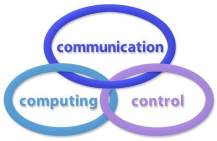
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 3, Month: June, Year: 2026
Article Number: 7399, https://doi.org/10.15837/ijccc.2026.3.7399



CCC Publications

Explainable Tomato and Pepper Leaf Disease Detection Using YOLOv12 and Grad-CAM

B. Tej, S. Bouaafia, M. Hajjaji, A. Mtibaa

Balkis Tej*
Automatic Signal and Image Processing Research Laboratory (LR13ES13)
National Engineering School of Monastir,
University of Monastir, Monastir, Tunisia.
*Corresponding author: balkis.tej@enim.u-monastir.tn

Soulef Bouaafia
Laboratory of Condensed Matter and Nanoscience (LR11ES40),
Faculty of Sciences of Monastir, University of Monastir, Tunisia
Higher Institute of Applied Sciences and Technology of Kairouan,
University of Kairouan, Kairouan, Tunisia
soulef.bouaafia@fsm.rnu.tn

Mohamed Ali Hajjaji
Research Laboratory in Algebra Numbers Theory and Intelligent Systems (RLANTIS),
University of Monastir, Tunisia.
Higher Institute of Applied Sciences and Technology of Sousse,
University of Sousse, Sousse, Tunisia. mohamedali.hajjaji@issatso.rnu.tn

Abdellatif Mtibaa
Systems Integration and Emerging Energies Laboratory (LR21ES14)
National Engineering School of Sfax, University of Sfax, Sfax, Tunisia.
abdellatif.mtibaa@enim.rnu.tn

Abstract

Deep learning–based object detection models have shown strong potential for automated plant disease detection from leaf images. Among these models, YOLO architectures are widely used due to their ability to achieve high detection accuracy while maintaining real-time performance. However, despite these advantages, the adoption of such systems in agricultural practice remains limited. One of the main reasons is that their predictions are often difficult to interpret, which can reduce the confidence of farmers and agricultural experts, who need to understand the basis of model predictions before relying on them for decision-making. To address this issue, this paper proposes an explainable deep learning framework that combines a YOLOv12-based detection model with explainable artificial intelligence techniques. The proposed approach is evaluated on a self-generated dataset of plant leaf images. The experimental results show that the proposed YOLO model achieves satisfactory detection performance, confirming its suitability for plant disease detection tasks. In addition to performance evaluation and to improve transparency, Gradient-weighted Class

Activation Mapping is employed to generate visual explanations of the model's predictions. The resulting heatmaps reveal that the network consistently concentrates on relevant diseased regions of the leaves, indicating that the detection decisions are guided by meaningful visual features. By combining accurate detection performance with visual explanations, the proposed framework aims to provide a more transparent and trustworthy solution for deep learning–based plant disease detection.

Keywords: object detection, plant disease, YOLO architecture, explainable AI.

1 Introduction

In Tunisia, tomato and pepper cultivation plays a major role in the agricultural sector and has been practiced for many years. These two crops are among the most widely grown vegetables and contribute significantly to the country's agricultural economy due to their high production levels and strong market demand. Tomato farming alone occupies nearly 24,500 hectares, with an annual production reaching approximately 1,416,000 tons [1]. Similarly, pepper cultivation extends over about 19,000 hectares, producing nearly 304,000 tons per year [2]. Even though tomato and pepper are very important crops, keeping their production stable has become increasingly difficult because of the spread of leaf diseases. These diseases are caused by different fungi, bacteria, and viruses, and they strongly affect both the quantity and the quality of the harvest. In Tunisia, the impact of leaf diseases has been particularly noticeable; for example, in 2021 the area dedicated to tomato cultivation declined by approximately 15% [3]. For farmers, this often means serious financial losses and growing concerns about food availability. Traditional approaches to plant disease identification rely largely on visual inspection and the extensive use of pesticides. These methods are time-consuming, labor-intensive, and susceptible to human error, which reduces their effectiveness in large-scale farming and poses potential risks to human health [4]. To overcome these difficulties, intelligent systems based on image analysis have become a practical alternative in modern agriculture. In recent years, deep learning (DL) has become a powerful tool for image-based analysis in agriculture, leading to significant improvements in detection accuracy and model robustness [5]. Convolutional neural networks (CNNs) have proven particularly effective at automatically extracting relevant visual patterns from raw images [6], eliminating the need for handcrafted features. Among deep-learning-based object detection methods, the You Only Look Once (YOLO) family stands out for its ability to perform fast and accurate detection in a single forward pass [7], [8]. Despite their success, CNN-based models often operate as black boxes, offering limited insight into their decision-making process [9]. This lack of interpretability can restrict their adoption in agricultural applications, where understanding the reasoning behind a diagnosis is essential for building trust and supporting informed decision-making by farmers and agronomists. To address this limitation, explainable AI (XAI) techniques are incorporated into plant disease detection DL models to make their predictions more transparent and understandable. In this study, we introduce an explainable YOLO-based system for tomato and pepper disease detection. The developed framework leverages the strong detection capabilities of the YOLO architecture while integrating explainability methods to provide reliable disease detection together with visual insights into the model's decision process. The contributions of this paper are:

- A custom dataset was constructed from images collected in greenhouse environments in Tunisia, focusing on tomato and pepper leaf diseases.
- The YOLOv12 architecture was employed to detect and localize leaf diseases.
- Grad-CAM was applied to produce visual heatmaps that show which parts of the leaf influenced the model's predictions.

The rest of the paper is organized as follows. Section 2 reviews related work in plant disease detection. Section 3 analyzes the materials and methods used in this study. Section 4 presents the experimental detection results along with the corresponding heatmap visualizations. Section 5 concludes this paper and points to future extensions of this work for improvement.

2 Related Works

2.1 Plant Disease Detection Using YOLO-Based Models

In recent years, YOLO-based object detection models have been increasingly used for plant disease detection. In recent years, YOLO-based object detection models have been increasingly explored for plant disease detection capability, combining high detection accuracy with real-time processing. Nguyen et al. [10] proposed an enhanced YOLO-based framework for plant disease detection by introducing α_k^cSiLU, a novel parameterized activation function designed to improve feature extraction in deep networks. The method was integrated into the lightweight YOLOv11n architecture and evaluated on tomato and cucumber leaf disease datasets collected under both controlled and real-field conditions. Experimental results showed that, with an optimal scaling factor (α_k^c= 1.05), the proposed activation function improved detection performance, achieving gains of up to 1.1% in mAP@50 compared to the standard SiLU activation. Wang and Liu [11] introduced TomatoGuard-YOLO, an improved YOLOv10-based model for tomato disease detection in real agricultural environments. Their approach combines a lightweight multi-path feature extraction module, a dynamic attention mechanism, and an optimized loss function to better focus on disease regions and handle class imbalance. The proposed model achieved an mAP of 94.23%. Abulizi et al. [12] proposed DM-YOLO, an improved YOLOv9-based model for tomato leaf disease detection in natural field conditions. The method enhances small lesion detection by introducing a dynamic upsampling strategy and improves localization accuracy through an optimized IoU-based loss function. Experimental results showed that the mAP increased by 1.5% compared to the standard YOLOv9. Wang et al. proposed BED-YOLO [13], an enhanced YOLOv10n-based model for tomato leaf disease detection under real field conditions. The method integrates deformable convolution, improved multi-scale feature fusion, and an attention mechanism to better capture small and complex lesion regions. Experimental results showed a clear performance gain, with the mAP increasing from 87.4% to 91.3%. Zheng et al. [14] proposed MSPB-YOLO, an improved YOLOv8-based framework for multi-site pepper blight disease detection. By integrating an attention module to suppress shallow noise, a RepGFPN structure for enhanced multi-scale feature fusion, and an optimized DIoU loss, the model achieved more accurate localization of disease regions across leaves, stems, and fruits. Experimental results indicated that MSPB-YOLO reached an mAP of 96.4%, outperforming the baseline YOLOv8 by 2.2%. Wang et al. proposed YOLO-Pepper, an enhanced YOLOv10n-based model for disease and pest detection in pepper crops. The approach improves small-target recognition and multi-scale feature fusion through adaptive feature extraction modules, a dynamic feature pyramid, and a dedicated small-object detection head. The results indicate that YOLO-Pepper achieved 94.26% mAP at real-time speed, outperforming the baseline YOLOv10n by 11.88%.

2.2 Explainable AI in Agricultural Applications

Beyond the detection performance of YOLO-based models for accurate plant disease detection, understanding how deep models make decisions has become increasingly important, leading to growing interest in Explainable AI techniques for plant disease analysis. Verma et al. [15] investigated the role of Explainable AI in plant disease detection by combining transfer learning with Grad-CAM++ to interpret deep learning model predictions. Using the PlantVillage dataset and multiple CNN architectures, the study showed that visual explanations help highlight disease-relevant regions on leaf images, improving transparency and confidence in model decisions. Özüpak et al. [16] proposed a hybrid deep learning framework combining MobileNetV2 and Vision Transformer (ViT) for maize leaf disease classification, with a strong emphasis on model interpretability through Explainable AI techniques. n addition to achieving high classification accuracy, the study employed Grad-CAM, LIME, and SHAP to visually explain model predictions and highlight disease-relevant regions on leaf images. Karimanzira [17] proposed a tomato leaf disease detection framework that combines deep learning with explainable AI to improve model transparency and practical usability. The approach integrates a Vision Transformer with cascaded group attention (ViT-CGA) and employs DLIME to generate stable and interpretable explanations of model predictions. The authors in [18] developed an approach for tea leaf disease classification using an ensemble of deep learning models, including ResNet50, MobileNet,

EfficientNetB0, and DenseNet121, to address challenges caused by low-resolution images and complex backgrounds. To enhance interpretability, Grad-CAM was employed to visualize disease-relevant regions, showing strong correspondence between highlighted areas and specific disease types. Karim et al. [19] proposed a deep learning–based framework for classifying plant diseases, focusing on model interpretability through explainable AI methodologies. By employing Grad-CAM visualizations, the study highlighted the specific leaf regions that contributed most to the model's predictions, enabling a better understanding of disease characteristics. Gopalan et al. [20] proposed a deep learning–based framework for corn leaf disease classification using the ResNet152 architecture. To enhance model interpretability, Grad-CAM was employed to visualize the image regions that most strongly influenced the model's predictions.

3 Materials and Methods

3.1 Proposed System Architecture

Figure 1 presents the overall workflow of the proposed plant disease detection and explainability system. The process begins with the acquisition and preparation of the dataset, where all images are manually annotated and resized to 640×640 pixels. The dataset is then divided into training and testing subsets following an 80%/20% split. To improve model robustness, the training set is further enriched through data augmentation techniques. Based on this dataset, multiple YOLOv12 variants are trained to detect and localize leaf diseases, and their performance is compared to identify the most effective model. The selected variant is then employed to generate predictions on unseen leaf images, producing bounding boxes and disease labels. The final stage incorporates explainable AI (XAI), where Grad-CAM heatmaps are generated to reveal the most discriminative leaf regions that influenced the model's decisions. This enhances transparency and provides visual justification for the predicted disease classes. Algorithm 1 summarizes the overall workflow for plant disease detection and explainability using the YOLOv12 model.

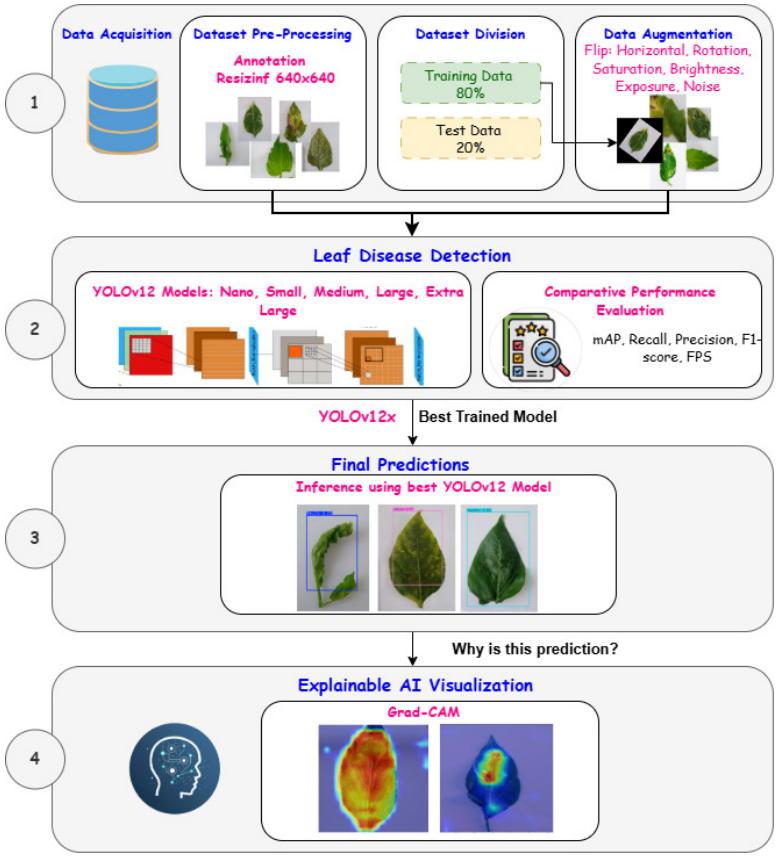


Figure 1: Overview of the proposed system

3.2 Dataset

3.2.1 Dataset Overview

Tomato and pepper crops are essential to Tunisian agriculture and have a big effect on the economy and industrial systems. However, they are highly susceptible to different diseases that can greatly lower productivity and quality. For this reason, our work concentrates on detecting diseases affecting tomato and pepper leaves. To build an appropriate dataset, images were acquired with the help of an agricultural expert who helped identify and describe the varied leaf conditions. We took all the samples in Bkalta, Tunisia, in tomato and pepper greenhouses using a mobile phone camera under controlled conditions. The final dataset has 522 leaf images that are divided into six categories: healthy, leaf miner, oidium, nutrient deficiency, mildew, and tomato yellow leaf curl virus (TYLCV) [21]. Figure 2 shows image examples from the dataset. TYLCV disease produces curled leaf edges, reduced leaf size, and pronounced yellowing. Oidium can lead to darkening, yellowing, and powder-like white spots. Leaf miner is caused by larvae tunneling inside the leaf tissue and creating thin yellow trails. Mildew begins as small circular white patches of fungal growth on the leaf surface, which later expands and turns the surrounding tissue yellow and eventually brown. Nutrient deficiency results from inadequate levels of micronutrients such as zinc (Zn), manganese (Mn), or iron (Fe), producing interveinal chlorosis while the veins remain green.

Algorithm 1 Plant Disease Detection and Explainability Using YOLOv12

Require: Tomato and pepper leaf image dataset
Ensure: Disease predictions and Grad-CAM heatmaps

Dataset Preparation
Acquire leaf images
for each image in the dataset **do**
    Annotate disease regions
    Resize image to 640×640
end for
Split the dataset into training and testing sets
Apply data augmentation to the training set
Model Training
for each YOLOv12 variant $\in \{n, s, m, l, x\}$ **do**
    Train the YOLOv12 model
    Evaluate performance using standard metrics
end for
Select the best-performing model: YOLOv12x
Disease Prediction
Perform inference on unseen leaf images using YOLOv12x
Generate bounding boxes and disease class labels
Explainable AI
Apply Grad-CAM to the selected YOLOv12x model
Generate activation heatmaps
Overlay heatmaps on original images =0

3.2.2 Dataset Preparation

Before being used for model training, the dataset is processed through multiple preparation stages. First, each image was manually annotated by drawing bounding boxes around the relevant leaf regions and assigning each one to its corresponding disease class or to the healthy class. The open-source Roboflow platform [22] was used for this labeling stage since it is an effective tool for managing classes and ensuring precise annotations. These annotations provide the ground-truth information required for training and evaluating the YOLOv12 variants. After annotation, the dataset was divided into training and test subsets. Due to the limited size of the original dataset, data augmentation was

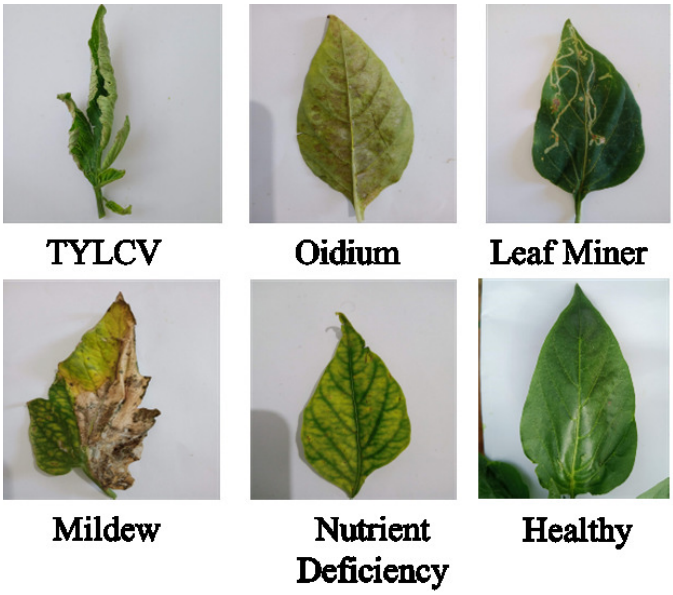


Figure 2: Dataset image examples

applied to increase the number of training samples and enhance the generalization capability of the models. The augmentation process expanded the dataset from 522 to 1352 images. The applied transformations included geometric operations such as rotation, horizontal and vertical flipping, and scaling, as well as photometric adjustments such as brightness variation. These transformations simulate realistic variations that may occur in practical agricultural environments, including changes in leaf orientation and illumination conditions. By introducing controlled variability into the training data, augmentation helps reduce overfitting and improves the robustness of the trained YOLOv12 models. Table 1 summarizes the different data augmentation techniques used.

Table 1: Data Augmentation Techniques

Augmentation Type	**Description**
Flip	Horizontal flipping applied to the input images.
90° Rotation	Clockwise and counterclockwise rotations of 90°.
Random Rotation	Random rotations within the range of −15° to +15°.
Saturation Adjustment	Variation of image saturation between −25% and +25%.
Brightness Adjustment	Brightness variation between −15% and +15%.
Exposure Modification	Exposure adjustment between −10% and +10%.
Blur	Gaussian blur applied with a kernel size of up to 1.4 px.
Noise	Random noise injected into up to 0.1% of the image pixels.

3.3 YOLOv12 Model Architecture

YOLOv12 is a recently introduced real-time object detection architecture [23]. Released in February 2025 by the Ultralytics team, it incorporates an attention-based design that significantly enhances both detection speed and accuracy. Similar to earlier YOLO generations, the YOLOv12 architecture is provided in five model scales (n, s, m, l, x), allowing it to be applied to a wide range of computer vision tasks, including object detection, instance segmentation, and oriented object detection [24]. YOLOv12 brings a major change to the YOLO family by being the first version to use an attention mechanism in an efficient way inside the architecture. Earlier YOLO models relied almost entirely on convolutional layers, but this new design shows that attention can work well even in real-time tasks. To make this possible, the authors introduced three main improvements: a regional attention module, a new residual feature-aggregation block called R-ELAN, and several adjustments to the structure of the network [25], as shown in Figure 3. The regional attention module is one of the key ideas. Instead of applying full self-attention, which is expensive, it splits the feature map into four long regions,

either vertically or horizontally. Working on these smaller areas reduces the cost of attention to about one-third of the original amount. Even with this simplification, the model still captures global information and keeps a wide receptive field, which helps it recognize objects more accurately. The second improvement, R-ELAN, helps with the difficulties that come when training attention-based models. It uses residual shortcuts with scaling factors and a compact feature-aggregation structure. This design avoids gradient problems that often appear in large networks and makes training more stable. At a higher level, the architecture includes several refinements. YOLOv12 removes positional encodings, adjusts the MLP expansion ratio to 1.2 instead of the usual 4, and introduces a separable convolution with a large 7×7 kernel to capture spatial information. These changes keep the model lightweight but still capable of modeling spatial relationships effectively. Memory access patterns were also optimized so that the model runs faster during inference. Additional improvements include the use of Flash Attention, which cuts down memory traffic and helps the model reach speeds closer to classical CNN-based YOLO versions. The backbone is also simplified: the final stage uses only one R-ELAN block instead of stacking multiple attention or convolution units. Moreover, all linear layers were replaced with convolution-batch normalization pairs, which makes the computations more efficient on modern hardware.

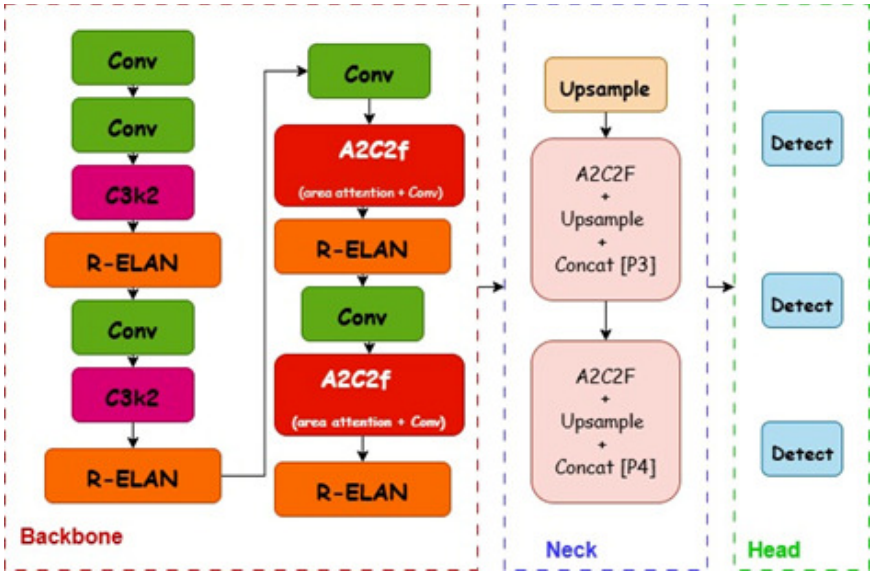


Figure 3: Yolov12 Architecture

3.4 Grad-CAM–Based Explainability for the YOLOv12 Model

Providing explanations is a key requirement for deploying AI models in plant disease analysis, because end-users need interpretable results to guide real agricultural decisions. Interpretability ensures that predictions are not treated as black-box outputs but are supported by clear visual evidence. Among the different interpretation approaches available, Grad-CAM (Gradient-Weighted Class Activation Mapping) is particularly effective, as it produces spatially focused visual explanations that highlight the image regions influencing the model's prediction [26]. It produces a heatmap that highlights the regions of the input image most responsible for the model's prediction. This heatmap is then overlaid onto the original image, allowing the model's decision to be interpreted visually [19]. Grad-CAM works by examining how the gradients of the predicted class interact with the activation maps in the last convolutional layer, allowing it to produce a coarse map that highlights the image regions that most influenced the model's decision [27]. The process involves three main steps: First, Grad-CAM computes the gradients, meaning it measures how the output score of the target class y^c varies regarding each feature map A_k in the chosen convolutional layer. Equation 1 presents this computation

$$\frac{\partial y^c}{\partial A_k} \tag{1}$$

These gradients indicate how sensitive the class score is to variations in the activation maps. Second, the calculated gradients are averaged over the spatial dimensions (i, j) to obtain a single importance weight α_k^c for each feature map as shown in equation 2.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_k^{ij}} \tag{2}$$

where Z denotes the total number of spatial locations in the feature map. Finally, the Grad-CAM heatmap is generated by computing athe feature maps using the weights and applying a ReLU function to keep only positive influences, as shown in equation 3

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c A_k\right) \tag{3}$$

The ReLU function keeps only the positive values, ensuring that the heatmap highlights features that contribute positively to the class prediction. The architecture of Grad-CAM model is shown in Figure 4. After an input image is passed through the CNN, the network generates a series of convolutional feature maps that capture the essential patterns associated with each disease category. These feature maps are then combined using weights that are derived from the activations of the fully connected α_k^c which are derived from the activations of the fully connected layer. The weighted feature maps are subsequently aggregated and processed with a ReLU activation. This step produces a relevance map—known as the Grad-CAM heatmap—that highlights the regions of the leaf most responsible for the model's decision. By overlaying this heatmap onto the original image, Grad-CAM provides an intuitive visual explanation of how the model arrived at its classification.

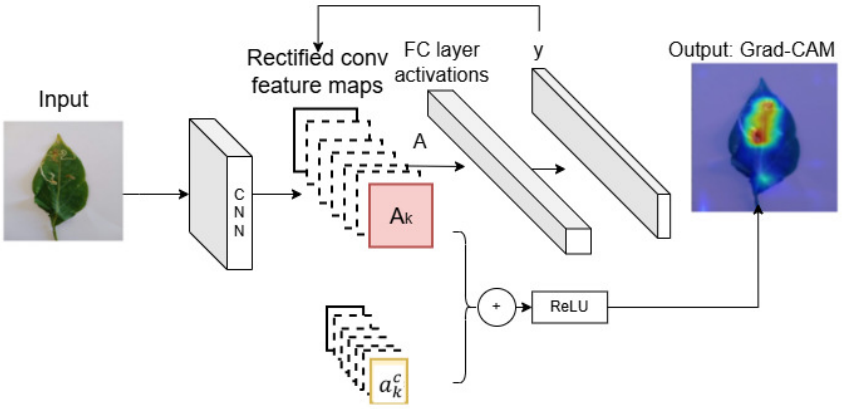


Figure 4: Grad-CAM architecture

4 Experimental Settings

4.1 Experimental Setups and Training Configuration

All YOLOv12 variants were trained using the prepared dataset described earlier in section 3.2. The images had already been separated into training and testing sets, with the split designed to preserve the proportion of each disease class and to maintain a strict separation between samples. The training was carried out on Google Colab Pro, a cloud-based environment equipped with high-performance GPUs and TPUs. This platform provides accelerated computation, extended session durations, and increased RAM, allowing the training of large deep-learning models without the need for a local GPU. Colab Pro's hardware acceleration significantly reduces training time and supports

Table 2: Training Hyperparameters

Parameter	Value
Image size	640 × 640
Optimizer	AdamW
Learning rate	0.01
Epochs	100
Batch size	16
Momentum	0.937
Weight decay	0.005

efficient experimentation. For optimization, the models were trained with the AdamW algorithm, initialized with a learning rate of 0.01. This rate was automatically adapted during training based on validation feedback to stabilize convergence. The models were trained for up to 100 epochs, with additional settings such as momentum of 0.937 and weight decay of 0.0005 to help regularize the training process. Table 2 summarizes the hyperparameters and training settings used during model training.

4.2 Detection Evaluation Metrics

Evaluating a model's performance is a crucial step in determining how well it performs a classification task. Several metrics are commonly used for this purpose, including mean Average Precision (mAP), precision, and recall. These indicators are calculated using the confusion matrix, which breaks down the model's predictions into four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [28]. These metrics are formally defined in Equations 4-8

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

$$AP = \int_0^1 P\,dR \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

Where TP represents the number of leaf images correctly assigned to their true disease class, TN refers to images correctly identified as not belonging to a specific class, FP corresponds to images assigned to a class they do not actually belong to, and FN denotes images that truly belong to a class but were misclassified into another category. In addition to these metrics, the inference speed of the model was evaluated using Frames Per Second (FPS), which measures how many images the system can process in one second. FPS is computed as shown in Equation 9.

$$FPS = \frac{1}{t_{\text{inference}}} \tag{9}$$

where $t_{\text{inference}}$ denotes the time required to process a single image.

5 Results and Discussion

5.1 Detection and Explainability Performance

YOLOv12 offers a scalable architecture where model depth and width can be adjusted to match different computational constraints and accuracy needs. This scaling results in five model variants:

YOLOv12-n, YOLOv12-s, YOLOv12-m, YOLOv12-l, and YOLOv12-x. Each version provides a different balance of speed and capacity. In terms of complexity, YOLOv12-n contains about 2.6M parameters, YOLOv12-s includes 9.3M, YOLOv12-m reaches 20.2M, YOLOv12-l has 26.4M, and the largest version, YOLOv12-x, incorporates 59.1M parameters. These variations make the YOLOv12 family flexible, allowing users to select the model that best fits their hardware capabilities and real-time processing requirements. The detection performance of all YOLOv12 variants is summarized in Table 3, using the evaluation metrics described in Section 4.2.

Table 3: Detection Performance of YOLOv12 Variants

Model	**mAP (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**FPS**
YOLOv12n	79.3	83.7	71.6	77.3	556
YOLOv12s	76.4	78.9	73.4	76.1	227
YOLOv12m	78.6	74.4	78.4	76.3	109
YOLOv12l	74.8	76.9	73.2	75.0	70
YOLOv12x	81.6	77.0	80.4	78.7	110

As expected, the lightweight versions such as YOLOv12n and YOLOv12s run extremely fast, reaching up to 556 FPS, but this speed comes at the cost of lower mAP, precision, and recall. The medium and large models offer a better balance, but their scores remain below those of YOLOv12x. Among all variants, YOLOv12x consistently achieves the highest mAP (81.6%), the strongest F1-score (78.4%), and solid recall, showing that it provides the most reliable and stable detection results. Even though it is not the fastest model, its accuracy gains make it the most suitable choice for a detailed interpretability study. For this reason, YOLOv12x is selected as the final model for generating predictions and applying Grad-CAM visual explanations, as it offers the best overall performance for analyzing plant leaf diseases in our work. Figure 5 presents the progression of training and validation metrics for the YOLOv12x model across 100 epochs. The loss curves for both training and validation steadily move downward, which indicates that the network is progressively reducing its error and learning meaningful representations from the data. At the same time, the performance metrics rise consistently throughout the epochs, showing that the detector becomes more confident and accurate as training advances. The validation curves follow the same trend as the training curves, suggesting that the model generalizes well and is not simply memorizing the training samples. The confusion matrix,

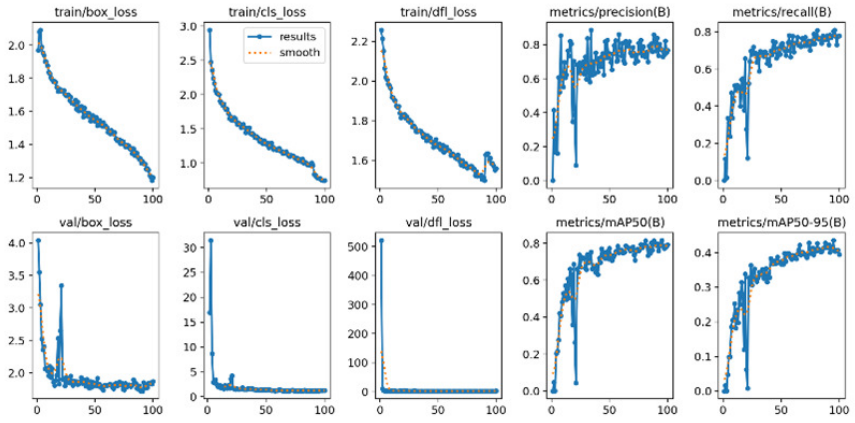


Figure 5: Evolution of Training and Validation Metrics for YOLOv12x

illustrated in Figure 6, complements these results by showing how the model performs on each specific class. The diagonal values are generally high, indicating that the model correctly predicts most samples for each class. Diseases such as healthy, leaf miner, and oidium are recognized with strong consistency, as shown by their dominant diagonal cells. Some misclassifications appear between visually similar

diseases; for example, a few TYLCV samples were mistaken for background, and some mildew cases were predicted as nutrient deficiencies. These confusions are expected because the symptoms share overlapping visual patterns, such as spots or discoloration.

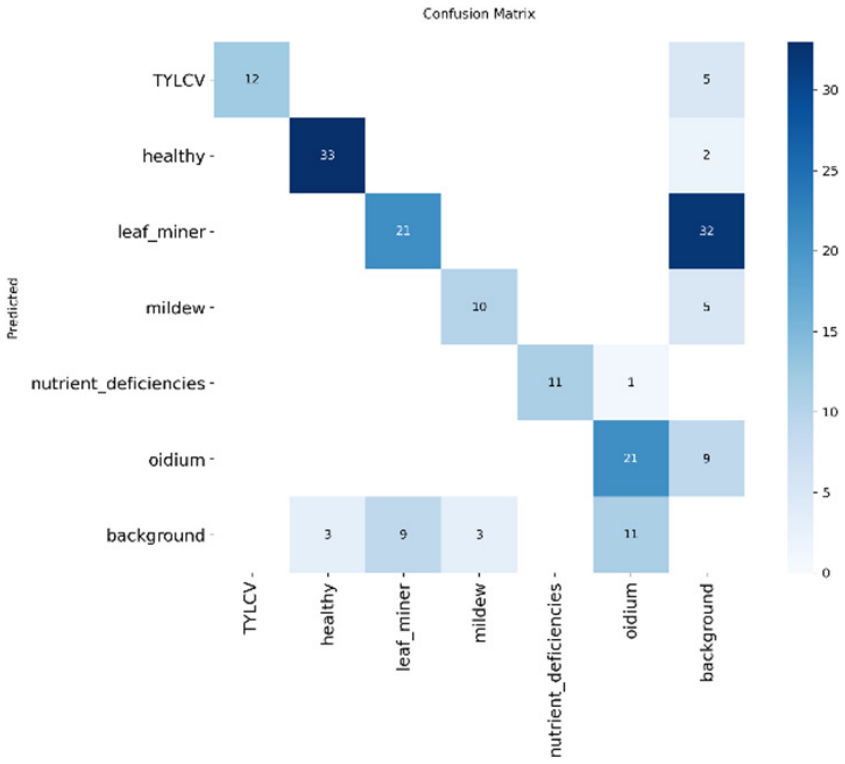


Figure 6: Confusion Matrix of Yolov12x

5.2 Visual Explainability of YOLOv12x Using Grad-CAM

Although modern deep learning models can achieve very high prediction accuracy, their internal reasoning processes are often difficult to interpret, which makes them behave like black-box systems. This lack of transparency can reduce user confidence, especially in applications such as agriculture, where practitioners need to understand the basis of automated decisions. For this reason, explainability plays an important role, as it provides insight into the model's decision-making process. To improve interpretability, Grad-CAM is applied to visualize the regions of the input image that contribute most strongly to the model's predictions. Figure 7 illustrates the explainability results. Each example contains three components: the original leaf image, the detection result produced by the YOLOv12x model, and the corresponding Grad-CAM heatmap. The detection image shows the predicted bounding box and disease label, while the heatmap highlights the areas that most influenced the model's decision. Regions with higher activation intensity and warmer colors indicate the leaf areas that the network considered most informative. The visual analysis reveals several consistent patterns across the disease categories. First, the activated regions largely coincide with the bounding boxes predicted by YOLOv12x, indicating strong consistency between the model's localization capability and its internal feature representation. For example, in the case of Mildew, the strongest activations correspond directly to the fungal lesions present on the leaf surface. Similarly, for Leaf Miner, the Grad-CAM responses are concentrated along the characteristic serpentine feeding tunnels, suggesting that the model relies on these distinctive visual patterns for classification. These observations indicate that the network focuses on meaningful disease symptoms rather than unrelated background information. Such behavior is important for validating the reliability of the model and demonstrates that the predictions are guided by biologically relevant features. By enabling visualization of the decision process, Grad-CAM provides an additional level of transparency that can support expert verification and increase trust in AI-assisted plant disease detection systems.

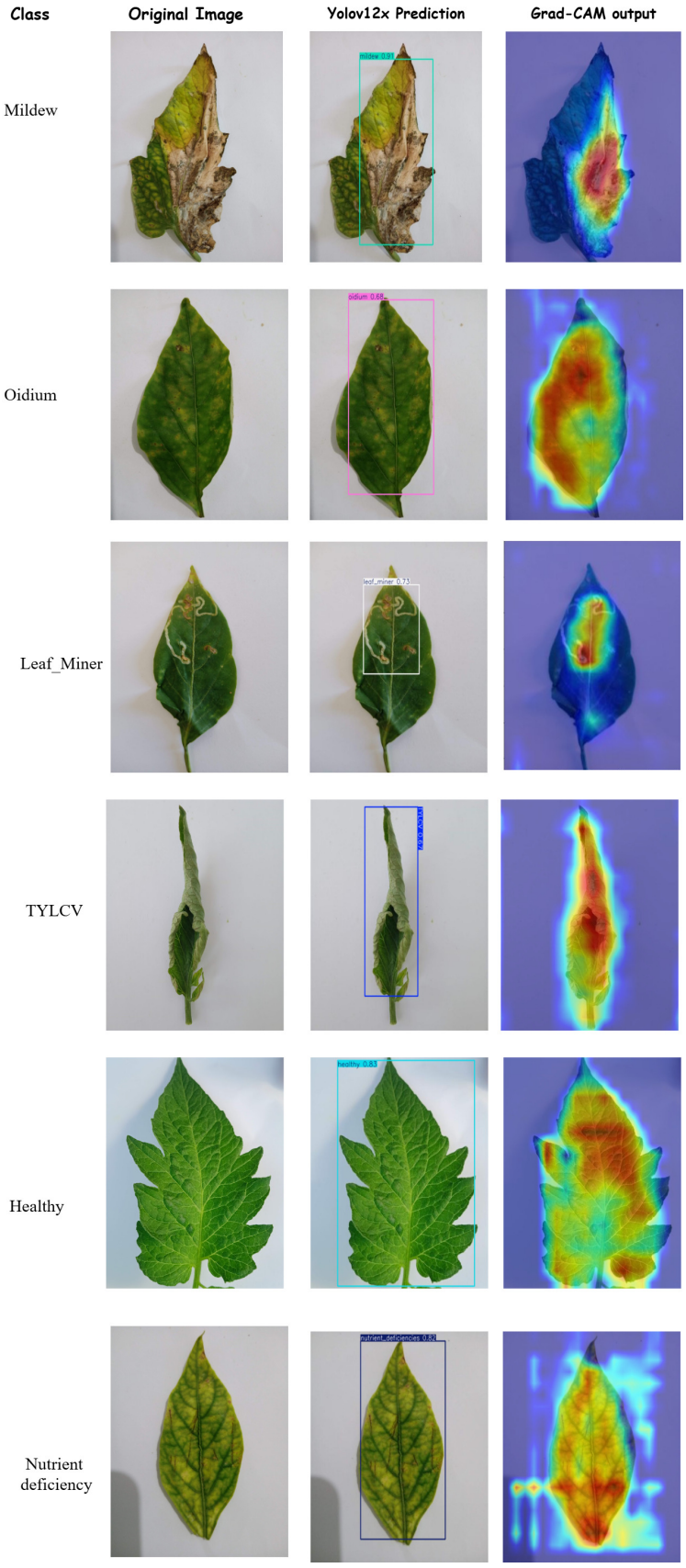


Figure 7: Qualitative results of YOLOv12x predictions and Grad-CAM visual explanations

6 Conclusion and Future Work

In this paper, we presented an explainable deep learning framework based on the YOLOv12 model for tomato and pepper leaf disease detection. Different variants of the YOLOv12 architecture were evaluated to determine the most suitable model in terms of detection accuracy and computational efficiency. The experimental results showed that YOLOv12x achieved the best overall performance among the tested variants. In addition to detection accuracy, model explainability was addressed by integrating Grad-CAM. The generated heatmaps consistently emphasized the leaf regions that played the most important role in the model's predictions. This indicates that the model relies on meaningful disease symptoms rather than background or irrelevant image areas. Such visual explanations improve transparency and are particularly valuable for building trust among agricultural practitioners, supporting the adoption of AI-assisted tools in precision agriculture. As future work, this study can be extended in several directions. First, the dataset will be enriched by collecting more diverse samples from additional cultivation sites and under varied environmental conditions. In addition, cross-dataset validation will be performed using independent datasets to further assess the generalization capability of the proposed framework. Second, explainability can be enhanced by exploring more advanced XAI techniques or combining multiple interpretation methods to provide deeper insights into the model's behavior. Finally, we aim to adapt the system for deployment in resource-limited environments. by applying optimization and model compression techniques.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Romdhane, A.; Riahi, A.; Piro, G.; Lenucci, M.S.; Hdider, C. (2023). Agronomic performance and nutraceutical quality of a tomato germplasm line selected under organic production system, *Horticulturae*, 9(4), 490, 2023.

[2] Ilahy, R.; R'him, T.; Tlili, I.; Hager, J. (2013). Effect of different shading levels on growth and yield parameters of a hot pepper (*Capsicum annuum* L.) cultivar 'Beldi' grown in Tunisia, *Food*, 7(Special Issue 1), 32–35, 2013.

[3] Abderrazek, M.B. (2025). Défis et enjeux du secteur de la transformation des tomates en Tunisie, *Tunisie Numérique*, 2025. Available online: https://www.tunisienumerique.com/defis-et-enjeux-du-secteur-de-la-transformation-des-tomates-en-tunisie/ (accessed on 16 December 2025).

[4] Gupta, H.K.; Shah, H.R. (2023). Deep learning-based approach to identify the potato leaf disease and help in mitigation using IoT, *SN Computer Science*, 4(4), 333, 2023.

[5] Brahimi, M.; Mahmoudi, S.; Boukhalfa, K.; Moussaoui, A. (2019). Deep interpretable architecture for plant diseases classification, in *Proceedings of the Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, IEEE, 111–116, 2019.

[6] Motiejauskas, M.; Dzemyda, G. (2024). Efficientnet convolutional neural network with gram matrices modules for predicting sadness emotion. *International Journal of Computers Communications and Control*, 19(5).

[7] Kaur, R.; Mittal, U.; Wadhawan, A.; Almogren, A.; Singla, J.; Bharany, S.; Hussen, S.; Rehman, A.U.; Al-Huqail, A.A. (2025). YOLO-LeafNet: a robust deep learning framework for multispecies plant disease detection with data augmentation, *Scientific Reports*, 15(1), 28513, 2025.

[8] Alhwaiti, Y.; Khan, M.; Asim, M.; Siddiqi, M.H.; Ishaq, M.; Alruwaili, M. (2025). Leveraging YOLO deep learning models to enhance plant disease identification, *Scientific Reports*, 15(1), 7969, 2025.

[9] Talaat, F.M.; Salem, M.; Shehata, M.; Shaban, W.M. (2025). An efficient explainable AI model for accurate brain tumor detection using MRI images, *Computer Modeling in Engineering & Sciences (CMES)*, 144(2), 2025.

[10] Nguyen, D.T.; Bui, T.D.; Ngo, T.M.; Ngo, U.Q. (2025). Improving YOLO-based plant disease detection using αSiLU: a novel activation function for smart agriculture, *AgriEngineering*, 7(9), 271, 2025.

[11] Wang, X.; Liu, J. (2025). TomatoGuard-YOLO: a novel efficient tomato disease detection method, *Frontiers in Plant Science*, 15, 1499278, 2025.

[12] Abulizi, A.; Ye, J.; Abudukelimu, H.; Guo, W. (2025). DM-YOLO: improved YOLOv9 model for tomato leaf disease detection, *Frontiers in Plant Science*, 15, 1473928, 2025.

[13] Wang, Q.; Yan, N.; Qin, Y.; Zhang, X.; Li, X. (2025). BED-YOLO: an enhanced YOLOv10n-based tomato leaf disease detection algorithm, *Sensors*, 25(9), 2882, 2025.

[14] Zheng, X.; Shao, Z.; Chen, Y.; Zeng, H.; Chen, J. (2025). MSPB-YOLO: high-precision detection algorithm of multi-site pepper blight disease based on improved YOLOv8, *Agronomy*, 15(4), 839, 2025.

[15] Jun, M.; Wang, K.; Liu, Z.; Fu, K.; Zhu, C.; Li, C.; Wang, Z.; Jia, G. (2025). P-YOLO11: an improved lightweight model for accurate detection of declining trees in poplar plantations, *Smart Agricultural Technology*, 101454, 2025.

[16] Özüpak, Y.; Alpsalaz, F.; Aslan, E.; Uzel, H. (2025). Hybrid deep learning model for maize leaf disease classification with explainable AI, *New Zealand Journal of Crop and Horticultural Science*, 1–23, 2025.

[17] Karimanzira, D. (2025). Context-aware tomato leaf disease detection using deep learning in an operational framework, *Electronics*, 14(4), 661, 2025.

[18] Ozturk, O.; Sarica, B.; Seker, D.Z. (2025). Interpretable and robust ensemble deep learning framework for tea leaf disease classification, *Horticulturae*, 11(4), 437, 2025.

[19] Karim, M.J.; Goni, M.O.F.; Nahiduzzaman, M.; Ahsan, M.; Haider, J.; Kowalski, M. (2024). Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM, *Scientific Reports*, 14(1), 16022, 2024.

[20] Gopalan, K.; Srinivasan, S.; Singh, M.; Mathivanan, S.K.; Moorthy, U. (2025). Corn leaf disease diagnosis: enhancing accuracy with ResNet152 and Grad-CAM for explainable AI, *BMC Plant Biology*, 25(1), 440, 2025.

[21] Balkis (2025). Leaf disease dataset, Available online via the Roboflow platform: https://app.roboflow.com/balkis/leaf_disease_dataset-ncmi5/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true (accessed on 04 December 2025).

[22] Roboflow (2025). Roboflow: computer vision tools for developers and enterprises, Available online: https://roboflow.com (accessed on 07 December 2025).

[23] Guo, D.; Yuan, G.; Liu, B.; Liu, Z.; Fen, L.; Zhang, D.; Wang, Z.; Tan, M.; Luo, D.; Guo, J. (2025). SMA-YOLO: an enhanced architecture for the detection of corn diseases based on YOLOv12, *Smart Agricultural Technology*, 101502, 2025.

[24] El-Geneedy, M.; El-Din Moustafa, H.; Khater, H.; Abd-Elsamee, S.; Gamel, S.A. (2025). Advanced real-time detection of acute ischemic stroke using YOLOv12, YOLOv11, and YOLO-NAS: a comparative study for multi-class classification, *Scientific Reports*, 15(1), 32546, 2025.

[25] Alif, M.A.R.; Hussain, M. (2025). YOLOv12: a breakdown of the key architectural features, *arXiv preprint arXiv:2502.14740*, 2025.

[26] Bamaqa, A.; Alahamade, W.O. (2025). A multi-phase framework for enhancing diagnostic accuracy and transparency in renal cell carcinoma grading using YOLOv8 and Grad-CAM, *Scientific Reports*, 15(1), 35370, 2025.

[27] Das, P.; Ortega, A. (2022). Gradient-weighted class activation mapping for spatio-temporal graph convolutional networks, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 4043–4047, 2022.

[28] Miller, C.; Portlock, T.; Nyaga, D.M.; O'Sullivan, J.M. (2024). A review of model evaluation metrics for machine learning in genetics and genomics, *Frontiers in Bioinformatics*, 4, 1457619, 2024.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Tej, B.; Bouaafia, S.; Hajjaji, M.; Mtibaa, A. Explainable Tomato and Pepper Leaf Disease Detection Using YOLOv12 and Grad-CAM, *International Journal of Computers Communications & Control*, 21(3), 7399, 2026.
https://doi.org/10.15837/ijccc.2026.3.7399