	47
YOLOv9t	52.5	91	49
YOLOv9s	53.8	90	49
YOLOv9m	<u>54.0</u>	90	50
YOLOv9c	53.3	90	<u>53</u>
YOLOv10n	48.2	**94**	48
YOLOv10s	50.2	92	47
YOLOv10m	50.0	91	49
YOLOv10l	50.7	91	51
YOLOv11n	52.2	91	51
YOLOv11s	51.5	<u>93</u>	52
YOLOv11m	51.8	90	48
YOLOv11l	51.7	92	51

Note: The best result is shown in **bold** and the second best in <u>underline</u>. ↑ indicates that higher values are better.

The results show that the YOLOv8s model outperformed the YOLOv10n model in diseases and pests detection on coffee leaves, achieving a mAP of 54.5%, an F1-Score of 54%, and coming second in recall with 93%. The YOLOv9m, YOLOv9s, and YOLOv8n models also achieved relevant results with 54%, 53.8%, and 53.6%, respectively. In general terms, the YOLOv8 and YOLOv9 versions obtained the best results when evaluating the mAP metric compared to the YOLOv10 and YOLOv11 versions.

An analysis of the different versions of YOLO according to the mAP metric shows that in version 8, the YOLOv8s variation is the model with the best detection performance. In version 9, YOLOv9m stands out among the others with a mAP of 54%. In version 10, the YOLOv10l variation achieves a mAP of 50.7%, showing the best performance within the version. Finally, the YOLOv11n variation achieved a mAP value of 52.2%, outperforming the other variations in version 11.

To analyze the efficiency of the models in the task of disease detection in coffee leaves, the average times required for the different stages of image processing were calculated. The stages include pre-processing, which corresponds to preparing the image before inference, inference, which represents the time taken by the neural network to process and detect objects, and post-processing, which involves applying techniques such as non-maximum suppression (NMS) to refine the results. The total time is obtained by adding up these three metrics and reflects the time taken by each model to perform the detection task. Table 6 shows the average processing times for the images in the test set.

Table 6. Average processing time of YOLO models for image detection.

Model	PreProcess (ms) ↓	Inference (ms) ↓	PostProcess (ms) ↓	Total Time (ms) ↓
YOLOv8n	1.4	**6.7**	1.6	**9.7**
YOLOv8s	<u>1.3</u>	8.5	1.6	11.4
YOLOv8m	1.7	15.6	2.2	19.5
YOLOv8l	**1.2**	24.5	1.6	27.3
YOLOv9t	**1.2**	14.2	1.3	16.7
YOLOv9s	<u>1.3</u>	16.9	1.3	19.5
YOLOv9m	**1.2**	18.6	1.7	21.5
YOLOv9c	**1.2**	20.0	1.4	22.6
YOLOv10n	**1.2**	9.2	<u>0.5</u>	10.9
YOLOv10s	<u>1.3</u>	9.7	<u>0.5</u>	11.5
YOLOv10m	**1.2**	15.0	<u>0.5</u>	16.7
YOLOv10l	**1.2**	23.3	**0.4**	24.9
YOLOv11n	<u>1.3</u>	<u>8.2</u>	1.3	<u>10.8</u>
YOLOv11s	<u>1.3</u>	9.1	1.4	11.8
YOLOv11m	**1.2**	14.7	1.3	17.2
YOLOv11l	**1.2**	19.6	1.4	22.2

Note: The best result is shown in **bold** and the second best in <u>underline</u>. ↓ indicates that lower values are better.

The results indicate that YOLOv8n was the fastest model in terms of total processing time, averaging 9.7 ms per image. Next, YOLOv11n (10.8 ms), YOLOv10n (10.9 ms), and YOLOv8s (11.4 ms) also showed reduced processing times compared to the other models, making them viable options for applications requiring real-time detection. On the other hand, the more robust models, such as YOLOv8l (27.3 ms) and YOLOv10l (24.9 ms), showed significantly higher times, which impacts performance in scenarios where speed of response is an important factor.

In general, it can be seen that in all YOLO versions, the fastest variants are those with the fewest parameters, as shown in Table 2.

From the quantitative results achieved, it can be seen that the YOLOv8s model achieves excellent results in terms of performance and image-processing time, proving to be a good model for the real-time detection task. Due to the excellent results in terms of processing time and detection performance, the YOLOv8n model also appears to be an agile alternative.

The performance differences observed among the YOLO variants can be attributed to architectural trade-offs between model complexity and feature-extraction capabilities.

YOLOv8s achieved the best balance between detection accuracy and inference time, likely due to its optimized backbone (CSPDarknet53+) and SE attention modules that enhance focus on relevant features without significantly increasing computational load. In contrast, although YOLOv10 and YOLOv11 introduce more advanced components—such as CSPNet++ and transformer-based necks—these models require more processing time and, in some cases, do not outperform simpler versions in terms of mAP. The YOLOv11 variants, for example, use CSP-DenseNet backbones and anchorless heads, which may offer theoretical improvements but can also lead to instability or overfitting when applied to datasets with limited diversity like BRACOL. Furthermore, models with a smaller number of parameters (e.g., YOLOv8n, YOLOv10n) are faster but tend to underperform in complex images due to limited representation power.

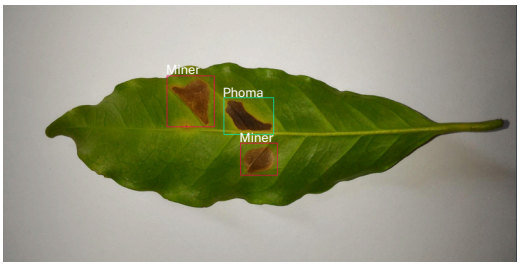
4.6. Qualitative Results

The qualitative analysis considers a visual appreciation of the accuracy of the model detections and the quality of the disease predictions in the samples. In this context, four samples were selected from the enhanced BRACOL test set containing at least one type of disease or pest. Specifically, sample 1 contains three large regions characterizing two different diseases. Sample 2 was chosen because it contains three different diseases occupying five medium regions on the leaf. In sample 3, there are nine small regions containing the same type of disease. Finally, sample 4 also has diseased regions of the same type as sample 3, but with different sizes. Within the YOLO models, the YOLOv8s, YOLOv9m, YOLOv10l, and YOLOv11n variations were used for the analysis, each one being the best variant of its version in terms of mAP.

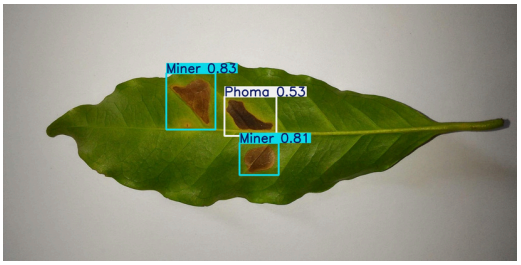
Figure 7 shows the ground truth (GT) of sample 1, containing two red markings indicating the presence of the miner pest, while a green marking indicates the region affected by the phoma disease. Also shown in the same figure are the visual results of disease detection in coffee leaves using the selected versions of YOLO. The bounding boxes indicate the regions detected, accompanied by the class labels and their respective confidence values. It can be seen that the YOLOv8s, YOLOv9m, and YOLOv10l versions have consistent results, with well-defined detections and relatively high confidence levels for both classes. In particular, YOLOv8s stood out for assigning high confidence to its predictions of the miner class, suggesting greater accuracy in detecting this particular disease. A similar behavior is shown by the YOLOv10l model and the phoma class. On the other hand, the YOLOv11n had its results compromised, showing a false positive in the detections for the phoma class.

Figure 8 shows the ground truth (GT) of sample 2, with three orange markings for cercospora disease, a green marking for phoma, and a red marking indicating the presence of the miner pest. The same figure shows the results of automatic detection using different versions of YOLO. It can be seen that the YOLOv8s and YOLOv10l versions obtained better detections for the cercospora class, while the YOLOv9m version showed multiple false positives, confusing the cercospora disease with the rust class. The YOLOv11n model had difficulties detecting cercospora, recognizing only phoma and miner with moderate levels of confidence.

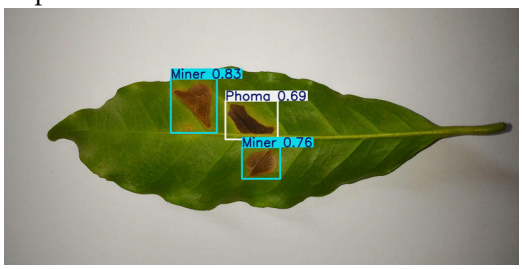
Figure 9 shows the ground truth (GT) of sample 3, indicating nine regions affected by rust disease on the leaf. Also shown are the results of automatic detection using different versions of YOLO. It can be seen that none of the versions managed to detect all nine regions. YOLOv8s and YOLOv9m were the most effective, identifying seven of the nine affected regions, although with variations in confidence levels. YOLOv10l and YOLOv11n, on the other hand, showed limitations, with fewer regions detected, indicating less sensitivity to diseases in smaller areas.



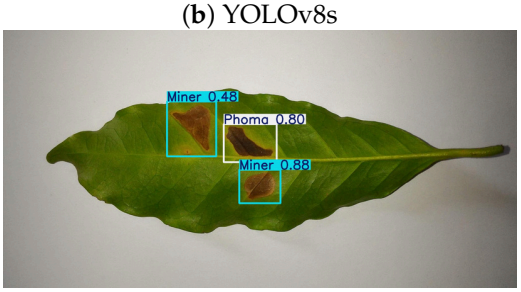
(**a**) GT Sample 1



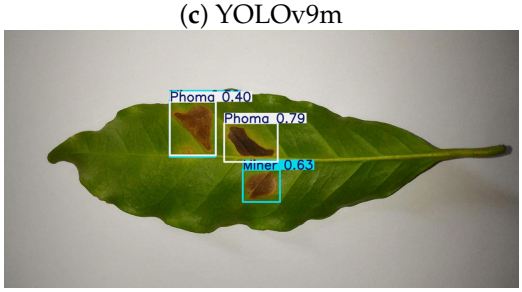
(**b**) YOLOv8s



(**c**) YOLOv9m

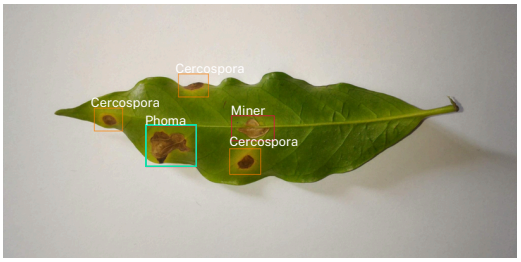


(**d**) YOLOv10l

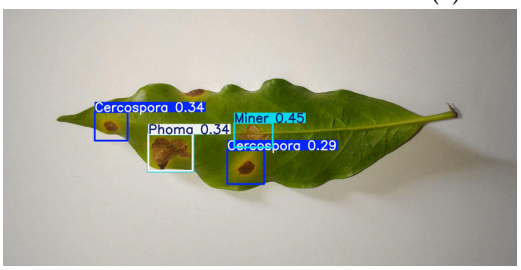


(**e**) YOLOv11n

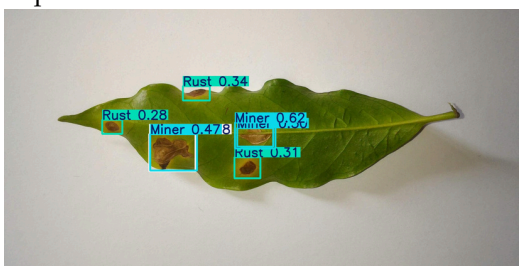
Figure 7. The GT of sample 1 and YOLO results from the detection of diseases and pests on coffee leaves. (**a**) GT of sample 1. (**b**) Detection of YOLOv8s. (**c**) Detection of YOLOv9m. (**d**) Detection of YOLOv10l. (**e**) Detection of YOLOv11n.



(**a**) GT Sample 2

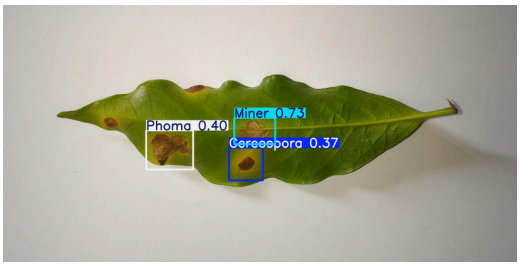


(**b**) YOLOv8s

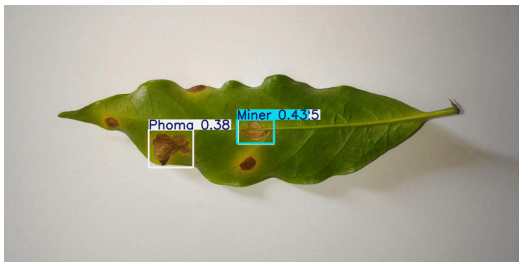


(**c**) YOLOv9m

Figure 8. *Cont.*

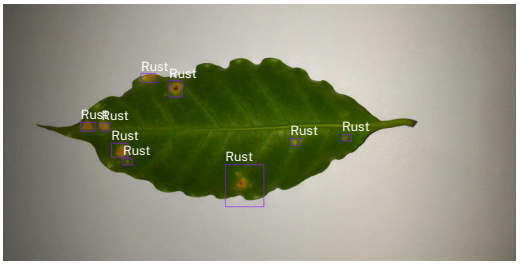


(**d**) YOLOv10l

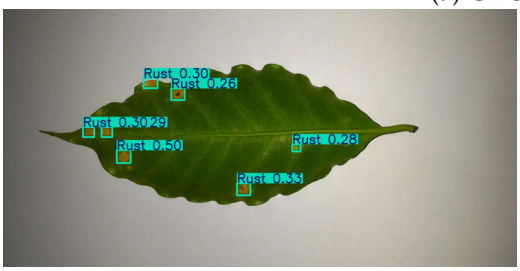


(**e**) YOLOv11n

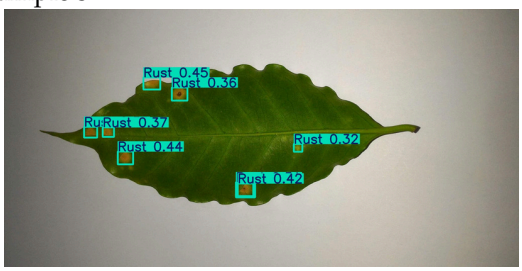
Figure 8. The GT of sample 2 and YOLO results from the detection of diseases and pests on coffee leaves. (**a**) GT of sample 2. (**b**) Detection of YOLOv8s. (**c**) Detection of YOLOv9m. (**d**) Detection of YOLOv10l. (**e**) Detection of YOLOv11n.



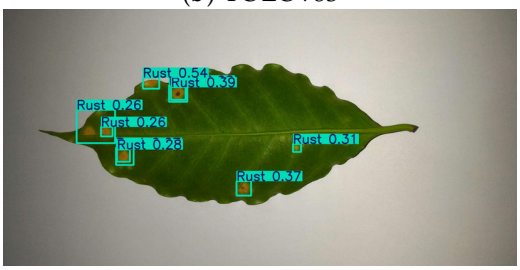
(**a**) GT Sample 3



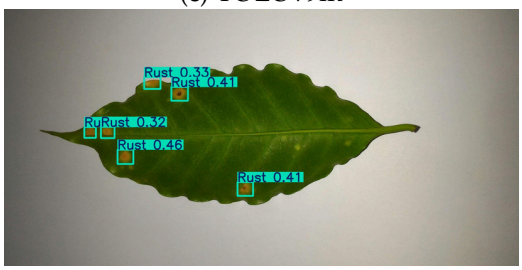
(**b**) YOLOv8s



(**c**) YOLOv9m



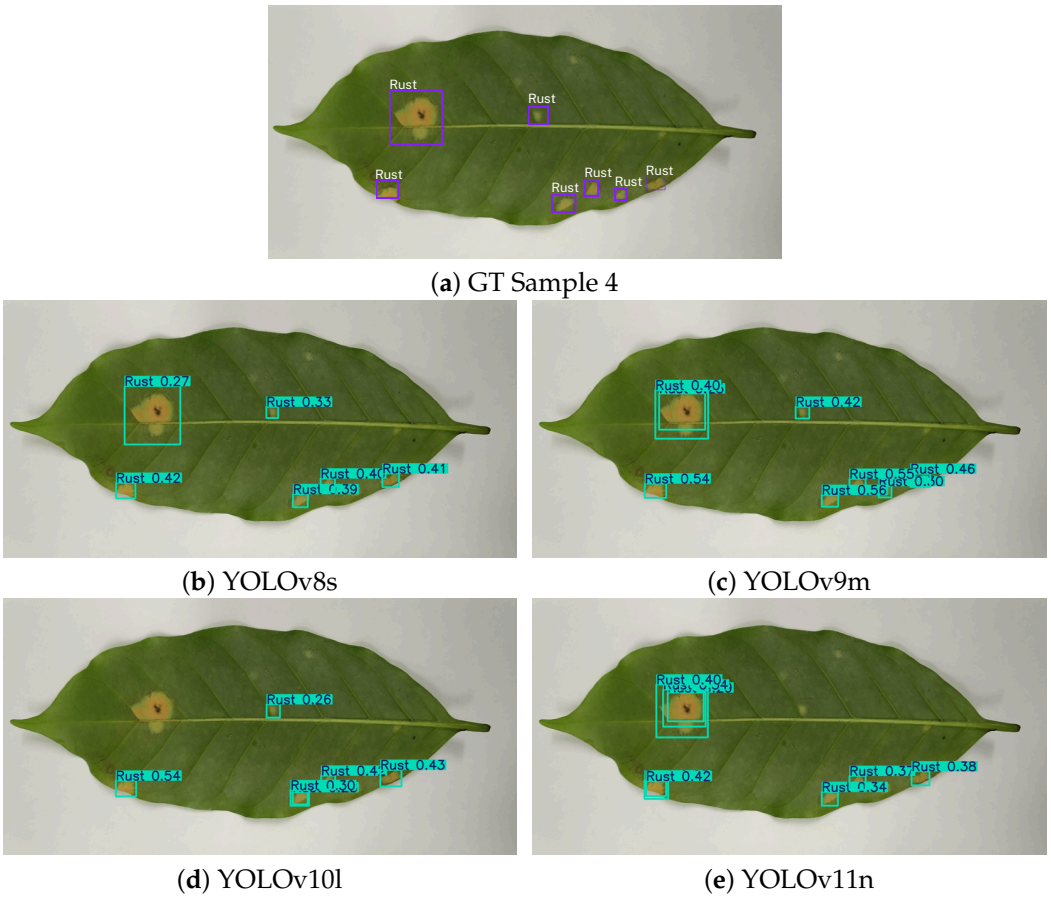
(**d**) YOLOv10l



(**e**) YOLOv11n

Figure 9. The GT of sample 3 and YOLO results from the detection of diseases and pests on coffee leaves. (**a**) GT of sample 3. (**b**) Detection of YOLOv8s. (**c**) Detection of YOLOv9m. (**d**) Detection of YOLOv10l. (**e**) Detection of YOLOv11n.

Figure 10 shows the ground truth (GT) of sample 4, where various regions of the leaf are marked, indicating the presence of the rust disease. The figure also shows the detection results using YOLO versions. It can be seen that the YOLOv8s and YOLOv9m models identified multiple regions with varying levels of confidence. The YOLOv10l model showed greater sensitivity when detecting smaller regions, but was the only one that failed to detect the largest region. YOLOv11n, on the other hand, correctly identified the disease occupying the largest region, which was detected three times. Also, the lack of detection of the smallest diseased region indicates the lower sensitivity of this model.



(**a**) GT Sample 4

(**b**) YOLOv8s (**c**) YOLOv9m

(**d**) YOLOv10l (**e**) YOLOv11n

Figure 10. The GT of sample 4 and YOLO results from the detection of diseases and pests on coffee leaves. (**a**) GT of sample 4. (**b**) Detection of YOLOv8s. (**c**) Detection of YOLOv9m. (**d**) Detection of YOLOv10l. (**e**) Detection of YOLOv11n.

The qualitative analysis describes the performance of the YOLO models in visually detecting diseases and pests on coffee leaves. In Figures 7–10, it can be seen that the YOLOv8s model showed consistency in the detection behavior of the four diseases, with well-defined bounding boxes and good confidence in the predictions, especially for the miner and rust classes regardless of the size of the location and the region affected. On the other hand, the YOLOv11n model showed difficulties in detecting cercospora and miner, generating predictions with low confidence and omitting some infected regions that were present in the ground truth. It was also observed that YOLOv9m had an intermediate performance, detecting instances of rust, but with low accuracy for the cercospora disease. YOLOv10l was also able to detect most instances of phoma and smaller regions of the rust class. However, the model showed some difficulty in differentiating cercospora from other diseases, as well as failing to detect larger regions of the rust class. In this scenario, the visual results reinforce the importance of a qualitative assessment complementing the quantitative analysis, highlighting the performance of YOLOv8s for detection.

5. Conclusions

This paper presents an analysis of single-shot models for the problem of visible detection of visible diseases and pests in Arabica coffee leaves. The analysis compared the performance of the YOLOv8, YOLOv9, YOLOv10, and YOLOv11 versions in miner, rust, phoma, and cercospora diseases detection. The experiments were carried out using the BRACOL dataset, which was revised and improved, with precise markings carried out by an expert in plant pathology. The analysis considers a quantitative and qualitative evaluation, taking into account the mAP, recall, F1-Score, and inference-time metrics, as well

as checking the visual precision of the detections made by the models. These metrics enabled a comprehensive assessment of the models' accuracy, efficiency, and generalization capacity, making them essential tools for disease detection.

From the analysis process carried out, it can be seen that the enhancement of the BRACOL dataset, as detailed in Table 1, brought significant improvements in the quality of the annotations. The total number of disease markings increased from 1899 to 8226, with the correction of inaccurate annotations and the addition of new affected regions that had not previously been identified. These improvements, made with the help of an expert in plant pathology, provided a more accurate representation of the diseases and pests present on coffee leaves. The experimental results indicated that the models trained with the improved dataset achieved consistent mAP, recall, and F1-Score values in detecting the diseases rust, miner, phoma, and cercospora. Additionally, the comparative analysis of the improved BRACOL dataset with the original dataset showed a qualitative contrast in the performance of the models evaluated using the modification.

The detection results achieved show that the YOLO models perform very well in detecting diseases and pests on coffee leaves, excelling both in quantitative metrics and in the visual evaluation of predictions. The comparison between the most recent versions of YOLO allowed for a detailed analysis of the differences in performance, highlighting their respective advantages and limitations in the scenario evaluated. The experimental results revealed that the YOLOv8 versions stand out significantly compared to the other models evaluated, both in terms of accuracy and prediction time. In particular, YOLOv8s showed the highest mAP with 54.5%, indicating its better ability to detect diseases. When evaluating the total detection time, the model reached 11.4 ms, managing to maintain an excellent prediction performance with significantly reduced times when compared to the other models. This attribute is fundamental for practical applications where the speed of object detection can be decisive for the efficiency of the system, such as autonomous or continuous monitoring systems. When evaluating the visual characteristics of detection, YOLOv8s demonstrates accurate performance in complex scenarios, where other models fail to identify diseases or present false detections.

The continuation of this research, together with the development of new technologies and resources, proposes for future work the expansion of the dataset by collecting images from different regions and environmental conditions to ensure greater robustness and the ability to generalize the models, as well as future collaborations with experts in plant pathology and agronomy to experimentally validate the model's accuracy in diagnosing diseases. In addition, there is interest in implementing fine-tuning mechanisms and adapting the model as new data are entered, ensuring continuous improvement in detection.

Another important future direction is addressing the class imbalance found in the enhanced BRACOL dataset, where diseases like rust and phoma appear much more often than miner and cercospora. Future work may include strategies such as data augmentation focused on the less frequent classes, adjusting the loss function to give them more weight, or using focal loss to help the model learn to detect these less common cases more effectively. Moreover, future studies could benefit from exploring and revisiting alternative architectures and techniques found in the literature, such as those presented in references [38,39]. These approaches may offer complementary advantages or improved performance in specific aspects of disease detection and localization, and their integration or comparison under a consistent experimental framework could further enrich the understanding of optimal strategies for plant disease identification.

Finally, expanding the comparison to include other object-detection architectures from the literature such as Faster R-CNN, EfficientDet, or MobileNet-based models presents a valuable direction for future research. Conducting these experiments under a uni-

fied evaluation protocol would allow for a more accurate and comprehensive assessment of how the YOLO models evaluated in this study compare within the broader state-of-the-art landscape.

Author Contributions: Conceptualization, J.F., C.S., T.P. and A.B.A.; methodology, J.F., C.S., O.C.J., T.P., A.B.A., R.F., F.P.-Q., L.G.S. and P.A.T.; software, J.F., C.S. and R.F.; validation, J.F., C.S., T.P. and A.B.A.; formal analysis, J.F. and C.S.; investigation, J.F., C.S., T.P., A.B.A. and F.P.-Q.; resources, A.B.A.; data curation, J.F., C.S., L.G.S. and P.A.T.; writing—original draft preparation, J.F., C.S. and T.P.; writing—review and editing, J.F., C.S., O.C.J., T.P., A.B.A. and F.P.-Q.; visualization, J.F. and C.S.; supervision, A.B.A. and F.P.-Q.; project administration, T.P., A.B.A. and F.P.-Q.; funding acquisition, A.B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the PAVIC Laboratory, University of Acre, Brazil.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The enhanced BRACOL dataset is available on the Kaggle platform (https://www.kaggle.com/datasets/jonatanfragoso/bracol-for-yolov8-detection) accessed on 3 March 2025.

Acknowledgments: The authors gratefully acknowledge support from the PAVIC Laboratory, and benefited from SUFRAMA fiscal incentives under Brazilian Law No. 8387/1991.

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

References

1. Quintam, C.P.R.; de Assunção, G.M. Perspectivas e desafios do agronegócio brasileiro frente ao mercado internacional. *RECIMA21-Rev. Científica Multidiscip.* **2023**, *4*, e473641.
2. Araújo, M.d.R.P.; da Silva, P.L.; da Rocha, A.P.S. Cafeicultura: evoluÇÃO do cafÉ no brasil, minas gerais e no municÍPIO de joÃO pinheiro–MG. *Rev. Contemp.* **2023**, *3*, 21683–21706. [CrossRef]
3. Naik, B.J.; Kim, S.C.; Seenaiah, R.; Basha, P.A.; Song, E.Y. Coffee cultivation techniques, impact of climate change on coffee production, role of nanoparticles and molecular markers in coffee crop improvement, and challenges. *J. Plant Biotechnol.* **2021**, *48*, 207–222. [CrossRef]
4. Yamashita, J.V.Y.B.; Leite, J.P.R. Coffee disease classification at the edge using deep learning. *Smart Agric. Technol.* **2023**, *4*, 100183. [CrossRef]
5. Paulos, E.B.; Woldeyohannis, M.M. Detection and classification of coffee leaf disease using deep learning. In Proceedings of the IEEE 2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA), Bahir Dar, Ethiopia, 28–30 November 2022; pp. 1–6.
6. Wang, S.; Xu, D.; Liang, H.; Bai, Y.; Li, X.; Zhou, J.; Su, C.; Wei, W. Advances in Deep Learning Applications for Plant Disease and Pest Detection: A Review. *Remote Sens.* **2025**, *17*, 698. [CrossRef]
7. Kamilaris, A.; Prenafeta-Boldú, F.X. A review of the use of convolutional neural networks in agriculture. *J. Agric. Sci.* **2018**, *156*, 312–322. [CrossRef]
8. Pinto, L.A.; Mary, L.; Dass, S. The real-time mobile application for identification of diseases in coffee leaves using the CNN model. In Proceedings of the IEEE 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; pp. 1694–1700.
9. Younesi, A.; Ansari, M.; Fazli, M.; Ejlali, A.; Shafique, M.; Henkel, J. A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends. *IEEE Access* **2024**, *12*, 41180–41218. [CrossRef]
10. Upadhyay, A.; Chandel, N.S.; Singh, K.P.; Chakraborty, S.K.; Nandede, B.M.; Kumar, M.; Subeesh, A.; Upendar, K.; Salem, A.; Elbeltagi, A. Deep learning and computer vision in plant disease detection: A comprehensive review of techniques, models, and trends in precision agriculture. *Artif. Intell. Rev.* **2025**, *58*, 1–64. [CrossRef]
11. Ali, M.L.; Zhang, Z. The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers* **2024**, *13*, 336. [CrossRef]
12. Cheng, D.; Zhao, Z.; Feng, J. Rice Diseases Identification Method Based on Improved YOLOv7-Tiny. *Agriculture* **2024**, *14*, 709. [CrossRef]

13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
14. 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, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
15. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
16. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; Springer Nature: Cham, Switzerland, 2024.
17. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
18. Khanam, R.; Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725.
19. Liu, T.H.; Nie, X.N.; Wu, J.M.; Zhang, D.; Liu, W.; Cheng, Y.F.; Zheng, Y.; Qiu, J.; Qi, L. Pineapple (Ananas comosus) fruit detection and localization in natural environment based on binocular stereo vision and improved YOLOv3 model. *Precis. Agric.* **2023**, *24*, 139–160. [CrossRef]
20. Abid, M.S.Z.; Jahan, B.; Al Mamun, A.; Hossen, M.J.; Mazumder, S.H. Bangladeshi crops leaf disease detection using YOLOv8. *Heliyon* **2024**, *10*, e36694. [CrossRef]
21. Jepkoech, J.; Mugo, D.M.; Kenduiywo, B.K.; Too, E.C. Arabica coffee leaf images dataset for coffee leaf disease detection and classification. *Data Brief* **2021**, *36*, 107142. [CrossRef]
22. Esgario, J.G.; Krohling, R.A.; Ventura, J.A. Deep learning for classification and severity estimation of coffee leaf biotic stress. *Comput. Electron. Agric.* **2020**, *169*, 105162. [CrossRef]
23. Yu, J.; Zhang, B. MDP-YOLO: A LIGHTWEIGHT YOLOV5S ALGORITHM FOR MULTI-SCALE PEST DETECTION. *Eng. Agrícola* **2023**, *43*, e20230065. [CrossRef]
24. Sá, P.C.A.; Quezia, A.; Marcus, C.; Júnior, C.L.; Maciel, A.M.; Bastos-Filho, C. YOLOv8 para Controle de Produção Pós-colheita e Beneficiamento de Frutos. *Rev. Eng. Pesqui. Apl.* **2024**, *9*, 115–122. [CrossRef]
25. Surya, V.; Santhi, S. Smart Agriculture: Advanced Rice Disease Detection Using YOLOv10 for Enhanced Crop Health Management. In Proceedings of the IEEE 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 4–6 December 2024; pp. 1733–1737.
26. Li, A.; Wang, C.; Ji, T.; Wang, Q.; Zhang, T. D3-YOLOv10: Improved YOLOv10-Based Lightweight Tomato Detection Algorithm Under Facility Scenario. *Agriculture* **2024**, *14*, 2268. [CrossRef]
27. Soeb, M.J.A.; Jubayer, M.F.; Tarin, T.A.; Al Mamun, M.R.; Ruhad, F.M.; Parven, A.; Mubarak, N.M.; Karri, S.L.; Meftaul, I.M. Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Sci. Rep.* **2023**, *13*, 6078. [CrossRef] [PubMed]
28. Javierto, D.P.P.; Martin, J.D.Z.; Villaverde, J.F. Robusta Coffee Leaf Detection based on YOLOv3-MobileNetv2 model. In Proceedings of the 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Manila, Philippines, 28–30 November 2021; pp. 1–6.
29. Albuquerque, L.D.; Guedes, E.B. Coffee Plant Leaf Disease Detection for Digital Agriculture. *J. Interact. Syst.* **2024**, *15*, 220–233. [CrossRef]
30. Nawaz, M.; Nazir, T.; Javed, A.; Amin, S.T.; Jeribi, F.; Tahir, A. CoffeeNet: A deep learning approach for coffee plant leaves diseases recognition. *Expert Syst. Appl.* **2024**, *237*, 121481. [CrossRef]
31. Ribeyre, F.; Avelino, J. Impact of field pests and diseases on coffee quality. In *Specialty Coffee: Managing Quality*; International Plant Nutrition Institute; IPNI [Southeast Asia]: Penang, Malaysia, 2012; pp. 151–176.
32. Jegham, N.; Koh, C.Y.; Abdelatti, M.; Hendawi, A. Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors. *arXiv* **2024**, arXiv:2411.00201.
33. Sharma, A.; Kumar, V.; Longchamps, L. Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. *Smart Agric. Technol.* **2024**, *9*, 100648. [CrossRef]
34. Bento, J.; Paixão, T.; Alvarez, A.B. Performance Evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 for Stamp Detection in Scanned Documents. *Appl. Sci.* **2025**, *15*, 3154. [CrossRef]
35. Sani, A.R.; Zolfagharian, A.; Kouzani, A.Z. Automated defects detection in extrusion 3D printing using YOLO models. *J. Intell. Manuf.* **2024**, *35*, 1–21. [CrossRef]
36. Padilla, R.; Netto, S.L.; Da Silva, E.A. A survey on performance metrics for object-detection algorithms. In Proceedings of the IEEE 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.

37. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Poznanski, J.; Yu, L.; Rai, P.; Ferriday, R.; et al. ultralytics/yolov5: V3. 0. *Zenodo* **2020** Available online: https://github.com/ultralytics/yolov5 (accessed on 5 March 2025). [CrossRef]
38. Tang, H.; Li, Z.; Zhang, D.; He, S.; Tang, J. Divide-and-Conquer: Confluent Triple-Flow Network for RGB-T Salient Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *47*, 1958–1974. [CrossRef]
39. Zhang, X.; Yin, B.; Lin, Z.; Hou, Q.; Fan, D.P.; Cheng, M.M. Referring camouflaged object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 3597–3610. [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.