Robust two-stage object detection using YOLOv5 for enhancing tomato leaf disease detection

Endang Suryawati[1], Syifa Auliyah Hasanah[2], Raden Sandra Yuwana[1], Jimmy Abdel Kadar[1], Hilman Ferdinandus Pardede[1]
[1]Research Center for Artificial Intelligence and Cyber Security, National Research and Innovation Agency, Bandung, Indonesia
[2]Department of Statistics, Faculty of Mathematics and Natural Sciences, Padjadjaran University, Sumedang, Indonesia

Article Info

Article history:

Received Aug 13, 2024
Revised Feb 10, 2025
Accepted Mar 15, 2025

Keywords:

Convolutional neural network
Deep learning
Plant disease detection
Two-stage object detection
YOLO

ABSTRACT

Deep learning facilitates human activities across various sectors, including agriculture. Early disease detection in plants, such as tomato plant that are susceptible to diseases, is critical because it helps farmers reduce losses and control the disease spread more effectively. However, the ability of machine to recognize diseased leaf objects is also influenced by the quality of data. Data collected directly from the field typically yields lower accuracy due to challenges faced in machine interpretation. To address this challenge, we propose a two-stage detection architecture for identifying infected tomato plant classes, leveraging YOLOv5 to detect objects within the images obtained from the field. We use Inception-V3 for classifying objects into known classes. Additionally, we employ a combination of two dataset: PlantDocs which represent field data, and PlantVillage dataset which serves as a cleaner dataset. Our experimental results indicate that the use of YOLOv5 in handling data under actual field conditions can enhance model performance, although the accuracy value is moderate (62.50 %).

Corresponding Author:

Endang Suryawati
Research Center for Artificial Intelligence and Cyber Security, National Research and Innovation Agency
Sangkuriang St., KST Samaun Samadikun, Bandung, Indonesia
Email: enda029@brin.go.id

1. INTRODUCTION

Early disease detection in plants enables farmers to minimize losses and more effectively control the spread of disease [1]. Certain plants, particularly tomatoes, are vulnerable to a variety of diseases that can reduce crop productivity and fruit quality. Bacterial spot, late blight, leaf mold, septoria leaf spot, and spider mites are among the diseases that affect tomato plants. Consequently, early detection of diseases in tomato plants is crucial to minimizing losses [2].

Existing research indicates significant advancements in developing systems for identifying and classifying plant diseases using machine learning methods that utilize images of infected leaves. Initially, these methods relied on manual feature extraction, demanding expert knowledge and limiting the quality and relevance of features. Algorithms such as support vector machines, decision trees, k-nearest neighbors, naïve Bayes, and random forests have demonstrated the potential of traditional machine learning in agricultural applications [3], [4]. The advent of deep learning has revolutionized traditional machine learning through automatic feature extraction [5], greatly improving classification accuracy. Deep learning, first introduced in 1943 [6], continues to evolve and is widely applied across various domains, including text recognition [7], [8], speech recognition [9], [10], and image recognition [11], [12]. One of the deep learning architectures commonly used for image classification is the convolutional neural network (CNN). CNN leverages the movement of convolution kernels to classify objects based on visual features such as color, texture, and leaf edges. This approach delivered superior performance for various image data tasks while progressively superseding traditional machine learning methods [11]. However, for small dataset use cases, the traditional machine learning still outperforms [13].

In agriculture and plantation applications, existing research indicates that CNNs are capable of improving classification accuracy through various popular architectures. Different CNN architectures have been widely used to identify plants or classify plant diseases. AlexNet, GoogleNet, and VGG-16 each possess distinct characteristics for identifying and classifying diseased leaves [14]. The Inception architecture has been applied for classifying fruit plants [15], lung cancer [16], and plant diseases [17]–[20]. CNN architectures categorized as skip connection architectures, have also been used for classifying plant diseases such as ResNet [19], [21]–[23], and DenseNet [19], [21], [24], and also DenseNet for detecting plant nutrient deficiencies [25]. Other CNN architectures, developed for improved performance and efficiency, include ComNet [26], EfficientNet [19], [21], [24], MobileNet [20]–[22], [27], and InceptionResNet [21], [22]. In its development, some researchers have proposed models categorized under the detector family, namely one-stage and two-stage object detection. YOLO is recognized as a popular one-stage object detection model, while the region-based CNN family falls into two-stage object detection. Wu *et al.* [28] applies two learning models, YOLOv5 and EfficientNetV2, to classify tomato leaf diseases.

Nevertheless, many studies on plant disease classification rely heavily on clean datasets, which enable models to achieve high accuracy. However, most datasets found in real-world environments are captured under uncontrolled, real-world conditions, unlike laboratory datasets. We refer to such datasets as "dirty datasets," representing real-world conditions. Often, models struggle to perform well when tested on dirty datasets. This situation presents a challenging task for machines, which must recognize and classify objects from real-condition data into predefined categories.

Based on this background, our research focuses on improving model performance, particularly when tested with real-world (dirty) datasets, to develop a robust model for classifying tomato plant leaf diseases. There are several crucial factors to consider to address this research question. First, we employ an object detector and a classifier to propose a two-stage object detection architecture. Second, we leverage YOLOv5 to be integrated into the architecture as an object detector, performing pre-processing tasks before the data enters the classifier. For the preliminary research of our proposed architecture, we consider utilizing YOLOv5, which offers balanced performance, speed, a lightweight model, and adaptability for future requirements while also accounting for the constraints of our current hardware [27], [29]. We utilize Inception-V3 to classify detected objects from YOLOv5 into known tomato disease classes. The justification for choosing Inception-V3 as our baseline classifier is that it is quite efficient in terms of computational cost, has a simple design model, and is straightforward to study [30]. Many studies use this model as a baseline and achieve good performance. Third, we utilize the PlantDocs dataset to represent the challenges of real-world conditions (dirty datasets). Meanwhile, the PlantVillage dataset is used to validate the findings of many studies that rely on clean datasets. PlantDocs and PlantVillage will be alternately used as training and testing data. However, we assume that the role of YOLOv5 in pre-processing tasks will be more effective when the model is trained and tested using the PlantDocs dataset. Fourth, we aim to assess whether YOLOv5 as a pre-processor can improve classifier performance. The classifier will be evaluated with and without YOLOv5 pre-processing.

2. METHOD

2.1. Inception-V3

Inception-V3 is a deep learning architecture that has achieved an accuracy of more than 78.1% on classification tasks involving 1000 classes on the ImageNet dataset [31]. This level of accuracy renders it suitable for various image recognition tasks. Several studies have been conducted to classify 28 flower species using the Inception-V3 architecture and transfer learning to enhance accuracy by retraining the flower category collection. Based on the experiment results, from the two datasets used, the Oxford-17 and Oxford-102 flower datasets, the resulting accuracy is 95%. This indicates that Inception-V3 performs well in image classification tasks, even with datasets containing numerous class categories [32].

Additionally, Inception is designed to deliver high performance results with a lower computational load compared to other architectures. This is achievable due to the fewer parameters in Inception compared to other architectures. Inception-V3 is an advancement of the earlier architecture, Inception-V1, introduced in 2014 as GoogLeNet [33]. Several modifications have been implemented in this architecture compared to its predecessor, including factorization into smaller convolutions, spatial factorization into asymmetric convolutions, utilization of auxiliary classifiers, and efficient grid reduction. Figure 1 show the Inception-V3 architecture generally. Overall, the Inception-V3 architecture comprises thirteen modules: one stem module, ten inception modules, two reduction modules, and one auxiliary classifier module. This combination of modules allows Inception-V3 to process images efficiently, capturing a wide range of features at multiple scales while maintaining a balance between computational cost and performance.

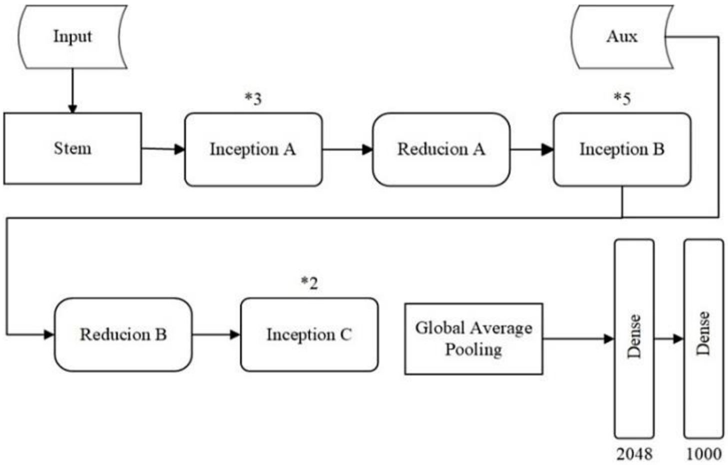

Figure 1. The inception-V3 architecture

2.2. YOLOv5

The architecture of CNN is excellent for classification. Nevertheless, object detection can be a good solution in certain cases to ensure that the classified images do not contain noise or other images outside the intended object. YOLOv5 is a method suitable for object detection. YOLOv5 is the evolution of the family of YOLO. Widely used, this object detection method balances speed and detection performance, and also offers a smaller model weight [27], enabling effective multi-scale object detection [29]. YOLOv5 also becomes a suitable and easy method to be modified for enhancements in further development needs [27], [29]. YOLOv5 can detect and classify multiple objects including humans, animals, and vehicles, in an image or video. Figure 2 is a depiction of YOLOv5 architecture.

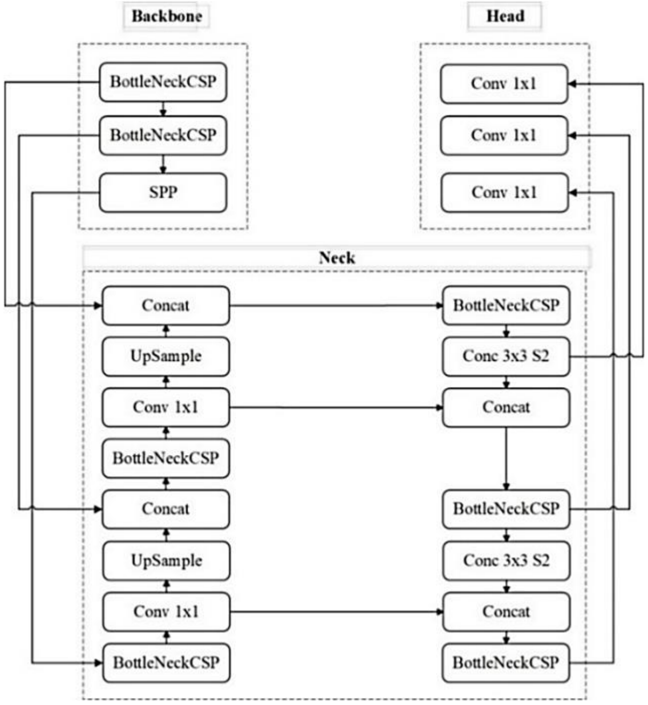

Figure 2. The YOLOv5 network architecture

YOLOv5 is available in five different sizes, based on the number of layers and parameters it possesses. This architecture comprises three main parts: the backbone, neck, and head. The backbone is responsible for forming features in the image, leveraging the CSPDarknet53 architecture, which is a modified version of Darknet. The cross-stage partial (CSP) structure helps overcome gradient problems by splitting the flow of gradients [34], reducing the number of parameters, and computing the load. In other words, the BottleneckCSP can handle the feature map extraction and reduce gradient information duplication in the CNN optimization process. Meanwhile, the spatial pyramid pooling (SPP) module enhances the detection of targets at different scales by aggregating features from multiple layers. The neck is the part that connects the backbone with the head, responsible for merging features from different scales. The head is responsible for detecting objects. Similar to other YOLO architectures, it uses YOLO layers to build this part. The output of this part includes bounding boxes and class probabilities.

2.3. YOLOv5 as pre-processing method for two-stages object detection architecture

As we have explained in the introduction section, we utilize YOLOv5 to support the pre-processing stage, including localizing and detecting objects within an image. This stage is the first step in the two-stage object detection process. The research commenced with the preprocessing stage, where we applied YOLOv5 to two datasets for object detection. We apply this process to the selected datasets, focusing the images on the important areas for easier and more accurate classification.

The first step in the object detection process in YOLOv5 involves extracting features from each dataset, using a resolution of 768×768×3 from the original backbone. This part splits each image into feature maps, each representing the image at different levels of abstraction. The neck part then concatenates these feature maps to aggregate information from various scales. After concatenation, a convolution process with 32 kernels transforms the concatenated feature maps into a 320×320×32 feature map. The head part then localizes and detects objects from these various scales of feature maps, ultimately producing classification results and object coordinates. Figure 3 illustrates the architecture of a two-stage object detection system, where the YOLOv5 object detection process forms an integral part of the entire system.

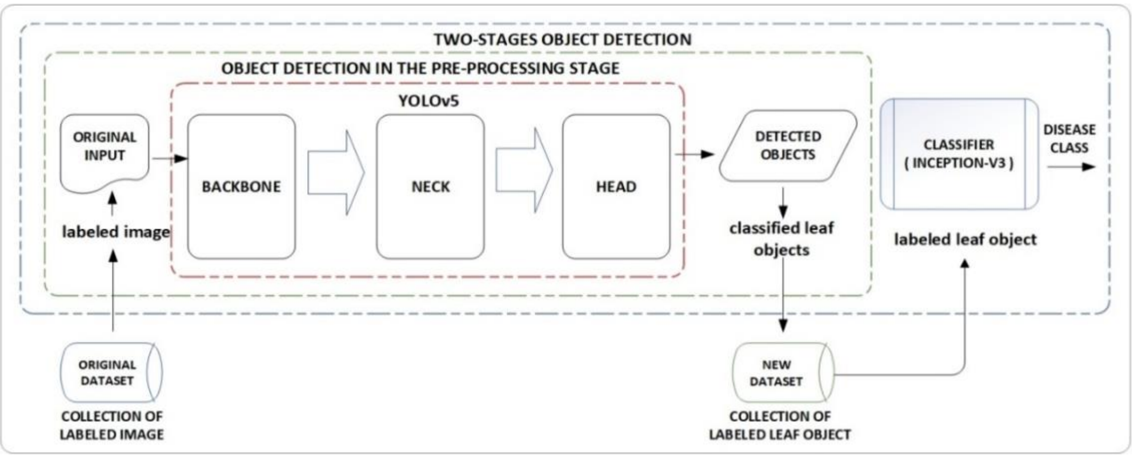
TWO-STAGES OBJECT DETECTION
OBJECT DETECTION IN THE PRE-PROCESSING STAGE
YOLOv5
ORIGINAL INPUT
BACKBONE
NECK
HEAD
DETECTED OBJECTS
CLASSIFIER (INCEPTION-V3)
DISEASE CLASS
labeled image
classified leaf objects
labeled leaf object
ORIGINAL DATASET
COLLECTION OF LABELED IMAGE
NEW DATASET
COLLECTION OF LABELED LEAF OBJECT

Figure 3. The proposed architecture of two-stages object detection

During object detection, we train the YOLOv5 model using the original dataset to generate a new dataset. This new dataset consists of classified leaf objects resulting from the detection process, specifically tomato leaf images classified as either diseased or healthy. The original dataset comprises tomato leaf images, both diseased and healthy, obtained from the PlantDocs and PlantVillage datasets. After object detection, we proceed to the classification stage. In this stage, we train the Inception-V3 model to classify tomato leaf diseases using the new dataset.

Interestingly, this study shows how YOLOv5, which is an object detector, contributes to the pre-processing stage to support the Inception-V3 classifier in identifying diseased tomato leaves. Using a dataset that accurately reflects real-world conditions, such as the PlantDocs dataset in Figure 4, significantly enhances its effectiveness. This figure illustrates a sample image from PlantDocs taken in field conditions. Through the object detection process, YOLOv5 localizes and detects three diseased leaf objects, then crops these leaves from the original image, resulting in three separate diseased leaf images, as shown on the right side of the arrow in Figure 4. The Inception-V3 classifier finds it easier to recognize and classify the leaves using these three individual leaf images, as opposed to using the original image on the left side.

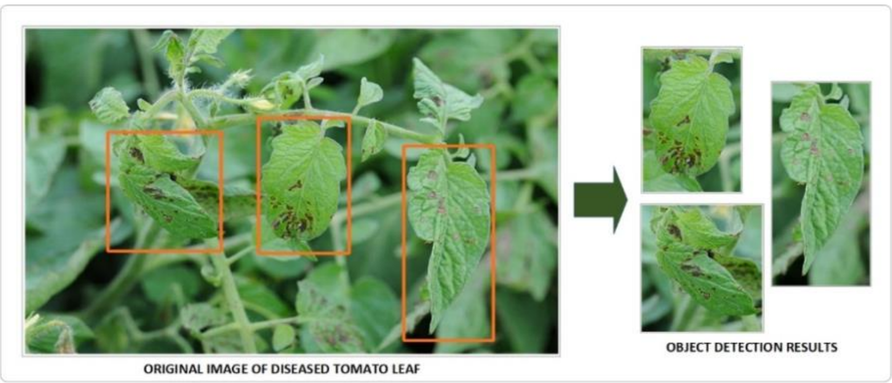

Figure 4. Object detection by YOLOv5 on PlantDocs sample image

We exclusively present the PlantDocs sample image for the object detection process, as it showcases the ability to detect and crop multiple leaf objects within an image into separate object images. However, we applied the same pre-processing using YOLOv5 to the PlantVillage dataset in our study, despite its classification as a clean dataset. PlantVillage images are well-organized tomato leaf images arranged in laboratory settings with uniform color backgrounds.

2.4. Dataset preparation

Preparing the data before using the datasets to train the model is another step in the pre-processing stage. Data preparation is essential for achieving excellent model performance. In our study, YOLOv5's object detection process produces the prepared data, which we refer to as the new dataset, as illustrated in Figure 4. As previously explained, we use two different datasets: the PlantVillage dataset and the PlantDocs dataset. The PlantVillage dataset consists of 38 class categories based on disease types for various plant species, totaling 54,303 images across all classes. The PlantDocs dataset consists of 2,598 images from 13 plant species, with a total of 17 class categories based on disease types. PlantDocs and PlantVillage are public datasets that are widely used for developing and testing plant disease detection models and can be accessed freely. Each of them has their own unique characteristics. Images samples for the PlantVillage dataset are shown in Figure 5, and Figure 6 illustrate sample of the images for the PlantDocs dataset.

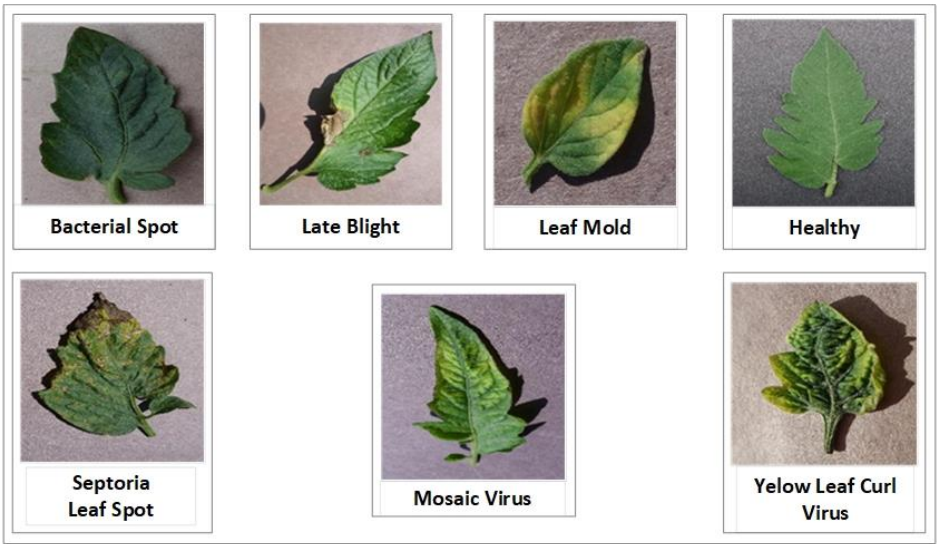

Figure 5. PlantVillage sample image

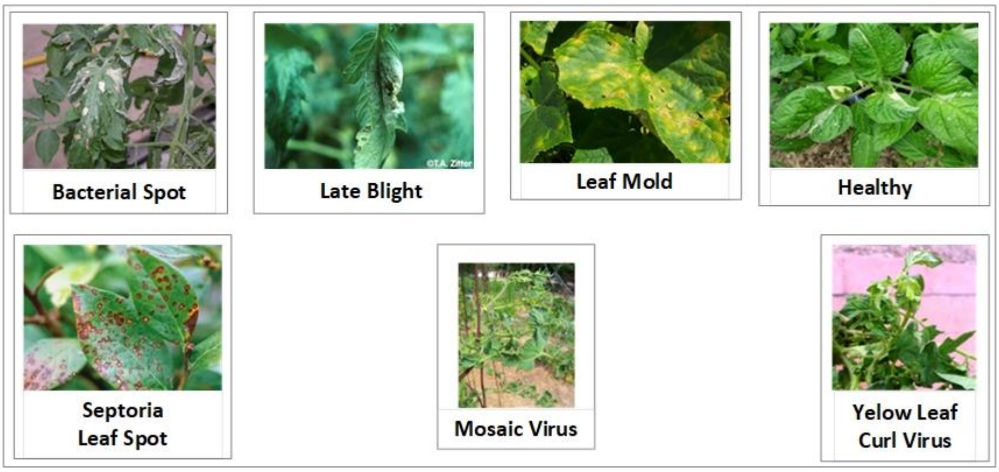

Figure 6. PlantDocs sample image

Both figures highlight the differences in image conditions between the two datasets. The PlantVillage dataset exhibits a relatively clean condition due to the process of capturing images takes place in a controlled environment setting. Meanwhile, the PlantDocs dataset contains images that may contain multiple leaves, each with varying backgrounds and lighting conditions. For our study, we use only tomato plants from each dataset, focusing on a subset of 6 disease classes, namely bacterial spot, late blight, leaf mold, septoria leaf spot, mosaic virus, yellow leaf curl virus, and 1 healthy class.

Table 1 shows the data distribution of the original datasets used as input for the object detection process. There are 12,357 images from the PlantVillage dataset and 648 images from the PlantDocs dataset. Since YOLOv5 can detect multiple classes in a single image, the number of instances in each class in the PlantDocs dataset has changed, as shown in Table 2. This change only occurs in the PlantDocs dataset because the PlantVillage dataset consists of single-leaf images.

Table 1. The data distribution for two original datasets

Disease class	PlantVillage	PlantDocs
Bacterial spot	1914	107
Late blight	1689	111
Leaf mold	857	91
Septoria leaf spot	1582	148
Mosaic virus	307	54
Yellow leaf curl virus	4671	75
Healthy	1337	62

Table 2. The data distribution for the PlantDocs dataset

Disease class	Original dataset	New dataset
Bacterial spot	107	265
Late blight	111	141
Leaf mold	91	368
Septoria leaf spot	148	195
Mosaic virus	54	482
Yellow leaf curl virus	75	1095
Healthy	62	582

2.5. Experimental setup

We divide each dataset into 80% training data and 20% testing data, respectively. The datasets we used contain images with various pixel sizes. Therefore, some pixel transformations or adjustments are required to adapt to the model architecture. We standardized all data sizes to 128×128 to ensure uniformity. We scale the pixel values from 0-255 to 0-1, speeding up the model's training and homogenizing the values in the data. We also use data augmentation to diversify the available data, allowing the model to learn from additional data during the training process. This can lead to better results by capturing targeted characteristics. Augmentation techniques used include shift, rotation, shear, zoom, and flip.

The images prepared in the pre-processing stage are then input into the Inception-V3 classifier. We use these images to train the model for optimal performance in classifying diseases in tomato plants. As outlined in our proposal, this study emphasizes leveraging YOLOv5 to detect objects within images, enhancing the pre-processing stage in a two-stage object detection architecture. Our hope is to improve Inception-V3 model performance in detecting tomato leaf diseases by incorporating YOLOv5.

To test our proposal, we train and test the Inception-V3 model using a combination of two datasets, allowing the model to learn from diverse data types. We alternately use these two datasets as training and testing data. Additionally, we train the Inception-V3 model without using YOLOv5 in the pre-processing stage, which we defined as our baseline architecture. In the baseline architecture, we directly train and test the Inception-V3 model using the two original datasets, by passing the YOLOv5 pre-processing stage. To support the training and testing of the model, we use the following hyperparameter settings: Adam optimizer with a learning rate of 1×10^{-4}, a batch size of 32, and 30 epochs per experiment.

3. RESULTS AND DISCUSSION

We divided the model performance results into two subsections: the first section, when the model uses PlantVillage as training data, and the second section, when the model uses PlantDocs as training data, including comparisons between the best proposed and its baseline. This comparison illustrates the effect of using YOLOv5 in the pre-processing stage on model performance. To clarify the terminology, "proposed" refers to the pre-processing method that uses YOLOv5, while "baseline" refers to the standard pre-processing method that does not use YOLOv5. We also use the term "PV" to refer to the PlantVillage dataset and "PD" to refer to the PlantDocs dataset.

3.1. Performance model based on PlantVillage as the training data

Table 3 demonstrates that the Inception-V3 model, trained and tested on the PlantVillage dataset, achieved an accuracy value of 98.28% for both our baseline and the proposed model. Conversely, testing the model with the PlantDoc dataset results in a decrease in its performance. However, as mentioned in the introduction, this outcome is not surprising, given that many classifications achieve high accuracy when using clean datasets, particularly for tomato plant diseases [14]. We also observe that in this case, the use of YOLOv5 does not significantly influence performance improvement.

Table 3. Model performance when trained by PlantVillage dataset

Testing data	Accuracy (%)	Architecture
PlantVillage	98.32	baseline
PlantVillage	98.32	Proposed
PlantDocs	21.54	basaeline
PlantDocs	15.28	Proposed

3.2. Performance model based on the PlantDocs as the training data

Based on the accuracy curves presented in Figures 7 and 8, we can observe that the model overfits with a high accuracy during training, but the validation process reveals a decline in the model's performance. It occurs when the model learns the training data too precisely, including noise, which negatively impacts its performance on testing data. In Figure 8, we can see our model performance through some of the curves with different levels of fluctuation. We observe that the proposed curve with a higher fluctuation indicates that the model has more difficulty in generalizing the learned features from the dirty PlantDocs dataset to the cleaner PlantVillage dataset. Overall, the use of the PlantDoc dataset tends to decrease model performance, both with the baseline and the proposed model.

Nevertheless, there is something intriguing to note in the results presented in Table 4. When the model is trained on the PlantDocs dataset and tested on the PlantVillage dataset, it shows a slight increase in accuracy of 13.9%. Similarly, training and testing the model on the PlantDoc dataset results in a greater accuracy increase of 27.08%, which causes the model's performance to achieve an accuracy of 62.50%. This demonstrates that using YOLOv5 can assist in improving accuracy, even though the results obtained are not very high.

When trained and tested on the field dataset (PlantDocs dataset), our proposed architecture achieved 62.50% accuracy. While this value isn't very high, it highlights YOLOv5's strength in object detection during pre-processing, which is key for identifying tomato plant leaf diseases. This underscores the importance of using real-condition datasets to build robust models. However, it is necessary to be attentive to preparing the dataset well, which needs systematic analysis of field-specific data variations and their influence on the model's error rates.

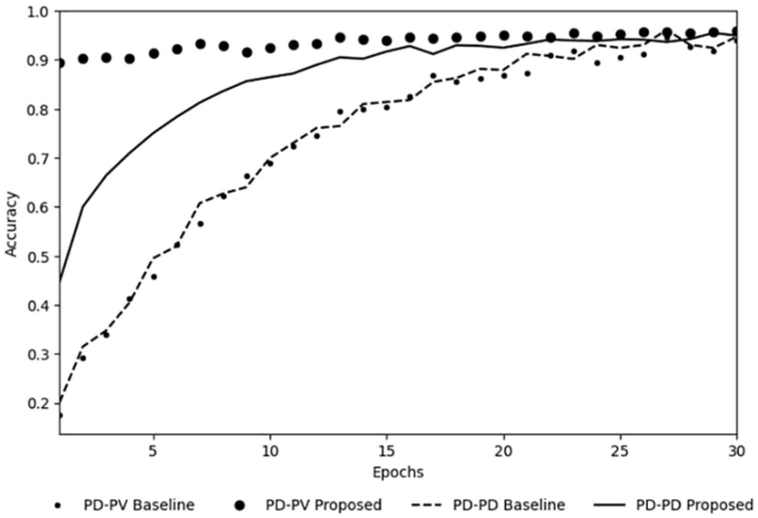

Figure 7. Training accuracy

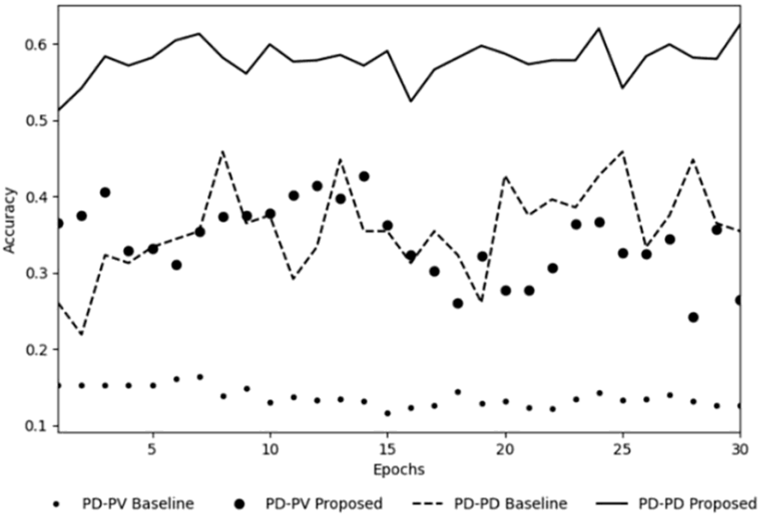

Figure 8. Validation accuracy

Table 4. Model performance when trained by PlantDocs dataset

Testing data	Accuracy (%)	Architecture
PlantVillage	12.59	baseline
PlantVillage	26.49	proposed
PlantDocs	35.42	basaeline
PlantDocs	62.50	Proposed

Many researchers in the previous study focus on achieving high accuracy using clean datasets with various CNN architectures, but our results suggest that popular CNNs struggle with real-condition datasets. To verify this, we tested several CNNs on the PlantDocs dataset, showing a drop in performance, as detailed in Table 5. The PD dataset encompasses images of diseases and unwanted objects, boasts a wide range of image sizes, and may include multiple leaves with a variety of backgrounds and lighting conditions. The convolutional layers find it challenging to extract features from the PD dataset, which frequently contains irrelevant background or noise. Therefore, the model has difficulty distinguishing disease spots or noise. It is necessary to apply a robust pre-processing technique to help localize the desired disease spots and separate them from the noise objects. To improve the quality of the noise and increase the amount of artificial data, we need to apply the augmentation technique. This will allow us to adapt the model to domains with different levels of variability [35]. Further, we can benefit from YOLOv5's capability in various augmentation techniques due to its ease of modification.

Table 5. Comparison of performance between our proposed and other CNN architectures

Architecture	Accuracy (%)
Resnet-50	41.07
DenseNet-121	43.57
MobileNet-V3	33.13
EfficientNet-V2	43.48
InceptionResNet-V2	32.71
Inception-V3	35.42
Our proposed	62.50

To develop a robust classification model, we need to train and test the model using a field dataset with high variability that represents the real environment, including various image backgrounds and noise [36]. We utilized the highly variable PlantDocs dataset for our proposed architectures [37], but Table 2 reveals an imbalance in the number of samples for each disease type. Class imbalance arises when one disease class dominates the dataset, leaving other diseases underrepresented. This imbalance makes the model difficult to generalize important features to new data, as it learns overly specific patterns and ignores more general ones.

For future work, it is necessary to increase the amount of data to ensure a balanced number for each class while also ensuring balanced variability [38], [39]. We also considered combining the dirty and clean datasets to aim at a balanced variability of the dataset. The use of YOLO still provides confidence as a robust pre-processor. YOLOv5 significantly aids the model in focusing on the extracted features. However, the use of background removal techniques needs to be involved to improve data quality. Without robust pre-processing, the model has difficulty extracting focused features, resulting in decreased accuracy [36]. For improving the model's ability to generalize features between domains with different variability, it is necessary to select the right domain adaptation technique and regularization technique. Furthermore, we must enhance the hyperparameter value settings for model training, including learning rate, optimizer, and batch size, while also implementing suitable regularization techniques.

4. CONCLUSION

In this study, we propose a two-stage detection architecture for identifying classes of infected tomato plants. In the pre-processing stage, we utilize YOLOv5 to detect objects within the images. The detected objects from the PlantDocs and PlantVillage dataset are then classified into known classes using Inception-V3 model. Our evaluation of two datasets confirms that our proposed architecture is more effective for diseased tomato plant detection, specifically when the classifier model is trained and tested using the PlantDocs dataset. In this case, YOLOv5 support our architecture for detecting regions of interest (ROI) and distinguishing important features from the noise which are present in the PlantDocs dataset. Although the experimental results show a moderate accuracy value (62.50 %), this research has the potential for future improvement. We need to prepare the dataset with balanced variability to achieved a more robust model for detecting diseased tomato leaves. Our hope is to develop a more accurate model by using a balanced dataset, a sophisticated pre-processor like YOLOv5, the appropriate regularization techniques, and the suitable domain adaptation technique.

FUNDING INFORMATION

This research was funded by Rumah Program Artificial Intelligence, Big Data, dan Teknologi Komputasi untuk Biodiversitas dan Citra Satelit-Electronics and Informatics Research Organization (OREI)-National Research and Innovation Agency (BRIN), Contract Number: 83/III.6.4/HK/2023, March 3, 2023, in collaboration with Department of Statistics, Faculty of Mathematics and Natural Sciences at Padjadjaran University.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Endang Suryawati	✓	✓	✓	✓	✓	✓			✓	✓		✓		
Syifa Auliyah Hasanah	✓	✓	✓	✓		✓	✓	✓	✓		✓			
Raden Sandra Yuwana						✓	✓	✓		✓	✓		✓	✓
Jimmy Abdel Kadar			✓	✓	✓			✓		✓	✓		✓	
Hilman Ferdinandus Pardede	✓	✓			✓	✓			✓	✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The datasets used in this study are publicly accessible and were obtained from open-source repositories, specifically:

- The data that support the findings of this study, in connection with the clean dataset are openly available in [PlantVillage dataset] at https://github.com/spMohanty/PlantVillage-Dataset
- The data that support the findings of this study that related to the dirty dataset are openly available in [PlantDoc] at https://github.com/pratikkayal/PlantDoc-Dataset

REFERENCES

[1] A. Abbas *et al.*, “Drones in plant disease assessment, efficient monitoring, and detection: a way forward to smart agriculture,” *Agronomy*, vol. 13, no. 6, pp. 1–26, 2023, doi: 10.3390/agronomy13061524.

[2] M. Bhandari, T. B. Shahi, A. Neupane, and K. B. Walsh, “BotanicX-ai: identification of tomato leaf diseases using an explanation-driven deep-learning model,” *Journal of Imaging*, vol. 9, no. 2, 2023, doi: 10.3390/jimaging9020053.

[3] B. S. Nawale and H. D. Gadade, “A systematic review: detecting plant diseases using machine learning techniques,” in *2023 11th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing (ICETET - SIP)*, IEEE, 2023, pp. 1–5, doi: 10.1109/ICETET-SIP58143.2023.10151590.

[4] T. S. Xian and R. Ngadiran, “Plant diseases classification using machine learning,” *Journal of Physics: Conference Series*, vol. 1962, no. 1, 2021, doi: 10.1088/1742-6596/1962/1/012024.

[5] M. Wu, J. Zhou, Y. Peng, S. Wang, and Y. Zhang, “Deep learning for image classification: a review,” in *Proceedings of 2023 International Conference on Medical Imaging and Computer-Aided Diagnosis (MICAD 2023)*, 2024, pp. 352–362, doi: 10.1007/978-981-97-1335-6_31.

[6] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943, doi: 10.1007/BF02478259.

[7] M. Eltay, A. Zidouri, and I. Ahmad, “Exploring deep learning approaches to recognize handwritten arabic texts,” *IEEE Access*, vol. 8, pp. 89882–89898, 2020, doi: 10.1109/ACCESS.2020.2994248.

[8] Z. Zhang *et al.*, “Dense residual network: enhancing global dense feature flow for character recognition,” *Neural Networks*, vol. 139, pp. 77–85, 2021, doi: 10.1016/j.neunet.2021.02.005.

[9] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, “Audio-visual speech recognition using deep learning,” *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015, doi: 10.1007/s10489-014-0629-7.

[10] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech recognition using deep neural networks: a systematic review,” *IEEE Access*, vol. 7, pp. 19143–19165, 2019, doi: 10.1109/ACCESS.2019.2896880.

[11] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A review of convolutional neural networks in computer vision,” *Artificial Intelligence Review*, vol. 57, no. 4, 2024, doi: 10.1007/s10462-024-10721-6.

[12] Y. Li, “Research and application of deep learning in image recognition,” in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, IEEE, 2022, pp. 994–999, doi: 10.1109/ICPECA53709.2022.9718847.

[13] P. Wang, E. Fan, and P. Wang, “Comparative analysis of image classification algorithms based on traditional machine learning and deep learning,” *Pattern Recognition Letters*, vol. 141, pp. 61–67, 2021, doi: 10.1016/j.patrec.2020.07.042.

[14] E. Suryawati, R. Sustika, R. S. Yuwana, A. Subekti, and H. F. Pardede, “Deep structured convolutional neural network for tomato diseases detection,” in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, IEEE, 2018, pp. 385–390, doi: 10.1109/ICACSIS.2018.8618169.

[15] P. Sumari, A. M. Kassim, S. Q. Ong, G. Nair, A. D. Ragheed, and N. F. Aminuddin, “Classification of jackfruit and cempedak using convolutional neural network and transfer learning,” *IAES International Journal of Artificial Intelligence*, vol. 11, no. 4, pp. 1353–1361, 2022, doi: 10.11591/ijai.v11.i4.pp1353-1361.

[16] Y. S. Kumaran, J. J. Jeya, T. R. Mahesh, S. B. Khan, S. Alzahrani, and M. Alojail, "Explainable lung cancer classification with ensemble transfer learning of vgg16, resnet50 and inceptionv3 using grad-cam," *BMC Medical Imaging*, vol. 24, no. 1, 2024, doi: 10.1186/s12880-024-01345-x.

[17] S. Sagar and J. Singh, "An experimental study of tomato viral leaf diseases detection using machine learning classification techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 451–461, 2023, doi: 10.11591/eei.v12i1.4385.

[18] V. K. Vishnoi, K. Kumar, B. Kumar, S. Mohan, and A. A. Khan, "Detection of apple plant diseases using leaf images through convolutional neural network," *IEEE Access*, vol. 11, no. 4, pp. 6594–6609, Apr. 2023, doi: 10.1109/ACCESS.2022.3232917.

[19] W. Shafik, A. Tufail, A. Namoun, L. C. De Silva, and R. A. A. H. M. Apong, "A systematic literature review on plant disease detection: motivations, classification techniques, datasets, challenges, and future trends," *IEEE Access*, vol. 11, pp. 59174–59203, 2023, doi: 10.1109/ACCESS.2023.3284760.

[20] A. Sohel, M. S. Shakil, S. M. T. Siddiquee, A. Al Marouf, J. G. Rokne, and R. Alhajj, "Enhanced potato pest identification: a deep learning approach for identifying potato pests," *IEEE Access*, vol. 12, pp. 172149–172161, 2024, doi: 10.1109/ACCESS.2024.3488730.

[21] D. Novtahaning, H. A. Shah, and J. M. Kang, "Deep learning ensemble-based automated and high-performing recognition of coffee leaf disease," *Agriculture*, vol. 12, no. 11, 2022, doi: 10.3390/agriculture12111909.

[22] F. Tang, R. R. Porle, H. Tung Yew, and F. Wong, "Identification of maize diseases based on dynamic convolution and tri-attention mechanism," *IEEE Access*, vol. 13, pp. 6834–6844, 2025, doi: 10.1109/ACCESS.2025.3525661.

[23] N. Zhang, H. Wu, H. Zhu, Y. Deng, and X. Han, "Tomato disease classification and identification method based on multimodal fusion deep learning," *Agriculture*, vol. 12, no. 12, 2022, doi: 10.3390/agriculture12122014.

[24] R. Rani, J. Sahoo, S. Bellamkonda, S. Kumar, and S. K. Pippal, "Role of artificial intelligence in agriculture: an analysis and advancements with focus on plant diseases," *IEEE Access*, vol. 11, pp. 137999–138019, 2023, doi: 10.1109/ACCESS.2023.3339375.

[25] M. S. H. Talukder and A. K. Sarkar, "Nutrients deficiency diagnosis of rice crop by weighted average ensemble learning," *Smart Agricultural Technology*, vol. 4, 2023, doi: 10.1016/j.atech.2022.100155.

[26] H. F. Pardede *et al.*, "Plant diseases detection with low resolution data using nested skip connections," *Journal of Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00332-7.

[27] J. Zhang, Z. Chen, G. Yan, Y. Wang, and B. Hu, "Faster and lightweight: an improved yolov5 object detector for remote sensing images," *Remote Sensing*, vol. 15, no. 20, 2023, doi: 10.3390/rs15204974.

[28] X. Wu, X. Li, S. Kong, Y. Zhao, and L. Peng, "Application of efficientnetv2 and yolov5 for tomato leaf disease identification," in *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, IEEE, pp. 150–158, 2022, doi: 10.1109/CACML55074.2022.00033.

[29] J. H. Kim, N. Kim, Y. W. Park, and C. S. Won, "Object detection and classification based on yolo-v5 with improved maritime dataset," *Journal of Marine Science and Engineering*, vol. 10, no. 3, 2022, doi: 10.3390/jmse10030377.

[30] L. Alzubaidi *et al.*, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00444-8.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[32] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, IEEE, pp. 783–787, 2017, doi: 10.1109/ICIVC.2017.7984661.

[33] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[34] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: a new backbone that can enhance learning capability of cnn," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.

[35] A. Fuentes, S. Yoon, T. Kim, and D. S. Park, "Open set self and across domain adaptation for tomato disease recognition with deep learning techniques," *Frontiers in Plant Science*, vol. 12, 2021, doi: 10.3389/fpls.2021.758027.

[36] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Computers and Electronics in Agriculture*, vol. 153, pp. 46–53, 2018, doi: 10.1016/j.compag.2018.08.013.

[37] J. G. A. Barbedo, "Deep learning applied to plant pathology: the problem of data representativeness," *Tropical Plant Pathology*, vol. 47, no. 1, pp. 85–94, 2022, doi: 10.1007/s40858-021-00459-9.

[38] M. Xu, S. Yoon, A. Fuentes, J. Yang, and D. S. Park, "Style-consistent image translation: a novel data augmentation paradigm to improve plant disease recognition," *Frontiers in Plant Science*, vol. 12, 2022, doi: 10.3389/fpls.2021.773142.

[39] G. Fenu and F. M. Malloci, "Evaluating impacts between laboratory and field-collected datasets for plant disease classification," *Agronomy*, vol. 12, no. 10, 2022, doi: 10.3390/agronomy12102359.

BIOGRAPHIES OF AUTHORS

Endang Suryawati received a Master's degree from the School of Electrical Engineering and Informatics at the Bandung Institute of Technology. She is currently working as a researcher at the Artificial Intelligence and Cybersecurity Research Center, National Research and Innovation Agency Indonesia. Her research interests include machine learning, pattern recognition, and image processing. She can be contacted at email: enda029@brin.go.id.

Syifa Auliyah Hasanah completed both her bachelor's and master's degrees in statistics and applied statistics from Padjadjaran University in 2023. She has served as a research assistant at the Research Center for Artificial Intelligence and Cyber Security at the National Research and Innovation Agency of Indonesia. She is passionate about research in the fields of applied statistics, big data, computer vision, and machine learning. She can be contacted at email: aulihassyifa@gmail.com.

Raden Sandra Yuwana obtain her master degree from Department of Informatics, Bandung Institute of Technology. Currently she is working as a researcher at Artificial Intelligence and Cybersecurity Research Center, National Research and Innovation Agency, Indonesia. Her research interests include machine learning, ontology, artificial intelligence, and NLP. She can be contacted at email: rade018@brin.go.id.

Jimmy Abdel Kadar received his M.Sc. in natural resources at IT from the Bogor Agricultural Institute. Currently working at the National Research and Innovation Agency, as a researcher. His areas of interest are artificial intelligence, CNN-LSTM, and facial recognition. He can be contacted at email: jimmy.abdel.kadar@brin.go.id.

Hilman Ferdinandus Pardede received his bachelor degree in electrical engineering from University of Indonesia. His master degree is from The University of Western Australia and his doctoral degree is obtained from Tokyo Institute of Technology in Computer Science. He is currently a research professor at Research Center for Data and Information, The National Research and Innovation Agency. His research interests include machine learning, multimedia signal processing, pattern recognition, and artificial intelligence. He can be contacted at email: hilm003@brin.go.id.