VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

COMPARATIVE EVALUATION OF YOLOV5–YOLOV11 MODELS FOR DETECTING NUTRIENT DEFICIENCY IN CHILI SEEDLINGS

Rangga Pebrianto[1*]; Agus Buono[1]; Heru Sukoco[1]; Aziz Kustiyo[1]; Muhamad Syukur[2]

School of Data Science, Mathematics and Informatics[1]
Department of Agronomy and Horticulture, Faculty of Agriculture[2]
IPB University, Bogor, Indonesia[1,2]
https://www.ipb.ac.id/id/[1,2]
ranggapebrianto@apps.ipb.ac.id*, agusbuono@apps.ipb.ac.id, hsrkom@apps.ipb.ac.id,
azizku@apps.ipb.ac.id, muhsyukur@apps.ipb.ac.id

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *Nutrient deficiencies during the seedling stage of chili plants can reduce crop productivity, while conventional identification methods remain subjective and costly. This study compares YOLOv5 to YOLOv11 object detection models for detecting nutrient deficiency symptoms in Bonita chili seedling leaves, including complete nutrition, nitrogen deficiency, phosphorus deficiency, potassium deficiency, and NPK deficiency. The final dataset comprised 4,173 images derived from 1,739 original annotated leaf images through controlled dataset preparation, including split-before-augmentation, laboratory validation of nutrient conditions, and expert-reviewed labeling. All YOLO models were trained and evaluated using the same dataset partition and comparable experimental settings. Performance was assessed using mAP@0.5, computational complexity (FLOPs), inference speed, and model size. The results show that all evaluated models achieved high detection performance, with differences mainly appearing in computational efficiency and the balance between accuracy and speed. YOLOv10s and YOLOv11s obtained the highest mAP@0.5 in this experiment, whereas YOLOv8s showed a competitive balance between accuracy, inference speed, and model compactness. These findings indicate that recent YOLO developments are promising for fine-grained nutrient deficiency detection in computer vision–based precision agriculture.*

Keywords: *Chili Leaves, Comparison, Nutrient Detection, YOLO.*

Intisari— *Kekurangan unsur hara pada fase pembibitan cabai dapat menurunkan produktivitas tanaman, sementara metode identifikasi konvensional masih bersifat subjektif dan mahal. Penelitian ini membandingkan model deteksi objek YOLOv5 hingga YOLOv11 untuk mendeteksi gejala defisiensi hara pada daun bibit cabai Bonita, meliputi kondisi nutrisi lengkap, defisiensi nitrogen, fosfor, kalium, dan NPK. Dataset akhir terdiri atas 4.173 citra yang berasal dari 1.739 citra daun beranotasi asli melalui proses persiapan dataset terkontrol, termasuk pembagian data sebelum augmentasi, validasi laboratorium kondisi hara, dan peninjauan anotasi oleh pakar. Seluruh model YOLO dilatih dan dievaluasi menggunakan partisi dataset yang sama serta pengaturan eksperimen yang sebanding. Kinerja dievaluasi menggunakan mAP@0.5, kompleksitas komputasi (FLOPs), kecepatan inferensi, dan ukuran model. Hasil menunjukkan bahwa seluruh model yang diuji mencapai performa deteksi tinggi, dengan perbedaan utama pada efisiensi komputasi dan keseimbangan antara akurasi dan kecepatan. YOLOv10s dan YOLOv11s memperoleh nilai mAP@0.5 tertinggi pada eksperimen ini, sedangkan YOLOv8s menunjukkan keseimbangan yang kompetitif antara akurasi, kecepatan inferensi, dan kekompakan model. Temuan ini menunjukkan bahwa perkembangan terbaru arsitektur YOLO menjanjikan untuk deteksi defisiensi hara yang bersifat fine-grained dalam sistem pertanian presisi berbasis visi komputer.*

Kata Kunci: *Daun Cabai, Perbandingan, Deteksi Nutrisi, YOLO*

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263

INTRODUCTION

Chili is an important horticultural commodity in Indonesia because it contributes to food supply, farmer income, and national agricultural value [1], [2], [3]. However, chili seedlings are highly sensitive to nutrient availability during the early growth stage. Nutrient deficiency can alter leaf morphology, suppress growth, reduce fruit formation, and ultimately decrease crop productivity [4]. Early and reliable identification of nutrient deficiency symptoms is therefore important for efficient and sustainable chili cultivation. In conventional practice, nutrient deficiency identification still relies largely on visual observation and laboratory analysis. Visual assessment is subjective, depends on farmer experience, and is prone to misinterpretation because several nutrient deficiencies exhibit similar morphological symptoms [5], [6]. Laboratory analysis is more reliable, but it requires additional cost, time, and supporting facilities that are often unavailable to small-scale farmers [7], [8].

These limitations motivate the development of faster and more objective image-based detection methods. In this study, the chili seedling leaf dataset was established through controlled experiments, laboratory analysis of the growing medium and leaf tissue, and expert validation to support reliable labeling. Recent advances in deep learning–based computer vision have enabled automated analysis of plant conditions from leaf images [9]. Among the available approaches, the You Only Look Once (YOLO) family is widely used for object detection because it combines high detection performance with efficient inference [10], [11], [12]. As the YOLO family has evolved, newer versions have introduced design refinements intended to improve feature extraction, efficiency, and practical deployment [13], [14], [15]. YOLO formulates object detection as a single regression task by predicting bounding boxes and class probabilities directly from the entire image in one forward pass [16], [17], [12]. This single-stage formulation makes YOLO suitable for real-time applications and has encouraged its broad adoption in agricultural vision tasks.

Earlier YOLO generations established the technical foundation for later developments across different detection settings [18]. YOLOv3 introduced multi-scale detection with the Darknet-53 backbone [19], while YOLOv4 improved training efficiency and detection robustness through CSPDarknet-based design and optimized training strategies [20]. These developments provided an important baseline for subsequent comparisons. Modern YOLO variants from YOLOv5 onward further diversified the design space. YOLOv5 popularized a modular PyTorch-based workflow that simplified experimentation and deployment [21], [22]. Later versions, including YOLOv6 and YOLOv7, emphasized efficiency and improved feature aggregation [23], [24], [25], whereas YOLOv8 adopted an anchor-free design within the Ultralytics framework [26]. Subsequent variants, including YOLOv9, YOLOv10, and YOLOv11, introduced additional refinements aimed at improving training efficiency, inference behavior, and feature representation [27], [19], [28], [29], [30], [31], [32]. These advances have enabled YOLO to evolve as one of the leading approaches to object detection, as reflected in its iterations summarized in Table 1.

Table 1. Evolution of YOLO Models

Model	Release Year	Key Contribution	Framework
YOLOv3	2018	Multi-scale detection with Darknet-53 backbone	Darknet
YOLOv4	2020	CSPDarknet-53 backbone and optimized training strategies	Darknet
YOLOv5	2020	Modular PyTorch implementation for efficient training and deployment	PyTorch
YOLOv6	2022	Efficiency-oriented design for real-time object detection	PyTorch
YOLOv7	2022	Improved feature aggregation and training optimization	PyTorch
YOLOv8	2023	Anchor-free Ultralytics architecture	PyTorch
YOLOv9	2024	PGI and GELAN-based design	PyTorch
YOLOv10	2024	Design refinements to reduce post-processing dependency	PyTorch
YOLOv11	2024	Further efficiency-oriented refinements in the Ultralytics model family	PyTorch

Source: (Fathurrahman [17], 2025)

Table 1 summarizes the main development trajectory from YOLOv3 to YOLOv11. In this study, YOLOv5 to YOLOv11 were selected because these versions represent the modern YOLO family, are openly available, and are widely used in current experimentation workflows, including the official Ultralytics documentation [28]. Earlier versions were retained only as historical context [17]. Several previous studies have demonstrated the usefulness of YOLO-based models in agricultural and non-agricultural detection tasks. In agriculture,

[33] compared YOLOv5 to YOLOv11 for detecting Coleoptera-related damage on bean leaves under natural field conditions. Their results showed that performance differences across YOLO variants can depend strongly on dataset characteristics and optimization choices. However, the study focused on pest-related symptoms rather than nutrient deficiency patterns, which are usually subtler and more visually overlapping.

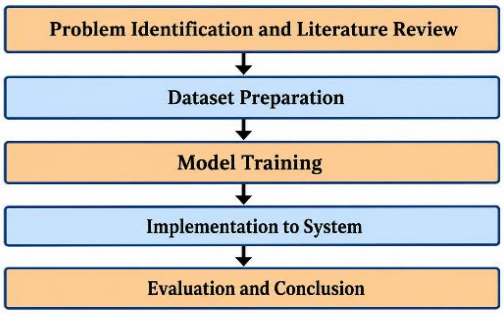
In the urban domain, [17] compared YOLOv3 to YOLOv11 for empty parking-slot detection and reported a different trade-off pattern, with YOLOv7 emphasizing accuracy and YOLOv5 emphasizing inference speed. This comparison is relevant because it shows that newer YOLO versions are not automatically superior across all tasks, especially when the visual characteristics of the target objects differ substantially. Based on recent literature, several research gaps remain. First, many agricultural studies focus on pests or diseases with more explicit visual markers, while nutrient deficiencies often appear as subtle changes in color, chlorosis, and leaf morphology. Second, systematic comparison of YOLOv5 to YOLOv11 for nutrient deficiency detection in chili seedlings remains limited. Third, prior comparisons often use different datasets, validation protocols, and performance settings, making direct interpretation difficult. These gaps justify a controlled comparison using the same dataset and evaluation procedure.

Most existing studies also evaluate only one or two YOLO variants, such as YOLOv5 or YOLOv8, so the effect of YOLO architectural evolution on fine-grained nutrient deficiency detection is still not well documented. This issue is important because differences in feature extraction, multi-scale fusion, and computational efficiency may affect the ability of a model to recognize subtle symptoms on small seedling leaves. Therefore, this study benchmarks YOLOv5 to YOLOv11 for single-object bounding-box detection of nutrient deficiencies in Bonita chili seedling leaves. The evaluation uses precision, recall, and mean Average Precision (mAP) on a homogeneous dataset and standardized testing protocol. The study is intended to contribute empirical evidence for selecting YOLO architectures in fine-grained agricultural detection tasks.

YOLOv5 was chosen as the starting point of the comparison because it marks the transition to the modern PyTorch-based YOLO ecosystem and provides a stable baseline for later versions [21], [34]. From YOLOv5 onward, many implementations report comparable evaluation outputs such as precision, recall, mAP, FLOPs, and inference speed, which supports a more consistent benchmark setting. Accordingly, this study focuses on YOLOv5 to YOLOv11 as the most relevant range for contemporary comparative analysis. Accordingly, this study analyzes the comparative performance of YOLOv5 to YOLOv11 for detecting nutrient deficiencies in Bonita chili seedling leaves under the same dataset partition and evaluation settings. The expected contribution is both theoretical, through empirical mapping of YOLO architectural development, and practical, through identification of models that are more suitable for accurate and efficient nutrient deficiency detection.

MATERIALS AND METHODS

This research was conducted through several systematic stages, as illustrated in Figure 1.



Source: (Research Results, 2026)
Figure 1. Research Stages

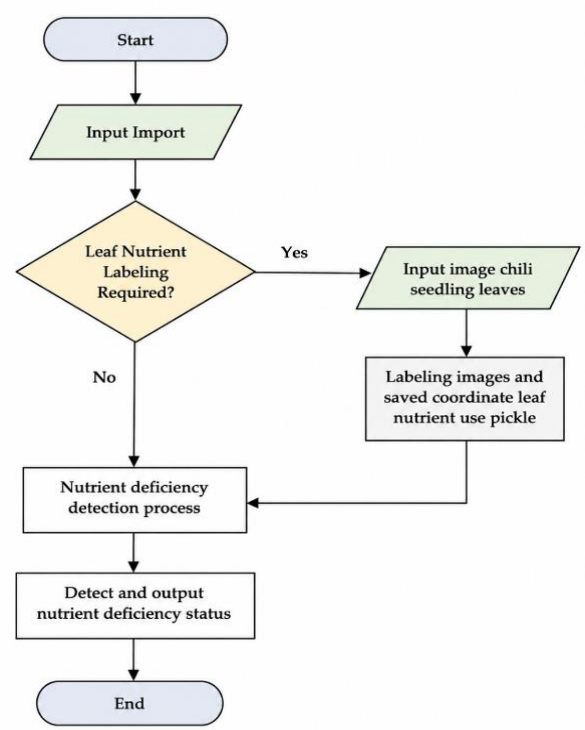
The workflow in Figure 1 summarizes the main stages of the study, including problem identification, literature review, dataset preparation, model training, and comparative evaluation. The dataset was developed to represent nutrient deficiency conditions in Bonita chili seedling leaves under controlled acquisition settings. Next, YOLOv5 to YOLOv11 were trained and evaluated using the same dataset partition and the same main training settings to support a fair comparison. Model performance was analyzed using precision, recall, mean Average Precision (mAP), and inference speed, followed by comparative interpretation of accuracy and computational efficiency across model versions. In this study, the task is defined consistently as single-object bounding-box detection. Each image contains one primary leaf object, and the model predicts the bounding box and the corresponding nutrient deficiency class in a single forward pass.

System Design

The workflow of the nutrient detection system design on chili leaves used in this study is presented in Figure 2.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263



Source: (Research Results, 2026)
Figure 2. Design System Workflow

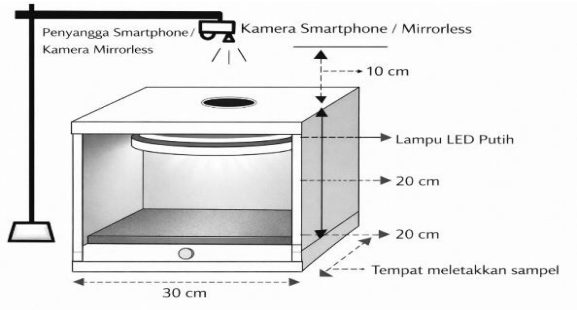
Figure 2 outlines the methodological workflow of the proposed nutrient deficiency detection system. The process begins with controlled image acquisition, continues with manual annotation of leaf objects and dataset preparation, and proceeds to model training and inference, where the system predicts the nutrient class, confidence score, and bounding-box location for each input image.

1. Image Acquisition
   Leaf images are collected under controlled lighting conditions to preserve symptom-relevant visual features, including color, texture, and chlorosis patterns.
2. Annotation and Label Verification
   Each image is reviewed to ensure that the leaf object has been annotated with a single bounding box and the appropriate nutrient class before being used for model development.
3. Dataset Preparation
   The annotated images are organized for preprocessing, augmentation of the training subset, and partitioning into training, validation, and testing data.
4. Model Training
   The labeled images are used to train YOLO models so that the network can learn the spatial and visual characteristics associated with each nutrient condition.
5. Inference
   During inference, the trained model detects the leaf object and predicts the corresponding nutrient category from the visual features in the input image.
6. Detection Output
   The final output consists of a predicted nutrient class, confidence score, and bounding-box visualization to support interpretation of nutrient conditions in Bonita chili seedlings.

Dataset

The dataset used in this study was compiled through controlled experimental stages to ensure reliable class labels. Biological experiments were conducted using a Completely Randomized Design (CRD) on Bonita chili seedlings under five nutrient treatments: complete nutrition (NPK), no nitrogen (−N), no phosphorus (−P), no potassium (−K), and no NPK. These treatments were designed to generate systematic variation in nutrient status as the basis for the image classes.

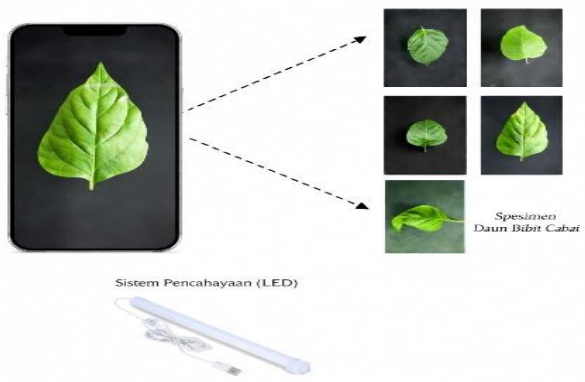
1. Image Acquisition
   To verify the actual nutrient conditions of the plants, laboratory analysis of the growing medium was performed before and after treatment, together with leaf tissue analysis. The results of these laboratory measurements were used as biological ground truth, so the assigned labels were not based solely on visual inspection. During annotation, the whole visible leaf was enclosed by a single bounding box and assigned one nutrient class. Images with unclear visual presentation or ambiguous labeling were reviewed during quality control to maintain annotation reliability [19], [35]. The image dataset in this study was obtained through an image acquisition system with controlled lighting, where each image represents one leaf as a unit of analysis. The scheme of the image acquisition system with controlled lighting and the configuration of the camera distance to the object is shown in Figure 3.



Source: (Research Results, 2026)
Figure 3. Image capture system scheme

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263
JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

And here is an example of an image of chili seedling leaves acquired using controlled lighting, shown in Figure 4.



Source: (Research Results, 2026)
Figure 4. Illustration of leaf image results

All images collected were then annotated using a bounding box approach to mark leaf areas as detection objects. Although each image generally contains a single leaf object, an object detection approach using bounding boxes was selected instead of a pure image classification approach. This decision was made to allow the model to explicitly learn the spatial localization of leaf regions that exhibit nutrient deficiency symptoms. In practical agricultural scenarios, leaf images may contain background elements, partial leaves, or multiple leaves captured in a single frame. The object detection framework enables the model to simultaneously localize and classify nutrient deficiency symptoms, improving robustness and enabling potential extension to more complex field conditions. This annotation approach is commonly used in object detection dataset development because it represents both object location and class information simultaneously [36].

The final dataset used in this study comprised 4,173 leaf images distributed across five nutrient classes: complete nutrition, nitrogen deficiency, phosphorus deficiency, potassium deficiency, and NPK deficiency. This final total included augmented training images. The original dataset contained 1,739 annotated images obtained from controlled experimental observations. To improve training diversity and generalization, the original dataset was first split into training, validation, and testing subsets using a 70%/20%/10% ratio. This produced 1,217 original training images, 348 validation images, and 174 testing images. Data augmentation was then applied only to the training subset, with three outputs generated per training image, resulting in 3,651 training images. Because the validation and testing subsets were kept in their original form, the final dataset size became 4,173 images. Across the five nutrient classes, the dataset remained relatively balanced before and after augmentation, so no single class dominated the training or evaluation subsets.

In addition, the dataset was collected using nine different camera devices with varying sensor resolutions and imaging characteristics. The use of multiple camera devices introduced variation in image resolution, color rendition, and sensor response. This variability increased dataset diversity and helped improve model robustness under different image acquisition conditions. The final data partition used in model development therefore consisted of 3,651 training images, 348 validation images, and 174 testing images. The split was performed before augmentation to avoid leakage between subsets, so no augmented derivative of a validation or testing image was included in the training data. The data were stratified to maintain a relatively balanced class distribution across all subsets [37], [35].

Source: (Research Results, 2026)
Figure 5. Collection of research dataset

Before model training, all images underwent preprocessing consisting of auto-orientation and resizing to 640 × 640 pixels using a fit-with-white-edges strategy. Data augmentation was applied only to the training subset to increase visual diversity and reduce overfitting. The

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263

augmentation settings included horizontal and vertical flipping, hue adjustment between −5° and +5°, saturation adjustment between −15% and +15%, brightness adjustment between −15% and +15%, blur up to 2 px, noise up to 0.04% of pixels, 90° clockwise and counter-clockwise rotation of the bounding box, and additional rotation between −15° and +15°. These settings followed common practices in YOLO-based object detection dataset development.

Experimental Setup

The research experiment setup integrates software and hardware to evaluate the performance of various versions of YOLO in detecting nutrient deficiencies in chili seedling leaves, with the following specifications:

Table 2. Hardware Specification

Component	Specification
Processor	Intel® Core™ Ultra 5 Processor 125H with AI Boost
Graphics	Intel® Arc™ Graphics (7 Xe Cores, 2.2 GHz)
RAM	16 GB LPDDR5X Dual Channel Memory
Storage	512 GB SSD NVMe Gen4
Display	14.0-inch 2.8K IPS Display (2880 × 1800), 16:10, sRGB 100%
Operating System	Windows 11 Home

Source: (Research Results, 2026)

Table 3. Software Specification

Software	Description
Operating System	Windows 11 64-bit
Programming Language	Python
Development Platform	Google Colab

Source: (Research Results, 2026)

Table 4. Equipment Specifications for data collection and laboratory testing:

Category	Specification
Camera Devices	Fujifilm X-20 Mirrorless Camera (12 MP), iPhone 13 Pro Max (12 MP), iPhone 11 (12 MP), Xiaomi 12T (108 MP), Redmi Note 11 Pro (108 MP), Redmi Note 8 (48 MP), Samsung Galaxy A73 (108 MP), Xiaomi 9 (48 MP), and Infinix X687 (64 MP)
Studio Equipment	Studio box
Planting Equipment	Growing media, seedling trays, N, P, and K fertilizers

Source: (Research Results, 2026)

Table 5. Laboratory Test Methods

Method	Purpose
Titrimetry	Determination of nitrogen (N) concentration
Gravimetry	Determination of phosphorus (P) and potassium (K) concentration
Spectrophotometry	Analysis of nutrient concentration
Atomic Absorption Spectroscopy (AAS)	Validation of nutrient concentration as biological ground truth

Source: (Research Results, 2026)

The dataset was collected through controlled image acquisition using nine camera devices, including mirrorless and smartphone cameras with different sensor resolutions. Images were captured inside a studio box to maintain consistent illumination and minimize background interference. This setting was used to preserve leaf color, texture, and symptom appearance while still introducing moderate variability from different sensors.

To guarantee the reliability of the nutritional labels, chili seedlings were cultivated under controlled experimental conditions using a Completely Randomized Design (CRD) with specific nutrient treatments, including complete nutrition (NPK), nitrogen deficiency (−N), phosphorus deficiency (−P), potassium deficiency (−K), and combined nutrient deficiency (−NPK). The nutritional conditions of the plants were validated through laboratory analysis of the growing media and leaf tissues using titrimetry, gravimetry, spectrophotometry, and Atomic Absorption Spectroscopy (AAS). These laboratory measurements served as the biological ground truth for dataset labeling.

All images were resized to 640 × 640 pixels and formatted for the YOLO training pipeline. The original 1,739 annotated images were first split into 1,217 training images, 348 validation images, and 174 testing images using a 70%/20%/10% ratio. Augmentation was applied only to the training subset, generating three outputs per original training image, so the final partition used for modeling consisted of 3,651 training images, 348 validation images, and 174 testing images. This split-before-augmentation protocol was used to prevent leakage between training and evaluation subsets.

For model training, all YOLO variants were trained under the same main experimental conditions to support a fair comparison. Each model was trained for 100 epochs using an input size of 640 × 640 pixels and a batch size of 16. Transfer learning with pretrained weights was used to accelerate convergence. The optimizer was AdamW with an initial learning rate of 0.001111, momentum of 0.9, and weight decay of 0.0005. A cosine learning-rate scheduler with a warm-up phase was used to stabilize the early stage of training. Although different YOLO versions rely on version-specific publicly available implementations, all models were evaluated under the same dataset partition, the same image size, the same main training configuration, and the same evaluation thresholds to reduce procedural variation across architectures.

The training process used standard detection loss components for bounding-box regression, classification, and localization refinement. In the implementation used in this study, these components included Complete Intersection over Union (CIoU) loss for bounding-box regression, Binary Cross Entropy (BCE) loss for classification, and Distribution Focal Loss (DFL) for localization refinement. Automatic Mixed Precision (AMP) was enabled to improve computational efficiency. During evaluation, the confidence threshold was set to 0.25 and the Intersection over Union (IoU) threshold to 0.5. A fixed random seed of 42 was applied to maintain consistent data shuffling and initialization. Each YOLO configuration was trained once using the same data split and the same main hyperparameter settings. All FPS measurements were obtained using the same input resolution (640 × 640) and the same runtime environment so that the reported inference speed reflected comparable benchmarking conditions across models.

Performance Metrics

To evaluate the detection performance of the YOLO models, several standard object detection metrics were employed, including Precision, Recall, mean Average Precision (mAP), and inference speed measured in frames per second (FPS). Precision represents the proportion of correctly predicted positive detections, while Recall measures the ability of the model to detect all relevant objects in the dataset. These metrics are derived from the number of True Positive (TP), False Positive (FP), and False Negative (FN) predictions.

The primary evaluation metric used in this study is mean Average Precision (mAP). Two variants of mAP are reported: mAP@0.5 and mAP@0.5:0.95. The mAP@0.5 metric evaluates detection performance when the Intersection over Union (IoU) threshold is set to 0.5, indicating whether predicted bounding boxes sufficiently overlap with the ground truth annotations. Meanwhile, mAP@0.5:0.95 represents the mean average precision calculated across multiple IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. This metric provides a more comprehensive evaluation because it considers both object localization accuracy and detection robustness across different IoU levels.

In addition to detection performance, computational efficiency was also evaluated using inference speed measured in frames per second (FPS). All evaluation metrics were computed using the standard YOLO evaluation protocol during the validation process to ensure fair and consistent comparison across all tested YOLO architectures.

Table 6. Measurement Scenario

Measurement	Quantity
True Positive (TP)	A chili seedling leaf exhibiting a specific nutrient deficiency condition is correctly localized by a bounding box and classified into the corresponding nutrient class with sufficient IoU overlap.
False Positive (FP)	A chili seedling leaf is incorrectly detected or classified into a nutrient deficiency class that does not correspond to its actual nutritional condition.
False Negative (FN)	A chili seedling leaf with a specific nutrient deficiency condition is not detected by the model or is misclassified into an incorrect nutrient class.

Source: (Research Results, 2026)

Note: In multi-class object detection, True Negative (TN) is not explicitly emphasized, because the evaluation focuses on object localization and correct class assignment. This evaluation metric provides a comprehensive overview of the YOLO model's performance in identifying and classifying the nutritional status of chili seedling leaves, taking into account object detection success, nutritional class accuracy, and detection errors, as formulated in Equation (1):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

In addition, the detection speed of each YOLO version is evaluated through FPS according to Equation (2):

$$FPS = \frac{N}{T} \quad (2)$$

Description:
N: total number of frames processed in one detection session.
T: total time required to process N frames, measured in seconds.

The selection of detection performance metrics together with inference speed reflects their relevance for evaluating nutrient deficiency detection systems intended for real-time agricultural applications. Precision, recall, and mAP describe how reliably the model localizes and classifies nutrient deficiency symptoms, whereas inference speed reflects operational efficiency during deployment [38]. Inference speed is especially important in practical monitoring

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263

scenarios because faster image processing supports more timely interpretation of crop conditions and more responsive decision-making in nutrient management [39].

Taken together, these metrics provide a broader basis for judging whether a model is both accurate and practically deployable in precision agriculture.

RESULTS AND DISCUSSION

The experimental results are presented in the following subsections.

Image Detection Results

The image detection results are shown in Figures 6:

(a) Bonita chili leaf image

(b) Detection result of complete nutrient condition on chili leaves

(c) Detection result of phosphorus deficiency on chili leaves

Source: (Research Results, 2026)

Figure 6. Examples of chili leaf images and detection results: (a) Bonita chili leaf image, (b) detection result of complete nutrient condition, and (c) detection result of phosphorus deficiency.

Figure 6 illustrates representative detection outputs for Bonita chili seedling leaves under different nutrient conditions. The figure includes the original leaf image and two example predictions corresponding to complete nutrition and phosphorus deficiency. The original image shows a clearly visible leaf separated from the background, providing a representative input for the detection model. In the complete-nutrition example, the model generates a bounding box that covers the leaf area accurately and assigns the class with a confidence score of 0.90.

The predicted result is consistent with the visual appearance of a healthy leaf, including uniform green coloration and intact leaf structure. In the phosphorus-deficiency example, the model also localizes the leaf accurately and predicts the class with a confidence score of 0.97. The predicted class is consistent with visible deficiency symptoms, such as paler coloration and reduced visual prominence of the leaf surface and vein structure. These examples indicate that the model can localize and classify leaf nutrient conditions under the controlled imaging setup used in this study. They also illustrate that visually distinctive nutrient patterns can be detected with high confidence.

Data preparation and Data Augmentation

Data preparation and augmentation were performed to improve the robustness and generalization ability of the YOLO-based model in detecting nutrient deficiencies (N, P, K, and NPK) in Bonita chili seedling leaves. Given that nutrient deficiency symptoms are highly dependent on subtle visual cues particularly leaf color, texture, and vein patterns each preprocessing and augmentation strategy was carefully selected to avoid altering critical characteristics that could lead to object detection errors.

Data preparation

The data preparation stage aims to standardize the input images while preserving the original morphology and color distribution of chili leaves. First, an automatic orientation process is applied to correct inconsistencies caused by camera metadata. This step ensures that all images are consistently aligned before model training, preventing orientation bias during feature learning.

Next, all images are resized to 640 × 640 pixels using a “fit with white edges” strategy. This approach is chosen to meet YOLO's square input requirements while avoiding geometric distortion of leaf shapes. Unlike cropping-based resizing, adding white edges preserves the intact leaf structure and maintains accurate bounding box alignment. This method is particularly suitable for close-up leaf datasets, where even minor shape deformations can affect detection accuracy.

The dataset consisted of five classes: complete nutrition, potassium deficiency, phosphorus deficiency, NPK deficiency, and nitrogen deficiency. Before augmentation, the original dataset of 1,739 images was split into 1,217 training images, 348 validation images, and 174 testing images. The validation subset was used to monitor model performance during training, whereas the testing subset was reserved for final evaluation.

The chili seedling leaf image dataset has been annotated using the YOLO annotation format, where each object is described in a text file (.txt) that includes class information and bounding box coordinates in the following format: (class_id), (x_center), (y_center), (width), (height). All coordinates are normalized based on image dimensions, so that the values are in the range of 0–1. Annotations are done manually using image labeling software, then validated by agronomists to ensure that the visual symptoms on the leaves match the defined nutrient deficiency categories.

In this study, labeling rules were determined based on the visual characteristics of nutritional deficiency symptoms in chili seedling leaves, including leaf discoloration, chlorosis patterns, necrosis, and modifications in leaf tissue shape. Each leaf object displaying specific symptoms was assigned a label corresponding to the detected nutritional category. To overcome the limited size of the original dataset, a 3× augmentation strategy was applied in Roboflow only to the training subset. In this setting, each original training image generated three outputs in total, producing 3,651 training images after augmentation. Because the validation and testing subsets were not augmented, the final dataset used in the study contained 4,173 images. This strategy preserved evaluation independence while improving training diversity.

Improvements in Boundary Box Levels

Augmentation was designed to improve spatial robustness and orientation invariance while preserving symptom characteristics. Horizontal and vertical flipping, rotation between −15° and +15°, 90° clockwise and counter-clockwise rotation, hue adjustment between −5° and +5°, saturation and brightness adjustment between −15% and +15%, blur up to 2 px, and noise up to 0.04% of pixels were applied to the training images.

Overall Impact

Overall, the data preparation and augmentation strategy is designed to maximize dataset diversity while preserving the intrinsic visual characteristics of nutrient deficiency symptoms. By maintaining color integrity, avoiding geometric distortion, and introducing realistic variations, the augmented dataset allows the YOLO model to learn more general and robust feature representations. This configuration effectively reduces overfitting, improves stability during training, and enhances detection performance across various shooting conditions, making it highly suitable for automated nutrient detection in chili seedling cultivation systems.

Comparison Table Results

The following is a comparison table of YOLOv5 to YOLOv11

Table 7. comparison results

Version	mAP@0.5	FLOPs (GFLOPs)	FPS (inference)	Parameter (M)	Model Size (MB)
YOLOv5s	0.984	15.8	~170 FPS	7.02 M	~28 MB
YOLOv6	0.985	45.17	~88 FPS	18.50 M	~74 MB
YOLOv7	0.871	105.2	~68 FPS	37.21 M	~149 MB
YOLOv8s	0.988	28.7	~131 FPS	11.13 M	~44 MB
YOLOv9 (gelan-c)	0.982	102.5	~30 FPS	25.41 M	~102 MB
YOLOv10s	0.991	21.4	~67.6 FPS	7.22 M	~29 MB
YOLOv11s	0.991	21.3	~82 FPS	9.41 M	~38 MB

Source: (Research Results, 2026)

Table 7 summarizes the comparative performance of YOLOv5 to YOLOv11 in terms of detection accuracy, computational complexity, inference speed, parameter count, and model size. Together, these metrics describe the trade-off between predictive performance and deployment cost. The models achieved generally high mAP@0.5 values, ranging from 0.871 to 0.991. YOLOv10s and YOLOv11s obtained the highest mAP@0.5 in this single-run comparison, while YOLOv8s, YOLOv5s, and YOLOv6 also showed competitive results. By contrast, YOLOv7 produced the lowest mAP@0.5, indicating that higher architectural complexity did not translate into better performance on this dataset. Clear differences were also observed in computational complexity. YOLOv7 and YOLOv9 (GELAN-C) required more than 100 GFLOPs, whereas YOLOv10s and YOLOv11s operated at approximately 21 GFLOPs while maintaining high detection performance. A similar pattern appeared in parameter count and model size. YOLOv5s, YOLOv10s, and YOLOv11s were comparatively

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263

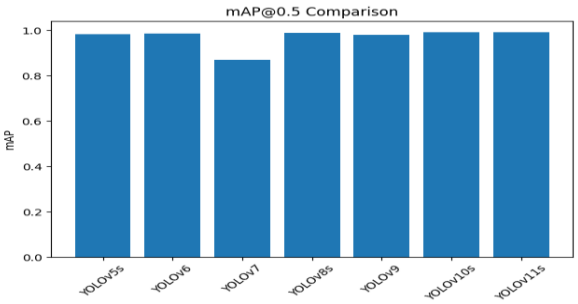
compact, whereas YOLOv7 and YOLOv9 required substantially larger parameter budgets and storage capacity.

These findings indicate that model selection should consider both accuracy and efficiency. Within the present experiment, YOLOv10s and YOLOv11s provided the strongest overall balance among the evaluated models; however, the observed differences among the top-performing models were small and should be interpreted as comparative tendencies rather than definitive superiority. Class-level inspection indicated that NPK deficiency was the most challenging category to distinguish, especially when combined symptoms produced diffuse visual transitions between healthy and deficient tissue. This difficulty likely arose from overlapping color changes, mixed symptom patterns, and less distinct visual boundaries compared with single-element deficiencies.

YOLOv5-v11 Comparison Chart Results

This section presents a graphical comparative analysis between YOLOv5 and YOLOv11 based on key performance indicators, including mAP@0.5, computational complexity (FLOPs), inference speed (FPS), number of model parameters, and estimated model size. This graphical visualization provides more intuitive insights into the performance trade-offs between YOLO variants when applied to the detection of nutrient deficiencies in Bonita chili seedlings using the npk_v15 dataset.

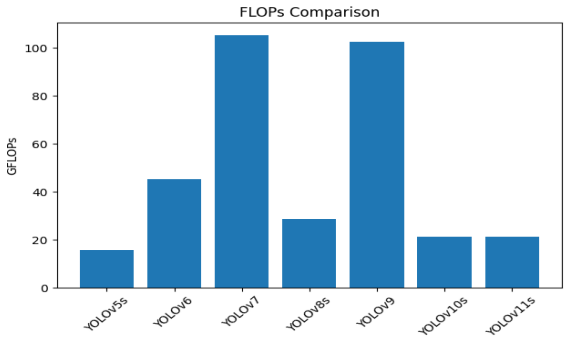
mAP@0.5 Comparison



Source: (Research Results, 2026)
Figure 9. mAP@0.5 comparison chart image

Figure 9 compares mAP@0.5 across the evaluated models. YOLOv10s and YOLOv11s achieved the highest values, while YOLOv8s also remained highly competitive. Because the gap among the top models was small, the figure is better interpreted as showing relative ranking under the present experimental setting rather than a large practical performance separation.

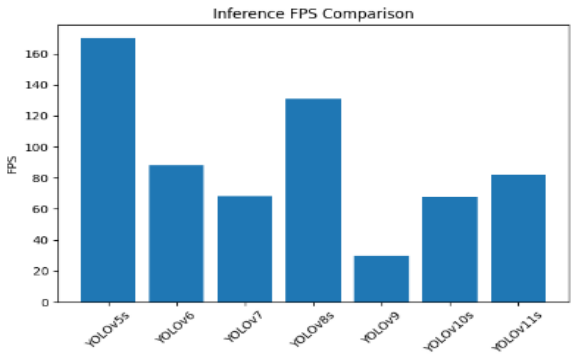
FLOPs Comparison



Source: (Research Results, 2026)
Figure 10. Comparison of computational complexity (GFLOPs) across YOLO models

Figure 10 shows that computational cost varied substantially across architectures. YOLOv7 and YOLOv9 required the largest GFLOPs, whereas YOLOv5 was the lightest model and YOLOv10–YOLOv11 remained comparatively efficient. This pattern suggests that architectural refinement can improve the accuracy–efficiency trade-off without simply increasing computational burden.

FPS Inference Comparison



Source: (Research Results, 2026)
Figure 11. Comparison of inference speed (FPS) across YOLO models

Figure 11 shows that YOLOv5 achieved the highest FPS, whereas YOLOv9 produced the lowest. YOLOv10 and YOLOv11 occupied an intermediate position, indicating that their higher accuracy was achieved with a moderate speed cost rather than maximum throughput.

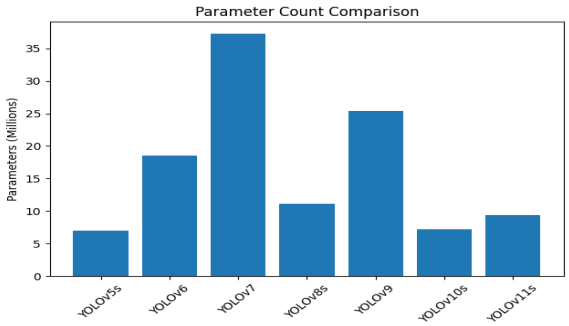
Comparison of Parameter Quantity

Figure 12 indicates that YOLOv7 had the largest parameter count, while YOLOv5 and YOLOv10 remained relatively compact. This result is consistent with the broader observation that

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

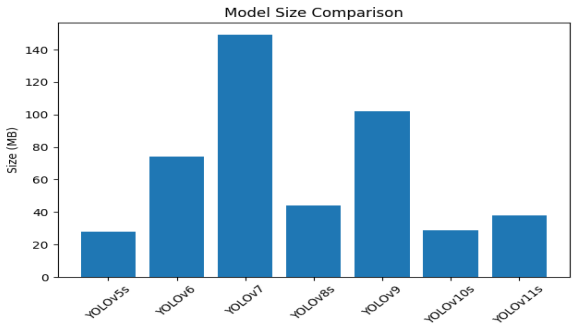
parameter growth alone did not guarantee superior detection performance on the nutrient deficiency dataset.



Source: (Research Results, 2026)
Figure 12. Comparison of trainable parameter counts across YOLO models

Comparison of Estimated Model Sizes

Figure 13 shows that model size followed a pattern similar to parameter count. Large models such as YOLOv7 and YOLOv9 may be less suitable for memory-limited deployment, whereas YOLOv5, YOLOv10, and YOLOv11 provide a more practical balance between model size and performance.

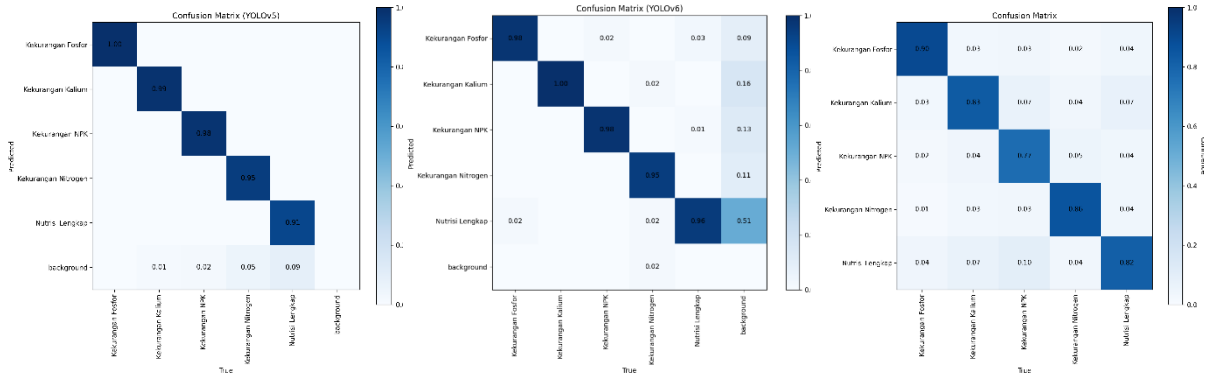


Source: (Research Results, 2026)
Figure 13. Comparison of model size (MB) across YOLO architectures

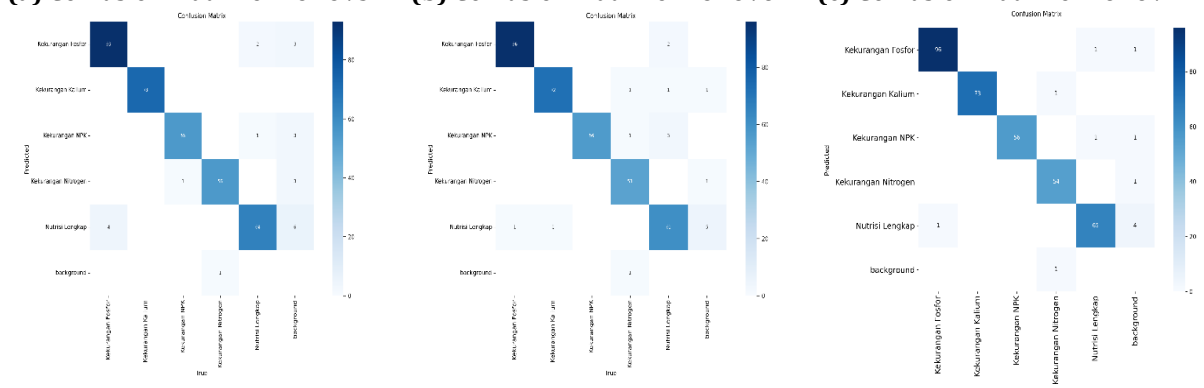
Overall Graphic Interpretation

The combined graphical analysis indicates that no single model simultaneously maximized every criterion. YOLOv10s and YOLOv11s offered strong accuracy with moderate computational demand, YOLOv5s emphasized inference efficiency, and YOLOv8s remained competitive across multiple metrics. This pattern is relevant for deployment decisions because the preferred model depends on whether the priority is absolute accuracy, model compactness, or inference speed.

Confusion Matrix



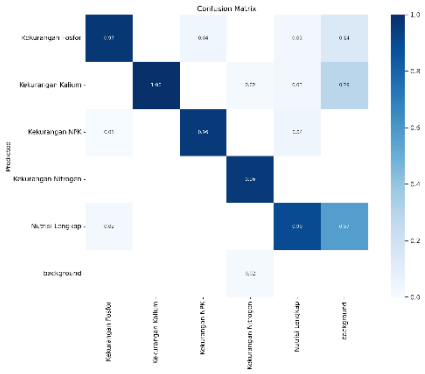
(a) Confusion Matrix of YOLOv5 **(b)** Confusion Matrix of YOLOv6 **(c)** Confusion Matrix of YOLOv7



(d) Confusion Matrix of YOLOv8 **(e)** Confusion Matrix of YOLOv9 **(f)** Confusion Matrix of YOLOv10

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.8263



(g) Confusion Matrix of YOLOv11

Source: (Research Results, 2026)

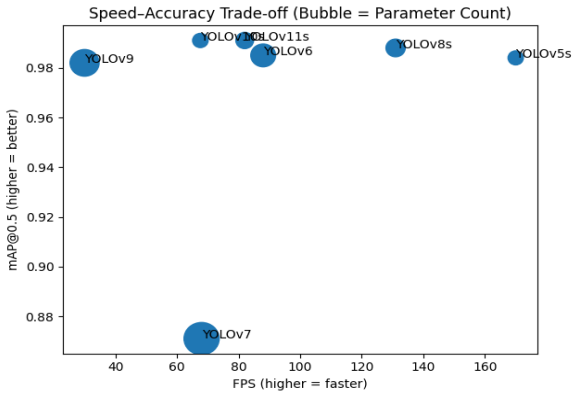
Figure 14. Confusion matrices of the evaluated YOLO models: (a) YOLOv5, (b) YOLOv6, (c) YOLOv7, (d) YOLOv8, (e) YOLOv9, (f) YOLOv10, and (g) YOLOv11.

The confusion matrices show that most models achieved strong class discrimination, as reflected by the dominance of diagonal values. However, the distribution of errors was not uniform across classes. In YOLOv5 and YOLOv6, most errors were concentrated in the Complete Nutrition class, which was occasionally confused with background or mild deficiency patterns. YOLOv7 displayed a more dispersed error distribution than the other models, particularly for NPK deficiency and potassium deficiency, indicating greater difficulty in separating visually similar classes. Newer models such as YOLOv8, YOLOv9, YOLOv10, and YOLOv11 showed more stable diagonal patterns, although minor confusion between NPK deficiency, nitrogen deficiency, and Complete Nutrition remained.

Across the evaluated models, NPK deficiency and Complete Nutrition were the most error-prone classes. This pattern is plausible because combined nutrient deficiency can produce overlapping symptoms, while healthy leaves may still resemble mildly deficient leaves in early stages. Overall, the confusion matrices suggest that YOLOv10, YOLOv11, and YOLOv8 achieved the most stable class separation in this experiment. Nevertheless, these results should be interpreted together with the small numerical differences in mAP@0.5 and the absence of repeated-run uncertainty estimates.

Speed Accuracy Results

This section analyzes the trade-off between inference speed and detection accuracy of YOLOv5–YOLOv11 using a graphical representation of speed–accuracy in the form of a bubble chart. This graph displays three main aspects simultaneously: inference speed (FPS) on the horizontal axis, detection mAP@0.5 on the vertical axis, and model complexity represented by bubble size, where larger bubbles indicate a higher number of model parameters.



Source: (Research Results, 2026)

Figure 15. Speed accuracy results image

The speed–accuracy graph illustrates the trade-off between detection performance and inference speed across the evaluated YOLO variants. Models in the upper-right area of the plot combine relatively high mAP@0.5 with relatively high FPS, whereas models in the lower-left area reflect a less favorable balance. YOLOv8s occupies a competitive position in this trade-off space because it combines high accuracy with high throughput. Although its mAP@0.5 is slightly lower than that of YOLOv10s and YOLOv11s, its overall balance remains attractive for practical deployment.

YOLOv9 (GELAN-C) provides strong detection accuracy but at the cost of considerably lower inference speed and higher model complexity, indicating that its performance advantage is limited by deployment efficiency. YOLOv7 occupies a less favorable region of the plot, with lower mAP@0.5 and relatively high

computational burden. This result reinforces the observation that larger model capacity was not advantageous for the present dataset. In contrast, YOLOv5s achieved the highest FPS, showing that lightweight architectures can still remain useful when real-time throughput is the main consideration. Overall, the speed–accuracy trade-off suggests that YOLOv10s and YOLOv11s were the strongest performers in this single experimental setting, while YOLOv8s and YOLOv5s remain practical alternatives depending on deployment priorities. These differences should be interpreted as relative tendencies rather than definitive rankings.

CONCLUSION

This study compared YOLOv5 to YOLOv11 for detecting nutrient deficiencies in Bonita chili seedling leaves across nitrogen deficiency, phosphorus deficiency, potassium deficiency, NPK deficiency, and complete nutrition classes. All evaluated models achieved high detection performance, but they differed in computational efficiency and class-level error patterns. In this experimental setting, YOLOv10s and YOLOv11s achieved the highest mAP@0.5, while YOLOv8s also remained highly competitive with a favorable balance between accuracy, speed, and model size. Class-level analysis indicated that single nutrient deficiencies were generally easier to recognize than combined deficiency patterns. NPK deficiency remained the most challenging class because its symptoms overlapped visually with several single-deficiency patterns, while the Complete Nutrition class was sometimes confused with mild deficiency cases. Overall, the results suggest that recent YOLO variants offer promising performance for nutrient deficiency detection in controlled chili seedling imagery. However, because the present comparison is based on one dataset partition and single training runs, the findings should be interpreted as empirical comparative evidence within this study rather than as universal rankings for all agricultural detection tasks.

ACKNOWLEDGEMENT

This research was funded by the Ministry of Education, Culture, Research and Technology, Republic of Indonesia (Grant No: 02420/BPPT/BPI.06/9/2023). The authors gratefully acknowledge the support of the Indonesian Education Scholarship (Beasiswa Pendidikan Indonesia/ BPI), the Center for Higher Education Funding and Assessment (Pelayanan Pembiayaan dan Asesmen Pendidikan Tinggi/ PPAPT), and the Endowment Fund for Education Agency (Lembaga Pengelola Dana Pendidikan/LPDP), Ministry of Finance, Republic of Indonesia.

REFERENCE

[1] M. T. Sundari, Darsono, J. Sutrisno, and E. Antriyandarti, “Analysis of chili farming in Indonesia,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 905, no. 1, 2021, doi: 10.1088/1755-1315/905/1/012046.

[2] C. M. Badgujar, A. Poulose, and H. Gan, “Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review,” *Comput. Electron. Agric.*, vol. 223, pp. 1–30, 2024, doi: 10.1016/j.compag.2024.109090.

[3] Nice Anjelin Gulo, Ayu Indah Purnama Mendrofa, Berliana Vivi Lestari Lase, Cynthia Florentina Mendrofa, Iman Viktor Telaumbanua, and Irwan Saham Laia, “Kurangnya Unsur Hara pada Tanaman Cabai Merah serta Pemeliharaannya,” *Tumbuh. Publ. Ilmu Sosiol. Pertan. Dan Ilmu Kehutan.*, vol. 1, no. 3, pp. 13–20, 2024, doi: 10.62951/tumbuhan.v1i3.112.

[4] J. Permatasari, A. U. Edmund, E. Sunardi, B. M. Laili, and M. S. Putri, “Evaluasi Kinerja YOLOv11 pada Deteksi Penyakit Tanaman Cabai Studi Komparatif dengan YOLOv8, YOLOv5, dan SSD,” *J. Teknol.*, vol. Vol. 25, N, 2025, [Online]. Available: https://e-jurnal.pnl.ac.id/teknologi/article/view/8400

[5] S. S. A. Begum and H. Syed, “GSAtt-CMNetV3: Pepper Leaf Disease Classification Using Osprey Optimization,” *IEEE Access*, vol. 12, no. January, pp. 32493–32506, 2024, doi: 10.1109/ACCESS.2024.3358833.

[6] D. T. Nguyen, T. D. Bui, T. M. Ngo, and U. Q. Ngo, “Improving YOLO-Based Plant Disease Detection Using αSILU: A Novel Activation Function for Smart Agriculture,” *AgriEngineering*, vol. 7, no. 9, pp. 1–25, 2025, doi: 10.3390/agriengineering7090271.

[7] R. Kaur *et al.*, “YOLO-LeafNet: a robust deep learning framework for multispecies plant disease detection with data augmentation,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–23, 2025, doi: 10.1038/s41598-025-14021-z.

[8] M. Shoaib *et al.*, “A deep learning-based model for plant lesion segmentation,

subtype identification, and survival probability estimation," 2022, *frontiersin.org*. doi: 10.3389/fpls.2022.1095547.

[9] M. Al-husaini, A. R. Raharja, V. Hafizh, C. Putra, and H. Hen, "Journal of Computer Networks , Architecture and High Performance Computing Enhanced Plant Disease Detection Using Computer Vision YOLOv11 : Pre- Trained Neural Network Model Application Journal of Computer Networks , Architecture and High Performance Comp," vol. 7, no. 1, pp. 82–95, 2025, doi: 10.30871/jaic.v9i3.9213.

[10] J. Sikati and J. C. Nouaze, "YOLO-NPK: A Lightweight Deep Network for Lettuce Nutrient Deficiency Classification Based on Improved YOLOv8 Nano †," 2023, *mdpi.com*. doi: 10.3390/ecsa-10-16256.

[11] F. Aldi, I. Nozomi, M. Hafizh, and T. Novita, "Comparative Analysis of YOLOv11 with Previous YOLO in the Detection of Human Bone Fractures," vol. 7, no. 3, pp. 777–790, 2025, doi: 10.47709/cnahpc.v7i3.6051.

[12] M. L. Ali, "The YOLO Framework : A Comprehensive Review of Evolution , Applications , and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.

[13] R. Sapkota and M. Karkee, "Comparing YOLOv11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment," arXiv preprint arXiv:2410.19869, 2025. doi: 10.48550/arXiv.2410.19869.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[15] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, pp. 1–18, 2023, doi: 10.1007/s44196-023-00302-w.

[16] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[17] M. Fathurrahman, A. Nugroho, A. Zein, and A. Wafi, "COMPARATIVE STUDY OF YOLO VERSIONS FOR DETECTING VACANT CAR PARKING SPACES," vol. 10, no. 4, pp. 833–848, 2025, doi: 10.33480/jitk.v10i4.6236.

[18] M. Hou, W. Hao, Y. Dong, and Y. Ji, "A detection method for the ridge beast based on improved YOLOv3 algorithm," *Herit. Sci.*, pp. 1–13, 2023, doi: 10.1186/s40494-023-00995-4.

[19] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, 2023, doi: 10.1109/JPROC.2023.3238524.

[20] M. A. R. Alif and M. Hussain, "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain," arXiv preprint arXiv:2406.10139, 2024. doi: 10.48550/arXiv.2406.10139.

[21] R. Khanam and M. Hussain, "What is YOLOv5: A deep look into the internal features of the popular object detector," arXiv preprint arXiv:2407.20892, 2024. doi: 10.48550/arXiv.2407.20892.

[22] Y. Lyu, "A Review of YOLO-Based Target Detection Methods," vol. 0, pp. 59–66, 2024, doi: 10.54254/2755-2721/80/2024CH0060.

[23] J. E. Gallagher and E. J. Oughton, "Surveying You Only Look Once (Yolo) Multispectral Object Detection Advancements , Applications And Challenges," *IEEE Access*, vol. PP, p. 1, 2025, doi: 10.1109/ACCESS.2025.3526458.

[24] M. Yang, X. Tong, and H. Chen, "Detection of Small Lesions on Grape Leaves Based on Improved YOLOv7," *Electronics*, vol. 13. no 2, 2024.

[25] M. Yaseen, "What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," arXiv preprint arXiv:2409.07813, 2024. doi: 10.48550/arXiv.2409.07813.

[26] X. Wang, C. Zhang, Z. Qiang, C. Liu, X. Wei, and F. Cheng, "A Coffee Plant Counting Method Based on Dual-Channel NMS and YOLOv9 Leveraging UAV Multispectral Imaging," Remote Sensing, vol. 16, no. 20, p. 3810, 2024. doi: 10.3390/rs16203810.

[27] Y. Wang, Q. Rong, and C. Hu, "Ripe Tomato Detection Algorithm Based on Improved YOLOv9," *Appl. Sci.*, vol. 14, no. 9, p. 3921,

2024, doi: 10.3390/app14093921.
[28] G. Jocher, "Ultralytics YOLO Docs." [Online]. Available: https://docs.ultralytics.com/models/. [Accessed: May 22, 2026].
[29] A. S. Geetha, M. Al, R. Alif, M. Hussain, and P. Allen, "Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection : Performance Metrics and Model Efficacy," pp. 1364–1382, 2024, doi: 10.3390/vehicles6030065.
[30] E. Nhancements, "YOLOV11 AN OVERVIEW OF THE KEY ARCHITECTURAL ENHANCEMENTS," vol. 2024, pp. 1–9, 2024.
[31] L. He, Y. Zhou, and L. Liu, "Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites," 2024.
[32] C. V Jul, C. Science, and H. Hd, "YOLOV5, YOLOV8 AND YOLOV10: THE GO - TO DETECTORS FOR REAL - TIME VISION," pp. 1–12, 2024.
[33] D. C. Rodríguez-Lira, D. M. Córdova-Esparza, J. M. Álvarez-Alvarado, J. A. Romero-González, J. Terven, and J. Rodríguez-Reséndiz, "Comparative Analysis of YOLO Models for Bean Leaf Disease Detection in Natural Environments," *AgriEngineering*, vol. 6, no. 4, pp. 4585–4603, 2024, doi: 10.3390/agriengineering6040262.
[34] N. Chitraningrum *et al.*, "Comparison Study of Corn Leaf Disease Detection based on Deep Learning YOLO-v5 and YOLO-v8," vol. 56, no. 1, pp. 61–70, 2024, doi: 10.5614/j.eng.technol.sci.2024.56.1.5.
[35] Kaur, J., Singh, W. "Tools, techniques, datasets and application areas for object detection in an image: a review." *Multimed Tools Appl,* vol. 81, pp. 38297–38351, 2022. https://doi.org/10.1007/s11042-022-13153-y.
[36] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010, doi: 10.1007/s11263-009-0275-4.
[37] J. Kaur, "Tools, techniques, datasets and application areas for object detection in an image: A review," *Multimed. Tools Appl.*, vol. 81, no. 27, pp. 38297–38351, 2022, doi: 10.1007/s11042-022-13491-9.
[38] J. Mijalkovic and A. Spognardi, "Reducing the False Negative Rate in Deep Learning Based Network Intrusion Detection Systems," *Algorithms*, vol. 15, no. 8, 2022, doi: 10.3390/a15080258.
[39] C. Mwitta and G. C. Rains, "Evaluation of Inference Performance of Deep Learning Models for Real-Time Weed Detection in an Embedded Computer," Sensors, vol. 24, no. 2, pp. 514, 2024, doi: 10.3390/s24020514.