ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Hybrid Deep Learning and Machine Learning Framework for Automated Tomato Leaf Disease Classification

Nguyen Viet Hung[1], Phi Dinh Huynh[2], Nguyen Thi Ngoc Thoa[3], Le Mai Nam[4],
Le Thi Huyen Trang[5] and Trong-Minh Hoang[6]

ABSTRACT

Tomato leaf diseases significantly impact crop productivity, necessitating accurate and efficient diagnostic tools. This study proposes a hybrid framework that integrates deep learning-based localization and segmentation with handcrafted feature extraction and classical machine learning for tomato leaf disease classification. Specifically, YOLOv8 is used for object detection and SAM for segmenting diseased regions. Features are then extracted using HSV color space, GLCM, and LBP descriptors. To address class imbalance, the SMOTE technique was applied, expanding the original 48,243-image dataset to 102,465 balanced samples across 11 disease categories. Multiple classifiers were evaluated, with Random Forest achieving the highest performance—over 90% accuracy and a macro F1-score of 0.90. Importantly, recall for minority classes improved markedly after balancing. The proposed system demonstrates strong potential for deployment in real-world agricultural environments due to its low computational cost and robustness under varying conditions. Future work will explore multi-crop generalization, real-time inference, and field validation under challenging conditions such as lighting variation and occlusion.

Article information:

Keywords: Machine Learning, Deep Learning, Tomato leaf diseases, YOLO (You Only Look Once), SMOTE (Synthetic Minority Oversampling Technique), Random Forest

Article history:
Received: September 11, 2025
Revised: October 16, 2025
Accepted: October 23, 2025
Published: October 31, 2025
(Online)

DOI: 10.37936/ecti-cit.2025194.262810

1. INTRODUCTION

Plant diseases, particularly fungal, bacterial, and viral infections, pose a significant threat to global agricultural productivity, with tomatoes being among the most vulnerable crops [1-3]. Early and accurate detection of these diseases is critical not only for reducing yield loss but also for enabling timely and effective intervention strategies. However, traditional diagnostic methods relying on visual inspection or laboratory analysis are often labor-intensive, time-consuming, and dependent on expert knowledge.

Recent advancements in machine learning (ML) and computer vision (CV) have revolutionized the field of plant disease detection, offering scalable and accurate solutions through automated analysis of leaf images [4-7]. Unlike conventional methods that rely heavily on farmer expertise and manual inspection, modern techniques leverage the power of deep learning (DL) and hybrid machine learning models to achieve superior precision and efficiency.

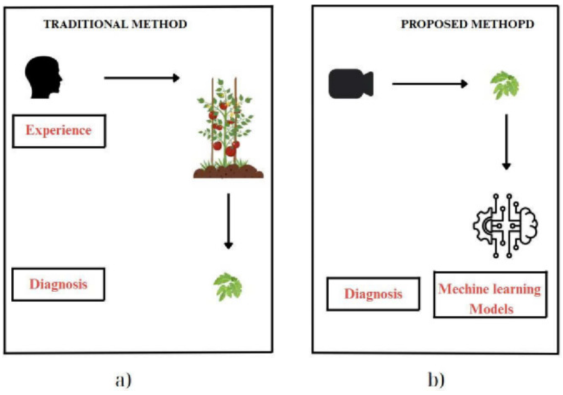


Fig.1: *Traditional and modern methods of identifying tomato leaf diseases.*

[1,2,3,4,5] The authors are with the East Asia University of Technology, BacNinh, Vietnam, E-mail: hungnv@eaut.edu.vn, 20222072@eaut.edu.vn, 20220752@eaut.edu.vn, namlm@eaut.edu.vn and tranglth@eaut.edu.vn

[6] The author is with the Posts and Telecommunications Institute of Technology, Hanoi, Vietnam. E-mail: hoangtrongminh@ptit.edu.vn

[1] Corresponding author: hungnv@eaut.edu.vn

Traditional approaches, as illustrated in Figure 1a, depend on visual diagnosis by farmers or experts. While valuable in certain contexts, these methods are often subjective, time-consuming, and limited in scalability. In contrast, modern systems (Figure 1b utilize high-resolution imaging devices paired with artificial intelligence (AI) models to identify diseases in tomato leaves with remarkable accuracy and speed. This technological advancement addresses the limitations of traditional methods, offering a robust solution for large-scale disease management in agriculture.

To further enhance predictive performance, recent studies have explored the integration of classical image processing techniques with state-of-the-art AI models. For instance, hybrid frameworks combining Gray-Level Co-occurrence Matrix (GLCM) with Convolutional Neural Networks (CNNs) [8] or GLCM and Local Binary Patterns (LBP) with Support Vector Machines (SVMs) [9] have shown promising results in detecting diseases like Early and Late Blight. These approaches highlight the potential of blending traditional feature extraction techniques with modern algorithms to improve accuracy and robustness.

Despite these advancements, there remain key challenges such as dataset variability, model generalizability, and the need for computational efficiency in resource-constrained environments. Addressing these challenges is critical to unlocking the full potential of AI-driven plant disease detection systems, making them accessible and reliable for real-world agricultural applications.

In this study, we propose a robust framework that integrates classical image processing techniques with traditional machine learning models to classify tomato leaf diseases. The framework utilizes segmented images produced by YOLO-SAM and extracts features such as HSV color statistics, Gray-Level Co-occurrence Matrix (GLCM) texture features, and Local Binary Patterns (LBP). These features are then used as inputs for multiple classifiers, including Random Forest, k-Nearest Neighbors (kNN), Support Vector Machines (SVM), and Naive Bayes.

The goal of this research is to evaluate and compare the performance of these classifiers in recognizing eleven distinct categories of tomato leaf conditions, including healthy leaves and common diseases. By employing performance visualization tools and dimensionality reduction techniques, we aim to uncover underlying data patterns and provide insights for the development of lightweight, interpretable, and cost-effective solutions in smart agriculture systems.

2. RELATED WORK

Image-based plant disease detection has become a widely researched area due to its potential for providing cost-effective and efficient diagnostic tools for agriculture. Traditional methods have primarily focused on handcrafted feature extraction techniques, leveraging the visual properties of leaves such as color, texture, and shape. Popular features include Haralick descriptors from Gray-Level Co-occurrence Matrix (GLCM), color histograms in HSV space, and Local Binary Patterns (LBP). These features are commonly used with classical machine learning classifiers, such as Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Random Forests (RF), and Naive Bayes, which have demonstrated promising results for identifying plant diseases [10, 11].

With the advent of deep learning, convolutional neural networks (CNNs) have emerged as the dominant approach in plant disease detection, owing to their ability to automatically extract meaningful features from raw image data. Transfer learning using pretrained models such as VGG, ResNet, and EfficientNet has further enhanced the accuracy of disease classification tasks [12, 13]. However, deep learning approaches are often resource-intensive, requiring large labeled datasets and substantial computational resources, which limits their application in low-resource environments or for real-time use.

In addition to classification, object detection and image segmentation have gained increasing attention as preprocessing steps to handle real-world challenges, such as complex backgrounds and overlapping leaves. Unsupervised clustering algorithms, such as Fuzzy C-Means (FCM) and K-Means, have been employed to isolate leaves from noisy backgrounds. For instance, the study in [14] demonstrated that FCM-based segmentation improved classification accuracy to approximately 85%. Similarly, advanced object detection models, including YOLO (You Only Look Once) and Faster R-CNN, have been utilized to detect and extract plant leaves effectively in complex scenarios [15]. Furthermore, segmentation models such as U-Net, Mask R-CNN, and DeepLab have been applied to identify infected regions on leaves, enhancing the relevance of extracted features for downstream classification tasks [16].

A recent advancement in segmentation is the introduction of the Segment Anything Model (SAM) by Meta AI in 2023. SAM has demonstrated remarkable performance in general-purpose segmentation tasks across diverse domains. However, its application in specialized areas, such as plant pathology, remains underexplored. Early efforts, including studies by [17,18], as well as the recent work by Islam *et al.* (2025) [14], have started to investigate SAM's potential for plant disease detection. These studies highlight that integrating SAM with object detection frameworks could significantly improve segmentation precision, particularly in agricultural settings.

Despite these advancements, there remains a gap in fully integrating modern object detection and segmentation frameworks with traditional machine learning pipelines to create lightweight, accurate, and resource-efficient systems. To address this, our study proposes a hybrid approach that combines YOLOv8 for object detection, SAM for precise leaf segmentation, and handcrafted feature extraction (HSV, GLCM, LBP) to train classical machine learning models [19]. This novel pipeline aims to achieve high accuracy while maintaining low computational complexity, making it particularly suitable for practical applications in resource-constrained environments.

3. METHODOLOGY

This section outlines the complete pipeline designed for tomato leaf disease classification, which integrates modern computer vision techniques with traditional machine learning classifiers. The overall workflow is illustrated in Figure 2.

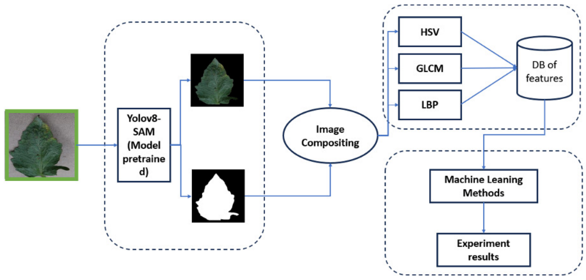


Fig.2: *Proposed system architecture integrating YOLOv8 and SAM with feature-based classification pipeline.*

The proposed Algorithm showed in Figure **??** outlines a comprehensive two-phase pipeline for automated tomato leaf disease classification, integrating deep learning-based segmentation and traditional machine learning-based classification.

Phase 1 – Leaf Segmentation:

This phase begins by feeding an input tomato leaf image I_i into the YOLOv8 object detector, which identifies a bounding box B_i around the leaf. To refine the segmentation, the Segment Anything Model (SAM) is prompted using the center point of B_i, thereby generating a precise binary mask M_i that delineates the leaf region. The original image is then element-wise multiplied by this mask to isolate the foreground leaf region $L_i = I_i * M_i$. This hybrid segmentation step is designed to combine YOLO's robust object localization with SAM's fine-grained pixel-wise segmentation, ensuring a cleaner background removal process prior to feature analysis.

Phase 2 – Feature Extraction and Training:

With the segmented leaf region L_i, a set of handcrafted features is extracted to describe the visual and structural characteristics of the leaf:

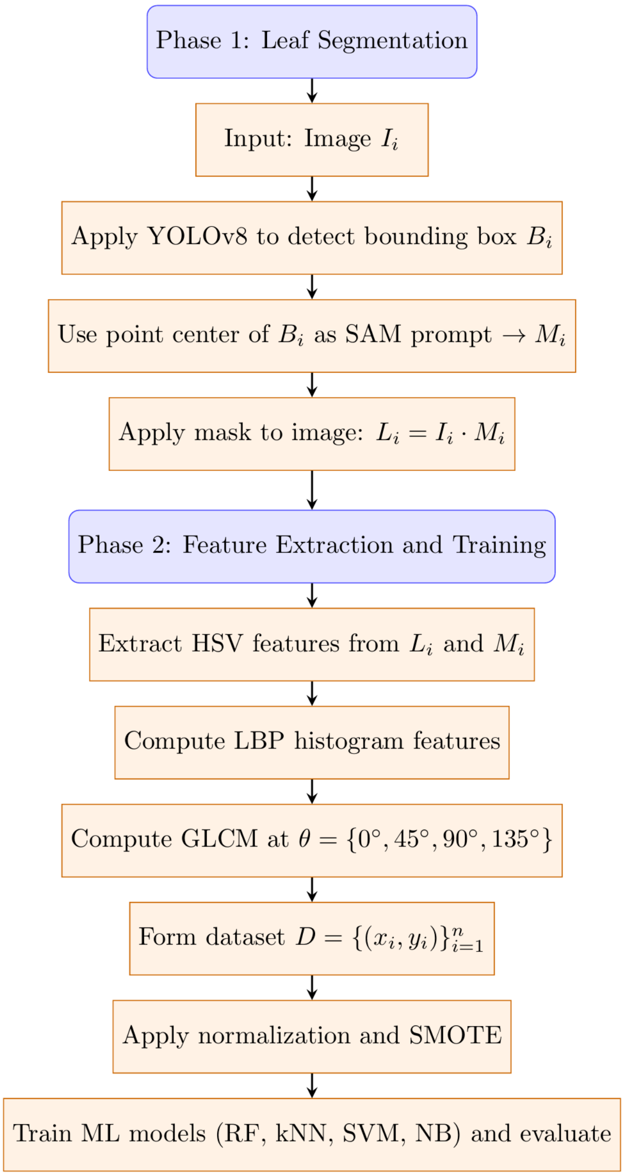


Fig.3: *Workflow of the proposed YOLO+SAM-based framework for tomato leaf disease classification.*

- Color features are derived in the HSV color space to capture variations in hue, saturation, and brightness that may correspond to disease symptoms.
- Texture descriptors include Local Binary Pattern (LBP) histograms, which capture local texture patterns, and Gray-Level Co-occurrence Matrix (GLCM) features computed across four directional offsets (0°, 45°, 90° and 135°) to represent spatial relationships of pixel intensities at multiple orientations.

Each feature vector x_i, along with its corresponding class label y_i, is stored to construct a dataset $\mathcal{D}$, where n denotes the total number of processed samples. This dataset serves as the basis for subsequent steps such as normalization, class balancing, and model training.

Specifically::

- i is the image (i = 1 for default)
- B_i is the bounding box of the image i
- M_i is the mask of the image i
- L_i is the result when I_i time M_i

Table 1: *The average results are measured after 3 runs with the seed (10/16/42). The YOLOv8+SAM framework achieves the highest mIoU (0.92) and Dice (0.94), outperforming Mask R-CNN (0.80/0.83) and U-Net (0.84/0.88).*

Model	**mIoU**	**Dice**
Mask R-CNN	±0.80	±0.83
U-Net	±0.84	±0.88
Yolov8+SAM	**±0.92**	**±0.94**

- D is the dataset with x_i, y_i are feature vector and corresponding label, respectively. n is the size of the dataset after feature extraction.

3.1 Input and Preprocessing

The input to the system is an RGB image of a tomato leaf, which may contain noise or complex backgrounds due to real-world conditions.

3.2 Object Detection and Segmentation

The first stage applies a YOLOv8 object detection model pretrained on agricultural datasets to detect the presence and location of tomato leaves in the image. The bounding box from YOLOv8 is then passed to the Segment Anything Model (SAM), which produces two outputs:

- A segmented foreground leaf image.
- A binary mask highlighting the exact region of the leaf.

This combination allows for both localization and accurate segmentation of the leaf, minimizing background interference.

The Yolov8 model which we collected from [20] and then further customized our dataset with the following configuration: epochs = 150, batch_size = 16, img_size = 256 × 256, learning_rate = 0.001, augmentation = True, optimizer = SGD with momentum = 0.937 and weight_decay = 0.0005. The SAM model is used from [21] and we do not customize any of its parameters.

At the same time, we also fine-tuned the U-Net and Mask R-CNN models to enhance segmentation performance. The configuration for the U-Net model is as follows: image size = 256 × 256, batch size = 8, number of epochs = 100, learning rate = 0.001, loss function = Dice loss, and backbone = ResNet34.

For the Mask R-CNN, we adopted a configuration optimized for small medical objects, consisting of: input image size = 256 × 256, batch size = 4, epochs = 100, learning rate = 0.0005, backbone = ResNet50, and optimizer = Adam. The model was fine-tuned on the dataset with pretrained COCO weights to accelerate convergence and improve generalization.

The Figure 4 provides a comparative visualization of segmentation results generated by several models, including Mask R-CNN and U-Net, all trained on the tomato leaf dataset. Among these, the proposed YOLOv8+SAM framework delivers the highest segmentation quality, effectively preserving fine-grained leaf contours and lesion details. In contrast, Mask R-CNN shows the greatest loss of detail, especially along the leaf boundaries and in small diseased regions. U-Net exhibits moderate performance, but it still falls short in spatial precision compared to the YOLOv8+SAM approach.

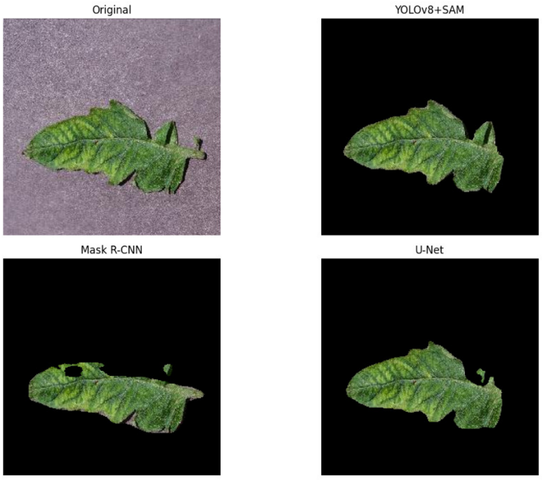


Fig.4: *Compare the segmentation method proposed with other models.*

As shown in Table 1, the YOLOv8+SAM framework achieves better accuracy than the other two models, so YOLOv8+SAM is chosen for the image segmentation stage. Next we apply an ROI filter, which works to filter $ROIs < 85\%$ to avoid generating noisy features.

3.3 Image Compositing

Using the segmentation mask and the cropped leaf image, an image compositing process is conducted to isolate the leaf region by removing the background. This step ensures that only relevant leaf regions are preserved for feature extraction.

3.4 Feature Extraction Methods

In order to capture both color and texture information of the segmented tomato leaves, we extracted features from three main descriptors: HSV color space, Gray-Level Co-occurrence Matrix (GLCM), and Local Binary Pattern (LBP). These features serve as inputs for the machine learning classifiers.

3.4.1 HSV Color Features

The RGB image is first converted to the HSV color space to better separate chromatic content (hue and saturation) from brightness (value), which is more robust under varying lighting conditions.

For each channel (H, S, V), the mean intensity is calculated by equation 1:

$$\mu_C = \frac{1}{N}\sum_{i=1}^{N} C_i, C \in \{H, S, V\} \tag{1}$$

Where:

- C_i is the intensity of pixel i in channel C
- N is total number of pixels

3.4.2 GLCM Texture Features

To capture texture information from segmented tomato leaf regions, we computed the Gray-Level Co-occurrence Matrix (GLCM) with a pixel distance of 1 and across four orientations: 0° (horizontal), 45° (diagonal), 90° (vertical), and 135° (anti-diagonal). These angles were chosen to account for directional variations in texture, as disease symptoms such as spotting, lesions, or mold can exhibit anisotropic patterns. By incorporating multiple directions, the extracted features become more rotation-invariant and robust to the leaf's orientation in the image. From the GLCM, we extract the following four features (These equations are 2, 3, 4 and 5):

$$\text{Contrast} = \sum_{ij} |i-j|^2 \times P(i,j) \tag{2}$$

$$\text{Homogeneity} = \sum_{ij} \frac{P(i,j)}{1+|i-j|} \tag{3}$$

$$\text{Energy} = \sum_{ij} P(i,j)^2 \tag{4}$$

$$\text{Correlation} = \sum_{ij} \frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i\sigma_j} \tag{5}$$

Where:

- $P(i,j)$ is the normalized GLCM value
- μ and σ are the means and standard deviations of the row and column sums.

The GLCM features were computed with a pixel distance of 1 and four directional angles (0°, 45°, 90°, and 135°), using 256 gray levels, symmetric normalization, and critically across orientations. Extracted texture descriptors include contrast, homogeneity, energy, and correlation.

3.4.3 Local Binary Pattern (LBP)

The LBP operator captures local texture by thresholding the neighborhood of each pixel with the center pixel's value. The LBP code is calculated by equation 6:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s(I_p - I_c) \times 2^p \tag{6}$$

Where:

- I_c is the intensity of the center pixel (x_c, y_c)
- I_p are the intensities of the P neighboring pixels
- s(x) = 1 if x $\geq$ 0, else s(x) = 0

We configure the LBP parameters as follows: Using a uniform sample with radius = 1 and P = 8 sampling points, generate a 10-bin normalized histogram for each occluded leaf region.

3.5 Classification

To evaluate the discriminative power of the extracted features, several traditional machine learning classifiers were employed. These models were selected due to their proven effectiveness in various image classification tasks, especially when working with limited data and handcrafted features. The classifiers used in this study include:

- Support Vector Machine (SVM)-[22]: A powerful supervised learning algorithm that constructs a hyperplane in a high-dimensional space to separate classes with the maximum margin. The radial basis function (RBF) kernel was utilized for non-linear separation.

- K-Nearest Neighbors (KNN)-[23]: A non-parametric method that classifies a data point based on the majority label of its **K** nearest neighbors. Euclidean distance was used to measure similarity.

- Random Forest (RF)-[24]: An ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the individual trees outputs. Variants with different values of max-depth, n-estimators, and max-features were explored.

- Gaussian Naive Bayes (GNB)-[25]: A probabilistic classifier based on Bayes theorem with the assumption of feature independence. It assumes that features follow a Gaussian distribution.

All models were trained and evaluated using stratified cross-validation to ensure balanced representation of classes. Performance was assessed through multiple metrics including accuracy, precision, recall, and F1-score (both macro and weighted averages). Hyperparameter tuning was conducted via grid search for SVM and RF models to optimize classification performance.

3.6 Data Collection

The dataset employed in this study consists of tomato leaf images sourced from publicly available and widely recognized repositories. Specifically, we utilized the TomatoVillage dataset [26], which contains annotated images of tomato leaves categorized into several classes, including early blight, late blight, tomato mosaic virus, and healthy samples. Owing to its standardized structure, high quality, and class diversity, this dataset is commonly used in plant disease classification research.

Each image in the dataset is associated with a class label representing the disease (or health condition). The dataset is organized into separate folders for each class, following a directory structure that supports

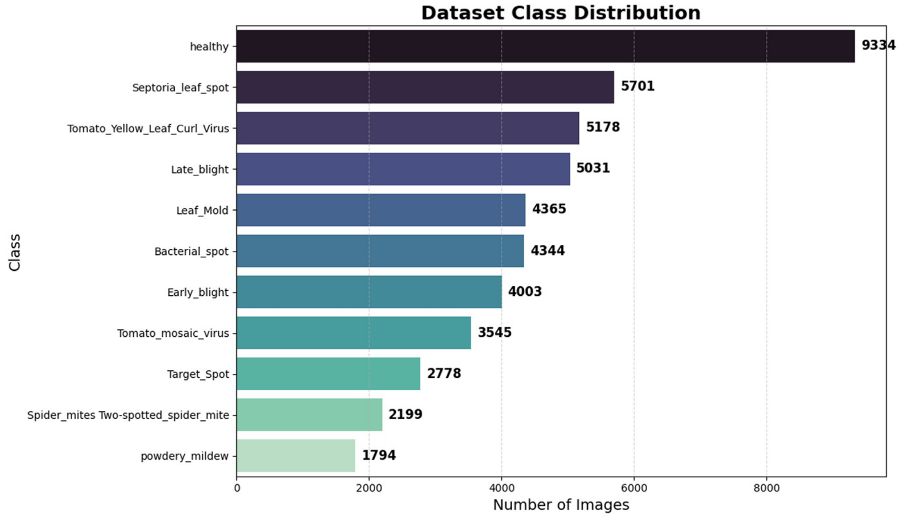


Fig.5: *Dataset Class Diagram.*

Fig.6: *Dataset after SMOTE.*

Fig.7: *t-SNE (t-distributed stochastic neighbor embedding) visualization.*

automatic loading and preprocessing, as shown in Figure 5. To increase generalization and minimize overfitting, the dataset is divided into training (80%) and validation (20%) subsets.

In addition, during training, SMOTE and subsequent data augmentation techniques (rotation, flipping, and enhancement) are applied to the training set of each fold to avoid data leakage to the test set. After applying the data augmentation techniques, the dataset will be distributed as shown in Figure 6. The SOMTE we use in this article is a function from the `imblearn` library in `Python`, the current parameter used is *SMOTE(random_state = 42)*.

4. EXPERIMENTS AND RESULTS

4.1 Experimental Setup

All experiments were conducted on a system equipped with an Intel Core i7-12700 CPU, 32GB of RAM, and an NVIDIA GeForce RTX 3060 GPU, running on 64-bit Window 11 Pro computer. The implementation was done using Python 3.10, with support from key libraries such as OpenCV, Scikit-learn, NumPy, and PyTorch. Image segmentation and object detection were facilitated by the YOLOv8 and Segment Anything Model (SAM) frameworks, while classification tasks were performed using scikit-learn classifiers.

The GLCM features were computed with a pixel distance of 1 and four directional angles (0°, 45°, 90°, and 135°), using 256 gray levels, symmetric normalization, and averaged across orientations. Extracted texture descriptors include contrast, homogeneity, energy, and correlation. Regarding to LBP, we used the uniform pattern with radius = 1 and P = 8 sampling points, generating a 10-bin normalized histogram for each masked leaf region.

4.2 Dataset and Preprocessing

The dataset used in this study consists of tomato leaf images, categorized into multiple classes based on disease type. Images were collected from publicly available repositories and agricultural image banks. Preprocessing involved resizing, denoising, and object-background separation using YOLOv8 and SAM. Segmented leaves were then normalized and resized uniformly to 224×224 pixels.

Feature extraction was conducted on the segmented leaf images using:

- **Color features** (HSV mean values)
- **Texture features** from GLCM: contrast, energy, homogeneity, correlation
- **Local Binary Pattern (LBP)** features: 10-bin histogram

The final feature vector was constructed by concatenating all extracted descriptors.

And then, we employed the Synthetic Minority Over-sampling Technique (SMOTE)-equation 7. SMOTE generates synthetic samples by interpolating feature vectors between each sample of minority class and its neighbors k nearest (typically $k = 5$). Specifically, new samples are created along the line segments connecting a minority sample and randomly selected neighbors in the feature space:

$$x_{new} = x_i + \lambda \times (x_{zi} - x_i), \lambda \in [0,1] \tag{7}$$

Where:

- x_i is a minority class sample
- x_{zi} is one of its k-nearest neighbors
- λ is a random number ranging 0 to 1

4.3 Data Visualization

To further investigate the separability of the extracted features, t-distributed stochastic neighbor embedding (t-SNE) was applied to project the high-dimensional feature space into two dimensions. As shown in Figure 7, the data points exhibit clear class-wise clustering, with samples from similar disease types forming compact regions, while individual classes remain largely separate. This confirms that the combined HSV, GLCM, and LBP descriptors capture meaningful visual and structural patterns relevant to tomato leaf disease discrimination.

4.4 Evaluate Metrics

To rigorously assess the classification performance of the proposed models, we utilized a comprehensive set of standard evaluation metrics, strictly adhering to the values and variable notations specified in reference [27].

- **Accuracy** reflects the overall proportion of correctly predicted samples out of all test instances.:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

- **Precision** measures the proportion of correctly predicted positive samples among all predicted positives:

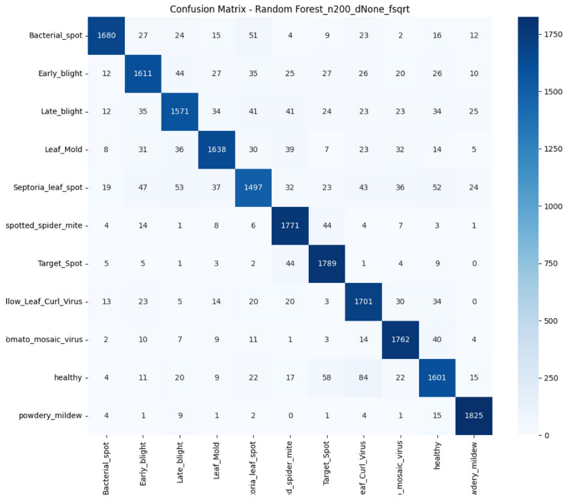
$$Precision = \frac{TP}{TP + FP} \tag{9}$$

- **Recall** (also known as sensitivity or true positive rate) quantifies the proportion of actual positive samples that are correctly identified:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

- **F1_Score** is the harmonic mean of precision and recall, offering a balanced evaluation between the two:

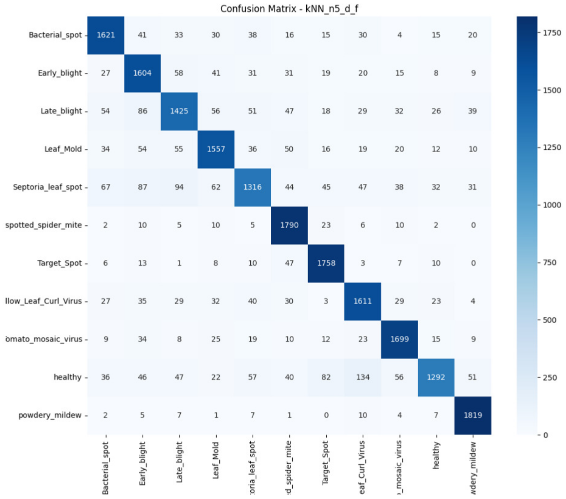
$$F1\_Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$



Confusion Matrix - Random Forest_n200_dNone_fsqrt

	Bacterial_spot	Early_blight	Late_blight	Leaf_Mold	oria_leaf_spot	d_spider_mite	Target_Spot	eaf_Curl_Virus	_mosaic_virus	healthy	wdery_mildew
Bacterial_spot	1680	27	24	15	51	4	9	23	2	16	12
Early_blight	12	1611	44	27	35	25	27	26	20	26	10
Late_blight	12	35	1571	34	41	41	24	23	23	34	25
Leaf_Mold	8	31	36	1638	30	39	7	23	32	14	5
Septoria_leaf_spot	19	47	53	37	1497	32	23	43	36	52	24
spotted_spider_mite	4	14	1	8	6	1771	44	4	7	3	1
Target_Spot	5	5	1	3	2	44	1789	1	4	9	0
llow_Leaf_Curl_Virus	13	23	5	14	20	20	3	1701	30	34	0
omato_mosaic_virus	2	10	7	9	11	1	3	14	1762	40	4
healthy	4	11	20	9	22	17	58	84	22	1601	15
powdery_mildew	4	1	9	1	2	0	1	4	1	15	1825

Fig.8: *Confusion Matrix of Random Forest.*



Confusion Matrix - kNN_n5_d_f

	Bacterial_spot	Early_blight	Late_blight	Leaf_Mold	oria_leaf_spot	d_spider_mite	Target_Spot	eaf_Curl_Virus	_mosaic_virus	healthy	wdery_mildew
Bacterial_spot	1621	41	33	30	38	16	15	30	4	15	20
Early_blight	27	1604	58	41	31	31	19	20	15	8	9
Late_blight	54	86	1425	56	51	47	18	29	32	26	39
Leaf_Mold	34	54	55	1557	36	50	16	19	20	12	10
Septoria_leaf_spot	67	87	94	62	1316	44	45	47	38	32	31
spotted_spider_mite	2	10	5	10	5	1790	23	6	10	2	0
Target_Spot	6	13	1	8	10	47	1758	3	7	10	0
llow_Leaf_Curl_Virus	27	35	29	32	40	30	3	1611	29	23	4
omato_mosaic_virus	9	34	8	25	19	10	12	23	1699	15	9
healthy	36	46	47	22	57	40	82	134	56	1292	51
powdery_mildew	2	5	7	1	7	1	0	10	4	7	1819

Fig.9: *Confusion Matrix of K-Nearest Neighbors.*

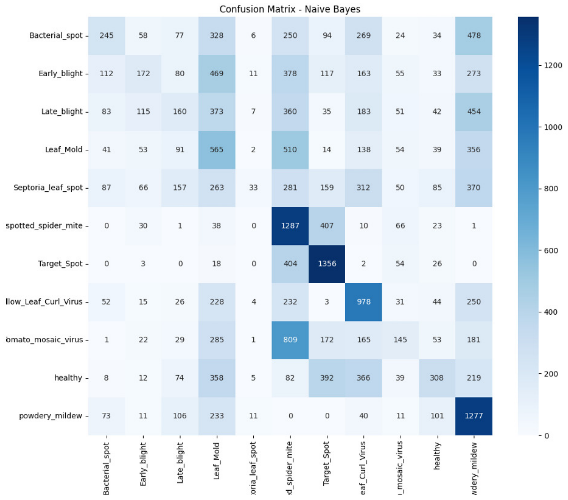
Where:

- TP is the number of true positives
- TN is the number of true negatives
- FP is the number of false positives
- FN is the number of false negatives

To provide a more comprehensive evaluation, these metrics were computed using both **macro averaging** and **weighted averaging**:

- **Macro-average** computes the metric independently for each class and averages the results, treating all classes equally regardless of their frequency.
- **Weighted-average** accounts for the number of instances per class (support), thus emphasizing the contribution of more frequent classes.

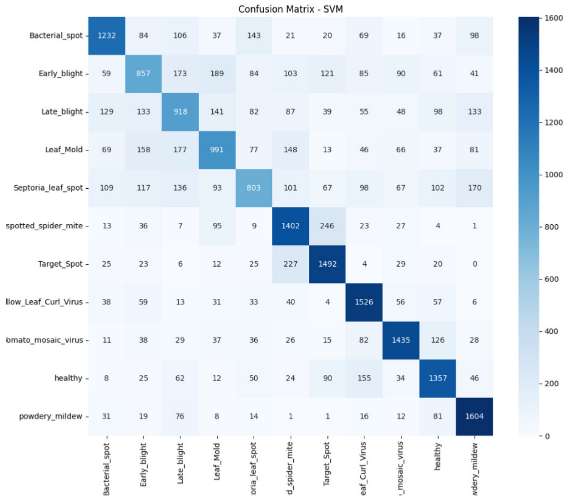
Confusion matrices (showed in Figures 8, 9, 10, 11) were also plotted for visual interpretation of classification results across different classes.



Confusion Matrix - Naive Bayes

	Bacterial_spot	Early_blight	Late_blight	Leaf_Mold	oria_leaf_spot	d_spider_mite	Target_Spot	eaf_Curl_Virus	_mosaic_virus	healthy	wdery_mildew
Bacterial_spot	245	58	77	328	6	250	94	269	24	34	478
Early_blight	112	172	80	469	11	378	117	163	55	33	273
Late_blight	83	115	160	373	7	360	35	183	51	42	454
Leaf_Mold	41	53	91	565	2	510	14	138	54	39	356
Septoria_leaf_spot	87	66	157	263	33	281	159	312	50	85	370
spotted_spider_mite	0	30	1	38	0	1287	407	10	66	23	1
Target_Spot	0	3	0	18	0	404	1356	2	54	26	0
llow_Leaf_Curl_Virus	52	15	26	228	4	232	3	978	31	44	250
omato_mosaic_virus	1	22	29	285	1	809	172	165	145	53	181
healthy	8	12	74	358	5	82	392	366	39	308	219
powdery_mildew	73	11	106	233	11	0	0	40	11	101	1277

Fig.10: *Confusion Matrix of Naive Bayes.*



Confusion Matrix - SVM

	Bacterial_spot	Early_blight	Late_blight	Leaf_Mold	oria_leaf_spot	d_spider_mite	Target_Spot	eaf_Curl_Virus	_mosaic_virus	healthy	wdery_mildew
Bacterial_spot	1232	84	106	37	143	21	20	69	16	37	98
Early_blight	59	857	173	189	84	103	121	85	90	61	41
Late_blight	129	133	918	141	82	87	39	55	48	98	133
Leaf_Mold	69	158	177	991	77	148	13	46	66	37	81
Septoria_leaf_spot	109	117	136	93	803	101	67	98	67	102	170
spotted_spider_mite	13	36	7	95	9	1402	246	23	27	4	1
Target_Spot	25	23	6	12	25	227	1492	4	29	20	0
llow_Leaf_Curl_Virus	38	59	13	31	33	40	4	1526	56	57	6
omato_mosaic_virus	11	38	29	37	36	26	15	82	1435	126	28
healthy	8	25	62	12	50	24	90	155	34	1357	46
powdery_mildew	31	19	76	8	14	1	1	16	12	81	1604

Fig.11: *Confusion Matrix of Support Vector Machine.*

4.5 Performance and Model Comparison

To evaluate the impact of SMOTE on model performance, we conducted experiments with and without applying SMOTE, as illustrated in Figure 12. Evaluation metrics, including Accuracy, Precision, Recall, and F1-Score, were averaged to compute a final composite score referred to as the "Score." The results indicate that applying SMOTE improved model accuracy by approximately 5% to 15% during the prediction process.

Table 2 presents the performance comparison of various classification models on the tomato leaf disease dataset. The models evaluated include k-Nearest Neighbors (kNN) with different values of k, Naive Bayes, Support Vector Machines (SVM), and Random Forest classifiers with varying numbers of estimators and tree depths.

The Random Forest classifier, employing 200 estimators without depth limitation, demonstrated supe-

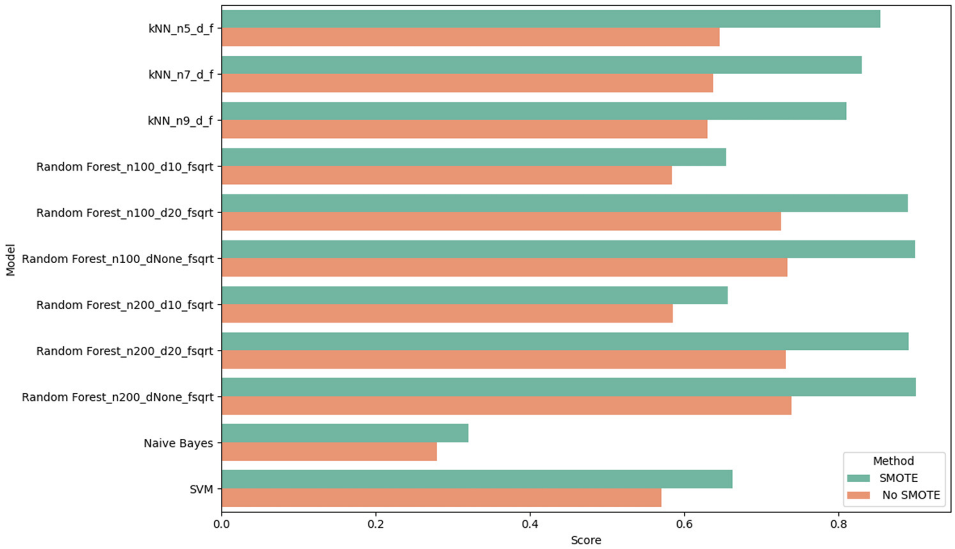


Fig.12: *Evaluation Comparison between using SMOTE and no using SMOTE.*

Table 2: *Evaluation results of models on the tomato leaf disease dataset.*

Model	n_estimators	max_depth	max_features	Accuracy (W+M)	Precision (W+M)	Recall (W+M)	F1-Score (W+M)
kNN	5	–	–	0.8535	0.8503	0.8445	0.8453
kNN	7	–	–	0.8304	0.8302	0.8304	0.8277
kNN	9	–	–	0.8106	0.8103	0.8106	0.8076
Naive Bayes	–	–	–	0.3184	0.3242	0.3185	0.2665
SVM	–	–	–	0.6644	0.6576	0.6646	0.6575
Random Forest	100	10	sqrt	0.6541	0.6541	0.6540	0.6460
Random Forest	100	20	sqrt	0.89	0.89	0.89	0.8893
Random Forest	100	None	sqrt	0.8990	0.8989	0.8990	0.8984
Random Forest	200	10	sqrt	0.6565	0.6564	0.6565	0.6484
Random Forest	200	20	sqrt	0.8910	0.8911	0.8910	0.8903
Random Forest	200	None	sqrt	**0.9022**	**0.9022**	**0.9022**	**0.9016**

rior predictive performance, achieving 90.22% across accuracy, precision, and recall, with a closely aligned F1-score of 90.16%. This performance coherence reflects the model's effective generalization on the balanced dataset.

On the other hand, our results indicate that integrating contemporary computer vision architectures with efficient lightweight classifiers can deliver strong performance, all while maintaining a lower computational footprint compared to conventional end-to-end deep learning frameworks. This methodology proves to be particularly advantageous for deployment in constrained environments, such as mobile or edge-based agricultural systems. The practicality and efficiency of this approach have also been corroborated by previous studies [28, 29].

Conversely, Naive Bayes performed poorly, likely due to its strong assumptions of feature independence, which are not suitable for this high-dimensional and complex dataset. SVM performed moderately, while kNN showed a performance decline as the value of k increased, suggesting that a smaller neighborhood yields better classification in this context.

All metrics are reported in weighted + macro-averaged (W+M) format to account for both global accuracy and per-class balance, particularly important given the original dataset's imbalance prior to SMOTE application.

Additionally, we measured the inference performance of the proposed pipeline on our local workstation to assess its computational efficiency and deployment feasibility. On an Intel Core i7-12700 (CPU-only) setup, the framework achieved an average inference time of 68 ms/image, with peak memory usage of 480 MB and a throughput of approximately 14.7 FPS. When executed on an NVIDIA RTX 3060 GPU, the average inference time decreased to 15 ms/image, while maintaining a modest memory footprint of around 710 MB, resulting in a throughput of $\approx$ 66 FPS.

These results demonstrate that the proposed YOLOv8 + SAM + ML pipeline is computationally lightweight and can process tomato leaf images in real time on standard desktop hardware. Although experiments were conducted on CPU and GPU systems, the obtained performance metrics strongly suggest that the model can be feasibly deployed on edge-class devices (e.g., Jetson Nano or similar embedded platforms), given their comparable compute capabilities and available memory. This indicates a clear potential for integration into mobile or IoT-based agricultural monitoring systems.

5. DISCUSSION

In this section, we summarize the results obtained and discuss the feasibility of our method.

On the one hand, the experimental results show that the hybrid framework achieves an effective balance between accuracy and computational efficiency in tomato leaf disease classification. Among all classifiers, Random Forest achieves the best performance (Accuracy = 0.9022, F1 = 0.9016), outperforming kNN (0.8535) and SVM (0.6644). This is due to its overall structure, which minimizes overfitting and models non-linear relationships between hand-crafted features such as HSV and GLCM. In contrast, Naive Bayes performs poorly (Accuracy = 0.3184), confirming that the independence assumption does not hold for this dataset. The SMOTE technique successfully alleviates the class imbalance, improving the recall for minority classes by approximately 5–15%. Meanwhile, the YOLOv8 + SAM segmentation stage achieves the best mIoU (0.92) and Dice (0.94) scores (Table 1) and maintains real-time throughput (66 FPS on GPU, 14.7 FPS on CPU) with memory consumption below 1 GB, confirming the lightweight nature of the system and its potential for real-time or edge deployment.

On the other hand, compared to end-to-end deep learning models, this hybrid framework demonstrates higher efficiency and stability under limited data conditions. Deep CNNs typically require large-scale labeled datasets and powerful GPUs, and they often overfit in the presence of background lighting or noise. In contrast, combining YOLOv8 + SAM for precise localization with interpretable color-texture descriptors yields a compact representation that generalizes well even with moderate data volumes.

Overall, the results support the practical claim that the proposed framework offers a favorable balance between accuracy, interpretability, and efficiency for real-world agricultural environments.

6. CONCLUSION

This study introduces a hybrid framework that integrates state-of-the-art object detection (YOLOv8) and segmentation (SAM) techniques with handcrafted feature extraction and conventional machine learning models for classifying tomato leaf diseases. The proposed approach effectively localizes and segments diseased regions, enabling the extraction of discriminative and interpretable features using HSV, GLCM, and LBP descriptors. Classical classifiers, particularly Random Forest, achieved high predictive performance, with the best model surpassing 90% accuracy.

Beyond its technical merits, the proposed system offers significant practical value in real-world agricultural applications. By decoupling feature extraction from end-to-end deep learning inference, the framework substantially reduces computational overhead, making it well-suited for deployment on mobile devices, edge computing platforms, or IoT-based crop monitoring systems. Such lightweight systems provide farmers with early disease detection tools even in resource-constrained rural environments.

For future research, we plan to extend this pipeline to support multi-crop disease diagnosis by incorporating datasets from other economically important plants, such as potato, grape, and cucumber. Furthermore, integrating deep feature representations via pretrained convolutional neural networks or transformer-based models and fusing them with handcrafted features could further enhance accuracy while preserving interpretability. We also intend to perform a comprehensive ablation and feature-importance analysis to better quantify each descriptor's contribution, and conduct cross-dataset benchmarking against recent state-of-the-art methods to strengthen comparative validation. Finally, we aim to validate the system's robustness under real-world field conditions, accounting for variations in lighting, background complexity, and partial occlusions.

AUTHOR CONTRIBUTIONS

Conceptualization, Nguyen Viet Hung and Trong-Minh Hoang; methodology, software, validation, Nguyen Viet Hung and Phi Dinh Huynh; formal analysis, Trong-Minh Hoang; investigation, Phi Dinh Huynh; data curation, Nguyen Thi Ngoc Thoa, Le Mai Nam, and Le Thi Huyen Trang; writing—original draft preparation, Nguyen Viet Hung, Phi Dinh Huynh, Nguyen Thi Ngoc Thoa, Le Mai Nam, Le Thi Huyen Trang, and Trong-Minh Hoang; writing—review and editing, Nguyen Viet Hung and Phi Dinh Huynh; visualization, Nguyen Viet Hung and Phi Dinh Huynh; supervision, Trong-Minh Hoang. All authors have read and agreed to the published version of the manuscript.

References

[1] A. Das, F. Pathan, J. R. Jim, M. M. Kabir and M. Mridha, “Deep learning-based classification, detection, and segmentation of tomato leaf diseases: A state-of-the-art review,” *Artificial Intelligence in Agriculture*, vol. 15, no. 2, pp. 192–220, 2025. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S258972172500025X`

[2] M. Shahid, U. B. Singh, T. Ilyas, D. Malviya, S. K. Vishwakarma, Z. Shafi, B. Yadav and H. V. Singh, “Bacterial inoculants for control of fungal diseases in solanum lycopersicum l.(tomatoes): a comprehensive overview,” *Rhizosphere microbes: biotic stress management*, pp. 311–339, 2022.

[3] S. N. Ong, S. Taheri, R. Y. Othman and C. H. Teo, “Viral disease of tomato crops (solanum lycopesicum l.): an overview,” *Journal of Plant*

Diseases and Protection, vol. 127, pp. 725–739, 2020.

[4] A. Bhargava, A. Shukla, O. P. Goswami, M. H. Alsharif, P. Uthansakul and M. Uthansakul, "Plant Leaf Disease Detection, Classification, and Diagnosis Using Computer Vision and Artificial Intelligence: A Review," in *IEEE Access*, vol. 12, pp. 37443-37469, 2024.

[5] L. C. Ngugi, M. Abelwahab and M. AboZahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition–a review," *Information processing in agriculture*, vol. 8, no. 1, pp. 27–51, 2021.

[6] N. V. Hung, N. A. Quan, N. Tan, T. T. Hai, D. T. Trung, L. M. Nam, B. T. Loan and N. T. T. Nga, "Building predictive smell models for virtual reality environments," *Computer Science and Automation*, vol. 24, no. 2, pp. 556–582, 2025.

[7] H. Tian, T. Wang, Y. Liu, X. Qiao and Y. Li, "Computer vision technology in agricultural automation—a review," *Information processing in agriculture*, vol. 7, no. 1, pp. 1–19, 2020.

[8] R. S. L. Kakarla *et al.*, "An implementation and design framework of disease detection and prediction of tomato plant leaves using gray level cooccurrence and convolutional neural networks," *International Information and Engineering Technology:Revue d'Intelligence Artificielle*, vol. 38, no.2, pp. 603-611, 2024.

[9] S. A. Kumar and S. Sasikala, "Disease Detection in Tomato Leaves using Machine Learning and Statistical Feature Fusion," *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Coimbatore, India, pp. 1-6, 2021.

[10] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational intelligence and neuroscience*, vol. 2016, no. 3289801, 2016.

[11] J. G. A. Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases," *SpringerPlus*, vol. 2, no. 1, pp. 1–11, 2013.

[12] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.

[13] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

[14] S. U. Islam, G. Ferraioli and V. Pascazio, "Tomato leaf detection, segmentation, and extraction in real-time environment for accurate disease detection," *AgriEngineering*, vol. 7, no. 4, p. 120, 2025.

[15] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," in *arXiv preprint* arXiv:2004.10934, 2020.

[16] A. Kirillov *et al.*, "Segment anything," *arXiv preprint* arXiv:2304.02643, 2023.

[17] H. P, C. MK, L. RH, R. C, S. K. G and S. K, "Thermal canopy segmentation in tomato plants: A novel approach with integration of yolov8-c and fastsam," *Smart Agricultural Technology*, vol. 10, p. 100806, 2025. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2772375525000401`

[18] Y. Li, D. Wang, C. Yuan, H. Li and J. Hu, "Enhancing agricultural image segmentation with an agricultural segment anything model adapter," *Sensors*, vol. 23, no. 18, 2023. [Online]. Available: `https://www.mdpi.com/1424-8220/23/18/7884`

[19] S. A. A. Qadri, N. -F. Huang, T. M. Wani and S. A. Bhat, "Advances and Challenges in Computer Vision for Image-Based Plant Disease Detection: A Comprehensive Survey of Machine and Deep Learning Approaches," in *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 2639-2670, 2025.

[20] A. ALIMAGNIDOKPO, "Tomato leaf disease detection using yolov8," `https://github.com/Arix-ALIMAGNIDOKPO/Tomato-Leaf-Disease-Dection-using-Yolov8`, 2023, accessed: 2025-10-16.

[21] F. Research, "segment-anything-2," `https://github.com/facebookresearch/segment-anything-2`, 2024, accessed: 2025-10-16.

[22] C. Cortes and V. Vapnik, Support-vector networks. Springer, 1995, vol. 20, no. 3.

[23] T. Cover and P. Hart, "Nearest neighbor pattern classification," in *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, January 1967.

[24] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[25] I. Rish, An empirical study of the naive Bayes classifier, 2001, vol. 3, no. 22.

[26] M. Gehlot, R. K. Saxena, and G. C. Gandhi, ""tomato-village": a dataset for end-to-end tomato disease detection in a real-world environment," *Multimedia Systems*, vol. 29, no. 6, pp. 3305–3328, 2023.

[27] H. Nguyen, T. N. Dao, N. S. Pham, T. L. Dang, T. D. Nguyen and T. H. Truong, "An accurate viewport estimation method for 360 video streaming using deep learning," *EAI Endorsed Trans. Ind. Networks Intell. Syst.*, vol. 9, no. 4, p. e2, 2022.

[28] S. P. Mohanty, D. P. Hughes and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, p. 01419, 2016.

[29] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi and U. R. Alo, "Machine learning for cognitive internet of things: A survey," in *IEEE Access*, vol. 6, pp. 35794–35817, 2018.

Nguyen Viet Hung received the B.Sc.degree in Bachelor of Informatics pedagogy from the faculty of Engineering Technology of Ha Tinh University, Vietnam, in 2012, the M.Sc. degree Master of Information Technology from the Faculty of Information Technology of Ho Chi Minh City University of Technology, Vietnam, in 2016, and has been learning the Ph.D. degree in Telecommunications Engineering from the Hanoi University of Science and Technology, Vietnam, in 2025. His research interests include multimedia communications, network security, artificial intelligence, traffic engineering in next-generation networks, QoE/QoS guarantee for network services, green networking, and applications.

Phi Dinh Huynh is an undergraduate student at the Faculty of Information Technology, East Asia University of Technology, and a Research Assistant at the Mobilab Laboratory since 2023. His research focuses on computer networking, computer vision, data mining, and intelligent systems. He is particularly interested in integrating autonomous vehicles with AI-driven technologies to enhance smart mobility and intelligent communication systems. With strong enthusiasm for research, she continues to explore the synergy between machine learning, computer vision, and networked systems.

Nguyen Thi Ngoc Thoa is currently an undergraduate student at the Faculty of Information Technology, East Asia University of Technology, and has been working as a Research Assistant at the Mobilab Laboratory since 2023. Her research interests focus on computer networking, computer vision, data and web mining, as well as the development of intelligent systems and applications.

Le Mai Nam is a Lecturer at the Faculty of Information Technology, East Asia University of Technology. He obtained his Bachelor's degree in Applied Mathematics and Informatics from the University of Natural Sciences, Vietnam National University, Hanoi, in 2007, and his Master's degree in Information Systems from the Military Technical Academy in 2015. His research interests include data structures and algorithms, optimization methods, and mobile application development. He has contributed to various projects and academic activities in the fields of computer science and software engineering, focusing on practical applications of computational optimization and mobile technologies in education and industry.

Le Thi Huyen Trang is a Lecturer at the Faculty of Information Technology, East Asia University of Technology. She holds a Bachelor's degree in Computer Engineering from the Posts and Telecommunications Institute of Technology (2011) and a Master of Science in Information Systems from the Military Technical Academy (2015). Her research interests include marine ecology, aquaculture site assessment, broodstock technology, remote sensing, and geographic information systems (GIS). She has published several academic papers on marine ecology and aquaculture site assessment in national journals and conferences.

Trong-Minh Hoang received the bachelor's degree in physic engineering (1994) and electronic and telecommunication engineering (1999) from Hanoi University of Science and Technology, the master's degree in electronic and telecommunication engineering (2003), and the Ph.D.'s degree in telecommunication engineering (2014) from Posts and Telecommunications Institute of Technology. He is currently a lecturer in the Telecommunication Faculty of Posts and Telecommunications Institute of Technology. He is the head of the telecommunication network department. He has been working on the issues surrounding the performance of wireless networks. His research interests include routing, security, and network performance in mobile edge computing, flying ad hoc networks, wireless mobile networks, and beyond 5G. He applies mathematical analysis to model and analyze the behavior of complex systems and uses discrete event simulation tools to provide comprehensive solutions. He is a member of the IEEE ComSoc and IEEE Circuits and Systems Society.