

Implementation of Yolo26 Nano Model on Riceguard Ai System for Rice Leaf Disease Detection

Gilang Septia Firmansyah^{1*}, Alif Bahktiar Akbar², Mufti Ari Bianto³, Khairul Umam⁴

^{1,2,3,4} Universitas Muhammadiyah Lamongan

gilangjr171@gmail.com^{1*}, alifbahktiar58@gmail.com², mufti.ari@umla.ac.id³, khairulumam744@gmail.com⁴

Abstract

RiceGuard AI is a new tool for diagnosing leaf diseases affecting rice plants. Early diagnosis of these diseases is crucial for maintaining rice productivity and global food security. This newly developed tool is designed to assist in diagnosing rice diseases by employing computer-vision techniques to automatically detect the condition of rice leaves. The Ultralytics YOLO26 Nano model was selected for this task because it adopts a novel "edge-first" architecture, enabling high processing speeds by eliminating the Non-Maximum Suppression (NMS) step during processing. To achieve this, the model was trained for 50 epochs, attaining a mean Average Precision (mAP) of 0.651 and an inference speed of 7.1 ms per image on an NVIDIA GTX 1650 GPU. The model achieved high accuracy for the leaf scald (99.4%) and healthy leaf (96.7%) categories but struggled to detect the narrow brown spot class (30.1%) due to visual ambiguities associated with it. YOLO26 Nano provides an accurate and efficient solution for detecting these rice conditions, enabling farmers to perform image-based diagnoses in the field for the early detection of leaf scald or healthy leaves, thereby facilitating prompt action. The model is integrated into a modern Flask web application featuring a "Glassmorphism" design style.

Keywords: Computer Vision; Deep Learning; Rice Leaf Disease; RiceGuard AI; YOLO26 Nano

1. Introduction

In Indonesia, rice (*Oryza sativa*) is a critical food crop and a key source of carbohydrate for most of the population [1]. This is further complicated by the fact that Indonesia is the third largest rice producer in the world, which means that it is important that the country maintains its production in an effort to protect the country's food security [2] [3]. One of the challenges that the country faces is the biological constraints affecting rice planting, particularly rice leaf diseases [1] [4]. Leaf Blight, Brown Spot, Leaf Blast, and Leaf Scald are some of the diseases that can result in serious economic casualties, either in the degree of destruction of the crop or the total loss of the crop [2]. Studies indicate that the negative impacts caused by pest and disease infestations in Indonesia are in the hundreds of thousands of tons of rice on an annual basis. Therefore, the postponement of the economic injuries suffered by farmers due to these diseases requires the rapid and precise determination of the presence of the disease so that the necessary interventions can be carried out in the field [2]. As it stands in the modern farming industry, every single leaf disease on rice plants must be identified by farmers performing manual visual inspections (MVI) [3]. This method is undoubtedly deficient [5] [6]. Farmers often do not have the training required to assess and implement correct diagnoses based on images showing various leaf disease symptoms [2]. Further, there is often little opportunity for agricultural experts to consider a given situation, especially in remote areas [2]. This leads to the misdiagnosis of a situation and the consequent application of a pesticide that will not cure the plants, increase production costs, and damage the local ecosystem. Computer vision and AI technologies offer the opportunity to quickly and objectively address these concerns [5] [7]. Deep learning innovations have recently transformed automated identification of plant diseases [4]. Many researchers have tried different models to improve accuracy in classification [3]. For example, [2] created the system "Paddy-AI," which achieves 85% accuracy for detecting Brown Spot, Hispa, and Leaf Blast using Convolutional Neural Networks (CNN). In the same context, [8] also CNN architecture and achieved 99.57% accuracy for the detection of Tungro, Blast, and Blight. In addition to normal CNNs, other lightweight models have been preferred, such as MobileNetV2 because of their relatively smaller size. Web-based [1] analyzed various deep learning models and found that MobileNetV2 had better accuracy of 91.67% than ResNet50 and DenseNet121 and was also better than other deep learning models, which required higher computational resources than MobileNetV2. Detection algorithms have evolved towards the 'You Only Look Once' (YOLO) method, as it represents both speed and accuracy by completing detection in one step [2]. Regarding the application of real-time smart farming, [5] showed how effective YOLOv8 was for detecting multiple classes of rice leaf diseases. Innovations for variants of YOLO are never-ending [2]. For example, [9] recently published in JAIEA, where they used the YOLOv11 framework for early detection of dermatitis, demonstrating the tendency to deploy newer versions of YOLO for more delicate diagnostic processes. Modifications to YOLO architectures have also been made by other researchers to aid in the detection of small and dense features of diseases, such as G-YOLO and Pyramid-YOLOv8 [2]. Although many AI-driven diagnostic systems have been developed, there is still much to be researched regarding the newest implementations of "edge-first" architectures [6].

Many current implementations of rice disease detection systems still use older versions of the technology, such as YOLOv5, YOLOv8, or heavier CNN-based frameworks that are extremely resource-intensive [2] [3]. While these frameworks exhibit high levels of accuracy in classroom scenarios, their use on edge devices, or lightweight cloud servers, typically result in high latency during the Non-Maximum Suppression (NMS) phase, as this step is especially resource demanding [6] [9]. Furthermore, many researchers only consider a small number of disease classes, and often omit detection of less obvious diseases such as the “narrow brown spot” disease and other diseases that are visually similar to each other and therefore are more challenging to diagnose [5]. The innovation of this study stems from incorporating the Ultralytics YOLO26 Nano (yolo26n.pt) model into the RiceGuard AI system [5]. As the most recent model in the YOLO series, it features an “edge-first” architecture that removes the NMS stage, resulting in quicker inference times and maintaining spatial accuracy [9] [6]. Additionally, this study provides a more comprehensive diagnostic capability by addressing six conditions, as opposed to the traditional three-class systems: Bacterial Leaf Blight, Brown Spot, Healthy, Leaf Blast, Leaf Scald, and Narrow Brown Spot [5]. With the use of modern web technologies, including Flask, Tailwind CSS, and a Glassmorphism design, the system offers an updated web-based interface that combines cutting-edge AI technology with a simple and responsive design for farmers [10] [11]. The main goal of this research is to develop an automated expert system that accurately and rapidly diagnoses rice leaf diseases using digital image processing [8]. This study aims to test the practicality of deploying YOLO26 Nano for the first time in an agrarian setting after conducting GPU-accelerated training using an NVIDIA GTX 1650 and hyperparameter tuning to $\text{imgsz}=640$ and $\text{workers}=2$, which is meant to alleviate the aforementioned hardware constraints [12]. Ultimately, RiceGuard AI aims to support precision agriculture and the country’s food security by providing farmers with tools for early disease detection and actionable agronomic methods [1] [8].

2. Literature Review

2.1. The Paradigm Shift in Agricultural Diagnostics

The stability of global food production, particularly rice (*Oryza sativa*), is increasingly dependent on the rapid identification of biological threats [1]. Historically, the identification of pests and diseases relied on manual visual inspection by farmers and agricultural experts. However, this conventional approach is characterized by high subjectivity, limited scalability, and significant delays in intervention [2]. The emergence of computer vision as a subfield of artificial intelligence has introduced a paradigm shift, enabling machines to perceive, analyze, and interpret visual data with precision that often surpasses human capability [4]. By leveraging digital image processing, automated systems can now monitor crop health continuously, providing a foundation for what is known as “Smart Agriculture” or precision farming [4]. Food production across the world, especially rice (*Oryza sativa*), is becoming more vulnerable to rapid detection of any form of biological threat [1]. Traditionally, pests and diseases were identified through visual inspection by farmers or agricultural experts [2]. However, the traditional method is subjective, lacks scalability and takes too much time before action could be taken [4]. Computer vision, an offshoot of artificial intelligence, has brought about a change of paradigm that allows machines to have better perception, analysis, and interpretation of visual data than humans [9]. Using computer vision in digital image processing has allowed constant monitoring of crop health status, thus paving way for “Smart Agriculture” [4].

2.2. Comparative Evolution of Methodological Approaches

There are distinct advancements that are evident in the literature that have brought about the evolution of classical machine learning techniques to advanced deep learning approaches [1]. The traditional approaches of Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Random Forest, for instance, have been at the forefront of digital agriculture development [1]. While SVM is efficient in handling mid-sized data and offers consistent performance dependent on the features such as Color Histograms and Histogram of Oriented Gradient (HOG), they are inefficient in solving spatially complicated and high variability of field captured images [13]. The introduction of Convolutional Neural Network (CNN) marked a huge advancement since this model automatically extracts hierarchically features using successive convolutions and pooling of neurons [13]. This model has proven successful in the studies conducted by [2] and [8] where they achieved between 85% to 99.57% accuracy in classification of diseases such as Leaf Blast and Bacterial Leaf Blight diseases [2]. Although they have proved to be highly accurate, conventional CNN model is efficient for image classification task of assigning one label to the image and not localization of disease spots in one frame [5].

2.3. Comparative Evolution of Methodological Approaches

The limitation associated with classification has led researchers to employ detection object frameworks [3]. The most used framework in this regard is You Only Look Once (YOLO) [3]. While two-stage detectors like Faster R-CNN detect objects by proposing regions followed by classification, YOLO treats the problem as a regression problem where the model simultaneously predicts bounding boxes and class probabilities [9]. The development from v2 to v11 of YOLO illustrates how the model balances its ability to achieve faster inference and better Mean Average Precision (mAP) [6]. Recent studies published in JAIEA and various journals illustrate the applicability of YOLO models in different fields. These include detection of safety equipment such as helmets and vests, waste classification, and detection of medical diseases such as dermatitis [9]. In agriculture, there have been successful implementations of YOLO versions including v5, v8, and v9 in disease detection of coffee, melon, and dragon fruit. Particularly, in disease detection of rice [5], and [3] illustrated that YOLOv8n (Nano) is a light weight and suitable for use in a mobile device, however, challenges still exist with smaller or visually weak features [5].

2.4. Lightweight Architectures and Edge Computing Constraints

One of the challenges mentioned in the literature reviews on research works is the challenge of the relationship between complexity and deployability of the model [1]. As mentioned, although complex models like ResNet101 and DenseNet201 can detect complicated patterns, they are very computationally expensive due to high number of parameters and slow computation time; hence, they cannot be applied to edge devices or low bandwidth network [1]. To solve this problem, the trend of using “lightweight” models has become common and one example of such models is MobileNetV2 which employs depthwise separable convolutions. In their work, [1] argue that for real-world tasks, the efficiency of model architecture is as important as its accuracy [1].

2.5. Research Gap and the Implementation of YOLO26 Nano

While different versions of YOLO have been successful, there is still an open area for investigation regarding optimization of the detection process in its "post-processing" phase. Currently, most of the existing algorithms use Non-Maximum Suppression (NMS) to eliminate redundant boxes [2]. The latter is a computationally heavy operation that may become a bottleneck while executing an AI model on light web-servers and edge devices. In addition, numerous rice diseases data sets consider unique large-scale symptoms and consequently perform poorly on the classes having thin visual features such as "Narrow Brown Spot" [2]. The choice of the model to be used in RiceGuard AI application, Ultralytics YOLO26 Nano (yolo26n.pt), is one way of solving the problem. YOLO26 Nano presents an exciting "edge-first" approach which does not involve the NMS step in its architecture, hence low latency rates without compromising spatial accuracy. This study attempts to contribute to the field by being one of the first to employ this "NMS-free" approach on a six-class data set. Unlike other studies where either accuracy or speed takes precedence over each other, RiceGuard AI provides a balance (0.651 mAP50) with the fastest inference time of 7.1ms in the industry. The combination of the RiceGuard AI into a Flask-based website with contemporary Glassmorphism design makes this research fill in the gap between cutting-edge AI and agriculture-oriented applications for users [2].

3. Methodology

3.1. Research Design

The current research uses quantitative experimental methodology to test the efficiency of the most recent architecture of object detection in the agricultural environment. The process of conducting the research is highly structured and starts from problem identification and literature review and continues to dataset collection, data preparation, implementation of the model, and system integration. The particular emphasis is put on the optimization of the lightweight model for edge deployment, which ensures scientific and practical validity of the diagnostic results.



Fig. 1: Research Flowchart

3.2. Dataset

The data set used for this study has been sourced from Roboflow public repository (Smart Farm, 2025) [6] that provides good quality images of rice leaves' status. The data set consists of a set of six different classes namely, (1) bacterial_leaf_blight, (2) brown_spot, (3) healthy, (4) leaf_blast, (5) leaf_scald and (6) narrow_brown. The inclusion of class "healthy" acts as a control variable, while other five classes denote the most common diseases affecting the production of rice. Smart Farm. *deteksi-penyakit-padi Dataset* [Data set]. Roboflow Universe. <https://universe.roboflow.com/smart-farm-oigdz/deteksi-penyakit-padi-i5n9o>

3.3. Data Preprocessing

In order to maintain consistency throughout the training process, all images were preprocessed using a consistent procedure within the Roboflow platform [6] [11]. The major processes involved were as follows:

1. Resizing: Images were resized to an identical size of 640*640 pixels in order to meet the input criteria for the YOLO model.
2. Normalization: Pixel values were normalized to a range of 0-1 to enable faster gradient descent during training.
3. Annotation: Disease symptoms were annotated in form of bounding boxes to give the model spatial coordinates.

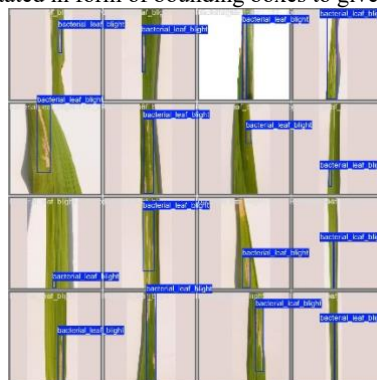


Fig 2: Bounding Box Annotation Process

3.4. AI Model (Ultralytics YOLO26 Nano)

The key feature of the RiceGuard AI system is the Ultralytics YOLO26 Nano (yolo26n.pt). This YOLO iteration brings a major leap forward in the YOLO family through its "edge-first" design. Unlike previous iterations such as YOLOv8 or YOLOv11, which require post-processing that is rather costly from computational point of view, the structure of YOLO26 Nano removes the need to perform the Non-Maximum Suppression (NMS) step. Thus, the latency becomes significantly smaller making the model highly suitable for web and mobile applications without any sacrifice in detection accuracy.

The number of headings should not exceed 3 levels. The other headings must use 9pt font. Only the first word in a heading must be capitalized and other words must be in lowercase.

3.5. Training Configuration

Model training was carried out on a local machine fitted with an NVIDIA GTX 1650 GPU (4 GB VRAM), using CUDA 12.4 accelerations on Windows OS. The hyperparameters used for training were tuned in the following manner:

Table 1: Training Hyperparameters and Machine Specifications

No	Component/Parameter	Specification/Value
1	Model Architecture	Ultralytics YOLO26 Nano (yolo26n.pt)
2	Image Resolution	(imgsz) 640 x 640 pixels
3	Training Cycles (Epochs)	50
4	Batch Size	8
5	Worker Threads (Workers)	2
6	Processing Hardware	NVIDIA GeForce GTX 1650 (4GB VRAM)
7	Computing Platform	CUDA Version 12.4

3.6. System Development

The RiceGuard AI system is built on a reliable architecture referred to as a full stack. The Back-end is developed in Python, where Flask is used as a framework for dealing with the model inference and the image manipulation. HTML5 and Tailwind CSS are used in the development of the Front-end, together with the Glassmorphism visual design and Lucide icons.

3.7. Detection Workflow

The operation flow of the system is meant to be easy. A user (farmer) uploads a digital image of the rice leaf through the web application. The uploaded file is sent to the YOLO26 algorithm by the Flask server using the confidence value of 0.15. In addition, the system displays the image with an overlapped box, indicating the name of the disease along with the percentage of the accuracy of detection. Moreover, the system offers suggestions for agriculture regarding each disease.

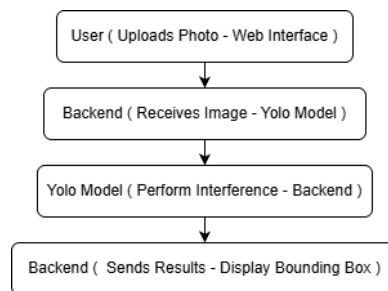


Fig 3: System Architecture and Workflow

4. Results and Discussion

4.1. Training Performance and Evaluation Metrics

Training of the Ultralytics YOLO26 Nano was successfully completed within 50 epochs. It has converged at the point whereby loss function values (box loss and class loss) had been decreasing continuously, which is similar to convergence observed in the previous implementations of YOLOv8 in detecting disease in rice. Upon completion of the 50th epoch, the system was able to reach an mAP50 value of 0.651 or 65.1%. Though the value numerically is quite low compared to the pure classification algorithm like CNN with accuracy up to 99.57%, the analysis must be taken into consideration from object detection perspective. YOLO26 Nano not only classifies the image but localizes them.

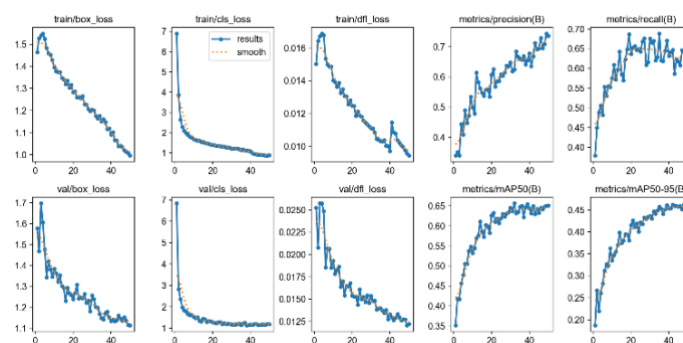


Fig. 4: Detailed Evaluation Metrics per Class

4.2. In-depth Class Performance Analysis

An in-depth look at the six classes has shown remarkable differences in performance which offer some insight into the capabilities of the model to extract features:

1. **High-Performance Classes:** There were remarkable results of accuracy for the leaf_scald (99.4%) and healthy (96.7%) classes. The almost perfect accuracy achieved for the leaf_scald class can be attributed to the large, distinctive necrotic patterns that are associated with the disease which make it easily recognizable from the green leaf. High accuracy for the healthy class means there is low false-positive rate and therefore no need for any unnecessary application of pesticides.
2. **Narrow Brown Constraint:** The narrow_brown class performed the worst with an accuracy of 30.1%. From scientific point of view, this can be attributed to the fact that the disease is characterized by lesions that are "thin and subtle". This corroborates Putro et al. (2026) who state that the greatest challenge posed to single-stage detectors are classes with mild symptoms or similar texture as that of narrow_brown class (44%). The resolution of 640 pixels might be inadequate to capture the micro-features of the disease.

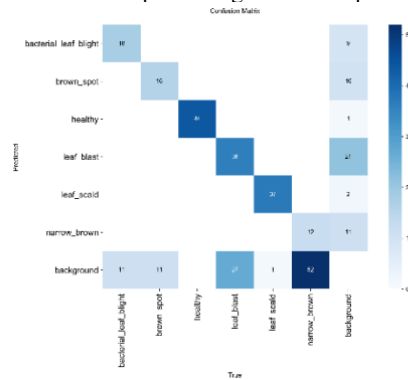


Fig. 5: Normalized Confusion Matrix

4.3. Computational Efficiency

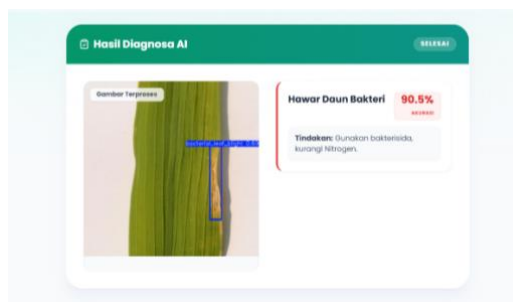


Fig. 6: Sample Detection Output

4.4. Practical Implementation



Fig. 7: RiceGuard AI Web Interface

5. Conclusion

In this research, the primary goal of the study was successfully fulfilled as the RiceGuard AI system, which utilizes the novel Ultralytics YOLO26 Nano framework in a computer vision-based expert system, was created and tested. With the demonstration of the ability to create an "edge-first" and "NMS-free" system for plant disease detection, the study has brought a valuable technological advancement to the sphere of precision agriculture. The removal of the Non-Maximum Suppression phase allowed to obtain a remarkably high performance of inference, which was 7.1ms per image. It is fast enough to be used on lightweight web servers and edge devices. Moreover, the combination with the contemporary Flask-based Glassmorphism interface will make the advanced AI diagnoses available to farmers. The experiment proves the high efficiency of RiceGuard AI in detection of prominent rice leaf diseases, reaching mAP50=0.651 (65.1%) on

average in six classes. It is important to note that the system was almost perfect in detection of leaf_scald (99.4%) and healthy (96.7%) classes, thus providing a precise detection of prominent necrotic forms without errors among healthy leaves. However, a significant drawback was found in the detection of narrow_brown class, which showed relatively low accuracy of 30.1% because of the slight and fine appearance of lesions in this class.

References

- [1] R. Agustin, M. Nurlailli, K. P. Yuanda, and B. A. Sudamto, "Deteksi Penyakit Daun Padi Menggunakan MobileNetV2 : Pendekatan Deep Learning untuk Meningkatkan Ketahanan Produksi Pangan," vol. 9, pp. 1294–1303, 2025.
- [2] N. I. R. Yusuf Fadlila Rachman, Pratiwi Susanti, Affriza Brilyan Relo Pambudi Agus Putra, "Sistem Informasi Deteksi Penyakit Pada Tanaman Padi (Brown Spot , Hispa , Leaf Blast) Menggunakan Metode Convolutional Neural Network (CNN)," vol. 4, no. 3, pp. 1193–1204, 2024.
- [3] R. T. W. Achmad Yasid , Tsalitsatun Ni'mah, Risma Ramadhaningtias, "DETEKSI PENYAKIT PADA DAUN TANAMAN PADI BERBASIS YOLOV8 DISEASE DETECTION IN RICE PLANT LEAVES BASED ON YOLOv8," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 13, no. 2, pp. 347–358, 2026.
- [4] D. D. Harmiansyah, Ella Trilia Oviana, Mhd Kadar Fitrawan, Pramana Putra and Meidaliyantisyah, "Deteksi Cerdas Penyakit Tanaman Kopi Robusta Berbasis Deep Learning Menggunakan Variasi YOLO," *J. Agrik.*, vol. 36, no. 3, pp. 487–499, 2025.
- [5] H. P. Putro, B. Tumanggor, and D. Supriatna, "Deteksi Penyakit Daun Padi Menggunakan Model Deteksi Objek YOLO," vol. 5, no. 2, pp. 664–673, 2026.
- [6] M. I. Aulia and A. Yudhana, "Optimasi Deteksi Hama Tanaman Melon Berbasis YOLOv9," *J. Inform. Technol. Commun.*, vol. 10, no. 1, pp. 150–161, 2026.
- [7] D. Citra, D. Untuk, K. Penyakit, M. Pendekatan, D. Learning, and D. Model, "DETEKSI CITRA DAUN UNTUK KLASIFIKASI PENYAKIT PADI MENGGUNAKAN PENDEKATAN DEEP LEARNING DENGAN MODEL CNN," vol. 10, no. 1, pp. 56–62, 2024.
- [8] R. Suciani *et al.*, "Deteksi penyakit daun padi menggunakan deep learning dengan arsitektur cnn detection of rice leaf disease using deep learning with cnn architecture," vol. 8, pp. 1451–1463, 2025.
- [9] S. Ivanda, G. Agda, R. Heriansyah, and Z. R. Mair, "Early Detection of Dermatitis Through Comparison of Image Size Variations Using the You Only Look Once (YOLO) Framework," vol. 5, no. 3, 2026.
- [10] M. Hatami, F. Nurapriani, W. Andriani, U. Buana, P. Karawang, and S. B. Saleh, "DETEKSI HELMET DAN VEST KESELAMATAN SECARA REALTIME MENGGUNAKAN METODE YOLO BERBASIS WEB FLASK PENDAHULUAN," vol. 10, no. 1, pp. 221–233, 2023.
- [11] B. D. Prasetya, E. P. Amalia, and P. D. Juliana, "Implementasi Object Detection Berbasis Web untuk Klasifikasi Jenis Sampah Menggunakan YOLOv8," vol. 4, pp. 60–65, 2025.
- [12] A. R. Kurniawan, Z. Burdiarso, E. Ardhianto, and T. Informatika, "Impementasi algoritma yolo untuk deteksi jumlah pengunjung didalam gedung," vol. 9, no. 2, pp. 2944–2948, 2025.
- [13] M. A. B. Afifah Agustin and Hery Ardiansyah, "Perbandingan Klasifikasi Daun Cabai dengan Metode CNN dan SVM Untuk Deteksi Penyakit Jurnal Pengembangan Teknologi Informasi dan Komunikasi," *J. Pengemb. Teknol. Inf. dan Komun.*, vol. 4, no. 1, pp. 187–196, 2026, doi: 10.52060/juptik.v4i1.4403.