Rapid Detection of Soybean Nutrient Deficiencies Using YOLOv8s: Advancing Precision Agriculture

Minsoo Jeong
Department of Applied Biosciences, Kyungpook National University
Sihyun Park
Department of Integrative Biotechnology, Kyungpook National University
Sook-Min Kwon
Department of Applied Biosciences, Kyungpook National University
KyeongMo Lim
Department of Applied Biosciences, Kyungpook National University
Da-Ryung Jung
Department of Applied Biosciences, Kyungpook National University
Hong-Seok Lee
Department of Southern Area Crop Science, NICS, RDA
Hei Jung Kim
NGS Core Facility, Kyungpook National University
Jae-Ho Shin
jhshin@knu.ac.kr

Department of Applied Biosciences, Kyungpook National University

---

Article

Keywords: Artificial intelligence, nutrient deficiency, soybeans, smart farming, YOLOv8

Posted Date: November 2nd, 2024

DOI: https://doi.org/10.21203/rs.3.rs-5300628/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on April 21st, 2025. See the published version at https://doi.org/10.1038/s41598-024-83295-6.

Rapid Detection of Soybean Nutrient Deficiencies Using YOLOv8s: Advancing Precision Agriculture

Minsoo Jeong[1], Sihyun Park[2], Sook-Min Kwon[2], KyeongMo Lim[1], Da-Ryung Jung[1], Hong-Seok Lee[3], Hei Jung Kim[4], Jae-Ho Shin[1,2,4]*

[1]*Department of Applied Biosciences, Kyungpook National University, Daegu 41566, Republic of Korea*

[2]*Department of Integrative Biotechnology, Kyungpook National University, Daegu 41566, Republic of Korea*

[3]*Department of Southern Area Crop Science, NICS, RDA, Milyang 50424, Republic of Korea*

[4]*NGS Core Facility, Kyungpook National University, Daegu 41566, Republic of Korea*

Corresponding author email: jhshin@knu.ac.kr

Abstract

Early detection of nutrient deficiencies is crucial for optimizing crop yields and ensuring sustainable agricultural practices. This study presents a novel application of the YOLOv8s object detection model for identifying nitrogen, phosphorus, and potassium deficiencies in soybean plants. Employing a unique dataset from a long-term nutrient-deficient field maintained for over 40 years, we trained and evaluated the model on 6,020 red, green, and blue images of soybean leaves exhibiting nutrient stress conditions. The YOLOv8s model achieved exceptional performance, with a mean average precision (mAP@0.5) of 99.18% during training and 98.51% for validation. Precision rates for individual nutrient deficiencies ranged from 90.03% to 96.54%, with highly accurate potassium deficiency detection. The model demonstrated robust generalization across diverse field conditions, processing images in 3.46 ms each, making it suitable for real-time applications. This research significantly advances the field of precision agriculture by providing a fast, accurate, and scalable method for detecting early nutrient deficiency in soybean crops, potentially revolutionizing fertilizer management practices and contributing to more sustainable farming systems.

Keywords

Artificial intelligence, nutrient deficiency, soybeans, smart farming, YOLOv8

1. Introduction

Agricultural productivity fundamentally depends on proper nutrient management[1,2]. Adequate nutrient supply is crucial for promoting plant growth and for maximizing crop yield. In various crops, such as soybeans, deficiencies in specific nutrients can significantly reduce productivity[3]. However, early detection of nutrient deficiencies in agricultural settings remains challenging[4,5]. These deficiencies often manifest through subtle changes that are difficult to identify with the naked eye and, if unaddressed, can lead to significant reductions in yield and crop quality[6,7].

Traditional agricultural methods offer several strategies to mitigate nutrient deficiencies, but technology that can quickly and accurately diagnose these problems is still limited[8,9]. This problem is acute in some countries, such as South Korea, where agriculture is predominantly individually managed, making it more difficult to balance nutrient deficiencies and over-application. The challenge lies in developing a method that can provide rapid, accurate, and early detection of nutrient deficiencies for timely interventions.

Recent advancements in deep learning technology offer promising solutions for automating and optimizing agricultural processes; nevertheless, their applications in nutrient deficiency detection remain limited. Deep learning models have demonstrated remarkable success in diagnosing plant diseases early, allowing farmers to respond quickly and effectively. However, despite this proven effectiveness for disease detection, fewer than 10 studies are available on using deep learning for nutrient deficiency diagnosis[4.] For instance, Ghosal et al.[10] achieved 90.3% accuracy in detecting various soybean conditions including potassium deficiency using a Deep-Convolutional neural network (CNN) model, but their study was limited to a small set of stress conditions. Similarly, Tran et al.[11] used CNNs for macronutrient deficiency classification in tomatoes, achieving 91% accuracy with an ensemble averaging approach, but their model's performance varied significantly depending on the specific nutrient deficiency. In the case of rice, Xu et al.[12] utilized a Densely Connected Convolutional Networks (DenseNet)121 model, achieving 97.44% accuracy on the test set for various nutrient deficiencies, but the model's performance in real-world field conditions remains uncertain.

This scarcity of research presents a significant opportunity for innovation. This study aims to bridge this gap by developing a deep learning-based system for the early detection of nutrient deficiencies in soybean crops. Thus, we propose a novel approach that combines state-of-the-art object detection algorithms with a unique long-term nutrient deficiency dataset, addressing the limitations of previous studies and pushing the boundaries of precision agriculture. To achieve this, we propose the application of You Only Look Once Version 8 (YOLOv8), a state-of-the-art object detection algorithm, to this agricultural challenge[13,14]. The YOLOv8 model offers significant advantages in real-time object detection, making it particularly suitable for rapid nutrient deficiency identification in field conditions. However, the effectiveness of such advanced algorithms heavily depends on the quality and representativeness of the training data. This is where our research offers a unique contribution.

We utilize a dataset derived from a field at the Gyeongsangbuk-do Agricultural Research and Extension Services, maintained in a nutrient-deficient state for over 40 years. This unprecedented long-term experimental setup provides a rare opportunity to study the effects of chronic nutrient stress on soybean crops, offering insights that short-term or greenhouse studies cannot capture. By applying this unique dataset, we aim to develop a model that can accurately detect nutrient deficiencies as they naturally occur in long-term agricultural settings, significantly advancing the field of precision agriculture.

The objectives of this research are as follows: 1) to assess changes in plant health and growth under nutrient-deficient conditions, 2) to capture and analyze phenotypic changes using Red-Green-Blue (RGB) image data, and 3) to develop and implement a detection model using YOLOv8. We hypothesize that specific nutrient deficiencies result in distinct, detectable changes in soybean leaf phenotypes, which can be effectively captured by RGB images and accurately identified using the YOLOv8-based model.

We expect our model, trained on the long-term nutrient deficiency dataset, to achieve detection accuracy of over 95% for each deficiency type and to outperform existing models in early detection and classification accuracy, particularly in field conditions. Our approach utilizes RGB imaging to capture the phenotypic changes associated with nutrient deficiencies, providing a rich and accessible data source for our deep learning model. This rare setting not only provides an unparalleled opportunity to study long-term nutrient deficiency impacts on crops but also lends exceptional credibility to our findings. Through this approach, we can potentially revolutionize how nutrient deficiencies are detected and managed in agricultural settings, offering a powerful tool for improving crop yields and sustainability. This approach combines innovative deep learning with comprehensive phenotypic analysis, advancing soybean crop management and enabling early, precise detection of nutrient deficiencies. The potential practical implications of this research are significant. If successful, farmers could use a smartphone application to scan soybean fields quickly, receiving real-time feedback on nutrient status. This approach could allow for targeted, efficient fertilizer application, reducing costs and environmental effects while optimizing yield.

In summary, this research aims to address a critical gap in applying deep learning to soybean nutrient management. This study contributes to the development of more efficient and sustainable agricultural practices by investigating the potential of YOLOv8 for nutrient deficiency detection. Although challenges may arise, the insight gained from this study can advance the understanding of nutrient deficiency detection in soybeans and serve as a foundation for similar applications in other crops. This research could offer new directions for precision agriculture and sustainable farming practices, significantly affecting the future of global agriculture.

2. Results

2.1. Model Training and Validation Performance

The performance of YOLOv8s during the training and validation phases was evaluated using multiple metrics over 300 epochs (Fig. 1). The analysis focused on box loss, classification loss, distribution focal loss, precision, recall, and mean average precision (mAP) at various thresholds for the intersection over union (IoU). The following results collectively demonstrated the effectiveness of YOLOv8s in learning to detect and classify nutrient deficiencies in soybean leaves, with robust performance across various evaluation metrics.

The box loss, indicative of the ability of the model to localize nutrient deficiency regions accurately, exhibited a consistent downward trend for the training and validation sets (Fig. 1a and b). The initial box loss of approximately 0.02 decreased to 0.005 by the 300th epoch, representing a 75% reduction.

This substantial improvement suggests enhanced bounding-box prediction accuracy as training progressed. Classification loss, reflecting the proficiency of the model in distinguishing between various nutrient deficiency classes, demonstrated similar improvements (Fig. 1c and d). Both training and validation classification losses converged to values below 0.01, indicating that the model achieved a high degree of class separability.

The distribution focal loss, which balances the contribution of easy and difficult examples during training, also exhibited a marked decrease over time (Fig. 1e and f). This trend suggests that the model became increasingly adept at handling varying degrees of detection difficulty across the dataset.

Precision and recall metrics (Fig. 1g and i) exhibited robust performance, with both values approaching 1.0 by the end of training. The high precision (>0.95) indicates a low false positive (FP) rate, whereas the high recall (>0.98) value suggests minimal false negatives (FNs), which is crucial for reliable nutrient deficiency detection in agricultural applications.

The mAP at the IoU thresholds of 0.5 and 0.5:0.95 (Fig. 1h and j) demonstrated consistent improvement throughout training. The mAP@0.5 exceeded 0.99, whereas mAP@0.5:0.95 surpassed 0.90, indicating the robust performance of the model across varying degrees of overlap between predicted and ground-truth bounding boxes.

The statistical analysis of these metrics revealed low standard deviations (<0.01) in the final 50 epochs, suggesting stable model convergence. The consistent performance across training and validation sets, with less than a 1% difference in the final metric values, indicates good generalization and minimal overfitting.

2.2. *Precision-Recall and F1-Confidence Curves*

The performance of the YOLOv8s model across various nutrient-deficient classes, including nitrogen deficiency (–N), phosphorus deficiency (–P), and potassium deficiency (–K), as well as complete nutrient leaves, is summarized in the precision-recall and F1-confidence curves presented in Figure 2.

The precision-confidence curve (Fig. 2a) indicates that the model achieved high precision across a wide range of confidence thresholds, particularly for –K leaves. The –N and –P classes also exhibited strong precision values, though slightly lower than that of the –K class. The normal class maintained nearly perfect precision across all confidence levels, demonstrating the ability of the model to differentiate between nutrient-deficient and normal leaves accurately.

The precision-recall curve (Fig. 2b) further highlights the high performance of the model, with mAP values close to 1.0 for all classes. This finding suggests that the model consistently detects nutrient deficiencies without significant trade-offs between precision and recall.

The F1-confidence curve (Fig. 2c) illustrates the balance between precision and recall, with high F1- scores across all confidence levels for each class. The recall-confidence curve (Fig. 2d) confirms that the model maintained high recall values, particularly for the normal and –K classes, indicating that the model was able to identify the true positive (TP) cases.

2.3. *Confusion Matrix Analysis*

Figure 3 illustrates a confusion matrix generated for each nutrient deficiency class to evaluate the performance of YOLOv8s further. The model demonstrated nearly perfect classification accuracy for –K and –P, with classification accuracy rates of 1.00 for both classes. The model also highly accurately classified –N at 0.97, although a small misclassification occurred between –N leaves and the normal class. Specifically, 3% of the normal leaves were incorrectly classified as –N, which may have been caused by the visual similarities between the two conditions in some instances.

2.4. *Detection Results for Nutrient Deficiencies*

Figure 4 illustrates examples of the detection results for YOLOv8s for nutrient deficiencies in soybean leaves. For –N leaves (Fig. 4a), the model detected deficient areas with confidence scores ranging from 0.44 to 0.92. In addition, –P leaves (Fig. 4b) were detected with confidence scores between 0.65 and 0.88, whereas –K leaves (Fig. 4c) had confidence scores as high as 0.92. Normal leaves (Fig. 4d) were consistently detected with confidence scores above 0.90. These results demonstrate the robustness of the proposed model in detecting nutrient deficiencies, with most predictions yielding high confidence levels.

2.5. Model Performance Summary

Table 1 summarizes the performance of YOLOv8s across the training and validation datasets. The model achieved an overall precision of 96.01% during training, with –N being detected with 93.75% precision, –P with 95.42%, and –K with 95.02%. The recall for the training set was 98.15%, indicating that the model identifies nutrient deficiencies accurately with minimal FNs.

During validation, the model maintained a high overall precision of 95.21%, with individual class precisions ranging from 90.03% for –N to 96.54% for –P. The overall recall during validation was 98.19%, confirming the generalizability of the model to unseen data. The mAP@0.5 values for training and validation were consistently above 99%, whereas the more stringent mAP@0.5:0.95 values remained above 90%, reflecting the ability of the model to perform well even under stricter IoU thresholds.

The performance of YOLOv8s was compared to prior studies using other models for nutrient deficiency detection, as outlined in Supplementary Table 1. For instance, the DenseNet169 model was applied to banana nutrient deficiency detection and achieved a precision of 99.12% and recall of 98.37%. In addition, the InceptionV3 model was used for lettuce nutrient deficiency detection, and a detection accuracy of 96.5% was reported with a precision of 95.7%.

In comparison, in this study, YOLOv8s demonstrated competitive results across all nutrient deficiency classes in soybeans, with the added benefit of detecting multiple deficiencies simultaneously. Additionally, the processing time of 3.46 ms per image for YOLOv8s supports its application in real-time field monitoring, a significant improvement over other models. For example, DenseNet201, which was used for rice nutrient deficiency detection, displayed slower processing times.

3. Discussion

Early detection of nutrient deficiencies in crops is crucial for optimizing productivity and ensuring sustainable agricultural practices[12,25]. However, traditional diagnostic methods are often time-consuming, costly, and prone to inaccuracy[4,19,26]. This study proposes the YOLOv8s object detection model to identify nutrient deficiencies rapidly and accurately in soybean crops to address these challenges.

The study results demonstrate a significant advancement in applying deep learning models to precision agriculture, particularly for the early detection of nutrient deficiencies in soybean crops. The YOLOv8s model can identify –N, –P, and –K with an mAP@0.5 of 98.51% in validation, underscoring the reliability and precision of the model in field conditions that closely replicate real-world agricultural environments.

The use of a long-term experimental field maintained for over 40 years with specific nutrient-deficient plots distinguishes this work. This unique dataset provides a realistic simulation of nutrient stress that few studies can replicate. Unlike short-term or artificially induced nutrient deficiencies often applied in other studies, this dataset captures the natural progression of deficiencies, offering more authentic phenotypic data for model training and evaluation.

The YOLOv8s model detected –P and –K with nearly perfect accuracy, achieving precision rates of 96.54% and 94.68%, respectively, during validation. These findings highlight the potential of the model to address nutrient management in large-scale agricultural operations. By identifying deficiencies early, farmers can apply targeted treatments, minimizing yield loss and environmental harm caused by overusing fertilizers.

However, this study also revealed areas for improvement, particularly in detecting –N. Although the model achieved a reasonable precision of 90.03% for –N, this value was lower than those for –P and –K. In addition, –N symptoms are often less visually distinct, especially in the initial stages, and this subtlety likely contributed to the reduced accuracy of the model. Addressing this limitation should be a crucial focus of future research. One potential solution is integrating multispectral or hyperspectral imaging, which can capture a broader range of phenotypic and physiological data than the standard RGB imagery[27–29].

Comparing the results to previous studies reveals a marked improvement in performance for soybean nutrient deficiency detection. The deep convolutional neural network model that Ghosal et al.[10] employed achieved an accuracy of 90.3%, whereas YOLOv8s reached an mAP@0.5 of 98.51%. This significant improvement can be attributed to advancements in model architecture and the comprehensive, long-term dataset.

Furthermore, the performance of the proposed model is competitive with or surpasses that of models applied for nutrient deficiency detection in other crops. For instance, the DenseNet201 model that Sharma et al.[23] employed for rice achieved 85% accuracy, whereas the EfficientNetB4 model that Espejo-Garcia et al.[19] employed for citrus trees reached 98.52% accuracy. The impressive performance of the proposed model across nutrient deficiencies suggests its potential applicability to a broader range of crops and deficiency types[21,30,31].

The rapid inference time of the proposed model (3.46 ms per image) is a critical factor that sets it apart from other models. This speed makes this model suitable for real-time applications, potentially allowing for the quick processing of large fields using drone-mounted cameras or other high-throughput imaging systems[32]. The combination of high accuracy and rapid processing offers new possibilities for precision agriculture, enabling more timely and targeted interventions in crop management[33].

This study focused on nutrient deficiencies; nevertheless, future iterations could be expanded to detect other crop stressors, such as water shortages or pest infestations[34,35]. Broadening the capabilities of this model can create a more comprehensive crop-health monitoring system that responds to multiple stress factors in real time.

The proposed model displays impressive performance in controlled experimental conditions; however, real-world implementation may present additional challenges. Varying soil types, microclimates, and interactions between multiple stressors could affect the performance of the model[36]. Future work should focus on validating and refining the proposed model under a broader range of field conditions to ensure its robustness and reliability in diverse agricultural settings.

Integrating this technology with other precision agriculture tools, such as variable-rate fertilizer applicators and farm management software, presents exciting possibilities. Creating a seamless flow of information from detection to action could further optimize resource use and crop management practices.

In conclusion, this research represents a meaningful advancement in the application of artificial intelligence and deep learning to agriculture. The YOLOv8s model is an effective tool for the early detection of nutrient deficiencies in soybeans, offering high accuracy and practical utility. This study lays the groundwork for future innovations in crop management by demonstrating the robustness of the model across real-world conditions and its potential for large-scale deployment. The ability of the proposed model to address nutrient deficiencies—one of the most critical challenges in agriculture—positions it as a valuable asset in the pursuit of sustainable farming practices and improved crop yields.

4. Methods

4.1. Experimental Design for the Soybean Nutrient Deficiency Study

This study was conducted at the nutrient test field of the Gyeongsangbuk-do Agricultural Research and Extension Services in Chilgok, Daegu, Republic of Korea (35°95′28.97"N, 128°56′16.01°E). This site has been dedicated to long-term nutrient deficiency experiments for N, P, and K for over 40 years[37], providing a unique opportunity to study the effects of prolonged nutrient stress on soybeans (Fig. 5). The experimental field was divided into four treatment plots, each covering an area of 70 m².

To prevent cross-contamination and ensure that each significant nutrient deficiency plot remained isolated, thick plastic barriers were driven approximately 1 m deep into the soil between plots. This setup ensured that nutrient treatments did not mix between plots. The prolonged treatment effectively reduced soil buffering capacity over time, simulating natural nutrient-deficient conditions that could occur in long-term agricultural settings.

4.2. Long-term Nutrient Treatment and Soil Analysis

This study uses the soybean variety Hwangkeum (*Glycine max* (L.) *Merrill*). Soybean planting and sowing were conducted yearly on June 20, maintaining consistency across growing seasons[38]. The plants were spaced 60 cm apart between rows and 15 cm apart within rows, following standard agricultural practices for soybean cultivation. The experiment employed a single plot design with four treatment groups. These included a control group, where all nutrients were applied, and three deficiency treatments: excluding N, excluding P, and excluding K. Fertilizer was applied in two stages to optimize nutrient availability throughout the growing season. In the field, 40% of the total N fertilizer was applied as a base fertilizer before planting[39]. The first and second top-dressing applications each involved 30% of the remaining N. The P and K were fully applied as base fertilizers[40]. The fertilizer was applied around the base of the plants, or applications were divided as needed, depending on plant growth and environmental conditions. Fertilization was applied according to the guidelines of the Rural Development Administration of Korea. The recommended application rates per 10 acres (1,000 m^2) were 12 kg for N, 8 kg for P (i.e., P_2O_5), and 8 kg for K (i.e., K_2O) (Table 2). This design allowed for the isolated study of the effects of the deficiency of each nutrient on soybean growth and development.

The soil was analyzed to evaluate the long-term effects of nutrient treatments on the physicochemical properties of the soil. Soil samples were collected from each treatment plot at a depth of 0-20 cm[41]. Soil physical and chemical analyses were conducted based on the soil standard analysis method by the National Institute of Agricultural Science and Technology, RDA, Korea[42].

The soil samples were analyzed for pH, electrical conductivity (EC), organic matter (OM), total N, ammonium (NH_4^+), nitrate (NO_3^-), available phosphorus (P_2O_5), and exchangeable bases (K, Ca, Mg, and Na)[43]. The pH and EC were determined in a 1:5 soil-to-water ratio suspension. Organic matter (OM) content was analyzed using the Tyurin method. The available phosphorus (P_2O_5) was measured using the Lancaster method. Total N was measured with an elemental analyzer. The ammonium (NH_4^+) and nitrate (NO_3^-) were analyzed using the Kjeldahl method for N determination. Exchangeable bases (K, Ca, Mg, Na) were analyzed by the 1.0 M NH_4OAc (pH 7.0) method.

Supplementary Table 2 presents the detailed results of this analysis. Statistical differences between treatments were analyzed using the one-way analysis of variance, followed by Tukey's honestly significant difference post hoc test. The letters in Supplementary Table 2 indicate the significant differences between treatments ($p < 0.05$).

4.3. Data Collection and Visual Representation

Phenotypic images of soybean leaves under various nutrient deficiency conditions were captured to represent the effects of each treatment visually. The RGB leaf images were taken twice over two growing seasons to capture early-stage phenotypes[44]: on August 13, 2021, and July 27, 2022, using a digital camera capable of 360° capture (Insta360 ONE RS F3.2, Shenzhen, Arashi Vision Inc., China).

A gray card was used for color balance in each image to ensure accurate color representation. Sampling was performed at two distinct sampling points in a checkerboard pattern to ensure reproducibility for the predictive models targeting biological subjects[45]. As listed in Table 3, 6,020 RGB images were obtained, distributed across the nutrient conditions. Figure 6 provides representative examples of these images, illustrating the typical symptoms associated with each nutrient deficiency condition.

These high-resolution images provide a clear visual representation of how nutrient deficiencies manifest in soybean leaves, aiding in identifying and understanding nutrient stress symptoms in field conditions. The large dataset, comprising 1,800 images of –N, 1,480 of –P, 1,460 of –K, and 1,280 of complete nutrient, ensures a robust representation of each nutrient state. This experimental setup (maintained consistently over four decades) effectively reduced soil buffering capacity, closely simulating natural nutrient-deficient conditions that could be encountered in long-term agricultural settings. The long-term nature of this study, combined with the comprehensive image dataset, provides valuable insight into the chronic effects of nutrient deficiencies on soybean growth, development, and leaf phenotypes, contributing significantly to the understanding of plant responses to prolonged nutrient stress.

4.4. Image Preprocessing and Dataset Construction

After collecting the initial set of 6,020 RGB images, we manually annotated the images using the LabelImg software[46]. Each image was annotated, designating four classes: –N, –P, –K, and complete nutrient. The annotation process included labeling truncated and difficult-to-identify objects to ensure a comprehensive representation of real-world scenarios. Following the annotation process, we employed preprocessing and augmentation techniques to enhance the dataset size and quality, addressing potential overfitting problems due to limited data.

The image augmentation process involved several techniques to create a more robust and diverse dataset[47]. We applied hue, saturation, and value adjustments, varying the hue by ±0.015, the saturation from 0.7 to 1.3 (±30%), and the value (brightness) from 0.4 to 1.6 (±60%). Geometric transformations were also implemented, including rotations of up to ±10°, translations up to 10% in any direction, and scaling from 0.5 to 1.5 (±50%). Additionally, we applied vertical and horizontal flipping, each with a 50% probability, and used mosaic augmentation with a factor of 1.0.

These augmentation techniques were applied randomly to each annotated image in the dataset, expanding the original dataset of 6,020 images to approximately 30,100 images. The augmented dataset maintained the original annotations, ensuring that each new image retained accurate class labels and object locations. The final augmented dataset was partitioned into three subsets: approximately 70% (21,070 images) for the training set, 20% (6,020 images) for the validation set, and 10% (3,010 images) for the testing set[48]. This partitioning strategy ensures robust model training, validation, and unbiased performance evaluation.

4.5. YOLOv8s Model Architecture, Configuration, and Training

4.5.1. Model Architecture

For nutrient deficiency detection in soybean leaves, we employed YOLOv8s[14]. The YOLOv8s architecture comprises an input terminal processing 640x640x3 pixel images, a backbone for feature extraction, a head for object detection, and detection modules. As illustrated in Figure 7, the backbone includes a stem layer and four stage layers with ConvModule and C2f components, progressively refining feature representations. The head uses top-down and bottom-up layers for multiscale feature processing, enabling detection at various scales. The detection module predicts bounding boxes, object classes, and confidence scores using specialized layers and activation functions.

4.5.2. Deep Learning Development Environment and Hyperparameter Tuning

The development environment consisted of an Intel(R) Xeon(R) Gold 5220R CPU, dual Nvidia Corporation GA106 (RTX A2000 12 GB) graphics processing units (GPUs), and 256 GiB of memory. The system operated on Linux 5.15.0-117-generic (Ubuntu SMP, x86_64). Python 3.10.14 served as the primary programming language, with PyTorch 2.4.0+cu124 used for deep learning development. The GPU-based computations were facilitated by CUDA 12.4, cuDNN 8.9.7, and OpenCV 4.10.0, ensuring optimal performance and compatibility (Table 4).

The YOLOv8 model was employed for object detection, with custom configurations tailored to the specific dataset. The dataset consisted of four classes: $k_{deficit}$, $n_{deficit}$, $p_{deficit}$, and *complete*, with class weights adjusted to [1.5, 1.5, 2.5, 1.0], respectively, to address potential class imbalances. Various hyperparameters were carefully tuned to optimize learning efficiency[49]. The total training epochs were set to 300, with a batch size of 16. The learning rate schedule involved an initial rate of 0.01 with a final learning rate factor of 0.2. Additional hyperparameters included a momentum of 0.937, weight decay of 0.0005, warm-up epochs of 3.0, warm-up momentum of 0.8, and a warm-up bias learning rate of 0.1. The loss function was configured with a box loss weight of 0.05, class loss weight of 0.5, and IoU threshold of 0.7. Data augmentation techniques were extensively applied to enhance model generalization. These included hue, saturation, and value color space adjustments with a hue variation of 0.015, saturation variation of 0.7, and value variation of 0.4. Geometric transformations were also employed, including rotations of up to 10°, translation of 0.1, and scale adjustments of 0.5. Vertical and horizontal flipping were applied with a probability of 0.5 each. Mosaic augmentation was fully used with a factor of 1.0, whereas MixUp and Copy-Paste augmentations were not used in this configuration[50].

4.6. Evaluation of Model Performance

To quantify the detection performance comprehensively of YOLOv8, we employed a wide range of evaluation metrics commonly employed in object detection tasks. These metrics include the precision, recall, F1-score, average precision (AP), mAP, IoU, loss, confidence, detection accuracy, kappa score, and specificity[14]. The following terms are used in the evaluation metrics:

- True positives (TPs): number of bounding boxes that the model accurately identified as the correct class.
- False positives (FPs): number of bounding boxes that the model misclassified or detected where there was no object.
- False negatives (FNs): number of objects that should have been detected but were missed by the model.
- True negatives (TNs): number of correctly identified absences of objects.

4.6.1. Precision, Recall, and F1-score

Precision measures the accuracy of the positive predictions made by the model[51]. A higher precision indicates that when the model predicts a nutrient deficiency, it is more likely to be correct. The calculation formula is as follows:

$$\textbf{Precision}\ (\boldsymbol{Pr}) = \frac{\textbf{Total correct classifications}}{\textbf{Total classifications made}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}.\ \textbf{Equation (1)}$$

Recall, also known as sensitivity, measures the proportion of TP cases that the model successfully predicts as positive[51]. The calculation formula for the recall is as follows:

$$\textbf{Recall} = \frac{\boldsymbol{TPs}}{\boldsymbol{Total\ positivies}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}.\ \textbf{Equation (2)}$$

A higher recall indicates that the model can capture more instances of nutrient deficiency, but it might also lead to more FPs.

The F1-score represents the harmonic average of the precision and recall[51]. The range of the F1-score is [0, 1]. An F1-score close to 1 indicates excellent model performance, whereas a score close to 0 indicates inferior performance. The formula for calculating the F1-score is

$$\textbf{F1–score} = \boldsymbol{2} \times \left(\frac{\boldsymbol{Precision \times Recall}}{\boldsymbol{Precision + Recall}}\right).\ \textbf{Equation (3)}$$

The range of F1-Score is [0, 1].

4.6.2. Average Precision and Mean Average Precision

The AP summarizes the precision-recall curve for a single class[16]:

$$\boldsymbol{AP} = \int_{\boldsymbol{0}}^{\boldsymbol{1}} \boldsymbol{P(r)dr},\ \textbf{Equation (4)}$$

Where P(r) is the Precision value at Recall value r.

where $P(r)$ is the precision value at recall value r. The mAP extends the AP to multiple classes:

$$\boldsymbol{mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i}, \textbf{ Equation (5)}$$

Where n is the number of classes.

In the evaluation, we applied two variations of the mAP. First, mAP@0.5IoU calculates the average precision at an IoU threshold of 0.5. Second, mAP@0.5:0.95IoU averages the AP over multiple IoU thresholds from 0.5 to 0.95, providing a more robust evaluation of the performance of the model across overlap scenarios.

4.6.3. Intersection over Union

The IoU represents the ratio of the intersection to the union of the model-predicted and ground-truth bounding boxes. The IoU values range from 0 to 1, with higher values indicating better localization accuracy of the detected nutrient deficiency regions. The calculation formula is as follows:

$$\boldsymbol{IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}}. \textbf{ Equation (6)}$$

$$\boldsymbol{mIoU = \frac{TP}{FN+TP+FP}}. \textbf{ Equation (7)}$$

4.6.4. Loss

The loss function measures the prediction error of the model[52]. For YOLO models, the loss typically combines localization, confidence, and classification losses. In the configuration, we set the box loss weight to 0.05 and class loss weight to 0.5. The categorical cross-entropy loss is

$$\boldsymbol{Loss = L = -\sum_{c=1}^{C} y_c \times \log(\widehat{y_c})}. \textbf{ Equation (8)}$$

4.6.5. Confidence

The confidence score indicates the certainty of the model in its predictions[53]. This score is typically output for each detected object and is applied in conjunction with an IoU threshold (set to 0.7 in this configuration) to determine the final detections.

4.6.6. Detection Accuracy

Detection Accuracy measures the overall correctness of the model's predictions[54]:

$$\boldsymbol{Accuracy = \frac{Total\ correct\ classifications}{Total\ samples} = \frac{\sum TP + \sum TN}{\sum TP + \sum TM + \sum FP + \sum FN}}. \textbf{ Equation (9)}$$

4.6.7. Kappa Score

The kappa score measures the agreement between the model predictions and ground truth, accounting for the possibility of agreement occurring by chance:

$$\boldsymbol{kappa\ score} = (\boldsymbol{p_o} - \boldsymbol{p_e})/(\mathbf{1} - \boldsymbol{p_e}).\ \textbf{Equation (10)}$$

where p_o represents the observed agreement, and p_e denotes the expected agreement by chance.

4.6.8. Specificity

Specificity measures the ability of the model to identify negative instances correctly:

$$\boldsymbol{Specificity} = \frac{\boldsymbol{Total\ true\ negatives}}{\boldsymbol{Total\ negatives}} = \frac{\boldsymbol{TN}}{\boldsymbol{TN+FP}}.\ \textbf{Equation (11)}$$

4.6.9. Model Fitness and Inference Speed

We evaluated the model fitness, which is a single value representing the overall performance of the model, considering various aspects of the predictions. A higher fitness score indicates better overall model performance. Additionally, we measured the inference speed in milliseconds per image, which is crucial for assessing the suitability of the model for real-time or nearly real-time applications in agricultural settings.

4.6.10. Evaluation Process

To assess the generalizability of the model, we conducted separate evaluations on the training and validation datasets. For both datasets, we focused on precision (%), recall (%) mAP@0.5IoU, and mAP@0.5:0.95IoU. We directly compared the performance and identified any potential overfitting or underfitting problems using the same set of metrics for the training and validation datasets. This comprehensive evaluation approach provides insight into how well the model generalizes to new, unseen data. This evaluation framework assesses the accuracy, precision, and efficiency of the model across various IoU thresholds and datasets. We ensure that the model performs well on the data on which it was trained and that it maintains its effectiveness when applied to new, unseen data in practical agricultural settings by considering the training and validation performance using the same metrics.

References

1. Havlin, J. L. & Heiniger, R. W. Soil Fertility Management for Better Crop Production. *Multidisciplinary Digital Publishing Institute* **10**, 1349 (2020).

2. Dotaniya, C. K. *et al.* Crop Performance and Soil Properties under Organic Nutrient Management. *Excellent Publishers* **9**, 1055–1065 (2020).

3. Bagale, S. Nutrient Management for Soybean Crops. *Hindawi Publishing Corporation* **2021**, 1–10 (2021).

4. Sudhakar, M. & Priya, R. M. S. Computer Vision Based Machine Learning and Deep Learning Approaches for Identification of Nutrient Deficiency in Crops: A Survey. **22**, 1387–1399 (2023).

5. Jose, A., Nandagopalan, S., Ubalanka, V. & Viswanath, D. P. Detection and classification of nutrient deficiencies in plants using machine learning. *IOP Publishing* **1850**, 12050 (2021).

6. Yi, J. *et al.* Deep Learning for Non-Invasive Diagnosis of Nutrient Deficiencies in Sugar Beet Using RGB Images. *Multidisciplinary Digital Publishing Institute* **20**, 5893 (2020).

7. Silva, A. L. De *et al.* Hyperspectral Imaging of Adaxial and Abaxial Leaf Surfaces as a Predictor of Macadamia Crop Nutrition. *Multidisciplinary Digital Publishing Institute* **12**, 558 (2023).

8. Payne, W. Z. & Kurouski, D. Raman-Based Diagnostics of Biotic and Abiotic Stresses in Plants. A Review. *Frontiers Media* vol. 11 Preprint at https://doi.org/10.3389/fpls.2020.616672 (2021).

9. Roper, J. M., Garcia, J. & Tsutsui, H. Emerging Technologies for Monitoring Plant Health in Vivo. *American Chemical Society* vol. 6 5101–5107 Preprint at https://doi.org/10.1021/acsomega.0c05850 (2021).

10. Ghosal, S. *et al.* An explainable deep machine vision framework for plant stress phenotyping. *Proc Natl Acad Sci U S A* **115**, 4613–4618 (2018).

11. Tran, T. T., Choi, J. W., Le, T. T. H. & Kim, J. W. A comparative study of deep CNN in forecasting and classifying the macronutrient deficiencies on development of tomato plant. *Applied Sciences (Switzerland)* **9**, (2019).

12. Xu, Z. *et al.* Using deep convolutional neural networks for image-based diagnosis of nutrient deficiencies in rice. *Comput Intell Neurosci* **2020**, (2020).

13. Alif, M. A. R. & Hussain, M. YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.2406.10139 (2024).

14. Khare, O., Gandhi, S., Rahalkar, A. M. & Mane, S. B. YOLOv8-Based Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes. (2023) doi:10.1109/punecon58714.2023.10449999.

15. Muthusamy, S. & Ramu, S. P. IncepV3Dense: Deep Ensemble Based Average Learning Strategy for Identification of Micro-Nutrient Deficiency in Banana Crop. *IEEE Access* **12**, 73779–73792 (2024).

16. Islam, S. *et al.* Nutrient Stress Symptom Detection in Cucumber Seedlings Using Segmented Regression and a Mask Region-Based Convolutional Neural Network Model. *Agriculture (Switzerland)* **14**, (2024).

17. Lu, J., Peng, K., Wang, Q. & Sun, C. Lettuce Plant Trace-Element-Deficiency Symptom Identification via Machine Vision Methods. *Agriculture (Switzerland)* **13**, (2023).

18. Taha, M. F. *et al.* Using Deep Convolutional Neural Network for Image-Based Diagnosis of Nutrient Deficiencies in Plants Grown in Aquaponics. *Chemosensors* **10**, (2022).

19. Espejo-Garcia, B., Malounas, I., Mylonas, N., Kasimati, A. & Fountas, S. Using EfficientNet and transfer learning for image-based diagnosis of nutrient deficiencies. *Comput Electron Agric* **196**, (2022).

20. Majdalawieh, M., Khan, S. & Islam, M. T. Using Deep Learning Model to Identify Iron Chlorosis in Plants. *IEEE Access* **11**, 46949–46955 (2023).

21. Wang, S., Zhang, Y. & Li, R. Citrus Yellow Shoot Disease Detection based on YOLOV5. **39**, 1291–1300 (2023).

22. Jesie, R. S. & Premi, M. S. G. Improved Tunicate Swarm Optimization Based Hybrid Convolutional Neural Network for Classification of Leaf Diseases and Nutrient Deficiencies in Rice (Oryza). *Agronomy* **14**, (2024).

23. Sharma, M. *et al.* Identification of rice leaf diseases and deficiency disorders using a novel DeepBatch technique. *Open Life Sci* **18**, (2023).

24. Azimi, S., Kaur, T. & Gandhi, T. K. A deep learning approach to measure stress level in plants due to Nitrogen deficiency. *Measurement (Lond)* **173**, (2021).

25. Waheed, A. *et al.* An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. *Elsevier BV* **175**, 105456 (2020).

26. Xu, Z. *et al.* Research Article Using Deep Convolutional Neural Networks for Image-Based Diagnosis of Nutrient Deficiencies in Rice. Preprint at (2020).

27. Abebe, A. M., Kim, Y., Kim, J.-Y., Kim, S. L. & Baek, J. Image-Based High-Throughput Phenotyping in Horticultural Crops. *Multidisciplinary Digital Publishing Institute* vol. 12 2061 Preprint at https://doi.org/10.3390/plants12102061 (2023).

28. Gaikwad, S. & Kothari, S. Multi-Spectral Imaging for Fruits and Vegetables. *Science and Information Organization* **13**, (2022).

29. Marzougui, A., McGee, R. J., Vleet, S. Van & Sankaran, S. Remote sensing for field pea yield estimation: A study of multi-scale data fusion approaches in phenomics. *Frontiers Media* **14**, (2023).

30. Wang, Y., Wang, H. & Peng, Z. Rice diseases detection and classification using attention based neural network and bayesian optimization. *Elsevier BV* **178**, 114770 (2021).

31. Guo, W., Feng, Q., Li, X., Yang, S. & Yang, J. Grape leaf disease detection based on attention mechanisms. *Chinese Society of Agricultural Engineering* **15**, 205–212 (2022).

32. Reis, D., Kupec, J., Hong, J. & Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.2305.09972 (2023).

33. Chin, R., Catal, C. & Kassahun, A. Plant disease detection using drones in precision agriculture. *Springer Science+Business Media* **24**, 1663–1682 (2023).

34. Abioye, A. E. *et al.* Precision Irrigation Management Using Machine Learning and Digital Farming Solutions. *Multidisciplinary Digital Publishing Institute* **4**, 70–103 (2022).

35. Cheng, D. *et al.* Precision agriculture management based on a surrogate model assisted multiobjective algorithmic framework. *Nature Portfolio* **13**, (2023).

36. Park, S.-H. *et al.* Development of a Soil Moisture Prediction Model Based on Recurrent Neural Network Long Short-Term Memory (RNN-LSTM) in Soybean Cultivation. *Multidisciplinary Digital Publishing Institute* **23**, 1976 (2023).

37. Son, H., Klingaite, J., Park, S. & Shin, J. H. Exploring the role and characterization of Burkholderia cepacia CD2: a promising eco-friendly microbial fertilizer isolated from long-term chemical fertilizer-free soil. *J Appl Biol Chem* **66**, 394–403 (2023).

38. Kim, M. *et al.* Genome assembly of the popular Korean soybean cultivar Hwangkeum. *Genetics Society of America* **11**, (2021).

39. Agehara, S. & do Nascimento Nunes, M. C. Season and Nitrogen Fertilization Effects on Yield and Physicochemical Attributes of Strawberry under Subtropical Climate Conditions. *Multidisciplinary Digital Publishing Institute* **11**, 1391 (2021).

40. Loha, G., Derese, M. & Gidago, G. Effect of Blended Nitrogen, Phosphorus, Sulfur, Boron, and Potassium Fertilizer Rates on Growth and Yield of Maize (Zea mays L.) at Sodo Zuriya District, Southern Ethiopia. *Hindawi Publishing Corporation* **2023**, 1–13 (2023).

41. Setubal, I. S. *et al.* Macro and Micro-Nutrient Accumulation and Partitioning in Soybean Affected by Water and Nitrogen Supply. *Multidisciplinary Digital Publishing Institute* **12**, 1898 (2023).

42. National Institute of Agricultural Science Technology (NIAST). *Methods of Analysis of Soil and Plant*. (National Institute of Agricultural Science Technology, Rural Development Administration, Suwon, Korea, 2000).

43. Pros, K., Kim, K., Lee, Y.-H., Park, J. W. & Han, G. H. Compositional Characteristics of Major Inorganic Anions in Soil and Plant Water Extracts from Various Farmlands. **55**, 424–432 (2022).

44. Singhvi, V., Lunga, L., Nidhi, P. S. A., Keum, C. & Prakash, V. J. High-Throughput Phenotyping using Computer Vision and Machine Learning. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.2407.06354 (2024).

45. Liu, G. S., Shenson, J. A., Farrell, J. & Blevins, N. H. Signal to noise ratio quantifies the contribution of spectral channels to classification of human head and neck tissues ex vivo using deep learning and multispectral imaging. *SPIE* **28**, (2023).

46. Wspanialy, P., Brooks, J. & Moussa, M. An Image Labeling Tool and Agricultural Dataset for Deep Learning. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.2004.03351 (2020).

47. Cossio, M. Augmenting Medical Imaging: A Comprehensive Catalogue of 65 Techniques for Enhanced Data Analysis. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.2303.01178 (2023).

48. Zoph, B. *et al.* Learning Data Augmentation Strategies for Object Detection. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.1906.11172 (2019).

49. Bertsimas, D. *et al.* Holistic Deep Learning. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.2110.15829 (2021).

50. Takahashi, R., Matsubara, T. & Uehara, K. Data Augmentation Using Random Image Cropping and Patching for Deep CNNs. *Institute of Electrical and Electronics Engineers* **30**, 2917–2931 (2019).

51. Padilla, R., Netto, S. L. & da Silva, E. A. B. A Survey on Performance Metrics for Object-Detection Algorithms. (2020) doi:10.1109/iwssip48289.2020.9145130.

52. Zheng, Z. *et al.* Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Cornell University* Preprint at https://doi.org/10.48550/arXiv.1911.08287 (2019).

53. Geng, Q., Huang, X., Zhou, Z. & Yang, R. A Network Structure to Explicitly Reduce Confusion Errors in Semantic Segmentation. *Cornell University* Preprint at https://doi.org/10.48550/arxiv.1808.00313 (2018).

54. Szegedy, C., Toshev, A. & Erhan, D. Deep Neural Networks for Object Detection. **26**, 2553–2561 (2013).

Acknowledgements

This work was carried out with the support of "Cooperative Research Program for Agriculture Science & Technology Development (Project No. PJ015697)" Rural Development Administration, Republic of Korea.

Funding

This work was carried out with the support of "Cooperative Research Program for Agriculture Science & Technology Development (Project No. PJ015697)" Rural Development Administration, Republic of Korea.

Author contributions

M.J. led the manuscript writing, data analysis, coding, plant and soil experiments, and conducted the initial analyses. S.P. contributed to the analysis of soil physicochemical properties and assisted with plant experiments. S.M.K. provided essential support in data preprocessing. K.M.L. assisted with plant experiments, data preprocessing, and contributed to the design of the plant experiments. H.S.L. provided the initial research idea. J.H.S. supervised the overall project.

Competing interests

The authors declare no competing interests.

Data availability statement

The datasets used and analyzed during the current study are available from the corresponding author upon request.

Additional Information

Correspondence and requests for materials should be addressed to M.J., J.H.S.

Figure 1. Training and validation performance metrics for YOLOv8s: (a) training box loss, (b) validation box loss, (c) training classification loss, (d) validation classification loss, (e) training distribution focal loss, (f) validation distribution focal loss, (g) precision, (h) mean average precision (mAP) at a threshold of 0.5, (i) recall, and (j) mAP@0.5:0.95. Each graph presents the performance of the model over 300 epochs, with loss values decreasing and precision, recall, and mAP metrics improving as training progresses. Letter (B) indicates that the metrics were calculated on a batch-wise basis.

Figure 2. Performance curves for YOLOv8s in detecting nutrient deficiencies in soybean leaves. (a) Precision-confidence curve displaying the precision values across confidence thresholds for potassium (K), nitrogen (N), and phosphorus (P) deficiencies and normal conditions. (b) Precision-recall curve comparing precision and recall for each nutrient condition. (c) F1-confidence curve displaying the F1-score across confidence levels for each class. (d) Recall-confidence curve depicting the recall values across confidence thresholds.

Figure 3. Confusion matrix presenting the classification performance of YOLOv8s in detecting nutrient deficiencies in soybean leaves. The matrix displays the actual class labels, potassium (K) deficiency, nitrogen (N) deficiency, and phosphorus (P) deficiency and normal, along the *y*-axis and the predicted class labels along the *x*-axis. The values in each cell represent the classification accuracy for each nutrient condition, with darker colors indicating higher accuracy.

Figure 4. Detection results of soybean leaves under nutrient deficiency conditions: (a) nitrogen deficiency, (b) phosphorus deficiency, (c) potassium deficiency, and (d) normal (control). Each image depicts the predicted bounding boxes, with labels indicating the predicted class and confidence score for each detection.

Figure 5. Overview of experimental field site and nutrient-deficient treatment plots. (a) Geographical location of the experimental field: Daegu, Republic of Korea. Images of the nutrient-deficient treatment plots: (b) nitrogen (–N), (c) phosphorus (–P), (d) potassium (–K) deficiencies, and (e) complete nutrient.

Figure 6. Images of soybean leaves under nutrient deficiency conditions: (a) nitrogen (–N), (b) phosphorus (–P), (c) potassium (–K) deficiencies, and (d) complete nutrient.

Figure 7. Network architecture diagram of YOLOv8s for nutrient deficiency detection in soybean leaves. The architecture consists of an input terminal, backbone, head, and detection modules using ConvModule, C2f, SPFF, and Concat.

Table 1. Performance metrics of YOLOv8s for nutrient deficiency detection in soybean leaves.

Dataset	Metric	–N	–P	–K	Complete nutrient	Total (mean)	Model fitness	Speed (ms image^{-1})

Training	Precision (%)	93.75	95.42	95.02	99.85	96.01	92.92	2.78
	Recall (%)	96.10	95.50	99.99	99.99	98.15		
	mAP@0.5IoU	98.85	98.90	99.46	99.50	99.18		
	mAP@0.5:0.95IoU	92.72	90.84	91.72	93.62	92.22		
Validation	Precision (%)	90.03	96.54	94.68	99.59	95.21	91.55	3.46
	Recall (%)	96.67	96.11	99.99	99.99	98.19		
	mAP@0.5	98.58	98.93	97.03	99.50	98.51		
	mAP@0.5:0.95	90.83	91.61	87.50	93.14	90.77		

Notes: mAP: mean average precision, IoU: intersection over union, N: nitrogen, P: phosphorus, K: potassium

Table 2. Composition of nutrients applied in soybean nutrient deficiency treatments (1 kg per 10 acres).

Nutrient treatments	Composition of nutrients (kg/10 acres)		
	Nitrogen (N)	Phosphorus (P)	Potassium (K)
Complete nutrient	12	8	6
N deficiency	-	8	6
P deficiency	12	-	6
K deficiency	12	8	-

Table 3. Typical symptoms of nutrient deficiencies in soybean plants with corresponding image counts.

Class	Typical symptoms	Images
–N	Older leaves gradually turn yellow, starting from the base of the plant, while newer leaves stay green but may become pale over time.	1,800
–P	Leaves, especially lower ones, develop dark green or purplish shades, often with stunted growth and delayed maturity.	1,480
–K	Leaf edges are yellow or brown with scorched-looking tips, and small necrotic spots may appear as the condition worsens.	1,460
Complete Nutrient	Leaves are evenly green, smooth, and free from any visible damage or discoloration.	1,280

Notes: N: nitrogen, P: phosphorus, K: potassium

Table 4. Experimental environment specifications used for model training and analysis.

Experimental environment	
Processor	Intel(R) Xeon(R) Gold 5220R CPU
Operating system	Linux 5.15.0-117-generic (Ubuntu SMP, x86_64)
RAM	256 GiB
Graphic card	Nvidia Corporation GA106 (RTX A2000 12 GB) (2EA)

Programming language	Python 3.10.14
Deep learning libraries	PyTorch 2.4.0+cu124
Software	CUDA 12.4 + CUDNN 8.9.7 + OpenCV 4.10.0 and Visual Studio

Algorithm 1. Procedure for training and evaluating YOLOv8s for nutrient deficiency detection.

1. Start.
2. Load and preprocess the dataset.
   a. Create a custom.yaml file specifying training and validation data paths, number of classes, class names, and class weights.
   b. Preprocess the images by resizing them to 640x640 for YOLOv8.
3. Data augmentation:
   a. Apply hue (0.015), saturation (0.7), and value (0.4) adjustments.
   b. Implement geometric transformations (rotation: 10°, translation: 0.1, scale: 0.5).
   c. Apply vertical and horizontal flipping with 0.5 probability each.
   d. Use mosaic augmentation with a factor of 1.0.
4. Build the YOLOv8 model:
   a. Import the YOLO model from Ultralytics.
   b. Load the pretrained YOLOv8s model.
   c. Set up model hyperparameters (learning rate: 0.01 to 0.2, batch size: 16, momentum: 0.937, and weight decay: 0.0005).
   d. Specify data augmentation techniques in the model configuration.
   e. Define the number of classes (4) and detection anchors.
5. Compile the model:
   a. Use pretrained YOLOv8 with custom weights.
   b. Specify the optimizer with the defined learning rate schedule.
   c. Set the loss function weights (box: 0.05 and class: 0.5).
   d. Specify the evaluation metrics to monitor during training (precision, recall, and mAP).
6. Train the model:
   a. Set the number of epochs to 300 and batch size to 16.
   b. Configure the warm-up settings (epochs: 3.0, momentum: 0.8, and bias learning rate: 0.1).
   c. Use multiple GPUs for training (device='0,1').
   d. Execute the model.train() function with all specified parameters.
7. Evaluate the model:
   a. Calculate the evaluation metrics on the validation set (mAP@0.5, mAP@0.5:0.95, precision, recall, and F1-score).
   b. Generate the per-class performance metrics.

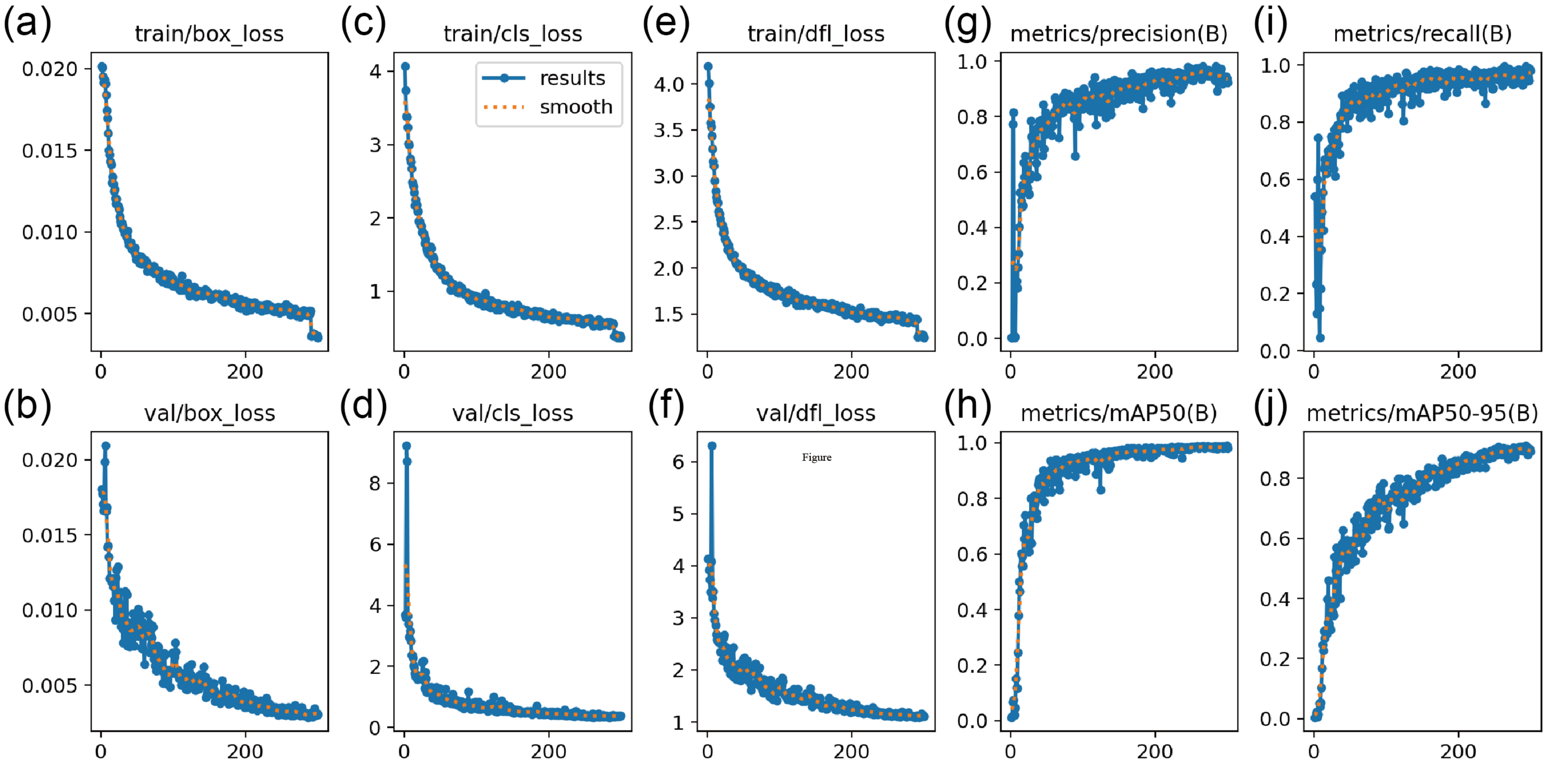
c. Evaluate the IoU with a threshold of 0.7.

8. Save the trained model for future use.

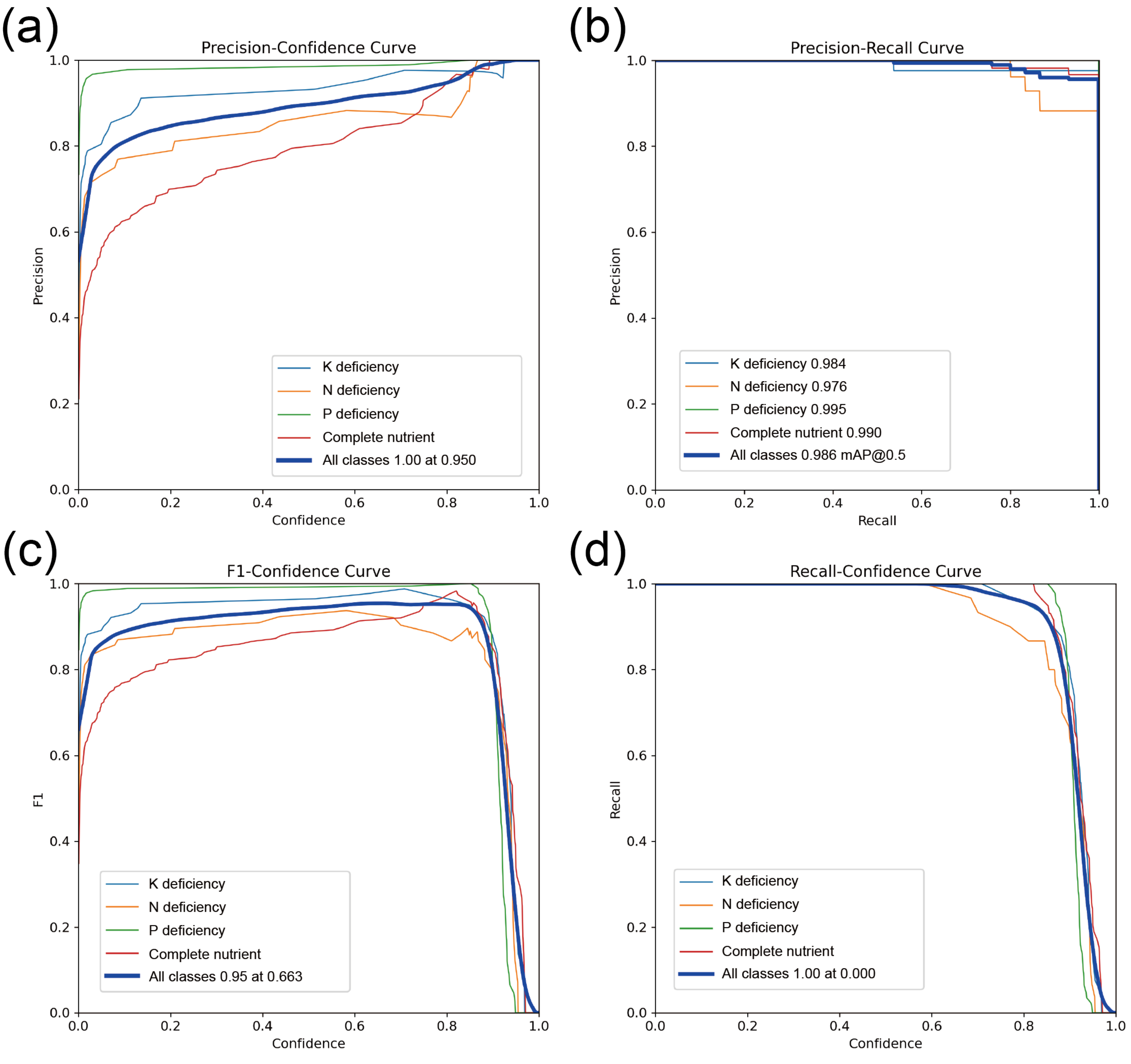
9. Output the results:

   a. Report the model performance, training speed, and evaluation metrics.

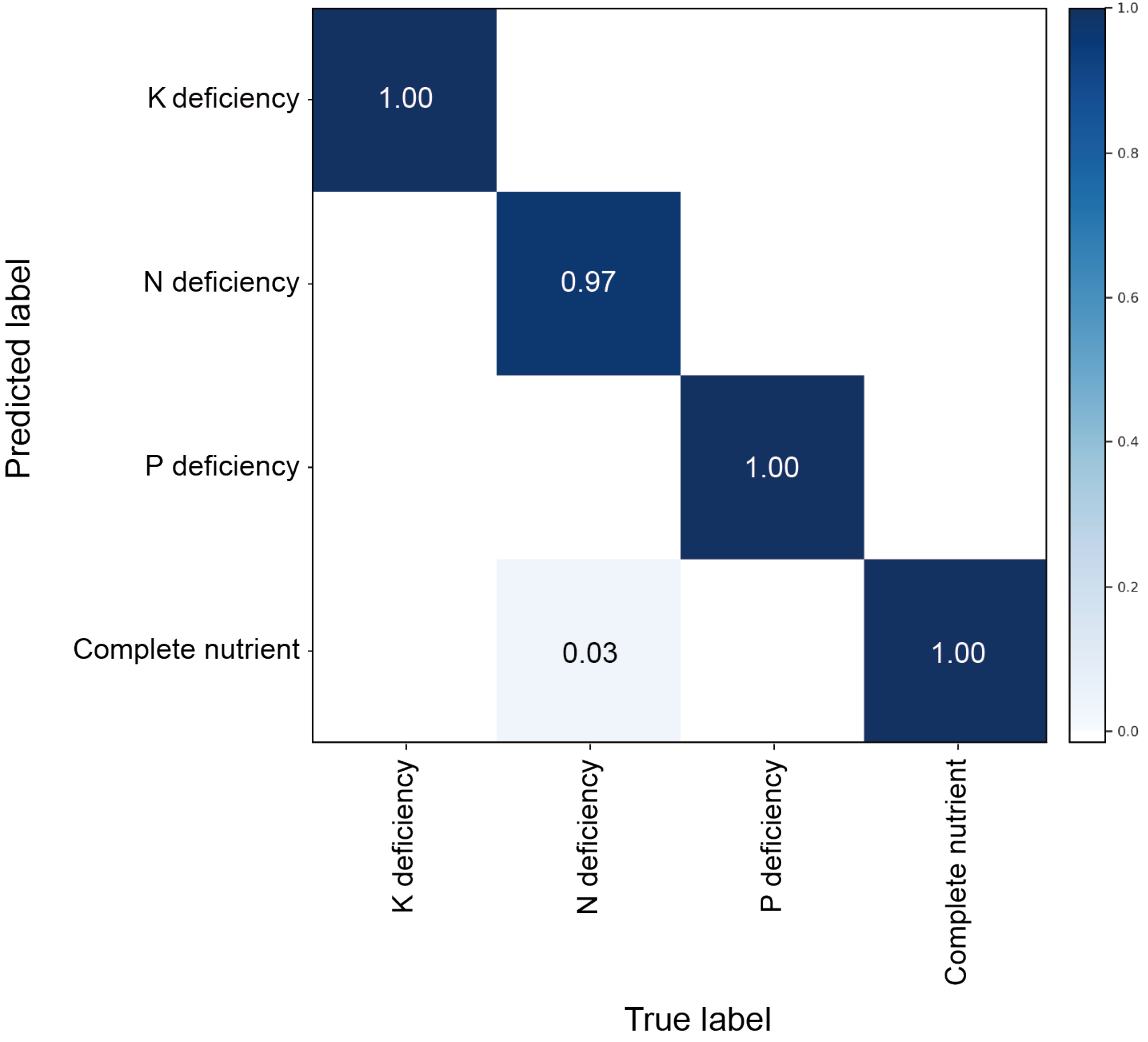
10. End.



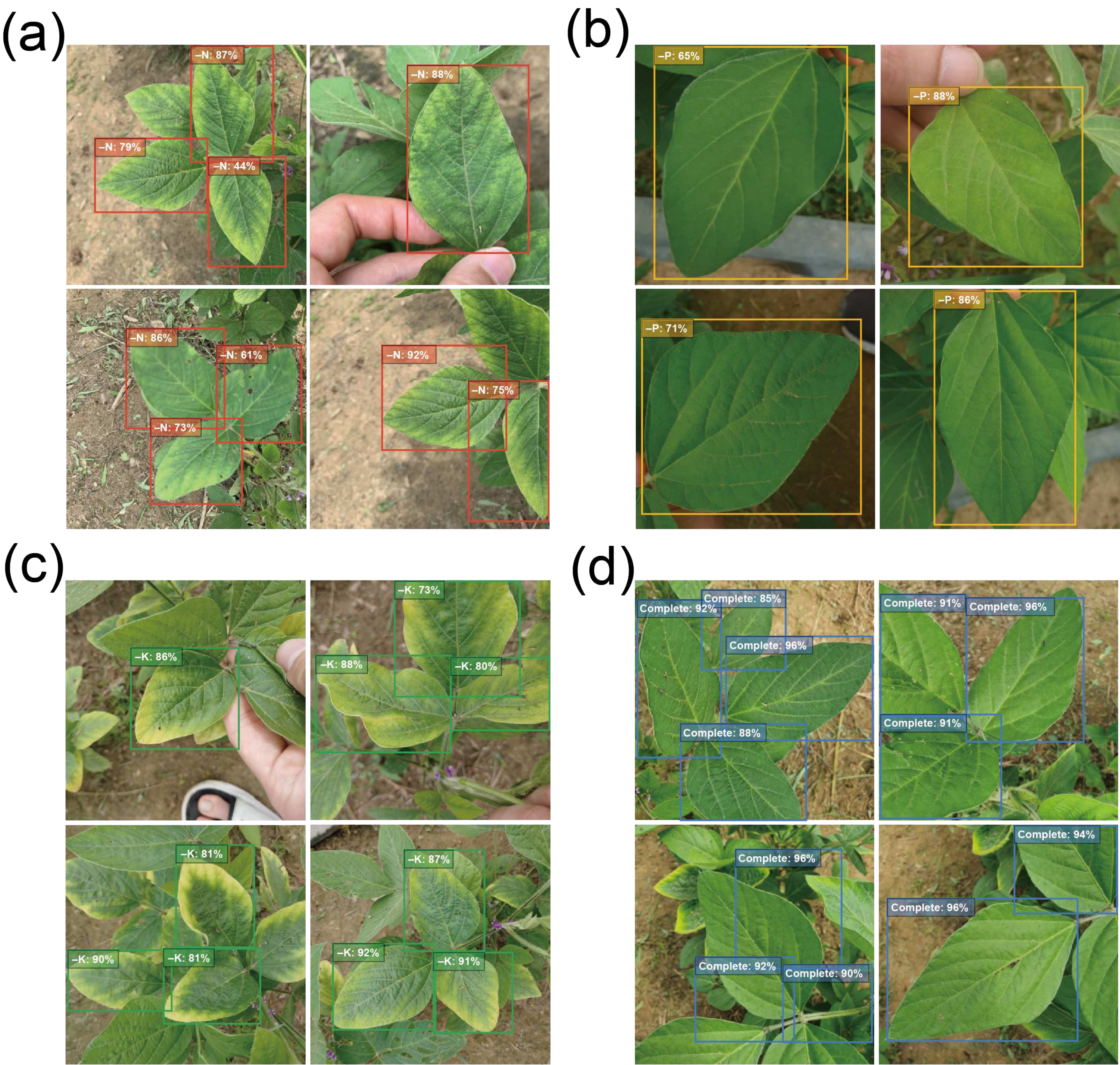
(a)
train/box_loss
(b)
val/box_loss
(c)
train/cls_loss
results
smooth
(d)
val/cls_loss
(e)
train/dfl_loss
(f)
val/dfl_loss
Figure
(g)
metrics/precision(B)
(h)
metrics/mAP50(B)
(i)
metrics/recall(B)
(j)
metrics/mAP50-95(B)



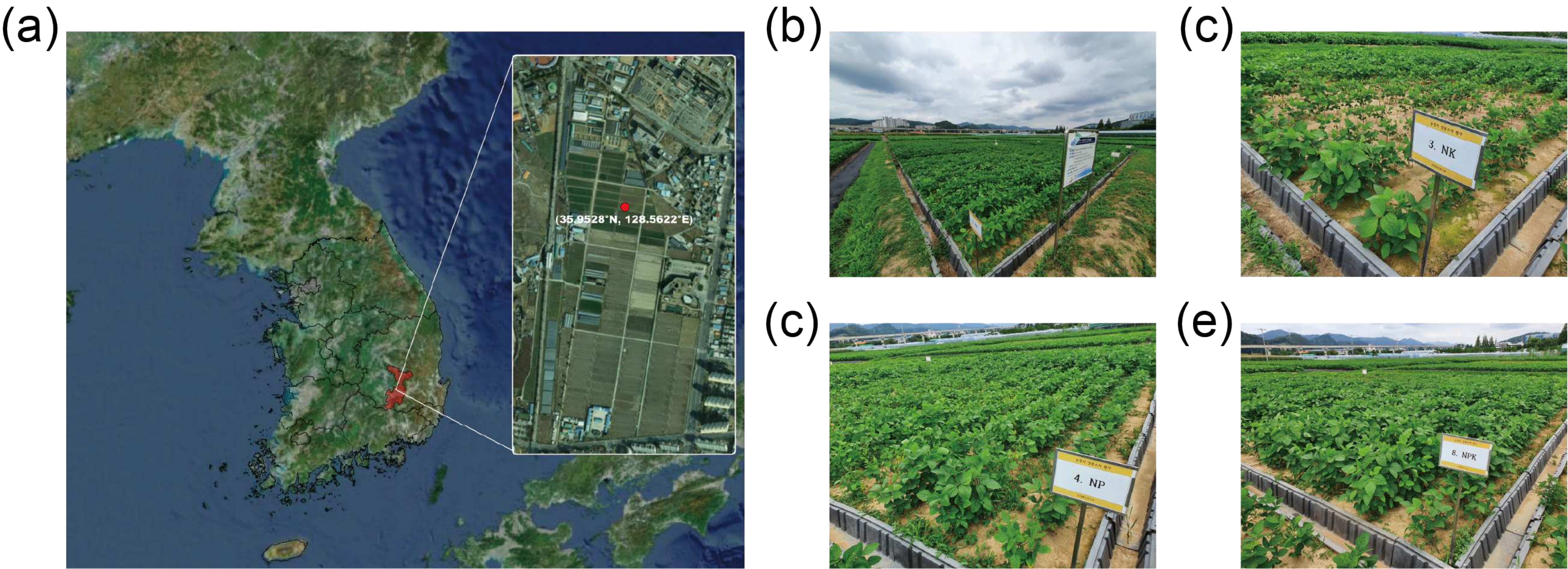
(a)
Precision-Confidence Curve
Precision
Confidence
K deficiency
N deficiency
P deficiency
Complete nutrient
All classes 1.00 at 0.950
(b)
Precision-Recall Curve
Precision
Recall
K deficiency 0.984
N deficiency 0.976
P deficiency 0.995
Complete nutrient 0.990
All classes 0.986 mAP@0.5
(c)
F1-Confidence Curve
F1
Confidence
K deficiency
N deficiency
P deficiency
Complete nutrient
All classes 0.95 at 0.663
(d)
Recall-Confidence Curve
Recall
Confidence
K deficiency
N deficiency
P deficiency
Complete nutrient
All classes 1.00 at 0.000

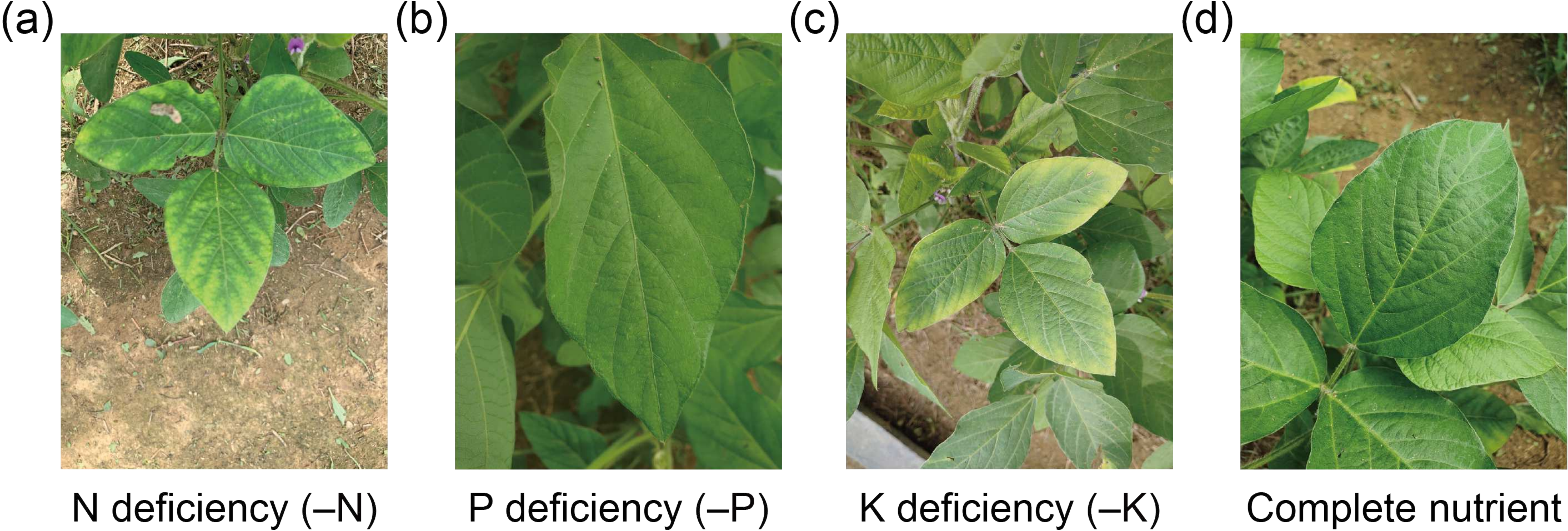


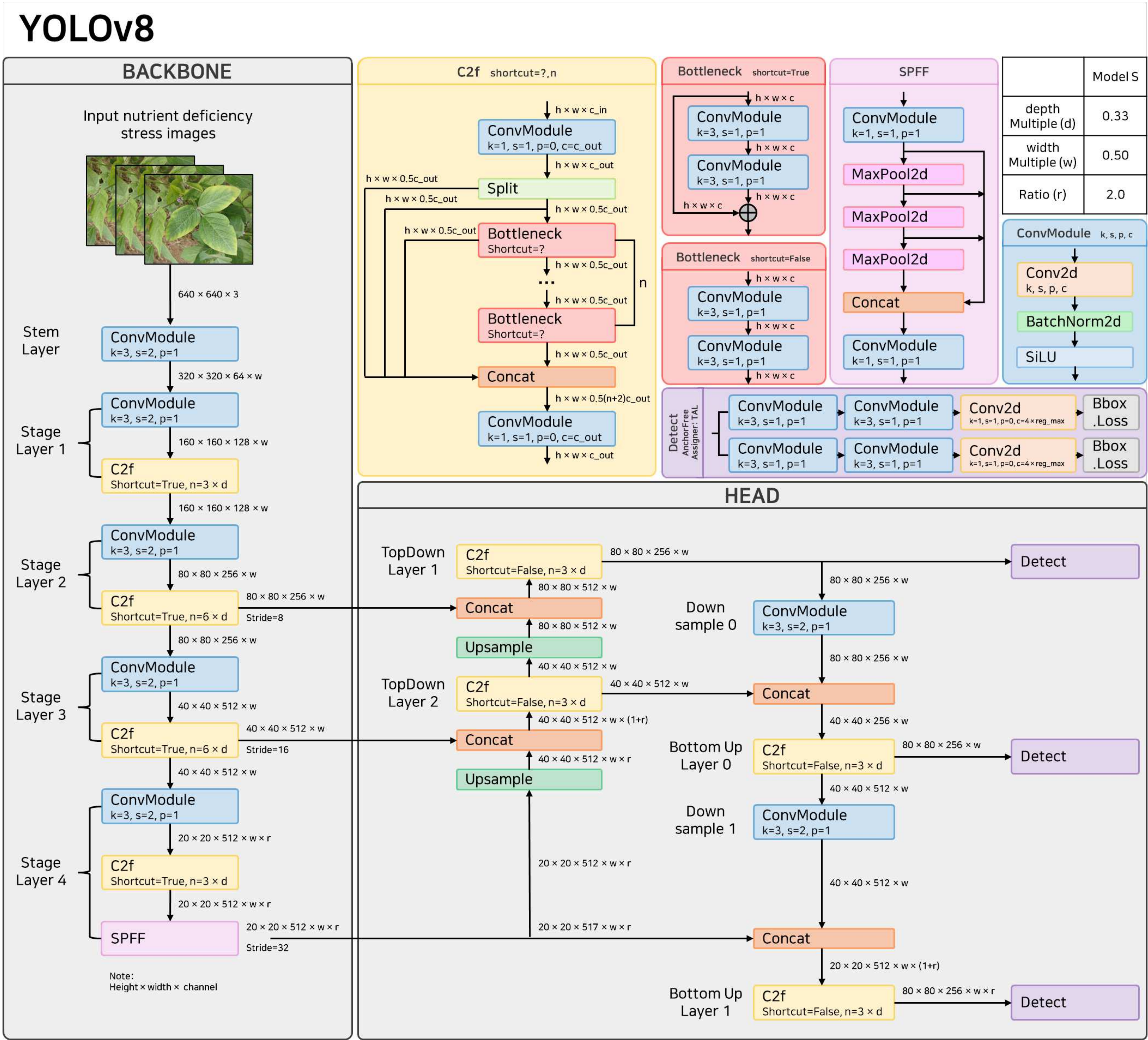
Predicted label
K deficiency
N deficiency
P deficiency
Complete nutrient
1.00
0.97
1.00
0.03
1.00
True label
K deficiency
N deficiency
P deficiency
Complete nutrient
1.0
0.8
0.6
0.4
0.2
0.0



(a)
–N: 87%
–N: 79%
–N: 44%
–N: 88%
–N: 86%
–N: 61%
–N: 73%
–N: 92%
–N: 75%
(b)
–P: 65%
–P: 88%
–P: 71%
–P: 86%
(c)
–K: 86%
–K: 73%
–K: 88%
–K: 80%
–K: 81%
–K: 90%
–K: 81%
–K: 87%
–K: 92%
–K: 91%
(d)
Complete: 85%
Complete: 92%
Complete: 96%
Complete: 88%
Complete: 91%
Complete: 96%
Complete: 91%
Complete: 96%
Complete: 92%
Complete: 90%
Complete: 94%
Complete: 96%


(a)
(35.9528°N, 128.5622°E)
(b)
(c)
3. NK
(c)
4. NP
(e)
8. NPK


(a)
(b)
(c)
(d)
N deficiency (–N)
P deficiency (–P)
K deficiency (–K)
Complete nutrient



YOLOv8
BACKBONE
Input nutrient deficiency stress images
640 × 640 × 3
Stem Layer
ConvModule k=3, s=2, p=1
320 × 320 × 64 × w
Stage Layer 1
ConvModule k=3, s=2, p=1
160 × 160 × 128 × w
C2f Shortcut=True, n=3 × d
160 × 160 × 128 × w
Stage Layer 2
ConvModule k=3, s=2, p=1
80 × 80 × 256 × w
C2f Shortcut=True, n=6 × d
80 × 80 × 256 × w
Stride=8
80 × 80 × 256 × w
Stage Layer 3
ConvModule k=3, s=2, p=1
40 × 40 × 512 × w
C2f Shortcut=True, n=6 × d
40 × 40 × 512 × w
Stride=16
40 × 40 × 512 × w
Stage Layer 4
ConvModule k=3, s=2, p=1
20 × 20 × 512 × w × r
C2f Shortcut=True, n=3 × d
20 × 20 × 512 × w × r
SPFF
20 × 20 × 512 × w × r
Stride=32
Note: Height × width × channel
C2f shortcut=?,n
h × w × c_in
ConvModule k=1, s=1, p=0, c=c_out
h × w × c_out
Split
h × w × 0.5c_out
h × w × 0.5c_out
h × w × 0.5c_out
h × w × 0.5c_out
Bottleneck Shortcut=?
h × w × 0.5c_out
n
h × w × 0.5c_out
Bottleneck Shortcut=?
h × w × 0.5c_out
Concat
h × w × 0.5(n+2)c_out
ConvModule k=1, s=1, p=0, c=c_out
h × w × c_out
Bottleneck shortcut=True
h × w × c
ConvModule k=3, s=1, p=1
h × w × c
ConvModule k=3, s=1, p=1
h × w × c
h × w × c
Bottleneck shortcut=False
h × w × c
ConvModule k=3, s=1, p=1
h × w × c
ConvModule k=3, s=1, p=1
h × w × c
SPFF
ConvModule k=1, s=1, p=1
MaxPool2d
MaxPool2d
MaxPool2d
Concat
ConvModule k=1, s=1, p=1
Model S
depth Multiple (d) 0.33
width Multiple (w) 0.50
Ratio (r) 2.0
ConvModule k, s, p, c
Conv2d k, s, p, c
BatchNorm2d
SiLU
Detect AnchorFree Assigner: TAL
ConvModule k=3, s=1, p=1
ConvModule k=3, s=1, p=1
Conv2d k=1, s=1, p=0, c=4×reg_max
Bbox .Loss
ConvModule k=3, s=1, p=1
ConvModule k=3, s=1, p=1
Conv2d k=1, s=1, p=0, c=4×reg_max
Bbox .Loss
HEAD
TopDown Layer 1
C2f Shortcut=False, n=3 × d
80 × 80 × 256 × w
Detect
80 × 80 × 512 × w
Concat
80 × 80 × 512 × w
Upsample
40 × 40 × 512 × w
TopDown Layer 2
C2f Shortcut=False, n=3 × d
40 × 40 × 512 × w
40 × 40 × 512 × w × (1+r)
Concat
40 × 40 × 512 × w × r
Upsample
20 × 20 × 512 × w × r
20 × 20 × 517 × w × r
80 × 80 × 256 × w
Down sample 0
ConvModule k=3, s=2, p=1
80 × 80 × 256 × w
Concat
40 × 40 × 256 × w
Bottom Up Layer 0
C2f Shortcut=False, n=3 × d
80 × 80 × 256 × w
Detect
40 × 40 × 512 × w
Down sample 1
ConvModule k=3, s=2, p=1
40 × 40 × 512 × w
Concat
20 × 20 × 512 × w × (1+r)
Bottom Up Layer 1
C2f Shortcut=False, n=3 × d
80 × 80 × 256 × w × r
Detect

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementarymetarialsubmission.pdf