EK-YOLOv5: An Enhanced Real-Time Deep Learning Model for Plant Disease Detection

Balkis TEJ[1]*, Soulef BOUAAFIA[2,3], Mohamed Ali HAJJAJI[4,5], Abdellatif MTIBAA[6]

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

[2] Laboratory of Condensed Matter and Nanoscience (LR11ES40), Faculty of Sciences of Monastir, University of Monastir, Monastir, Tunisia

[3] Higher Institute of Applied Sciences and Technology of Kairouan, University of Kairouan, Kairouan, Tunisia
soulef.bouaafia@fsm.rnu.tn

[4] Research Laboratory in Algebra Numbers Theory and Intelligent Systems (RLANTIS), University of Monastir, Monastir, Tunisia

[5] Higher Institute of Applied Sciences and Technology of Sousse, University of Sousse, Sousse, Tunisia
mohamedali.hajjaji@issatso.rnu.tn

[6] Systems Integration & Emerging Energies Laboratory (LR21ES14), National Engineering School of Sfax, University of Sfax, Sfax, Tunisia
abdellatif.mtibaa@enim.rnu.tn

Abstract: Tomato and pepper plants are essential crops in Tunisia's agricultural sector, yet their yield is often compromised by various foliar diseases. This study introduces EK-YOLOv5, an enhanced deep learning model derived from the YOLOv5 architecture, which was designed to improve the detection and localization of leaf diseases in tomato and pepper plants. The proposed EK-YOLOv5 model incorporates two major innovations: replacing the original SiLU activation function with an Exponential Linear Unit (ELU) in order to accelerate convergence and improve non-linearity handling, and employing the K-means++ algorithm for generating more representative anchor boxes for diverse lesion shapes and sizes. Both the baseline YOLOv5 model and the improved EK-YOLOv5 model were trained and tested on a custom dataset, and their performance was evaluated by using standard metrics such as precision, recall, the F1-score and mAP. The experimental results reveal that EK-YOLOv5 significantly outperforms the original model, achieving a mAP of 97.4% as compared with the value of 95.8% obtained by YOLOv5, along with marked improvements with regard to recall, precision, and the detection accuracy. These findings demonstrate the potential of EK-YOLOv5 as a reliable and efficient framework for smart agriculture applications in plant disease monitoring.

Keywords: Deep Learning, EK-YOLOv5 Network, Object Detection and Localization, Plant Leaf Disease.

1. Introduction

Tomato and pepper crops have a long-standing cultivation tradition in Tunisia, emerging as two principal types of vegetation that are essential to the agricultural landscape and economy of the nation. These crops possess a considerable economic significance owing to their extensive production and demand. Indeed, tomato growth encompasses a significant area of about 24,500 hectares, offering a yearly yield of 1,416,000 tons (Romdhane et al., 2023). In addition, pepper crops cover an area of approximately 19,000 hectares, providing a yearly yield of 304,000 tons (Ilahy et al., (2013). Nevertheless, the sustainable cultivation of tomatoes and peppers faces major challenges posed by the prevalence of many leaf diseases. These diseases, induced by several types of fungi, bacteria, and viruses, provide considerable challenges to crop production and quality, leading to significant financial losses for farmers and pose a danger to food safety and livelihoods. The classical methods of detecting plant diseases have predominantly depended on manual visual inspection techniques. Specialized professionals conduct visual inspections of crops for indicators such as a change of color, disfiguring, or abnormalities, which may signify crop disease. These methods, however, feature numerous constraints. First, the visual examination is interpretive (Dai et al., 2023), as various specialists may interpret disease manifestation differently, which leads to differences in the identification of a disease. Moreover, it is a lengthy process (Gupta & Shah, 2023), especially in extensive farming, and might not be practical for an efficient disease handling. Furthermore, an apparent disease manifestation may not be consistently visible, particularly in the early phases of illness progression, leading to a delayed exploration and intervention. To address these constraints, innovative methods for automating and improving the detection and identifying the location of pepper and tomato diseases are needed. Computer vision and artificial intelligence have become essential tools

https://doi.org/10.24846/v35i2y202607

for addressing the challenges of automating the detection and localization of tomato and pepper diseases. With continuous progress in machine learning, numerous computational approaches have been introduced to support agricultural disease diagnosis. Traditional classifiers, including support vector machines (SVM) (Singh & Kaur, 2019), K-nearest neighbors (KNN), Random Forest (Panigrahi et al., 2020), and decision tree (DT) (Rajesh, Vardhan, & Sujihelen, 2020) were studied to examine and identify plant diseases. These algorithms are simple to implement and require relatively modest amounts of training data; however, their performance is highly dependent on manual feature extraction, which remains time-consuming and labor-intensive.

Recently, the emergence of Deep Learning (DL) has been seen as an important technique in the agricultural field, with a view to overcoming the limits of machine learning algorithms. Deep learning has introduced several powerful models into plant disease analysis. Convolutional Neural Network (CNN) based frameworks (Tej et al., 2024a) remain the primary choice for visual diagnosis tasks, whereas architectures like long-short-term memory (LSTM) (Graves, 2012), and recurrent neural networks (RNNs) (Zaremba et al., 2014) have been explored for capturing temporal or spatial dependencies in plant data. CNNs algorithms are the leading and most used ones among the different algorithms of deep learning for detecting plant diseases. Well-known architectures like Resnet (Anim-Ayeko et al., 2023), VGG (Nguyen, Nguyen & Ngo, 2022), and AlexNet (Simhadri & Kondaveeti, 2023), have been used in many studies and have shown effectiveness in precisely identifying plant diseases. Although these approaches are performant in classifying the diseases illustrated in an image, they cannot precisely localize the affected area on the leaf. Classification alone remains insufficient for an optimal management. Object detection methods address this limitation by not only identifying and classifying diseases in plant images but also by localizing the exact affected areas, thus providing more actionable information for targeted treatments.

The traditional object detection systems involved two separate steps. One successful system of this kind was the R-CNN method introduced by Girshick et al. (2014). The process generally consists of two distinct steps. Initially, a region proposal network (RPN) identifies potential regions of interest (RoIs) within the image containing objects. The second stage involves the refinement and classification of these RoIs through the application of techniques such as Fast R-CNN or Faster R-CNN. These models demonstrate an enhanced accuracy; however, they might have reduced inference speeds because of the extra processing steps required. Yet, for single-stage detectors, the model does not include the proposal generation phase used in traditional two-step systems. Instead, they perform classification and localization in one unified process. This enables the identification of the object category and the precise bounding box regression, leading to a decrease in the detection time. Among the most used object detection algorithms is the YOLO series. Several recent studies have focused on improving YOLOv5 for plant leaf disease detection and localization by optimizing specific components of the detection pipeline, such as anchor box generation and model configuration (Tej et al., 2024b). This paper presents an enhanced YOLOv5 architecture, called EK-YOLOv5, which is tailored for identifying and locating disease symptoms on tomato and pepper leaves. This is achieved with the help of self generated data obtained from leaf images captured in the Monastir area of Tunisia. The EK-YOLOv5 model is proposed not only for detecting diseases affecting tomato and pepper leaves, but also for exactly localizing the affected regions on leaves. The improvements specific to the EK-YOLOv5 model in comparison with the standard YOLOv5 model are as follows:

1. Modifying the CBS layer by replacing the Sigmoid-weighted Linear Unit (SiLU) activation function with the Exponential Linear Unit (ELU). The modified CBE block was applied across several parts of the architecture, including C3, Spatial Pyramid Pooling Fast (SPPF), and CBS (Convolution – Batch Normalization – SiLU) modules, and was incorporated throughout the feature extraction, feature-fusion, and prediction stages of the network;

2. Adjusting anchor box configuration by substituting the standard K-means clustering method with K-means++, to improve the optimization of initial clustering points for anchor box generation.

In addition, a dataset was developed, including images of the most significant leaf diseases affecting tomatoes and peppers in Tunisia.

The remainder of this paper is organized as follows. Section 2 presents the proposed methodology, including the proposed EK-YOLOv5 architecture and the dataset used in this study. Section 3 presents the experimental results and performance analysis. Further on, Section 4 provides a comparative evaluation involving EK-YOLOv5 and recent YOLO-based models. Finally, Section 5 concludes this paper and discusses future research directions.

2. Proposed Methodology

A summary of the proposed methodology for detecting plant leaf disease is illustrated in Figure 1.

Initially, a set of plant leaf images was captured using a smartphone camera in the field environment. The collected images were then processed during the data preparation stage, which included data cleaning to remove noise and irrelevant elements, leaf segmentation to isolate the plant leaves from the background, and data annotation using LabelImg, where the regions of interest are marked with bounding boxes and assigned to their classes.

Next, the dataset was partitioned into an 80/20 train–validation split. Due to the limited amount of available data, the training samples were enriched using augmentation methods to reduce the risk of overfitting. To further enhance the detection model's performance, an improved version of the YOLOv5 architecture, called EK-YOLOv5 was developed. In this version, the activation function was modified and the anchor boxes were optimized using the K-means++ algorithm to better adapt to the specific characteristics of the images in the analysed dataset. Finally, the model's performance was evaluated for the validation set using standardized metrics such as precision, recall, F1-score, and mAP.

The performance of EK-YOLOv5 was then compared with that of the standard YOLOv5 model to highlight the improvements achieved.

2.1 Proposed EK-YOLOv5 Architecture

YOLOv5 is a widely used one-stage object detection framework that follows a backbone–neck–head architecture (Shen et al., 2024) and includes several variants with different complexity levels (YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x) (Yan et al., 2021). Owing to its

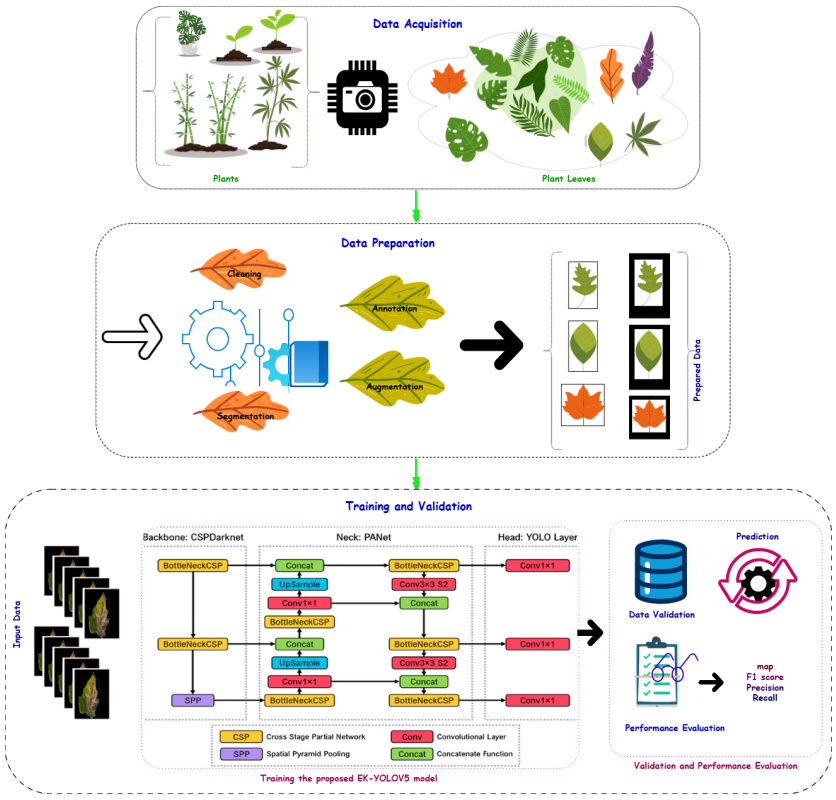

Figure 1. Proposed approach overview

favorable balance between detection accuracy and computational efficiency, YOLOv5 was selected as the baseline architecture in this study. This paper introduces EK-YOLOv5, an enhanced variant of the YOLOv5 architecture, that incorporates two targeted modifications aimed at improving the detection performance. The primary changes include replacing the SILU activation function with the ELU function. This modification aims to address the problem of gradient saturation in negative regions, promoting a more stable learning. Additionally, the detection anchors were optimized using the K-means++ algorithm to improve the initialization of clustering points for the anchor boxes. Figure 2 shows an overview of the proposed architecture.

2.1.1 Activation Function Modification

In YOLOv5-based detection, the choice of the activation and optimization functions is important. The activation functions add a non-linear behavior to the network, making it possible for the model to learn more complex relationships. This non-linearity is essential for neural networks to capture and represent the sophisticated relationships among different data, enabling them to perform tasks that linear models cannot achieve. The YOLOv5 architecture utilizes SiLU as its activation function, also known as Swish. Equations (1) and (2) give the formula for the SiLU function and its derivative, respectively.

$$SiLU(x) = x\frac{1}{1+e^{(-x)}} \tag{1}$$

$$SiLU'(x) = \frac{1}{1+e^{-x}} + \frac{xe^{(-x)}}{(1+e^{(-x)})^2} \tag{2}$$

However, a key limitation of the SiLU activation function is its saturation for negative inputs. In these cases, the derivative of SiLU becomes very small, approaching zero as x decreases. This can lead to the vanishing gradient problem, where the gradients are too weak to effectively update the network's weights. Consequently, learning may be hindered in specific regions of the feature space, particularly for negative values.

In contrast, ELU addresses this issue by handling negative inputs more efficiently, generating non-zero outputs, and avoiding saturation. This is expressed in equations (3) and (4):

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^{x}-1) & \text{if } x \le 0 \end{cases} \tag{3}$$

$$ELU'(x) = \begin{cases} 1 & \text{if } x > 0 \\ \alpha \cdot e^{x} & \text{if } x \le 0 \end{cases} \tag{4}$$

A key advantage of ELU is its improved performance with negative values in comparison with SiLU. While SiLU tends to saturate and produce very small outputs for negative inputs,

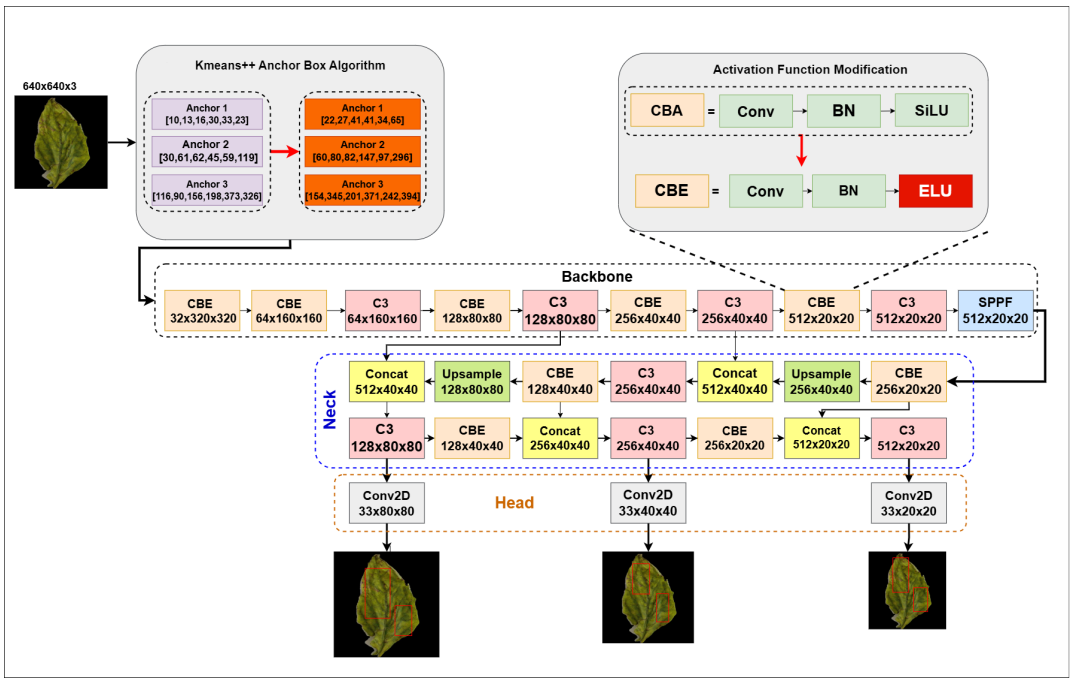

Figure 2. Proposed EK-YOLOv5 architecture

ELU generates non-zero negative outputs and maintains a meaningful derivative, even for values lower than zero. This helps prevent the vanishing gradient problem noticed with SiLU, allowing the model to learn more effectively from negative features. Moreover, the exponential component of ELU for negative inputs enables a faster recovery of neurons that could otherwise become stuck in a saturated state with SiLU, enhancing both the model's stability and convergence speed during training. In the proposed architecture, the SiLU activation function was replaced with ELU across the key components of the model. Specifically, the CBE (Conv + BatchNorm + ELU) block was implemented in place of the original CBS (Conv + BatchNorm + SiLU) block, applying it consistently throughout all the CBS blocks of EK-YOLOv5 for both the backbone and neck.

2.1.2 Optimization of Anchor Box Clustering with Kmeans++

In the YOLO architecture, anchor boxes play an important role in object detection. These boxes serve as predefined reference points for locating objects within an image. They are used for helping the model better handle objects that vary in scale and appearance. Instead of predicting the bounding box coordinates from scratch, the model uses these anchor boxes as a starting point and makes adjustments to match the actual size and shape of the object. Anchor boxes are typically defined by their dimensions (width and height), and these dimensions are often selected based on the characteristics of the objects present in the training dataset. By using anchor boxes as predefined reference bounding boxes, the model can more effectively detect objects of different scales and shapes within an image. Due to the specific characteristics of the analysed dataset, it was necessary to adjust the initial anchor values to better fit the data and the desired anchor sizes. In the standard YOLOv5 algorithm, anchor boxes are generated using the K-means clustering algorithm applied to the bounding boxes of the training dataset. In this context, each bounding box in the dataset is treated as a data point for clustering, and the resulting cluster centers correspond to the optimized anchor boxes used by the YOLO detector. K-means is a clustering algorithm that relies on distance metrics to assess the similarity among data points based on their proximity to one another. It attributes a higher similarity to items that are nearer to each other in space, resulting in the creation of clusters characterized by a high internal cohesion and a low similarity between distinct clusters. However, a limitation of the K-means algorithm lies in the fact that it selects the initial center points for the K pre-defined anchor boxes in the dataset randomly. If these initial points are not carefully chosen, this can result in suboptimal clustering, which can significantly affect the mean average precision (mAP) during training. To overcome this challenge, the K-means++ approach was adopted to generate the anchor boxes. K-means++ improves the initialization process by selecting the cluster centers sequentially, favoring the points that are further away from the already selected centers. This method increases the possibility to obtain centers that cover different regions of the data and represent their distribution more effectively, leading to more accurate clustering results and an improved anchor initialization, which ultimately enhances the overall model performance.

In the K-means++ clustering method, the first cluster center is initialized by randomly selecting a data point from the set. The remaining cluster centers are then selected iteratively based on the following approach: for each sample point x_i in the dataset, the probability of selecting a given point as the next center depends on its distance from the closest center already selected, represented as D(x). In practice, this means that the points located at larger distances from the existing centers are more likely to be selected during initialization. This relationship is given by:

$$P(x) = \frac{D(x)^2}{\sum_{i=1}^{N} D(x_i)^2} \tag{5}$$

where D(x) represents the distance between a sample bounding box x_i and the closest selected cluster center, defined as:

$$D(x) = \min_{c_j \in C}(1 - \mathrm{IoU}(c_j, x_i)) \tag{6}$$

Here, IoU(c_j, x_i) represents the Intersection over Union between the cluster center bounding box c_j and the sample bounding box x_i. The point with the highest probability becomes the next center. The algorithm repeats this step until the required K centers are obtained. For each sample x_i, the distance between each sample and all K centers is computed, and the sample is placed in the cluster with the nearest center. The cluster centers are then adjusted iteratively until their positions stabilize

and no longer change. Algorithm 1 represents the K-means++ algorithm adopted for optimizing anchor box clustering in the proposed approach.

Algorithm 1. K-means++ Anchor Box Optimization for EK-YOLOv5

Input: Dataset D of bounding boxes (x_i), Number of clusters K

Output: Set of optimized anchor boxes (clusters) $C = \{c_1, c_2, ..., c_k\}$

1: Randomly select the first cluster center c_1 from dataset D

2: For each remaining sample $x_i \in$ D, compute the IoU distance D(x) to the nearest selected cluster center c_{init} using equation (6)

3: Compute the probability P(x) for each sample x_i to be selected as the next cluster center using equation (5)

4: Select the next cluster center based on the highest probability P(x)

5: Repeat steps 2-4 until K cluster centers are selected

6: For each sample x_i, assign it to the cluster with the nearest cluster center c_j:

$$Cluster(x_i) = \arg\min_{c_j \in C} Dist(x_i, c_j) \quad (7)$$

7: Update each cluster center to be the mean of all the points assigned to that cluster:

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \quad (8)$$

where $|C_j|$ denotes the number of the elements of cluster C_j.

8: Repeat steps 6 and 7 until cluster centers stabilize

9: Return: Optimized anchor boxes (cluster centers) $C = \{c_1, c_2, ..., c_k\}$

2.2 Dataset

2.2.1 Dataset Description

Deep learning models strongly depend on the availability of representative datasets for achieving a reliable performance. While many existing studies rely on the PlantVillage dataset for tomato and pepper leaf disease detection, this dataset does not reflect regional variations in disease appearance caused by differences in environmental and agricultural conditions. In particular, several tomato and pepper diseases commonly observed in Tunisia are not included in publicly available datasets. For this reason, a dedicated dataset adapted to the Tunisian agricultural context was developed. The images were collected from tomato and pepper greenhouses located in Bekalta, Tunisia, using an OPPO A30 smartphone camera. The acquisition process was supervised by an agricultural engineer to ensure an accurate disease identification. The resulting dataset consists of 522 leaf images covering five disease classes (Leaf Miner, Oidium, Mildew, Nutrient Deficiency, and TYLCV) as well as a healthy class (Tej, 2025). It should be noted that some images contain multiple disease types, meaning that a single image may contribute to more than one class in the distribution analysis.

2.2.2 Dataset Preparation

This phase includes two main steps performed before model training: leaf segmentation and data augmentation, which aim to enhance data quality and improve model robustness.

a. Leaf Segmentation

To reduce background interference and shadow effects, a leaf segmentation step was applied prior to training. The images were converted from the RGB color space to LAB to enhance the leaf–background separation. Otsu's thresholding was applied to the selected channels to generate binary masks, followed by morphological operations to suppress noise. The resulting masks were used for extracting clean leaf regions from the original images, providing focused inputs for disease detection. Figure 3 illustrates representative examples of images from the dataset after segmentation, along with the number of images per class.

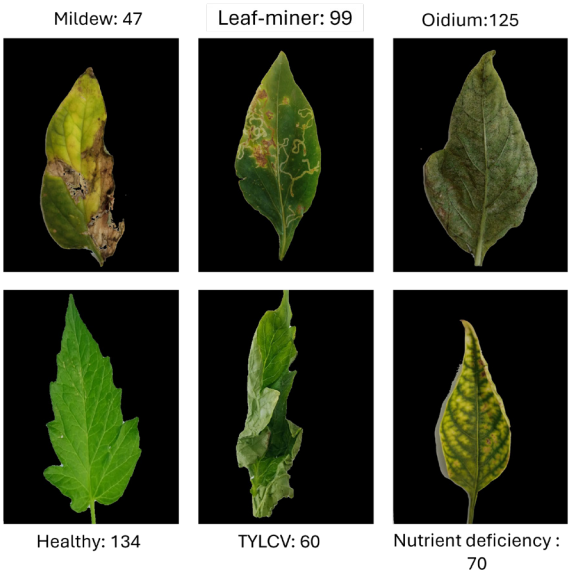

Figure 3. Sample images from the dataset after segmentation

b. Data Augmentation

To address the limited dataset size and class imbalance, data augmentation was applied to the training set. The employed transformations include rotation, horizontal and vertical flipping, brightness and saturation adjustment, Gaussian

blur, noise injection, and resizing to 640 × 640. As a result, the training set increased from 415 to 1245 images, while the validation set remained unchanged at 107 images.

3. Numerical Results

3.1 Experimental Setup and Training Parameters

The Google Colab platform was used for carrying out the training phase in the experimental setup. Google Colab is a simple and robust platform for model training. It allowed to benefit from the high-performance GPUs without requiring a costly hardware. The cloud-based architecture of Google Colab provided abundant computational resources and a reliable runtime environment during training and tests. This made it possible to effectively test and explore various model setups, hyperparameters, and training methodologies. For this study, the data was split into two parts: 80% was used for training the model, and 20% was used for evaluating it. The YOLOv5 model follows a defined configuration and training procedure. Once the dataset is annotated, the model will be adjusted to suit the dataset and hardware configuration. Training EK-YOLOv5 requires two YAML files: one for dataset definition and class labels, and one for the model architecture configuration. Table 1 illustrates the hyperparameters and training parameters for training the model.

Table 1. Training Hyperparameters

Parameters	Value
Image Size	640×640
Learning Rate	0.01
Batch Size	16
Epochs	50
Optimizer	SGD
Weight Decay	0.0005
Momentum	0.937

3.2 Evaluation Metrics

To assess the efficacy and performance of object identification models, evaluation measures are required. In the context of YOLOv5, multiple metrics were used for evaluating the trained model's accuracy and resilience. The most often used evaluation metrics are precision, recall, F1-score, average precision (AP) and mean average precision (mAP). The formulas for calculating these metrics are rendered in equations (9)-(13) (Khalid et al., 2023; Shang, Zhang & Song, 2022):

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \tag{9}$$

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{10}$$

$$F1 - score = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Re}\,call}{\mathrm{Pr}\,ecision + \mathrm{Re}\,call} \tag{11}$$

$$AP = \int_0^1 P \cdot Rdr \tag{12}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{13}$$

3.3 Results and Discussion

The architecture of the YOLOv5 network can be scaled by adjusting its width and depth, resulting in five distinct versions: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These variants have parameter counts of 1.9 million, 7.30 million, 21.4 million, 47.1 million, and 87.8 million, respectively. Typically, the models with fewer parameters provide faster computation times at the cost of a reduced accuracy. Choosing the appropriate model variant is essential for maximizing the potential of YOLOv5. In this study, all YOLOv5 variants were trained on the proposed dataset with data augmentation, using pre-trained models. The results, included in Table 2, demonstrate that YOLOv5x achieved the highest detection accuracy. Considering both precision and efficiency in detecting plant leaf diseases, YOLOv5x was selected as the base network for

Table 2. Performance Comparison for the YOLOv5 Variants

Model	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
Recall	88.3 %	92.8%	88.8%	91.8%	92.8%
Precision (%)	71.2%	76.4%	90.4%	77.3%	88.7%
F1-Score	78.83%	83.8%	89.6%	83.92%	90.7%
mAP	90.4%	94.8%	93.6%	95.6%	95.8%
Size (MB)	3.7	15	41	90	170
GigaFLOPs	4.2	15.8	49	109.1	203.9

detecting diseases in tomato and pepper leaves. Further modifications were made to its architecture in order to enhance its performance.

3.4 Ablation Study

Table 3 summarizes the experimental results for the ablation study, providing a comprehensive comparison between the standard YOLOv5x model and its enhanced variants integrating K-means++ clustering, and the ELU activation function, along with the full EK-YOLOv5x architecture. The Ablation experiments revealed that the variant enhanced with K-means++ achives a higher mAP of 96.1% (as compared to 95.8%), while replacing SiLU with ELU raises the accuracy to 96.9%. The complete EK-YOLOv5x model yields the most significant advancements, boosting the precision from 88.7% to 94.6%, elevating the F1-score from 90.7% to 92.0%, and achieving a mAP of 97.4%. These results demonstrate that each enhancement contributes positively, and that their combination provides a more accurate and efficient system for plant leaf disease detection.

3.5 Loss Analysis for the EK-YOLOv5 model

For evaluating the performance of the EK-YOLOv5 model, loss metrics are fundamental in determining how accurately the model's predictions correspond to the actual data.

The total loss is expressed as the sum of three components:

$$Total = L_{cls} + L_{obj} + L_{box} \quad (15)$$

where L_{cls} is the classification loss, used for classifying the detected objects into their respective categories, L_{obj} is the object confidence loss, used for measuring how well the model predicts the presence of an object in a bounding box and L_{box} represents the Box Regression Loss. Although the total loss defines the overall optimization objective, examining each component individually provides a deeper insight into the contribution of each task to the training process.

Figure 4 provides a visual representation of the loss metrics during the training and validation phases, showing final training losses of 0.0273 (box), 0.011 (objectness) and 0.0035 (classification), with the corresponding validation losses of 0.0253, 0.005, and 0.0308, respectively. The smooth decrease and convergence of the three loss components during both the training and validation phases indicate a stable optimization behavior and confirm the effectiveness of the proposed EK-YOLOv5 model.

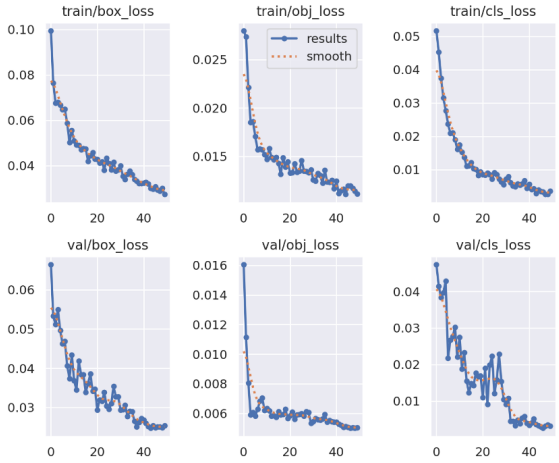

Figure 4. Training and validation loss using the EK-YOLOv5 model

3.6 EK-YOLOv5 Training Performance

Figure 5 illustrates the precision–recall curve for the EK-YOLOv5 model, illustrating how recall and precision vary across different probability thresholds for each leaf disease class. The overall PR curve (blue) reaches a mAP of 0.974 at the IoU value of 0.5, indicating that the model maintains a strong balance between precision and recall across all classes.

Table 3. Comparative Ablation Studies

Model	Recall (%)	Precision (%)	F1-Score (%)	mAP (%)
Standard YOLOv5x	92.8	88.7	90.7	95.8
YOLOv5x + K-means++	93.1	88.9	90.95	96.1
YOLOv5x + ELU	92.0	92.7	92.34	96.9
EK-YOLOv5x	91.3	94.6	92.0	97.4

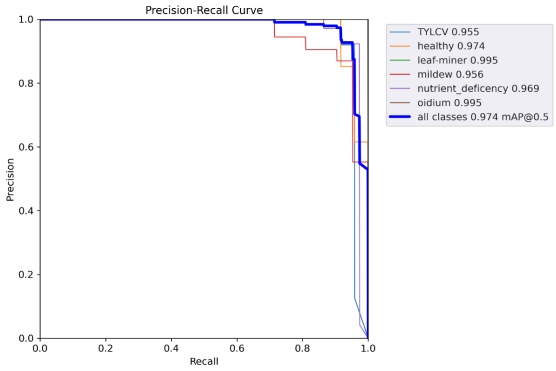

Figure 5. PR curve

Figure 6 illustrates the EK-YOLOv5 model's accurate detection of various plant conditions, such as nutrient deficiency, TYLCV, leaf-miner and oidium, along with healthy leaves, each labeled based on high-confidence predictions, demonstrating its reliability in handling diverse and complex cases.

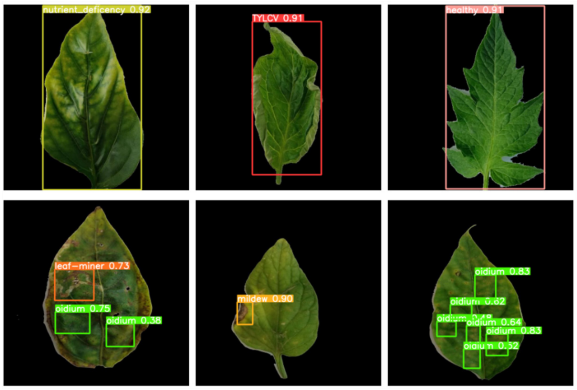

Figure 6. EK-YOLOv5 detection results

4. Comparative Performance Evaluation Involving EK-YOLOv5 and Recent YOLO Models

In this study, EK-YOLOv5 is compared with several state-of-the-art YOLO variants, including YOLOv8x (Jocher, Chaurasia & Qiu, 2023), YOLOv10x (Wang et al., 2024), and YOLOv11x (Jocher & Qiu, 2024), under the same experimental conditions and using the same self-generated dataset. As it can be seen in Table 4, EK-YOLOv5 achieves a precision of 94.6%, a recall of 91.3%, a F1-score of 92.0%, and a mAP of 97.4%, outperforming YOLOv8x (mAP = 95.0%), YOLOv11x (mAP = 92.9%), and YOLOv10x (mAP = 87.6%). These results demonstrate that the proposed activation function modification and anchor box optimization significantly improve the detection accuracy and localization performance, while maintaining a competitive recall in comparison with the more recent YOLO models.

5. Conclusion and future directions

This study introduced the EK-YOLOv5 architecture, an enhanced version of YOLOv5 developed in order to improve leaf disease detection by using a self-generated dataset. The proposed model integrates two key modifications: replacing the SiLU activation function with ELU to improve the learning efficiency and applying the K-means++ algorithm for a more optimal anchor box initialization. The experimental results demonstrate that EK-YOLOv5 achieves a mAP of 97.4%, which is higher than the value of 95.8% achieved by the standard YOLOv5, confirming the effectiveness of these enhancements in boosting the detection accuracy and overall performance.

Future research will explore optimization and compression techniques, such as quantization and binarization, to enable an efficient FPGA-based deployment of the disease detection system. These methods aim to reduce the computational load, memory usage, and power consumption while maintaining an acceptable accuracy, allowing the model to operate effectively in resource-constrained environments and guiding the evaluation of trade-offs between performance and hardware efficiency.

Table 4. Performance comparison involving recent YOLO models

Model	mAP (%)	Recall (%)	Precision (%)	F1-score (%)
YOLOv8x	95.0	91.0	83.7	87.1
YOLOv10x	87.6	85.2	74.9	79.7
YOLO11x	92.9	86.7	79.5	82.9
EK-YOLOv5x	**97.4**	**91.3**	**94.6**	**92.0**

REFERENCES

Anim-Ayeko, A.O., Schillaci, C. & Lipani, A. (2023) Automatic blight disease detection in potato (solanum tuberosum l.) and tomato (solanum lycopersicum, l. 1753) plants using deep learning. *Smart Agricultural Technology*. 4, 100178. https://doi.org/10.1016/j.atech.2023.100178.

Dai, G., Fan, J., Tian, Z. et al. (2023) PPLC-Net: Neural network-based plant disease identification model supported by weather data augmentation and multi-level attention mechanism. *Journal of King Saud University- Computer and Information Sciences.* 35(5). https://doi.org/10.1016/j.jksuci.2023.101555.

Girshick, R., Donahue, J., Darrell, T. et al. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 18-20 June 1996, San Francisco, USA*. New York, USA, IEEE. pp. 580–587.

Graves, A. (2012) Long short-term memory. In: *Supervised sequence labelling with recurrent neural networks*. Berlin, Heidelberg, Springer, pp. 37–45. https://doi.org/10.1007/978-3-642-24797-2_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. https://doi.org/10.1007/s42979-023-01758-5.

Ilahy, R., R'him, T., Tlili, I. et al. (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.

Jocher, G., Chaurasia, A. & Qiu, J. (2023) *YOLO by Ultralytics (Version 8.0.0). Computer software. GitHub.* https://github.com/ultralytics/ultralytics [Accessed 15th May 2026].

Jocher, G. & Qiu, J. (2024) *Ultralytics*. https://github.com/ultralytics/ultralytics [Accessed 17th May 2026].

Khalid, S., Oqaibi, H.M., Aqib, M. et al. (2023) Small pests detection in field crops using deep learning object detection. *Sustainability*. 15(8), 6815. https://doi.org/10.3390/su15086815.

Nguyen, T.-H., Nguyen, T.-N. & Ngo, B.-V. (2022) A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease. *AgriEngineering.* 4(4), 871–887. https://doi.org/10.3390 /agriengineering4040056.

Panigrahi, K.P., Das, H., Sahoo, A.K., et al. (2020) Maize leaf disease detection and classification using machine learning algorithms. In: *Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019*. Singapore, Springer, pp. 659–669. https://doi.org/10.1007/978-981-15-2414-1_66.

Rajesh, B., Vardhan, M.V.S. & Sujihelen, L. (2020) Leaf disease detection and classification by decision tree. In: 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI), *15–17 June 2020, Tirunelveli, India*. New York, USA, IEEE. https://doi.org/10.1109/ICOEI48184.2020.914 2988.

Romdhane, A. Riahi, A., Piro, G. et al. (2023) Agronomic performance and nutraceutical quality of a tomato germplasm line selected under organic production system. *Horticulturae*. 9(4), 490. https://doi.org/10.3390/horticulturae9040490.

Shang, Y., Zhang, Q. & Song, H. (2022) Application of deep learning using yolov5s to apple flower detection in natural scenes. *Transactions of the Chinese Society of Agricultural Engineering.* 38(9), 222-229. https://doi.org/10.11975/j.issn.1002-6819.2022.09. 024.

Shen, X., Shao, C., Cheng, D. et al. (2024) YOLOv5-POS: research on cabbage pose prediction method based on multi-task perception technology. *Frontiers in Plant Science.* 15, 1455687. https://doi.org/10.3389/ fpls.2024.1455687.

Simhadri, C.G. & Kondaveeti, H.K. (2023) Automatic recognition of rice leaf diseases using transfer learning. *Agronomy*. 13(4), 961. https://doi.org/10.3390/agronomy13040961.

Singh, J. & Kaur, H. (2019) Plant disease detection based on region-based segmentation and KNN classifier. In: *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, *16-17 May 2018, Palladam, India* (*Lecture Notes in Computational Vision and Biomechanics*, vol. 30). Cham, Springer, pp. 1667–1675. https://doi.org/10.1007/978-3-030-00665-5_154.

Tej, B., Bouaafia, S., Hajjaji, M.A. et al. (2024a) AI-based smart agriculture 4.0 system for plant diseases detection in Tunisia. *Signal, Image and Video Processing*. 8(Suppl. 1), 97-111. https://doi.org/10.1007/s11760 -024-03134-z.

Tej, B., Bouaafia, S., Hajjaji, M.A. et al. (2024b) An improved YOLOv5 for accurate detection and localization of tomato and pepper leaf diseases. Res.Square [Preprint]. https://doi.org/10.3389/ fpls.2024.149 3322

Tej, B. (2025) *Leaf Disease Detection Dataset [Dataset]. Roboflow*. https://universe.roboflow.com/balkis/leaf_disease_dataset-ncmi5 [Accessed 15th May 2026].

Wang, A., Chen, H., Liu, L. et al. (2024) Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*. 37, 107984-108011. https://doi.org/10.52202/079017-3429.

Yan, B., Fan, P., Lei, X. et al. (2021) A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sensing.* **13**(9), 1619. https://doi.org/10.3390/rs13091619.

Zaremba, W., Sutskever, I. & Vinyals, O. (2014) Recurrent neural network regularization. [Preprint] https://arxiv.org/abs/1409.2329.