mathematics

MDPI

Article

Deep Learning Architecture for Tomato Plant Leaf Detection in Images Captured in Complex Outdoor Environments

Andros Meraz-Hernández, Jorge Fuentes-Pacheco, Andrea Magadán-Salazar *, Raúl Pinto-Elías and Nimrod González-Franco

Tecnológico Nacional de México, Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Cuernavaca 62490, Morelos, Mexico; d20ce040@cenidet.tecnm.mx (A.M.-H.); jorge.fp@cenidet.tecnm.mx (J.F.-P.); raul.pe@cenidet.tecnm.mx (R.P.-E.); nimrod.gf@cenidet.tecnm.mx (N.G.-F.)

* Correspondence: andrea.ms@cenidet.tecnm.mx

Abstract

The detection of plant constituents is a crucial issue in precision agriculture, as monitoring these enables the automatic analysis of factors such as growth rate, health status, and crop yield. Tomatoes (*Solanum* sp.) are an economically and nutritionally important crop in Mexico and worldwide, which is why automatic monitoring of these plants is of great interest. Detecting leaves on images of outdoor tomato plants is challenging due to the significant variability in the visual appearance of leaves. Factors like overlapping leaves, variations in lighting, and environmental conditions further complicate the task of detection. This paper proposes modifications to the Yolov11n architecture to improve the detection of tomato leaves in images of complex outdoor environments by incorporating attention modules, transformers, and WIoUv3 loss for bounding box regression. The results show that our proposal led to a 26.75% decrease in the number of parameters and a 7.94% decrease in the number of FLOPs compared with the original version of Yolov11n. Our proposed model outperformed Yolov11n and Yolov12n architectures in recall, F1-measure, and mAP@50 metrics.

Keywords: object detection; tomato plant leaves; deep learning; convolutional neural network; attention mechanism

MSC: 68T07

Academic Editor: Shaozi Li

Received: 15 June 2025
Revised: 4 July 2025
Accepted: 7 July 2025
Published: 22 July 2025

Citation: Meraz-Hernández, A.; Fuentes-Pacheco, J.; Magadán-Salazar, A.; Pinto-Elías, R.; González-Franco, N. Deep Learning Architecture for Tomato Plant Leaf Detection in Images Captured in Complex Outdoor Environments. *Mathematics* **2025**, *13*, 2338. https://doi.org/10.3390/math13152338

1. Introduction

The introduction of artificial intelligence in agriculture has revolutionized the way problems are solved. Due to its ability to learn hierarchical representations of data, deep learning (DL) is the most suitable technique for developing computer vision systems with high performance and reliability, even on data very different from those used in training [1]. Detection and segmentation are fundamental tools in many vision systems [2]. An object detector identifies and categorizes specific regions of pixels within the image, assigning a class label to each object and drawing a box that encloses it. Segmentation classifies each image pixel (considering features such as color, texture, and shape) and groups them according to their class.

Detecting and counting leaves in RGB images is essential in precision agriculture (PA), because it indicates the growth rate, health status, and crop yield of plants. Automated detection and count reduce the need for manual inspections, which are costly and time-consuming in extensive agricultural areas. We aimed to detect tomato (*Solanum* sp.) plant

leaves in RGB images captured in uncontrolled conditions. Tomato plants have compound leaves that vary in shape and size depending on the genetic characteristics of the variety and the growing conditions. Each leaf has seven to nine leaflets, measuring 4 to 60 mm in length and 3 to 40 mm in width. Generally, they are green in color, glandular pubescent on the upper surface, and exhibit an ashen appearance on the underside [3].

Mexico is the world's leading supplier of tomatoes, holding a 25.11% share of the international market in terms of the value of global exports. In 2016, Mexican tomatoes accounted for 90.67% of imports to the United States and 65.31% to Canada, contributing 3.46% to the gross domestic product (GDP). By 2030, global demand is estimated to increase from 8.92 to 11.78 million metric tons, representing a cumulative growth of 32.10%. Meanwhile, national tomato production may increase from 3.35 to 7.56 million metric tons, yielding a cumulative increase of 125.80% [4].

Various DL models based on semantic segmentation have been used to determine leaf regions. The images used in these works typically contain a single dominant leaf. A person using a cell phone carefully captures an image of the leaf with a uniform background for easy analysis [5–7], or the background may be complex [8]. However, when the camera is handheld or mounted on a robot, semantic segmentation is no longer helpful, and we must resort to instance-based segmentation, as the images contain different numbers and types of leaves [9,10]. Nevertheless, we do not always require precise leaf segmentation and only need the leaf's location in the image. Object detection may be more appropriate for robotic localization and inspection, as well as for monitoring systems and mobile device applications in greenhouse and open-field growing environments.

Object detection is faster than semantic or instance-based segmentation, which makes it ideal for real-time applications. In addition, image labeling for object detection is more straightforward, as it involves only setting bounding boxes around objects, and there is no need to apply a mask per object. Likewise, object detection could be the first stage of a process to segment each leaf quickly and efficiently, thus avoiding the segmentation of the entire image and generating a more accurate extraction of the leaf [11].

Images of leaves captured in real-world situations can be affected by changes in camera pose, variability in illumination, and variations in the size, shape, and texture of the leaves. Images may include several leaves overlapping and under different focuses or may contain other background objects like weeds, soil, fruits, and branches. Table 1 lists the challenges of detecting tomato leaves in images: [12–17]. In each case, we show the regions of interest predicted (red boxes) by an object detector, which could be true positives, false positives, or false negatives.

Table 1. Challenges in detecting plant leaves in uncontrolled environments.

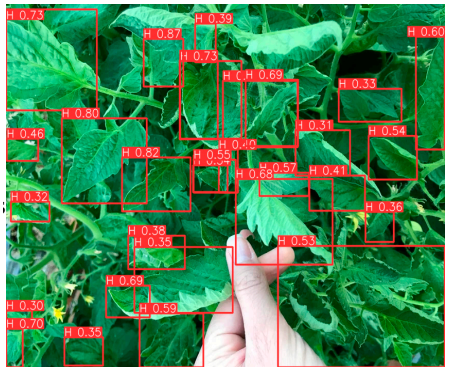
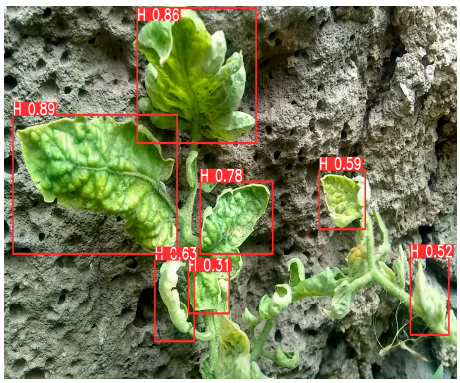
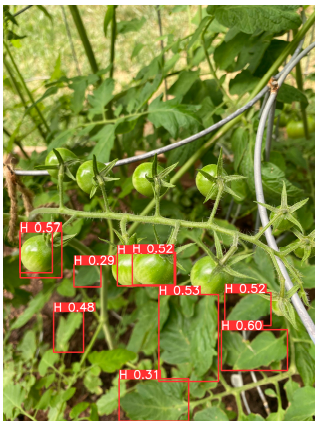
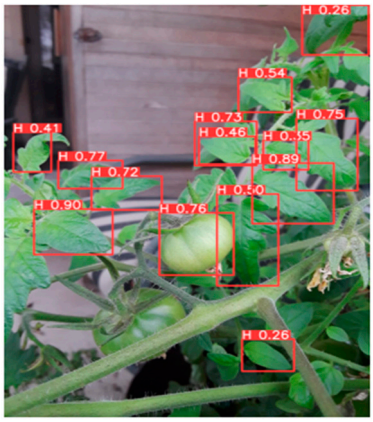
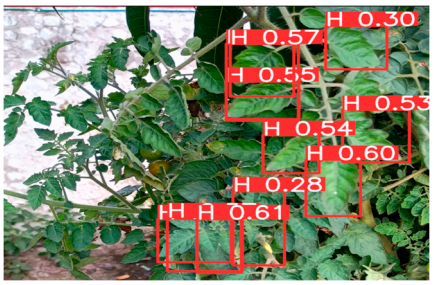
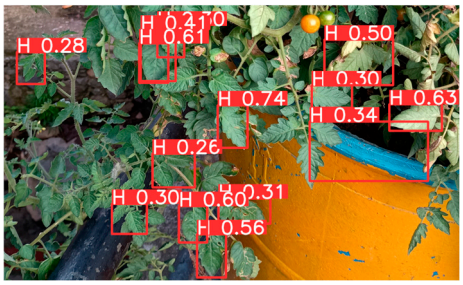
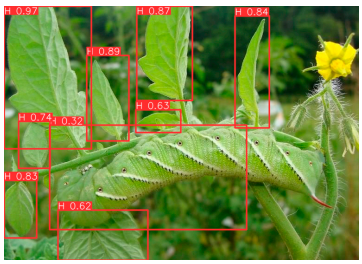
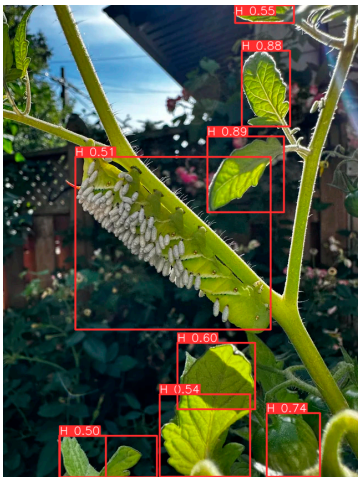
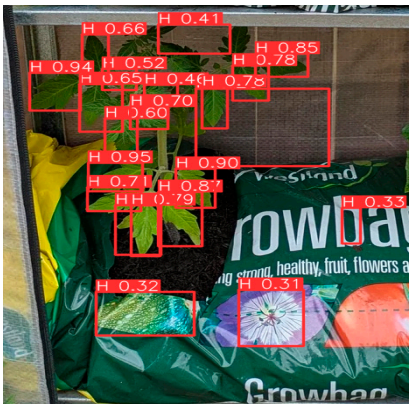
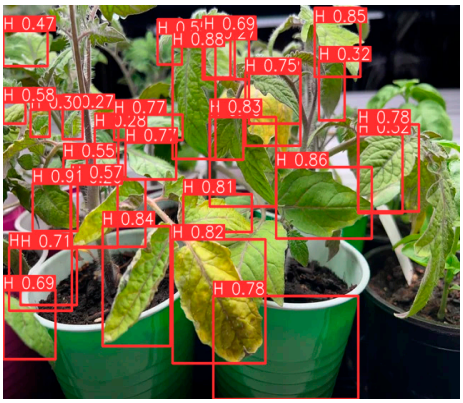
Category	Examples
Different tomato leaf shapes, colors, and textures	 

Table 1. *Cont.*

Category	Examples	
Green tomatoes		
Densely overlapping		
Insects		
Green artificial objects		

This paper focuses on detecting tomato plant leaves using RGB images as a general approach, which requires locating each complete leaf with bounding boxes without emphasizing plant pathological diseases or physiological disorders (such as environmental, nutritional, or physical factors). The main contributions of this paper are as follows:

- An object detection model based on Yolov11n [18] is presented, incorporating attention and transformer modules to detect leaves of tomato (*Solanum* sp.) plants in RGB images. We generated a model with fewer trainable parameters and FLOPs (floating-point operations per second), improving its performance compared with the original Yolov11n and Yolov12n [19].
- A dataset of images of tomato plants obtained from various sources, with their respective ground truth (labels for leaf detection).
- A qualitative and quantitative evaluation of leaf images collected from the Internet featuring various environments, changes in illumination, camera types and poses, and conditions of the leaves (including spider webs, insects, diseases, and partial occlusions).

The paper consists of the following sections: Section 2 reviews related work, Section 3 describes the proposed architecture, Section 4 presents an experimental evaluation, Section 5 draws a discussion, and Section 6 presents conclusions.

2. Related Work

Object detection in plant images using deep learning has been employed for various purposes. For example, Miao et al. [20] detected the tips of leaves to determine the numbers of leaves on maize and sorghum plants and to classify them as damaged or undamaged. Hayati et al. [21] detected tip-burn stress in lettuce grown indoors, enabling the administration of quick treatment. Cho et al. [22] used a ground robot to detect flowers and branching points by analyzing RGB and depth images to obtain plant growth indicators. Shu et al. [23] detected polygonal regions of corn leaf areas infected by pests, with a special emphasis on tiny areas. Hirahara et al. [24] estimated the distances between plant shoots in grape crops in images captured both during the day and at night. All these studies demonstrate the importance of object detection in plant development analysis, including the identification of patterns associated with growth and the early detection of diseases. In contrast, our state-of-the-art study focuses solely on detecting whole leaves.

Researchers generally categorize deep neural networks for object detection into two-stage and one-stage models. Two-stage models divide the detection task into region proposal and classification, resulting in higher accuracy at the cost of double processing. In contrast, one-stage models perform these processes in a single step, allowing their use in real time in resource-constrained device applications, albeit at the expense of accuracy. Excellent reviews of state-of-the-art object detection for agriculture in general can be found in the literature [25–27]. Both types of models have been used to detect leaves in images. The most relevant related papers are outlined below, organized according to the previously defined categories.

2.1. Two-Stage Models

In the two-stage models, Zhang et al. [28] used a Faster R-CNN to detect healthy and diseased tomato leaves. They performed k-means clustering to determine the anchor sizes and tested different residual networks as the backbone. The authors took the images under laboratory conditions, with only one leaf per image. Wang et al. [29] also used a modified Faster R-CNN model with a convolutional block attention mechanism and a region-proposal network module to detect densely overlapping sweet potato leaves. The attention mechanism enabled the extraction of highly relevant features from leaves by

merging spatial and cross-channel information. In contrast, the region-proposal mechanism reduced false detections, especially in cases with densely distributed leaves. Despite improvements made to the original network, the model continued to produce many false positives due to the presence of leaves with severe occlusions, and because the young leaves of the plant of interest were very similar in size and appearance to weeds.

2.2. One-Stage Models

Many studies focus on designing models for leaf detection that consider feature extraction at multiple resolutions, since plants can be at different stages of growth. Lu et al. [30] used the CenterNet network structure and adapted CBAM attention modules and ASPP modules to detect rosette-shaped plant leaves grown in greenhouses. Although this model can deal with plants of different ages and leaves of various sizes, the plant images lack complex backgrounds, varied camera angles, and variable lighting conditions. Xu et al. [31] proposed a vision system for estimating maize yield based on counting plant leaf growth in the field using drone images. They first segmented the background plant using a Mask R-CNN, and then the segmented image was passed to a one-stage model to detect and count the leaves. They categorized the leaves into two classes: fully unfolded leaves and newly emerged leaves, showing poor performance on the newly emerged leaves because they were small regions of the image. The authors excluded cases of folded corn leaves from their analysis and used only nadir imagery. Using a specific model for segmentation and another for localization allowed the efficient detection of maize leaves. However, this strategy requires more computational resources to be put into production.

The idea of integrating neural networks from the Yolo family with transformers has gained considerable interest. Li et al. [32] detected sweet potato leaves in field conditions by enhancing the Yolov5 neural network with the addition of two types of transformer blocks, namely the Vision Transformer and Swin Transformer, which enabled combining local and global features to improve detection accuracy. The authors considered different heights at which the plants were captured, but they always maintained a zenith angle of the camera. He et al. [33] detected maize plant leaves to compute the leaf azimuth angle as a means of improving crop yield through aerial top-view images. The authors also used the Yolov5 model as a basis, to which they added a Swin Transformer, successfully detecting maize leaves with oriented bounding boxes under field conditions. We selected Yolov11n as our base model because it has demonstrated better performance with fewer parameters than Yolov5. In our case, we used horizontal bounding boxes because tomato leaves do not have an elongated shape.

Single-stage models are the ideal choice for developing applications that require real-time operation. Ning et al. [34] improved the Yolov8 model to detect and count corn leaves. The authors replaced the backbone of Yolov8 with a StarNet network and a CAFM fusion module to combine local information obtained by convolutions and spatial information generated by global attention mechanisms. A StarBlock module was also incorporated into the neck section, and a lightweight shared convolutional detection header was used in the final section of the network. Despite its inference speed and good performance, the model was tested only with plants in indoor environments.

Currently, there is a trend toward using transformer elements to design architectures for object detection in agriculture, such as detecting buds [35], fruit ripeness [36], or entire weed plants [37]. However, we did not find any articles that described using this type of neural network specifically for leaf detection. In general, the most representative transformer-based architectures are RT-DETR [38], Yolo-world [39], and Yolov12. The RT-DETR model uses a CNN as a backbone for feature extraction and a transformer-based encoder-decoder to capture global context for real-time object detection. The smallest

version of RT-DETR (R18) has 20 M trainable parameters and a count of 60 GFLOPS for inference. Meanwhile, Yolov11n has only 2.6 M parameters and runs at 6.3 GFLOPS, making this model 10 times lighter than the R18 version. Yolo-world is an open-vocabulary real-time object detector that allows textual descriptions to be merged with the visual features extracted by Yolov8. It can also specify new classes using text without retraining the model. However, Yolo-world's performance depends heavily on the quality of the prompts. Yolov12 is the latest version of the Yolo's family, which primarily incorporates an attention mechanism to speed up the detection process, even more so than RT-DETR models.

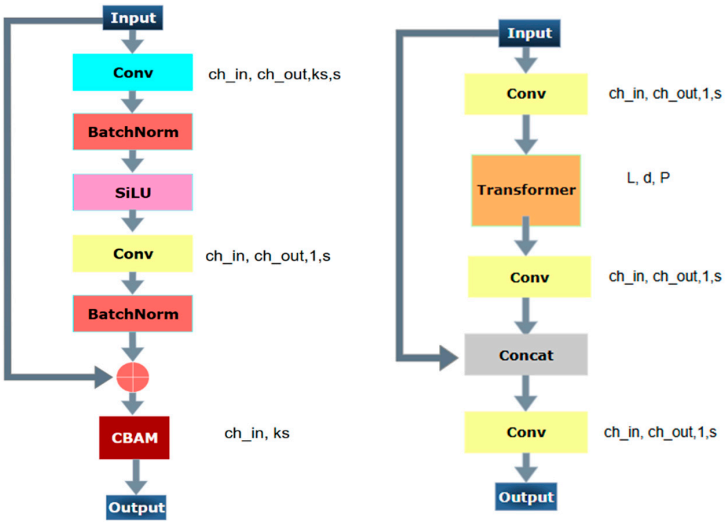
3. The Proposed Architecture

This section presents the modifications made to the Yolov11n [12] network to reduce the model size and address issues in leaf detection using plant images captured in uncontrolled natural environments. We decided to use Yolov11n as our base model because it is an optimized version of version 8 that replaces the C2f modules with C3k2 modules, which improve feature refinement at a lower computational cost, making it suitable for implementation on devices with limited hardware resources.

3.1. Deep Learning Architecture for Detecting Tomato Plant Leaves

We adopted two MobileViT [40] architecture blocks: MobileNetv2 (MV2) and MobileViT. The original MV2 block extracts features with lower computational costs due to the use of depthwise separable convolution and inverted residuals. In addition, similar to previous research [23], we integrated a convolutional block attention module (CBAM) [41] into this block; this improves the intermediate feature map through spatial and channel attention submodules, thereby enhancing the inference of the convolutional neural network. CBAM is a special attention module for feed-forward convolutional neural networks that are easily integrable, lightweight, and end-to-end trainable. All other MV2 modules were preserved. In the case of tomato leaf detection in complex environments, the channel attention module analyzes each input channel, representing the extracted features (e.g., color, texture, shape). It prioritizes the most relevant channels to discriminate the leaf regions from the background and among themselves. The spatial attention module enables the detection of leaves regardless of their distribution or apparent size in the image.

The original MobileViT block integrates convolutional operations with a transformer block containing self-attention layers and feed-forward modules to calculate local spatial relationships, learn global spatial correlations, and generate more complex features. We removed the first convolutional layer from this block, which used 3×3 filters, leaving two convolutional layers of 1×1 filters before and after the transformer. This modification reduced the number of training parameters by eliminating the need to calculate local spatial relationships from the start. We observed that, although local dependencies were not initially modeled, the performance of the architecture was not compromised; however, the computational cost was significantly reduced. The 1×1 convolutional layers were maintained to reduce the computational load on the transformer and subsequent modules. The resulting feature map is concatenated with the input map and passed to a 3×3 convolutional layer to finally consider local spatial relationships. Figure 1 shows the modified versions of the MV2 and MobileViT blocks used in our proposal. The words in the figure represent the following abbreviations: *ch_in*: input channel, *ch_out*: output channel, *s*: stride, *L*: number of stacking transformer blocks, *d*: latent vector size, *P*: resolution of each image patch, and *ks*: kernel size.



(**a**) MV2 (modified) (**b**) MobileViT (modified)

Figure 1. Blocks proposed for inclusion in the Yolov11n network: (**a**) MV2 (modified); (**b**) MobileViT (modified).

We placed the two modified blocks in the last layers of the Yolov11n backbone, the first part of the neural network, contributing to refining the spatial features of the maps (MV2) and merging local and global characteristics (MobileViT) to improve the performance of the original model in the object detection task. Figure 2 illustrates the modifications made to the backbone of the Yolov11n network, where the input feature maps' sizes of each layer are also specified.

(**a**) Original backbone (**b**) Modified backbone

Figure 2. Changes made to the backbone structure of Yolov11n: (**a**) original backbone; (**b**) modified backbone. The arrows pointing to the right make connections to the neck part of the network, which is not shown because we have not modified the neck or head sections.

Table 2 presents the main differences between the original and modified modules in the backbone section of the Yolov11n network. The changes made consisted of replacing a Conv layer with an MV2 block, a C3K2 layer with a MobileViT block, and a C2PSA layer with another MobileViT block. As a result, the number of trainable parameters has been reduced by 65.34%, and the number of MFLOPs is decreased by 63.56%.

Table 2. Main changes made to the Yolov11n network.

Index and Location	Original Yolov11n			Modified Yolov11n		
	Module	**Nb. of Parameters**	**FLOPs(M)**	**New Module**	**Nb. of Parameters**	**FLOPs(M)**
7 (Backbone)	Conv	295,424	49.92	MV2	34,658	3.10
8 (Backbone)	C3k2	346,112	58.49	MobileViT	83,648	17.85
10 (Backbone)	C2PSA	249,728	47.68	MobileViT	190,656	35.94
Total		891,264	156.09		308,962	56.89

3.2. Replacement of the Loss Function

The original Yolov11n version uses distribution focal loss (DFL) and complete intersection over union (CIoU) loss functions during training, which evaluate the precision of detecting the object of interest in the image. DFL allows handling the coordinates of the bounding boxes as discrete distributions. DFL calculates the discrepancies between the predicted and target distributions, significantly improving inference coordinates. On the other hand, CIoU uses the distance between the centroids, the overlap, and the aspect ratio of the bounding boxes to mitigate issues associated with the detection task.

The images from our study primarily consider complex scenarios where multiple bounding boxes overlap due to dense clusters of leaves, which negatively impact the training and performance of the model in locating targets. Figure 3 shows an image containing a plant with numerous leaves, resulting in significant overlap between the different bounding boxes being adjusted to fit each leaf. The areas with a more saturated yellow color indicate instances that are more difficult to detect due to the presence of other bounding boxes in the same image area. For this reason, we replaced the CIoU function with WIoUv3 [42], an improved version of wise intersection over union (WIoU) [43]. The WIoU loss dynamically adjusts the weights of the different components (distance, overlap, and aspect ratio), thereby better adapting to the characteristics of the dataset and dealing with the challenges of both large and small objects as well as low-quality labeled instances. Unlike WIoUv1, the WIoUv3 metric incorporates a non-monotonic focal coefficient β. The WIoUv3 loss focuses on quantifying the quality of anchor boxes, helping the model to focus more on anchor boxes of ordinary quality. The WIoUv3 function is defined in Equations (1)–(3). The WIoUv1 function is explicitly stated in Equations (4) through (6). Figure 4 illustrates the variables required to calculate Equation (5).

$$L_{WIoUv3} = r \times L_{WIoUv1} \tag{1}$$

$$r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{2}$$

$$\beta = \frac{L^{*}_{IoU}}{\overline{L_{IoU}}} \in [0, +\infty) \tag{3}$$

$$L_{WIoUv1} = R_{WIoU} \times L_{IoU} \tag{4}$$

$$R_{WIoU} = \exp\left(\frac{\left(b^{gt}_{c_x} - b_{c_x}\right)^2 + \left(b^{gt}_{c_y} - b_{c_y}\right)^2}{\left((c_w)^2 + (c_h)^2\right)}\right) \tag{5}$$

$$L_{IoU} = 1 - IoU \tag{6}$$

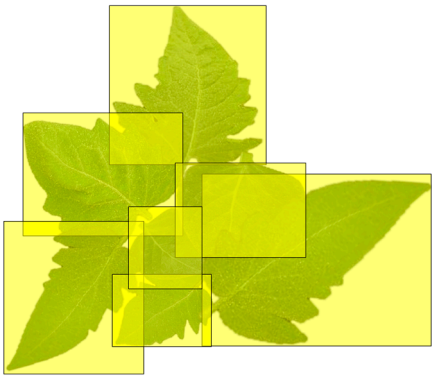


Figure 3. Overlapping bounding boxes occur when there are groups of leaves in the same image. The bounding boxes with intense yellow tone areas are the most difficult to detect as they include parts of other leaves.

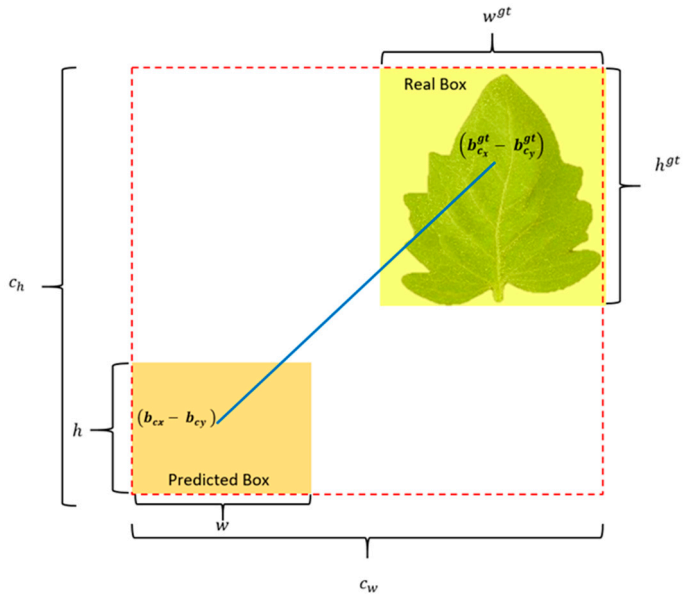


Figure 4. Main variables used in the computation of WIoUv1: the smallest enclosing box (red dotted box), and the distance between the ground truth and predicted bounding boxes' centroids (blue line).

4. Experimental Evaluation

This section presents the analyzed dataset, an ablation study to demonstrate the performance of the proposed model, and quantitative and qualitative analysis. We also present cases where the models still performed poorly.

4.1. Dataset Creation

The PlantVillage dataset is one of the most popular datasets used in precision agriculture for leaf classification and detection, due to its extensive collection of images of different plant species, labeling, and accessibility. However, all its leaf images were captured indoors under controlled lighting and background conditions.

We created the proposed dataset from various sources hosted on the Roboflow platform and from an internet image search. Only a small sample from specific repositories was considered, because many images were duplicated or did not meet the criteria of our study. All image labels were reviewed or created manually. Table 3 details the distribution and characteristics of the images. Our complete dataset included 3,000 images and 27,397 instances of leaves. Our focus was on tomato (*Solanum* sp.) plants, where the images

had the following features: diverse backgrounds, variable lighting, occlusions, different camera types used during collection, and a variable distance between the leaves and the camera's lens. The leaf blades show diversity in color, texture, and shape. Figure 5 presents a small set of images within the proposed dataset.

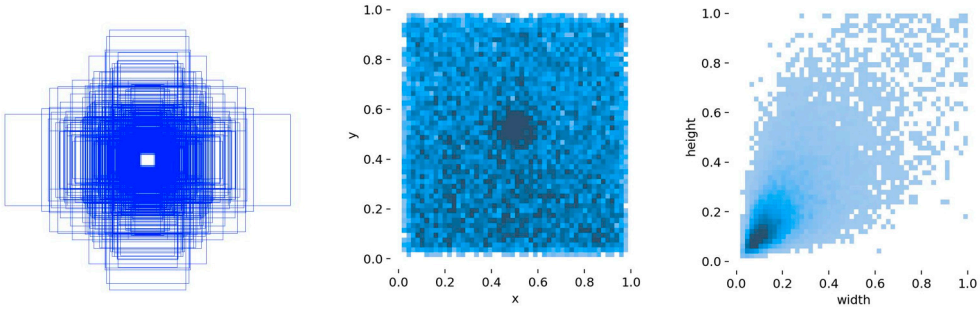
Table 3. Sources of images used to create our custom dataset.

	N.b. of Images	Image Sizes	Task Type	Url
A4	906	416 × 416	detection	http://bit.ly/4209xML (accessed on 6 July)
ACVP	20/27	3024 × 4032	detection	https://bit.ly/4ikDUUU (accessed on 6 July)
T7	44/542	416 × 416	detection	https://bit.ly/4bNEsAe (accessed on 6 July)
TT	48/94	416 × 416	detection	https://bit.ly/4bD39zf (accessed on 6 July)
DC	50/133	2304 × 3456	classification	https://bit.ly/4ij4UnR (accessed on 6 July)
NCVP	64/318	640 × 640	detection	http://bit.ly/4bD3ulv (accessed on 6 July)
TCVP	105/1482	640 × 640	detection	https://bit.ly/4iAJ6nl (accessed on 6 July)
TDL940	51/2255	640 × 640	detection	No longer available
PCVP	100/380	640 × 640	detection	https://bit.ly/4iFOcig (accessed on 6 July)
TLD	50/4132	640 × 640	detection	https://bit.ly/4kALfRX (accessed on 6 July)
Internet search	1562	416 × 416	-	-
Total	3000	variable	detection	

Figure 5. Examples of images that constitute the proposed dataset. Each column presents five images from the respective dataset. The final column displays our dataset, which merges the other datasets.

To generate the ground truth for the dataset, we used the application CVAT (Computer Vision Annotation Tool) available at [44] (UI version: 2.40.0). This process was carried out carefully by hand so that each bounding box fitted perfectly to the boundaries of each leaf. Multiple people validated all bounding boxes to minimize errors as much as possible. We labelled all tomato leaves contained in each image, regardless of their size and distribution in the image. In the special case of occluded leaves, we used the criterion that all visible segments (regardless of their area size) of the leaf must be labeled. Figure 6 shows the spatial and statistical characteristics of the 27,397 ground truth bounding boxes in our dataset. Figure 6a illustrates the distribution of bounding boxes in a normalized space between 0 and 1, indicating many objects of interest per image with significant overlap and variability in size. Figure 6b shows that most bounding boxes are centered in the images,

with fewer objects located at the edges. Figure 6c indicates that most bounding boxes are small (lower left corner) due to the presence of young leaves, leaves in the background, and mainly overlapping leaves.



(**a**) (**b**) (**c**)

Figure 6. Characteristics of the ground truth bounding boxes in our dataset. The figures show the following distributions of the bounding boxes in a normalized space: (**a**) position and size, (**b**) centroids, and (**c**) size (width vs. height).

4.2. Ablations and Configurations Study

We ran the experiments using Google Colab Pro, which uses a virtual environment equipped with an Intel(R) Xeon(R) CPU @ 2.20 GHz, 15 GB NVIDIA T4 GPU, 12.7GB of RAM, and 78 GB of hard disk space. We evaluated the performance of the proposed network using ablations. The ablations involved modifying the Yolov11n original architecture gradually to demonstrate the actual contribution of each of the proposed blocks. Therefore, the following three ablations are proposed: Ablation 1: Conv $\rightarrow$ MV2, Ablation 2: C3k2 $\rightarrow$ MobileViT, and Ablation 3: C2PSA $\rightarrow$ MobileViT, which represent the different proposed replacements. Table 4 details the configurations of the proposed ablations and configurations to examine which performs best. The *c1* configuration represents the Yolov11n original model.

Table 4. Experiments using Yolov11n as a base.

Nb. of Configuration	CIoU	WIoUv3	Ablation 1	Ablation 2	Ablation 3
1	√	X	X	X	X
2	√	X	√	X	X
3	√	X	√	√	X
4	√	X	√	√	√
5	X	√	X	X	X
6	X	√	√	X	X
7	X	√	√	√	X
8	X	√	√	√	√

Table 5 presents the primary hyperparameters and data augmentation techniques used to train the different models. The values for each variable were obtained through experimentation. We used the same values for all experiments presented in this paper and maintained the default values for the remaining.

Table 6 shows the performance of Yolov11n and its different modifications. We also compared the performance of the proposed models with the recently published Yolov12n model. We used the same training set, validation set, and hyperparameters during training for this experiment. The values in bold indicate the best value obtained in each evaluated network performance variable. As the defined ablations were incorporated, the number of parameters decreased without compromising the model's performance. We selected the model generated from configuration *c8* (3 ablations + WIoUv3) since the results of the recall, F1 measure, and mAP@50 metrics were comparable with those obtained by the Yolov11n

base model (configuration *c1*), but with a 26.75% reduction in the number of parameters and a 7.94% reduction in the number of FLOPs. The main innovation of Yolov12 is the incorporation of a module named attention-centric, based on ViTs [13].

Table 5. Main parameters used in the training process.

Used In	Parameter	Value
Model training	Batch size	47
	Epochs	500
	Image size	(416, 416)
	Optimizer	SGD
	Lr scheduler	Cosine annealing
	Initial and final learning rate	(0.01, 0.0001)
Data augmentation	hsv_h	0.0
	hsv_s	0.0
	hsv_v	0.0
	degrees	0.0
	translate	0.12431
	scale	0.07643
	shear	0.0
	perspective	0.0
	flipud	0.63
	fliplr	0.63
	mosaic	0.63
	mixup	0.0
	Close mosaic	50

Table 6. Comparison of the performance of the Yolov11n base model versus its proposed modifications and Yolov12n.

Nb. of Configuration		Nb. of Parameters	FLOPs (G)	%				
				Precision	Recall	F1-Measure	mAP@50	mAP@50-95
Yolov11n	*c1*	2,582,347	6.3	84.60	79.70	82.28	88.40	65.20
	c2	2,289,069	6.1	83.80	79.40	81.53	88.10	65.20
	c3	1,959,693	5.8	84.20	80.10	82.09	88.30	65.30
	c4	1,891,501	5.8	85.00	79.90	82.37	88.50	**65.60**
	c5	2,582,347	6.3	82.60	79.10	80.81	87.60	63.80
	c6	2,289,069	6.1	84.80	78.80	81.68	88.00	64.80
	c7	1,959,693	5.8	**85.50**	78.80	82.00	88.00	64.80
	c8	**1,891,501**	**5.8**	**83.80**	**81.70**	**82.73**	**88.50**	**65.40**

Our model *c8* was selected as the final model since it reduces the number of parameters by 36.52% and the number of GFLOPs by 8.62% compared with Yolov11n.

4.3. Quantitative Analysis of Our Proposal

We evaluated our proposed model using *k*-fold cross-validation. We chose this technique due to the size of the dataset and the amount of hardware resources required for training the model. The value of *k* was set to 5 to maintain an 80–20 partition for training and testing, respectively, in all cases. Table 7 illustrates how the partitions were created, with a different test set for each case. The entire dataset was divided into five subsets, generating five splits; a subset was selected for testing and the rest for training. Each fold had a different test subset.

Table 7. Dataset separation using *k*-fold cross-validation. The green squares represent the test sets, and the orange squares indicate the training sets.

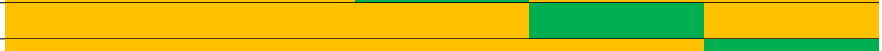
Fold	All Dataset
f1	
f2	
f3	
f4	
f5	

Table 8 displays the performance of the proposed model (*c8*) trained and tested with different folds. The mean values of recall, F1-measure, mAP@50, and mAP@50–95 metrics show relatively low standard deviations, with values less than one in all cases. This behavior indicates low dispersion in the evaluation results, thereby reinforcing the reliability of the model across different test subsets. Although the precision metric maintains a high average value, its standard deviation is slightly above one, which could be related to the model's greater sensitivity to specific variations in the test data.

Table 8. Performance of the proposed model when trained and tested on different image sets.

Fold	%				
	Precision	**Recall**	**F1-Measure**	**mAP@50**	**mAP@50-95**
f1	81.70	79.30	80.48	87.20	63.70
f2	84.70	78.80	81.64	87.30	64.60
f3	84.30	78.50	81.29	87.30	64.30
f4	84.90	79.40	82.05	88.60	65.40
f5	83.90	79.60	81.69	87.40	64.30
mean ± std	83.90 ± 1.15	79.12 ± 0.40	81.43 ± 0.5	87.56 ± 0.50	64.46 ± 0.54

Table 9 presents the performance of our proposal compared to recent versions of Yolo. Our model has the lowest number of parameters and GFLOPs, making it more suitable for devices with limited hardware resources. Compared with Yolov12n, an architecture that also has attention modules, our model reduces the parameters by 35.17% and GFLOPs by 8.62%. Additionally, the performance metrics obtained with our proposal are comparable to those of versions of Yolov8n, Yolov11n, and Yolov12n. Our model showed a slight reduction of 1.40% in precision compared to Yolov12n, suggesting an increase in the detection of other objects that are not of interest being classified as leaves (false positives). However, Yolov12n generated more false negatives than our proposal (low recall), leaving several leaves undetected.

Table 9. Comparison of the performance of our proposal with recent versions of Yolo, such as Yolov8n, Yolov10n, Yolov11n, and Yolov12n.

Model	Nb. of Parameters	FLOPs (G)	%				
			Precision	**Recall**	**F1-Measure**	**mAP@50**	**mAP@50-95**
Yolov12n	2,556,923	6.30	85.20	78.30	81.60	87.60	64.70
Yolov11n	2,582,347	6.30	84.60	79.70	82.28	88.40	65.20
Yolov10n	2,265,363	6.50	83.20	79.30	81.20	87.10	64.20
Yolov8n	3,005,843	8.10	83.40	80.50	81.92	87.70	64.80
Our proposal	1,891,501	5.80	83.80	81.70	82.73	88.50	65.40

4.4. Qualitative Analysis

The main objective of this analysis is to evaluate the inference quality of our proposal, the Yolov11n and Yolov12n models, across five case studies for detecting tomato plant leaves in different uncontrolled natural environments.

Table 10 displays the detections made in the images by the different models, considering different shapes, colors, and textures of the leaves. The tomato plant in the first row has diseased leaves, showing varying degrees of overlap and discoloration. In the background, the leaf blades are partially visible due to the camera's field of view. Yolov12n and Yolov11n have issues detecting the central leaves correctly due to overlap and leaf shape. The proposed model performs better as it detects the region of each leaf more accurately. A leaf that shows both its upper and lower surfaces because it is slightly folded is typically detected by all models in two parts. In the second row, the plant leaves are curling, which significantly alters the shape of the leaf. This condition may result from various factors such as water stress, extreme temperatures, nutrient deficiency, pests or diseases, agrochemical damage, or it could even be a natural feature of the leaf [45]. Though the leaves are visible, Yolov12n and Yolov11n struggle to detect the leaves (overlapping and inaccurate bounding boxes). The proposed model do not have the superposition problem. Nevertheless, a blade leaf was not detected.

Table 10. Evaluation of the model's performance on images with different tomato leaf shapes, colors, and textures.

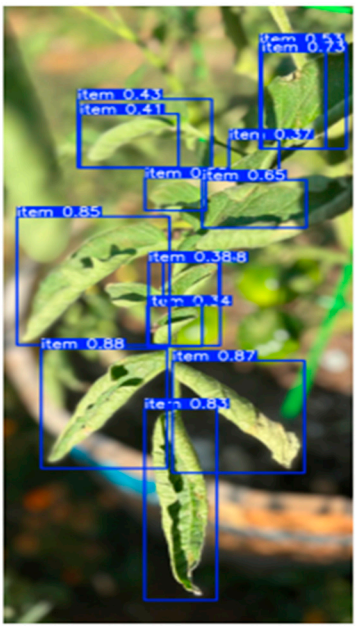
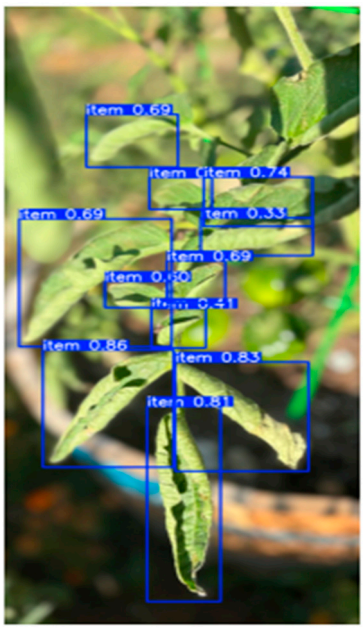
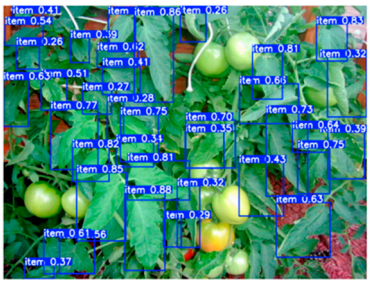
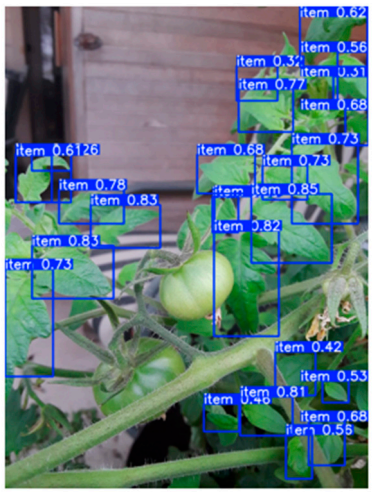
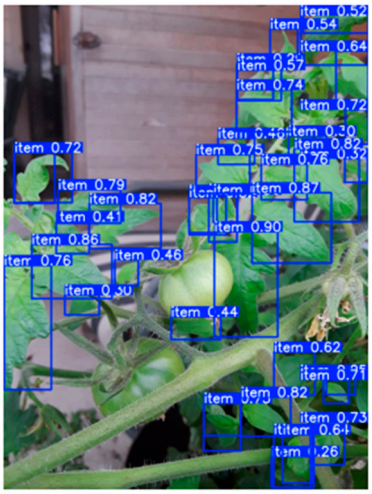
Models		
Yolov12n	**Yolov11n**	**Our model**
		
		

Table 11 shows the results of detections in images of plants with many overlapping leaves and the presence of green tomatoes. Row one illustrates a scene that is too complex for the models to detect each leaf and exclude the green tomatoes properly [46]. In this case, the image was captured at a greater distance from the plant, and the tomatoes also caused partial occlusions of the leaves. Our model was more precise, correctly detecting more leaves than the other two.

Table 11. Evaluation of the models' performance on images with green tomatoes and dense leaf overlap.

Models		
Yolov12n	**Yolov11n**	**Our Model**
		
		

Other elements, like worms, may also be found in plants. Depending on the lighting conditions and the position of the insect, it can be very similar in color and shape to the leaves. In the first row of Table 12, Yolov12n detects a worm as a leaf, while Yolov11n and our proposal do not [15]. In the second case, both Yolov12n and Yolov11n detect part of the worm as a leaf [47]. In both cases, our method correctly excluded the regions of the images occupied by worms, but it presented a false positive when detecting a tiny green tomato in the first case.

Table 13 shows the results obtained in the presence of green artificial objects. In the first row, the inference of Yolov12n is inaccurate due to the detection of green sacks as leaves in certain regions [48]. Moreover, Yolov12n and Yolov11n detect a different category of leaf, which result in false positive cases. In contrast, Yolo11n and the proposed model successfully discard these objects. In the second case, both Yolov11n and Yolov12n fail to consider part of the pot as a leaf [49]. The attention modules used in our proposal enable the highlighting of more relevant features learned in the convolutional layers.

Table 12. Evaluation of the models' performance on images with insects.

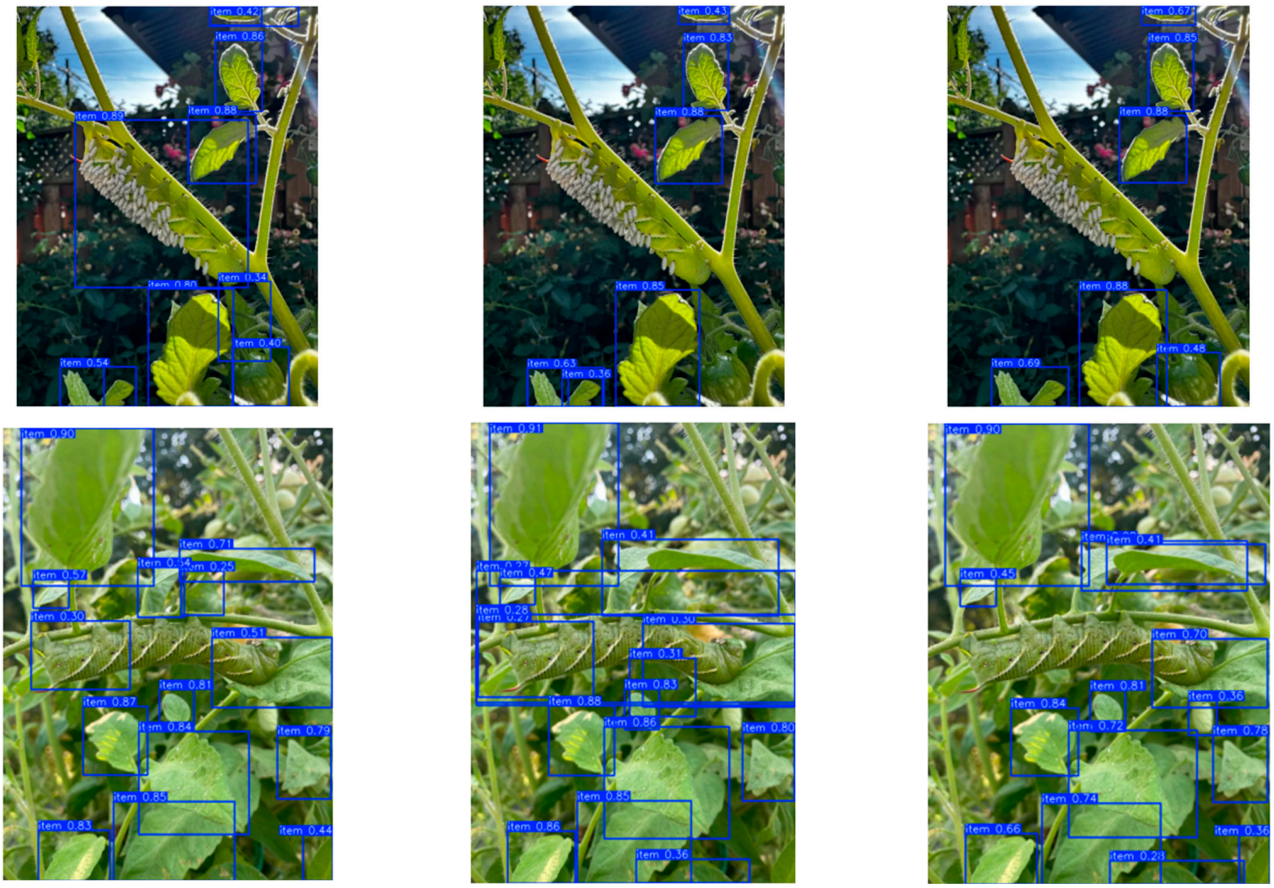
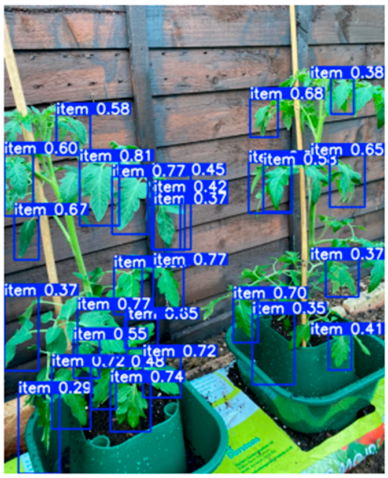
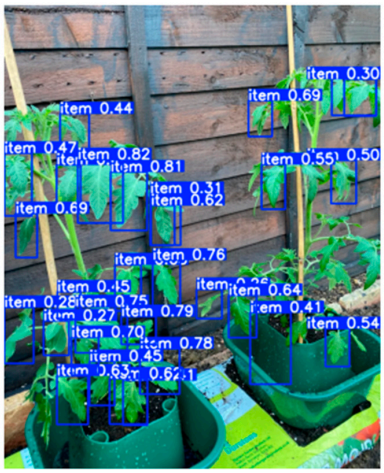
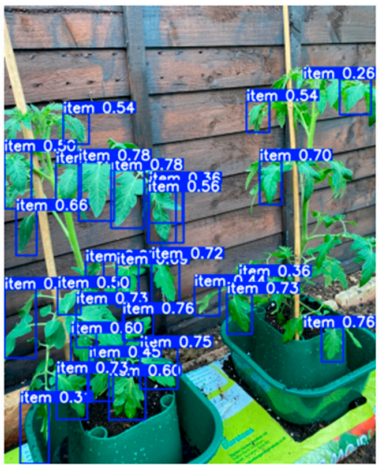
Models		
Yolov12n	**Yolov11n**	**Our Model**
		

Table 13. Evaluation of the model's performance on images with green artificial objects.

Models		
Yolov12n	**Yolov11n**	**Our Model**
		

Table 13. *Cont.*

Models		
Yolov12n	**Yolov11n**	**Our Model**
		

4.5. Limitations of the Models

Although our model generated good qualitative results compared with other versions of Yolo, we identified some limitations in the following cases, as shown in Table 14. *Distant views:* All models failed when images were captured from a long distance. Yolov12n and Yolov11n generated many bounding boxes, but most were false positives, detecting green tomatoes and groups of leaves. *Mixed leaf species:* The three models failed to differentiate between tomato leaves and those from other plant species. Row 3 shows an image with a weed plant in the foreground and tomato leaves in the background. *Artificial leaf prints*: In the third row, the image shows a plastic bag with some printed leaves and arrows (recycling symbol). Both Yolov12n and Yolov11n detect the arrows, but the proposed model does not detect other different shapes. In conclusion, the three models demonstrate inaccuracies and the presence of false positive cases, as the targets do not correspond to natural leaves.

Table 14. Performance of the models analyzed in problematic cases.

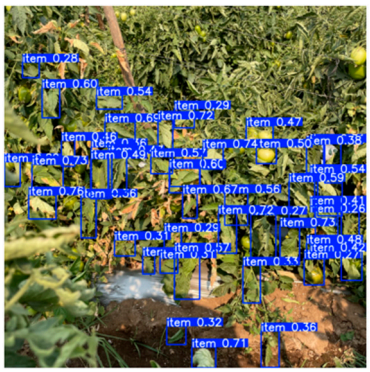
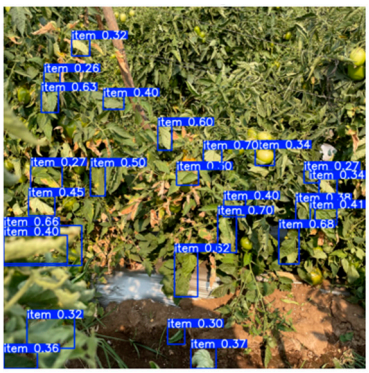
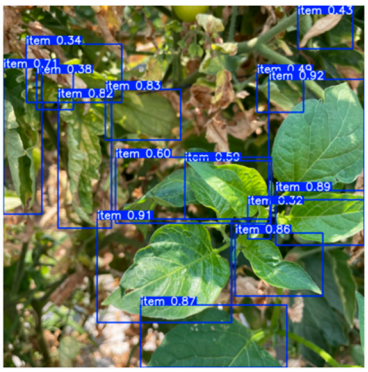
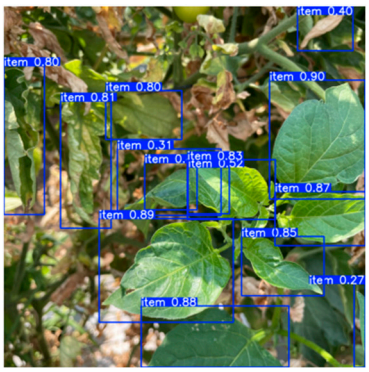
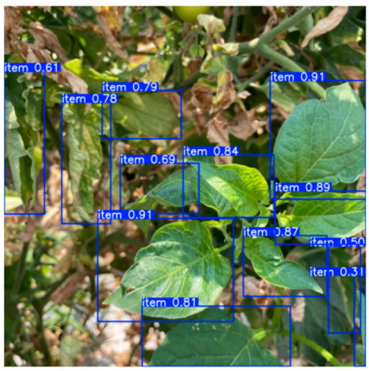
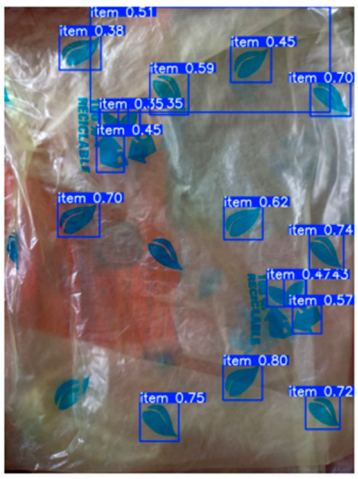
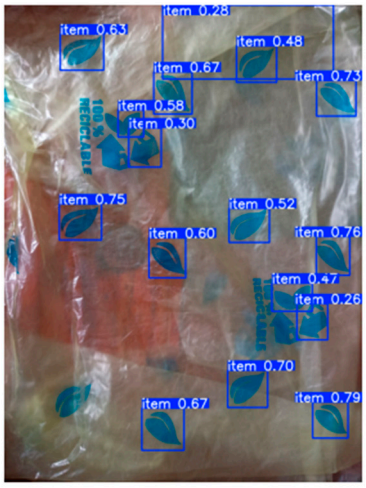
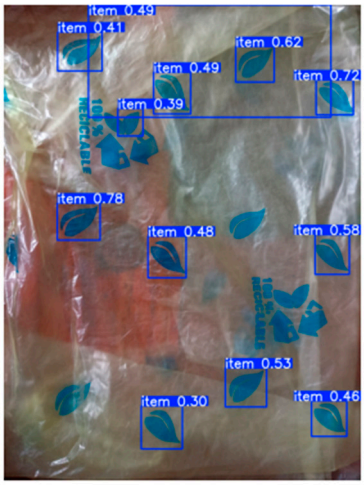
Models		
Yolov12n	**Yolov11n**	**Our Model**
		

Table 14. *Cont.*

Models		
Yolov12n	**Yolov11n**	**Our Model**
		
		

5. Discussion

The study of the state of the art revealed a trend toward improving or proposing new one-stage models over two-stage models, due to their ease of deployment on devices with limited hardware resources, as evidenced by a greater number of publications on this topic. We used Yolov11n as the base architecture and replaced different elements to compress it, both in terms of the number of parameters and computational cost (FLOPs). This modified version extracts better features by fusing local and global spatial relationships.

The incorporation of CBAM and transformer modules into the Yolov11n architecture enables a better combination of local and global dependencies, which is of great importance for addressing the problem at hand. CBAM helps to give greater relevance to the areas of the image belonging to the leaves. The integration of modified MobileViT attention modules into the original backbone enabled the refinement of feature maps. The MV2 block helped reduce the model size, the storage required, and the number of FLOPs. Meanwhile, replacing the original function (CIoU) with WIoUv3 allowed us to modify the way the target location is estimated based on a strategic selection of anchor box quality, which improves performance in detecting leaves that are heavily overlapping.

The ablation study helped identify the strengths and weaknesses of each of the elements analyzed. For example, the *c5* configuration in Table 6 indicates that the WIoUv3 function alone is insufficient to improve model performance. We achieved the best performance by using the three selected modules to extract better features (CBAM, MobileViT, and MV2), alongside WIoUv3. With an effective feature description, WIoUv3 enables the

efficient discrimination of highly atypical or low-quality anchor boxes, and consequently, better estimation of leaf locations.

As noted in the qualitative analysis, accurately detecting tomato plant leaves in images, considering complex environments, remains a challenging problem because the physical characteristics of leaves, such as size, color, texture, and shape, vary significantly depending on the environment. Furthermore, a cluster of leaf blades can give rise to countless ambiguities due to partial occlusions and shadows that alter the visual appearance of the leaf.

For images containing leaves with diverse shapes, colors, and textures, Yolov12n has trouble correctly locating objects of interest, generating redundant bounding boxes on the same leaf, albeit with a much lower confidence level. This problem occurs to a lesser extent in the proposed model or Yolov11n.

In the specific case of dense leaf overlap, both Yolov11n and Yolov12n presented problems in correctly detecting the occluded leaves, even though images were captured at a short distance from the plant. These models generated bounding boxes that included more than two objects, while the proposed model better resolved these cases, achieving a 60.50% mAP@50-95. All models failed when there were leaves with folds near the apex, decreasing the precision metric. The models perform better when the entire leaf blade region is visible in the image.

The performance of the proposed model is comparable to that of the Yolov12n, Yolov11n, Yolov10n, and Yolov8n models, but requires much less storage space and fewer floating-point operations to make inferences. This feature allows our model to be more easily implemented on devices with limited resources, such as cell phones or robots that can be easily transported to agricultural environments. Our model showed 1.2% and 3.4% increases in recall compared with Yolov8n and Yolov12n, respectively. The presence of fewer false negatives when using our model reduces the probability of ignoring leaves that are present in the images.

Several issues still need to be addressed to achieve better leaf detection, such as considering distant shots, differentiating leaves of interest from those of other plant species, and discarding non-natural plant representations. Likewise, our proposed model and the others analyzed still presented high numbers of false positives due to all the issues described above. We plan to modify the input of our architecture in the future to integrate additional information, such as depth maps and synthetic or real near-infrared (NIR) imaging, similar to [16,50]. The addition of a fourth channel to the RGB images, incorporating information from the NIR sensor, would provide a better representation of feature maps, as all artificial objects that reflect less radiation would be immediately excluded. Similarly, data from the depth sensor would help the model learn to differentiate leaves on different planes. We could even explore combining NIR and depth information to determine whether the problem can be better modeled.

6. Conclusions

Detecting tomato leaves in uncontrolled environments is a challenging task because plants are naturally subject to a variety of biotic and abiotic factors, resulting in leaves of different sizes, shapes, colors, and textures. Additionally, the leaves of plants are arranged in various ways to minimize the shade produced between them, thereby optimizing photosynthesis. This phenomenon is known as phyllotaxy, which causes partial or total occlusion of the leaves, further increasing the difficulty of correctly detecting them.

This paper presents a deep learning architecture based on Yolov11n for tomato leaf detection in images in uncontrolled environments. We propose incorporating an attention module into the original Yolov11n architecture. The proposed architecture was evaluated

both qualitatively and quantitatively by creating a dataset comprising a diverse range of tomato leaf images captured under various conditions and cameras. The proposed model performed comparably to Yolov11n and Yolov12n but required fewer trainable parameters and performed fewer floating-point operations. The incorporation of attention into the Yolov11n architecture enabled it to focus on the relevant features and regions of the leaves, leading to better results in challenging and varied scenarios. We identified other challenging cases that need to be addressed to achieve better detection of plant leaves, such as distant views, mixed leaf species, and artificial leaf prints.

Author Contributions: The authors of this work contributed as follows: conceptualization, A.M.-H., J.F.-P., A.M.-S., R.P.-E., and N.G.-F.; writing—original draft preparation, A.M.-H. and J.F.-P.; methodology, A.M.-H. and A.M.-S.; writing—review and editing, J.F.-P., A.M.-S., R.P.-E., and N.G.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: We make the data and code used available at https://github.com/andros1206/Leaf-Detection, accessed on 6 July 2025.

Acknowledgments: This research has been made possible thanks to generous support from the TecNM/Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) and Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) of México.

Conflicts of Interest: The authors A.M.-H., J.F.-P., A.M.-S., R.P.-E., and N.G.-F declare no conflicts of interest.

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. Torralba, A.; Isola, P.; Freeman, W.T. *Foundations of Computer Vision*; MIT Press: Cambridge, MA, USA, 2024; ISBN 978-0-262-04897-2.
3. International Plant Genetic Resources Institute. *Descriptors for Tomato (Lycopersicon spp.)*; IPGRI: Rome, Italy, 1996; Available online: https://hdl.handle.net/10568/73041 (accessed on 14 June 2025).
4. SAGARPA (Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación). *Planeación Agrícola Nacional 2017–2030: Jitomate Mexicano*; SAGARPA: Ciudad de México, México, 2017; Available online: https://www.gob.mx/cms/uploads/attachment/file/257077/Potencial-Jitomate.pdf (accessed on 14 June 2025).
5. Chowdhury, M.E.H.; Rahman, T.; Khandakar, A.; Ayari, M.A.; Khan, A.U.; Khan, M.S.; Al-Emadi, N.; Reaz, M.B.I.; Islam, M.T.; Ali, S.H.M. Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques. *AgriEngineering* **2021**, *3*, 294–312. [CrossRef]
6. Shoaib, M.; Hussain, T.; Shah, B.; Ullah, I.; Shah, S.M.; Ali, F.; Park, S.H. Deep Learning-Based Segmentation and Classification of Leaf Images for Detection of Tomato Plant Disease. *Front. Plant Sci.* **2022**, *13*, 1031748. [CrossRef] [PubMed]
7. Ozcan, T.; Polat, E. BorB: A Novel Image Segmentation Technique for Improving Plant Disease Classification with Deep Learning Models. *IEEE Access* **2025**, *13*, 71822–71839. [CrossRef]
8. Ngugi, L.C.; Abdelwahab, M.; Abo-Zahhad, M. Tomato Leaf Segmentation Algorithms for Mobile Phone Applications Using Deep Learning. *Comput. Electron. Agric.* **2020**, *178*, 105788. [CrossRef]
9. Bhagat, S.; Kokare, M.; Haswani, V.; Hambarde, P.; Kamble, R. Eff-UNet++: A Novel Architecture for Plant Leaf Segmentation and Counting. *Ecol. Inform.* **2022**, *68*, 101583. [CrossRef]
10. Guo, R.; Qu, L.; Niu, D.; Li, Z.; Yue, J. LeafMask: Towards Greater Accuracy on Leaf Segmentation. *arXiv* **2021**, arXiv:2108.03568. [CrossRef]
11. Yao, J.; Wang, Y.; Xiang, Y.; Yang, J.; Zhu, Y.; Li, X.; Li, S.; Zhang, J.; Gong, G. Two-Stage Detection Algorithm for Kiwifruit Leaf Diseases Based on Deep Learning. *Plants* **2022**, *11*, 768. [CrossRef] [PubMed]
12. u/UnrealElvis. *Do My Tomatoes Need More Water?* Reddit. 2018. Available online: https://www.reddit.com/r/gardening/comments/8ygjjy/do_my_tomatoes_need_more_water/?tl=es-419 (accessed on 3 July 2025).
13. u/No-Arm-1678. *Forever Green Tomatoes?* Reddit, r/tomatoes. 2023. Available online: https://www.reddit.com/r/tomatoes/comments/158rpcu/forever_green_tomatoes/ (accessed on 3 July 2025).

14. Day, E.R. *Tomato hornworm (Manduca quinquemaculata)*; Insect Images, The University of Georgia—Warnell School of Forestry & Natural Resources, 2013; Image No. 5520230. Available online: https://www.insectimages.org/browse/detail.cfm?imgnum=5520230 (accessed on 3 July 2025).
15. u/15okgep. *What Is This Caterpillar on My Tomato Plant?* Reddit, r/whatisthisbug. 2023 (est.). Available online: https://www.reddit.com/r/whatisthisbug/comments/15okgep/what_is_this_caterpillar_on_my_tomato_plant/ (accessed on 6 July 2025).
16. u/Ninja10. *What Happened to My Cucumber and Tomato?* Reddit, r/vegetablegardening. 2025. Available online: https://www.reddit.com/r/vegetablegardening/comments/1lqw75n/what_happened_to_my_cucumber_and_tomato/ (accessed on 6 July 2025).
17. u/UsernameUnavailable. *Tomatoes with Yellowing Leaves and Black Spots?* Reddit, r/vegetablegardening. 2023 (est.). Available online: https://www.reddit.com/r/vegetablegardening/comments/12h186f/tomatoes_with_yellowing_leaves_and_black_spots/ (accessed on 6 July 2025).
18. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725. [CrossRef]
19. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv* **2025**, arXiv:2502.12524. [CrossRef]
20. Miao, C.; Guo, A.; Thompson, A.M.; Yang, J.; Ge, Y.; Schnable, J.C. Automation of Leaf Counting in Maize and Sorghum Using Deep Learning. *Plant Phenom. J.* **2021**, *4*, e20022. [CrossRef]
21. Hamidon, M.H.; Ahamed, T. Detection of Tip-Burn Stress on Lettuce Grown in an Indoor Environment Using Deep Learning Algorithms. *Sensors* **2022**, *22*, 7251. [CrossRef] [PubMed]
22. Cho, S.; Kim, T.; Jung, D.-H.; Park, S.H.; Na, Y.; Ihn, Y.S.; Kim, K. Plant Growth Information Measurement Based on Object Detection and Image Fusion Using a Smart Farm Robot. *Comput. Electron. Agric.* **2023**, *207*, 107703. [CrossRef]
23. Zhu, R.; Hao, F.; Ma, D. Research on Polygon Pest-Infected Leaf Region Detection Based on YOLOv8. *Agriculture* **2023**, *13*, 2253. [CrossRef]
24. Hirahara, K.; Nakane, C.; Ebisawa, H.; Kuroda, T.; Iwaki, Y.; Utsumi, T.; Nomura, Y.; Koike, M.; Mineno, H. D4: Text-Guided Diffusion Model-Based Domain Adaptive Data Augmentation for Vineyard Shoot Detection. *Comput. Electron. Agric.* **2025**, *230*, 109849. [CrossRef]
25. Badgujar, C.M.; Poulose, A.; Gan, H. Agricultural Object Detection with You Only Look Once (YOLO) Algorithm: A Bibliometric and Systematic Literature Review. *Comput. Electron. Agric.* **2024**, *223*, 109090. [CrossRef]
26. Ariza-Sentís, M.; Vélez, S.; Martínez-Peña, R.; Baja, H.; Valente, J. Object Detection and Tracking in Precision Farming: A Systematic Review. *Comput. Electron. Agric.* **2024**, *219*, 108757. [CrossRef]
27. Huang, Y.; Qian, Y.; Wei, H.; Lu, Y.; Ling, B.; Qin, Y. A Survey of Deep Learning-Based Object Detection Methods in Crop Counting. *Comput. Electron. Agric.* **2023**, *215*, 108425. [CrossRef]
28. Deep Learning-Based Object Detection Improvement for Tomato Disease. Available online: https://ieeexplore.ieee.org/abstract/document/9044330 (accessed on 14 January 2025).
29. Wang, M.; Fu, B.; Fan, J.; Wang, Y.; Zhang, L.; Xia, C. Sweet Potato Leaf Detection in a Natural Scene Based on Faster R-CNN with a Visual Attention Mechanism and DIoU-NMS. *Ecol. Inform.* **2023**, *73*, 101931. [CrossRef]
30. Lu, S.; Song, Z.; Chen, W.; Qian, T.; Zhang, Y.; Chen, M.; Li, G. Counting Dense Leaves under Natural Environments via an Improved Deep-Learning-Based Object Detection Algorithm. *Agriculture* **2021**, *11*, 1003. [CrossRef]
31. Xu, X.; Wang, L.; Shu, M.; Liang, X.; Ghafoor, A.Z.; Liu, Y.; Ma, Y.; Zhu, J. Detection and Counting of Maize Leaves Based on Two-Stage Deep Learning with UAV-Based RGB Image. *Remote Sens.* **2022**, *14*, 5388. [CrossRef]
32. Li, X.; Fan, W.; Wang, Y.; Zhang, L.; Liu, Z.; Xia, C. Detecting Plant Leaves Based on Vision Transformer Enhanced YOLOv5. In Proceedings of the 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 22–24 July 2022; pp. 32–37.
33. He, W.; Gage, J.L.; Rellán-Álvarez, R.; Xiang, L. Swin-Roleaf: A New Method for Characterizing Leaf Azimuth Angle in Large-Scale Maize Plants. *Comput. Electron. Agric.* **2024**, *224*, 109120. [CrossRef]
34. Ning, S.; Tan, F.; Chen, X.; Li, X.; Shi, H.; Qiu, J. Lightweight Corn Leaf Detection and Counting Using Improved YOLOv8. *Sensors* **2024**, *24*, 5279. [CrossRef] [PubMed]
35. Chen, Y.; Guo, Y.; Li, J.; Zhou, B.; Chen, J.; Zhang, M.; Cui, Y.; Tang, J. RT-DETR-Tea: A Multi-Species Tea Bud Detection Model for Unstructured Environments. *Agriculture* **2024**, *14*, 2256. [CrossRef]
36. Wang, S.; Jiang, H.; Yang, J.; Ma, X.; Chen, J.; Li, Z.; Tang, X. Lightweight tomato ripeness detection algorithm based on the improved RT-DETR. *Front. Plant Sci.* **2024**, *15*, 1415297. [CrossRef] [PubMed]
37. Islam, T.; Sarker, T.T.; Ahmed, K.R.; Rankrape, C.B.; Gage, K. WeedVision: Multi-Stage Growth and Classification of Weeds using DETR and RetinaNet for Precision Agriculture. *arXiv* **2025**, arXiv:2502.14890. [CrossRef]
38. Allmendinger, A.; Saltık, A.O.; Peteinatos, G.G.; Stein, A.; Gerhards, R. Assessing the Capability of YOLO- and Transformer-based Object Detectors for Real-time Weed Detection. *arXiv* **2025**, arXiv:2501.17387. [CrossRef]

39. Mullins, C.C.; Esau, T.J.; Zaman, Q.U.; Toombs, C.L.; Hennessy, P.J. Leveraging Zero-Shot Detection Mechanisms to Accelerate Image Annotation for Machine Learning in Wild Blueberry (*Vaccinium angustifolium* Ait.). *Agronomy* **2024**, *14*, 2830. [CrossRef]
40. Mehta, S.; Rastegari, M. MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer. *arXiv* **2022**, arXiv:2110.02178. [CrossRef]
41. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521. [CrossRef]
42. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors* **2023**, *23*, 7190. [CrossRef] [PubMed]
43. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051. [CrossRef]
44. CVAT.ai Corporation. *Computer Vision Annotation Tool (CVAT) Zenodo*; CVAT.ai Corporation: Wilmington, DE, USA, 2024; Available online: https://www.cvat.ai (accessed on 3 July 2025).
45. u/UsernameUnavailable. *What's Going on with This Black Krim? Zone 9b SoCal*; Reddit, r/tomatoes. 2025 (est.). Available online: https://www.reddit.com/r/tomatoes/comments/1lqrkql/whats_going_on_with_this_black_krim_zone_9b_socal/ (accessed on 6 July 2025).
46. Aine, D. Our Tomato Crop; Flickr, Uploaded 30 August 2005, Photo taken 29 August 2005. Available online: https://www.flickr.com/photos/dainec/38406044 (accessed on 6 July 2025).
47. u/UsernameUnavailable. *Who Is This?* Reddit, r/gardening. 2023 (est.). Available online: https://www.reddit.com/r/gardening/comments/160zqa3/who_is_this/ (accessed on 6 July 2025).
48. u/UsernameUnavailable. *Droopy Sad Tomato Leaves?* Reddit, r/vegetablegardening. 2025 (est.). Available online: https://www.reddit.com/r/vegetablegardening/comments/1kd5fkr/droopy_sad_tomato_leaves/ (accessed on 6 July 2025).
49. u/Locke_Wiggin. *Successfully Planted Out My Tomatoes Today. 3 Gardeners Delight and 2 Rebellion Tomatoes.* Reddit, r/tomatoes. 2022–2023 (est.). Available online: https://www.reddit.com/r/tomatoes/comments/une7s9/successfully_planted_out_my_tomatoes_today_3/ (accessed on 6 July 2025).
50. Liu, C.; Feng, Q.; Sun, Y.; Li, Y.; Ru, M.; Xu, L. YOLACTFusion: An Instance Segmentation Method for RGB-NIR Multimodal Image Fusion Based on an Attention Mechanism. *Comput. Electron. Agric.* **2023**, *213*, 108186. [CrossRef]