Jurnal Elektronika dan Telekomunikasi (JET), Vol. 24, No. 2, December 2024, pp. 120-128
Accredited by KEMDIKBUDRISTEK, Decree No: **158/E/KPT/2021**
doi: 10.55981/jet.643

Object Detection Approach Using YOLOv5 For Plant Species Identification

Billi Clinton, Amperawan Amperawan, Tresna Dewi *

Department of Electrical Engineering
Politeknik Negeri Sriwijaya
Jl Srijaya Negara Bukit Besar Palembang 30139
Palembang, Indonesia

Abstract

Accurate identification of plant species is crucial for biodiversity conservation in modern agriculture and horticulture. However, manual identification methods often struggle with the complexity and overlapping visual characteristics of different plant species, making the process challenging. To address this issue, this research proposes using the YOLOv5 deep learning algorithm for automated plant species detection. The goal is to develop a model that is both effective and highly accurate in identifying plant species under various environmental conditions. The study utilized a dataset of 1,220 images representing nine plant species, such as *Alocasia Macrorrhizos*, *Cactus*, *Costus Spicatus*, *Euphorbia tirucalli*, and *Sansevieria*. The training process, which ran for 200 epochs and took approximately 53 minutes, resulted in a model with mean Average Precision (mAP) of 85.73%, precision of 98.27%, and recall of 94.36%. The model demonstrated strong performance, accurately identifying plant species in both single and multiple object scenarios. The findings confirm that the proposed YOLOv5-based model is highly effective for plant species identification, offering both accuracy and efficiency. The success of the model in detecting plant species makes it a valuable tool for biodiversity conservation efforts and further development of AI-driven plant recognition technologies.

Keywords*:* AI (Artificial intelligence), deep learning, YOLO (You Only Look Once), plant species identification.

I. INTRODUCTION

In contemporary agriculture and horticulture, biodiversity conservation plays a pivotal role in maintaining healthy ecosystems, and this effort hinges significantly on effective plant species identification. However, the complexity of plant identification lies in the diverse and often overlapping visual characteristics of species, which frequently results in errors when relying on manual methods [1], [2]. As a result, there is a growing need for highly efficient and accurate automated models that can assist in the reliable identification of plant species. In this context, Artificial Intelligence (AI) has shown great potential, particularly through the use of object detection techniques, to address these challenges in plant species identification [3], [4], [5], [6].

Among AI-driven approaches, the YOLO (You Only Look Once) algorithm has emerged as a leading solution for real-time object detection, known for its balance between speed and accuracy in processing large image datasets [7], [8], [9]. This research investigates the use of the YOLOv5 algorithm for plant species identification [10], [11], leveraging its advancements to improve efficiency and accuracy. A review of related works shows that AI, particularly deep learning and object detection techniques, have been successfully applied in similar tasks, establishing a strong foundation for further exploration.

* Corresponding Author.
Email: tresna_dewi@polsri.ac.id
Received: May 27, 2024 ; Revised: September 13, 2024
Accepted: October 07, 2024 ; Published: December 31, 2024

One notable example is the work by Wu *et al.* [12], who employed an improved YOLOv4 algorithm for detecting apple flowers in orchards. Their approach achieved a remarkable mean Average Precision (mAP) of 97.31% and a detection speed of 72.33 frames per second (fps), surpassing five other algorithms in both speed and accuracy. The model also demonstrated robustness to variations in apple tree species and lighting conditions, illustrating the effectiveness of deep learning models in real-world agricultural environments. This research highlights the potential of YOLO-based models in agricultural applications where accuracy and real-time performance are critical.

Similarly, Mardiyah [13] applied Convolutional Neural Networks (CNN) to classify images of gardens and rice fields, achieving a training accuracy of 96.25% and a validation accuracy of 75%. The CNN model, trained on a dataset of 100 images, used an 80:20 data split, further confirming the potential of AI in agriculture. Mardiyah's work demonstrates the versatility of AI models in diverse agricultural settings and underscores the importance of dataset size and quality in achieving high accuracy.

Sahla Muhammed Ali's comparative analysis of YOLOv3, YOLOv4, and YOLOv5 for sign language detection further supports the superiority of the YOLOv5 model [14]. Her research found that YOLOv5 achieved an F1-score of 0.655 and an mAP of 0.633, outperforming its predecessors. The study's findings reinforce the effectiveness of YOLOv5 in various object detection tasks, including those beyond agricultural applications. The comparative analysis of different YOLO versions indicates that YOLOv5's improvements in accuracy and speed make it an ideal candidate for tasks requiring real-time detection with high precision.

The study by Srivastava *et al.* [15], which compared several deep learning algorithms, including YOLO, SSD, and Faster R-CNN, further confirms the efficacy of YOLO models in object detection tasks. Srivastava *et al.* concluded that YOLOv3 outperformed SSD and Faster R-CNN in terms of speed and computational efficiency, establishing YOLO as a benchmark for future object detection applications. Moreover, Alfonso *et al.* [16] applied object detection models for tomato flower detection in greenhouse environments, achieving precise results with an average positional error of less than 0.3 across multiple targets. This research underscores the versatility and accuracy of object detection techniques like YOLO in agricultural scenarios.

In terms of real-time applications, the work by Paszke *et al.* [17] introduced the ENet architecture, a highly efficient neural network designed for low-latency tasks like semantic segmentation. ENet is 18× faster, requires 75× fewer FLOPs, and has 79× fewer parameters than comparable models while maintaining similar or better accuracy. The efficiency gains provided by ENet are particularly relevant for real-time plant species identification tasks, where both speed and accuracy are critical.

K. Wang's work on PANet (Path Aggregation Network) [1] introduced a novel few-shot image segmentation method based on prototype alignment, offering a potential avenue for improving plant identification models in cases where training data is limited. PANet's ability to handle few-shot learning scenarios could complement the real-time capabilities of YOLOv5, making it a valuable technique for scenarios where only a small number of plant species need to be identified. Additionally, Y. Tian *et al.* [2] demonstrated the effectiveness of using an improved YOLOv3 model for detecting apples at different growth stages in orchards. Their work showed that YOLO-based models could handle variations in growth stages and environmental conditions, making them well-suited for agricultural applications.

Building on these insights, the current research proposes a YOLOv5-based model for plant species identification. This model seeks to leverage the advancements in YOLOv5 to achieve high accuracy and real-time performance, while also addressing the unique challenges posed by plant species' diverse visual characteristics. The model will be tested on a curated dataset of plant species, focusing on key parameters such as mAP, detection speed, and robustness across varying environmental conditions. By refining the object detection capabilities of YOLOv5 and incorporating insights from previous studies, we aim to contribute to the development of faster, more accurate systems for plant species identification.

The significance of this research lies in its potential to enhance biodiversity conservation efforts by providing an efficient tool for plant species identification. By integrating state-of-the-art object detection algorithms like YOLOv5 with real-time application needs, this study aims to advance the technology available for plant recognition in agricultural and horticultural settings. Titled "Object Detection Approach Using YOLOv5 for Plant Species Identification," the research focuses on optimizing the YOLOv5 algorithm for practical use cases. The dataset employed will be limited to select plant species, as outlined in Table 1, with the aim of achieving meaningful improvements in detection speed and accuracy.

II. Methods

In this research, the detection of plant species was performed using a deep learning-based object detection algorithm to overcome the challenges posed by environmental variations such as background clutter and changes in illumination commonly encountered in open orchard settings [1], [18]. These factors often degrade the performance of traditional object detection methods, necessitating the adoption of more robust solutions. By leveraging deep learning, the study aims to improve the accuracy and resilience of the plant species detection process.

The YOLO framework was chosen for this task due to its reputation as a highly efficient one-stage object detection algorithm. Unlike traditional multi-stage detection systems, YOLO utilizes a single CNN [19], [20], [21] to process images and directly predict both the class labels and bounding box coordinates for detected objects. This end-to-end approach enables simultaneous object localization and classification, significantly enhancing detection speed without compromising accuracy [22].

In particular, YOLOv5 was selected for its modular scalability, offering five distinct model versions—YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra-large). These versions vary in terms of their convolutional width and depth, enabling the selection of models tailored to specific hardware capabilities and application requirements. For instance, YOLOv5n and YOLOv5s are optimized for low-resource environments, making them ideal for deployment on edge devices or embedded systems. Conversely, YOLOv5x, with its greater computational capacity, provides enhanced detection performance at the expense of processing speed, making it suitable for high-performance applications requiring superior accuracy.

In this study, the YOLOv5 algorithm was applied to build a robust plant species detection model. The model development process involved several stages, including data preprocessing, model training, and evaluation, as depicted in Figure 1. Through these steps, the study seeks to ensure that the model can maintain high detection accuracy and speed even in varying environmental conditions, thus contributing to advancements in automated plant species identification.

The research process begins by clearly defining the problem that needs to be addressed. Following this, a comprehensive review of relevant literature is conducted to identify appropriate methods and approaches for solving the identified issue. Once the most suitable approach is chosen, data collection takes place, focusing on gathering a diverse set of images from the surroundings, such as the home yard environment.

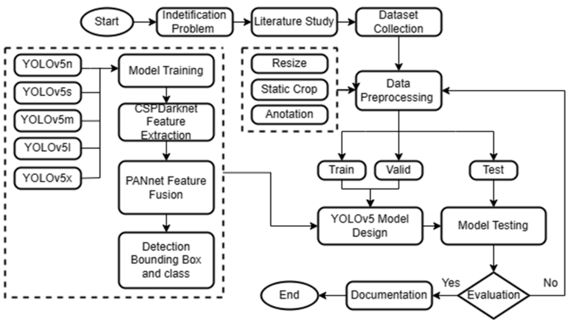

Figure 1. Flowchart workflow.

The collected dataset undergoes a preprocessing phase aimed at enhancing image quality and variability. During this stage, the images are resized, and static cropping techniques are applied. Afterward, the dataset is labeled and annotated to facilitate the training of the model, allowing it to predict and identify objects in the images accurately. The annotated dataset is then split into three subsets: training, validation, and testing datasets. The training and validation sets are utilized to build and refine the model, ensuring it achieves strong detection accuracy, while the testing set is reserved for evaluating the model's performance in real-world conditions.

The training process involves using the YOLOv5s model, a lightweight version of the YOLO family, known for its balance between speed and accuracy. During this stage, the model's architecture is enhanced with several feature extraction components, including CSPDarknet for backbone feature extraction and PANet for feature fusion. The model is then tasked with object classification and bounding box prediction for each identified object. Once trained, the YOLOv5s model is evaluated against the testing dataset to measure its detection accuracy across various scenarios.

After testing, the model is assessed to determine whether it meets the predefined accuracy benchmarks. If the results fall short, adjustments are made, often beginning with a return to the preprocessing stage to further improve the dataset quality. The final detection outcomes are documented, serving as a basis for potential future research and model improvements.

A. Identification Problem

The first step in addressing any challenge is identifying the problems that arise, as this helps in formulating an effective plan to solve them. In this research, the focus is on improving the identification of plant species. Several factors, such as environmental conditions and visual similarities between species, can cause the model to perform inaccurately. Therefore, the aim of this study is to develop a model that is both highly accurate and efficient in detecting plant types. The expectation is that this research will not only resolve current detection issues but also contribute valuable insights for further development and future studies.

B. State-of-the-Art

A thorough literature search is an essential part of the research process, helping to gather insights and information relevant to the studied topic. By reviewing existing literature, it becomes possible to identify various approaches and select the most suitable algorithm for addressing the problem at hand. This decision is informed by prior studies, which provide a foundation for understanding the strengths and weaknesses of different methodologies. Supporting references for this research are drawn from a wide range of sources, including journals, conference papers, books, websites, and other platforms, both from domestic and international contexts. This comprehensive approach ensures that the chosen solution is backed by well-established and diverse perspectives in the field.

C. Data Collecting

Image data collection for this research was conducted in the environment surrounding a residential area, focusing on capturing several types of plants. The details of the collected dataset are presented in Table 1. The images were taken using a Redmi Note 10s smartphone camera, with a resolution of 3472 × 3472 pixels. All photographs were captured during daylight hours to ensure consistent lighting conditions. The images were taken from specific angles to maximize clarity, and only the clearest images were selected for use. In total, 1,220 images were gathered for this dataset. Figure 2 presents sample images from the plant dataset used in this study.

D. Preprocessing Data

Once the collected and sorted data is ready, it first undergoes preprocessing. The preprocessing begins by resizing the images from their original dimensions of 3472 × 4624 pixels to a standardized 640 × 640 pixels.

TABLE 1
PLANT DATASET INFORMATION

Resolution	**(640 px) × (640 px)**	
Plant *Dataset*	*Alocasia Macrorrhizos*	122 images
(1,220 images)	*Cactus*	122 images
	Costus Spicatus	119 images
	Euphorbia tirucalli	122 images
	Excoecaria Cochinchinensi	119 images
	Frangipani	119 images
	Kalanchoe	126 images
	Sansevieria	231 images
	Syzygium paniculatum	140 images

Figure 2. Plant dataset collection.

After resizing, the next step involves trimming unnecessary parts of the images using a static crop feature. This cropping process helps enhance the accuracy of the images by focusing on relevant areas. Once the dataset is adequately refined, it moves to the labeling phase, commonly known as image annotation.

During the annotation process, each object in the image is given a "ground truth" bounding box, which accurately defines the position of the object according to its category or class. In this stage, labels are assigned based on the type of object present in each image. The annotation process was conducted using the Roboflow application, ensuring the dataset was properly prepared for the next steps in model training and evaluation.

E. Data Split

Once the data has been collected, it is divided into three categories: training, validation, and testing datasets. The majority of the data, 85%, is assigned to the training set to allow the model to learn effectively from a broad range of examples. Meanwhile, 10% of the data is reserved for validation, enabling adjustments and fine-tuning of the model during training. The remaining 5% is set aside for testing, ensuring that the final model's performance is evaluated on unseen data, providing an accurate measure of its ability to generalize. This distribution strategy is designed to optimize both model learning and performance assessment.

F. Training Model

The YOLOv5 architecture was employed to perform the model training, with the final weights generated from the process considered optimal for image detection during the testing phase. Table 2 provides detailed information about the YOLOv5 model training process, including the number of training images, batch size, training time estimation, and the number of epochs used. During the training, several performance metrics, including precision, recall, mAP, box loss, object loss (obj loss), and classification loss (cls loss), were recorded and displayed [23], [24]. The mAP graph specifically illustrates the accuracy of the predicted bounding box in estimating the true position of objects in the images. Additionally, three loss graphs were generated to track the model's performance in terms of bounding box placement, class prediction, and object presence detection. Google Collaboratory [25], was utilized as the platform for conducting the model training, providing the necessary computational resources.

TABLE 2
DETAIL MODEL TRAINING

Model (pt)	Image Training	Batchsize	Estimation Training(s)	Epoch
YOLOv5	1,220	32	53.52	200

Figure 3 illustrates the flow of a machine learning model architecture designed for object detection, specifically for identifying plant species. It begins with an input image, originally sized at 3472 × 4624 pixels, depicting different types of plants. These images are then processed through the core components of the model, which include the backbone, neck, and head. In typical object detection architectures, the backbone is responsible for extracting important features from the input image, usually employing CNN. The neck further refines these features, often through techniques like Feature Pyramid Networks (FPN), allowing the model to handle multi-scale object detection. The head of the model generates the final outputs, which consist of bounding boxes, classifications, and confidence scores for the objects detected in the image.

After passing through the model, the output images are resized to 640 × 640 pixels. The identified plant species are labeled and enclosed in colored bounding boxes to clearly indicate their location and classification. For example, the species *Sansevieria* is labeled with a green bounding box, *Alocasia Macrorrhizos* with a red bounding box, and *Euphorbia tirucalli* with an orange bounding box. This diagram represents the complete process of plant identification, from raw image input to the final detected and labeled output, showcasing the efficiency of the model in accurately detecting and classifying different plant species [26], [27], [28].

Figures 4 and 5 depict the proposed YOLOv5 architecture, which is composed of three main components: the backbone, neck, and head, each playing a crucial role in object detection.

The **backbone** of the YOLOv5 architecture is CSPDarknet53. This component functions as a CNN that extracts essential features from images. CSPDarknet53 enhances the efficiency and accuracy of the model by

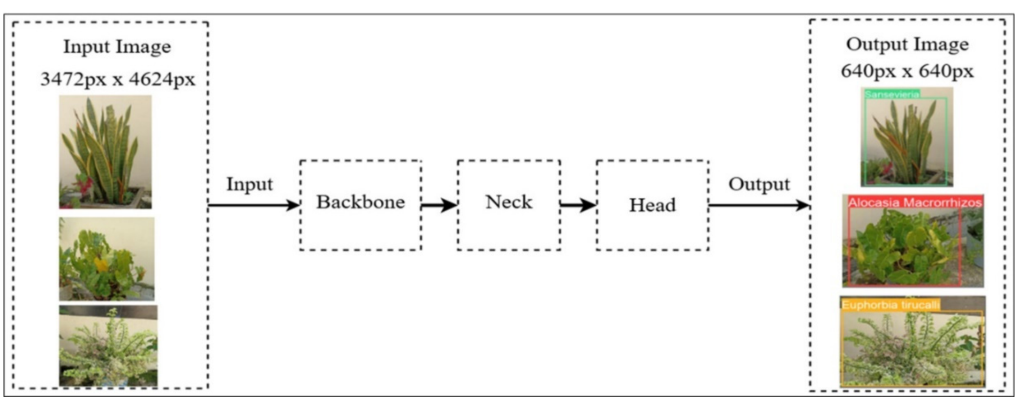

Figure 3. Block diagram YOLOv5.

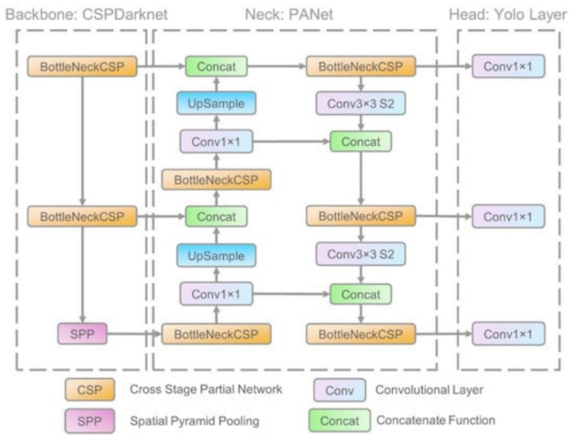

Figure 5. Detail of YOLOv5 architecture.

repeatedly separating and combining gradient information. By merging gradient changes into feature maps, it helps reduce the number of parameters, which in turn decreases the model size while maintaining high performance [29]. This design allows the model to process images with greater speed and precision.

The **neck** of the architecture is responsible for processing and combining the extracted features before passing them to the prediction layer. Since more complex networks increase the risk of information loss, YOLOv5 incorporates a Feature Pyramid Network (FPN) [30], which enhances the model's ability to detect small objects. A key component of this neck is PANet [31], which improves the flow of local information to higher layers, helping the model retain critical details for smaller object detection [32].

The **head** is where the actual object detection occurs. This component predicts the bounding box and class of each detected object [33]. YOLOv5 employs the same head structure as its predecessors, YOLOv3 and YOLOv4 [34], allowing it to generate three distinct output feature maps for multi-scale prediction. This approach improves the model's ability to detect objects of varying sizes, from small to large, with better efficiency and accuracy [32].

The YOLOv5 architecture starts by inputting an image into CSPDarknet53 for feature extraction. The extracted features are then processed through PANet, combining them before sending the processed data to the head for object detection [29]. YOLOv5 performs object detection across three different scales to account for the size variation of objects within the image. A 640 × 640-pixel image generates grids of different sizes: 80 × 80 for small objects, 40×40 for medium objects, and 20 × 20 for large objects. Each grid generates three anchor boxes of different sizes, resulting in a total of 25,200 bounding boxes ((80 × 80) + (40 × 40) + (20 × 20)) × 3 = 25,200.

To refine the detection results, Non-Max Suppression (NMS) [35], [36] is applied. This step selects the bounding box with the highest confidence when multiple boxes are generated for a single object [37], ensuring that only the most accurate prediction is retained.

G. Model Testing and Evaluation

In this study, the trained model is tested using new data that has not been used or included in the model training process. This step is crucial for evaluating the model's generalization ability and its effectiveness in accurately detecting plant species. The objective is for the model to achieve high accuracy with minimal loss, ensuring reliable detection performance.

To assess the model's performance, key evaluation metrics such as precision, recall, and mAP are used. These metrics are calculated based on objects detected with a confidence score of 0.5 or higher. To further verify the model's effectiveness, four main performance indicators are adopted: precision, recall, mAP, and detection speed.

A critical aspect of the evaluation is the use of the Intersection over Union (IOU) metric. IOU measures the overlap between the predicted bounding box and the ground truth bounding box. In this study, a prediction is considered correct when the IOU is greater than or equal to 0.5 (IOU ≧ 0.5). If the IOU is less than 0.5 (IOU < 0.5), it is treated as a false positive, whereas an IOU of 0 indicates a false negative. The precision, recall, and mAP are then calculated based on these cases, using the (1) – (4). In this context, mAP refers to the average value of the Average Precision (AP) when detecting plants, with higher mAP values indicating better detection accuracy

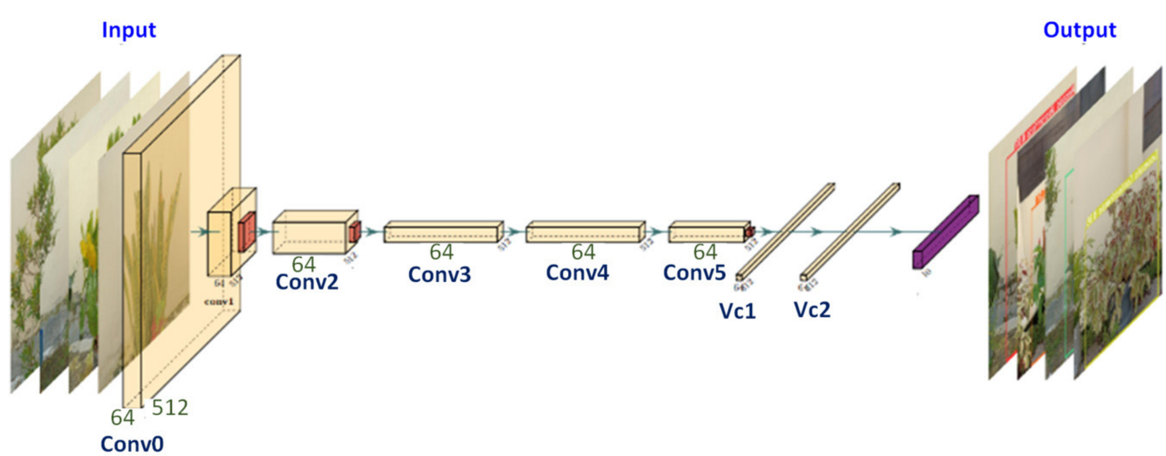

Figure 4. Proposed YOLOv5 architecture.

[12]. This approach ensures that the model's ability to detect plant species is rigorously tested and accurately measured, with the performance evaluation given as (1) - (4),

$$IOU\ (\mathrm{R}, \mathrm{R}') = \frac{|\mathrm{R} \cap \mathrm{R}'|}{|\mathrm{R} \cap \mathrm{R}'|} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \text{ x } 100\% \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \text{ x } 100\% \tag{3}$$

$$mAP = \frac{\sum_{c=1}^{C} AP(c)}{C} \text{ x } 100\% \tag{4}$$

where R is the detected area of the object's bounding box. R' is the actual area of the object's bounding box. TP, FP, and FN are the number of true positive cases, false positive cases, and false negative cases, respectively. C is the number of plant detection categories [12].

III. Result and Discussion

To demonstrate the effectiveness of the proposed technique, a sample of 10 plant images was used, and the results are presented in Table 3. The approach achieved precision and recall values of 98.27% and 94.36%, respectively, with an mAP of 85.73%. These results highlight the high precision and reliability of the model, showing its strong potential for accurate plant species identification. The impressive performance of the proposed method suggests that it can serve as a valuable technical reference for the development and enhancement of plant identification models in future research, particularly in applications requiring high accuracy and efficiency.

Table 3
Model testing result

Model	*Precision* (%)	*Recall* (%)	*mAP* (%)	Size Model (MB)
YOLOv5	98.27	94.36	85.73	14

During the system testing phase, the object detection performance is evaluated to assess the accuracy and effectiveness of the trained model in identifying plants. When a plant is detected, a bounding box is drawn around the object, accompanied by the probability score indicating the likelihood that the object is a plant. These detection results provide a visual representation of the model's ability to correctly identify plant species, as shown in Figure 6. This stage helps ensure that the trained weights are performing optimally and that the system can reliably detect plants with a high degree of confidence.

For this testing, the best-performing weights from the training phase were utilized. These weights were applied to the test dataset, and the predictions generated by the model were compared against the manually labeled data to evaluate accuracy and performance. Additionally, ten graphs depicting various performance metrics, such as precision, recall, and loss trends, that were generated during the training process are displayed in Figure 7. These graphs provide a visual representation of the model's training progression and offer insights into how the model has learned to make accurate predictions over time.

Figure 7 presents the evaluation metrics derived from the training results, which were used to assess the model's overall performance. The evaluation matrix reveals a consistent downward trend in the loss function values throughout the training process. The *x*-axis represents the number of epochs, while the *y*-axis indicates the threshold values for the different evaluation metrics. As training progresses, the loss function values for both the training and validation datasets steadily

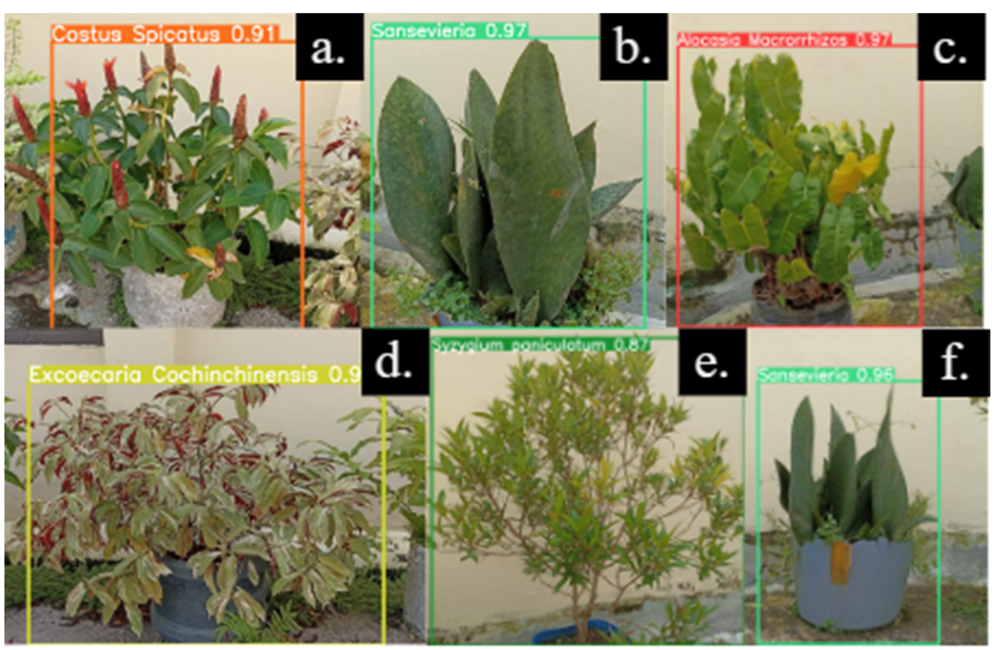

Figure 6. Model detection results for (a) costus spicatus, (b) sansiviera, (c) alocasia macrorrhizos, (d) excoecaria cochinchinensis, (e) syzygium paniculatum, and (f) sansiviera.

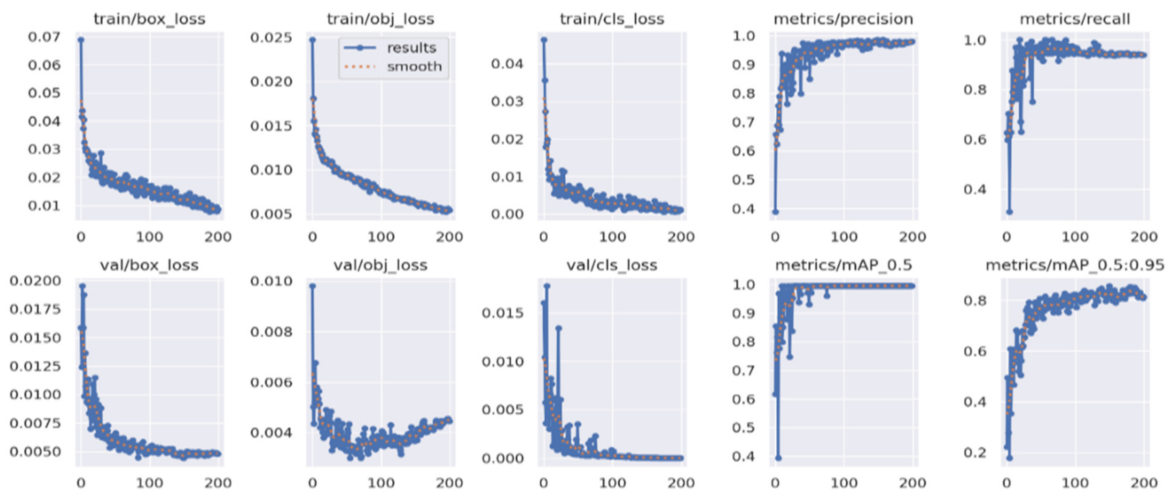

Figure 7. Training results graph.

decrease, indicating improved learning. Concurrently, key performance indicators such as accuracy, recall rate, and average precision show a gradual increase up to around the 10th epoch. After the 10th epoch, the validation data loss for classification (valid/cls_loss) decreases at an accelerated rate, resulting in a noticeable drop in accuracy. Between the 27th training batch and the 180th epoch, the validation data's object loss (valid/obj_loss) exhibited instability, leading to fluctuations in model performance. Despite these challenges in validation, the training data metrics (train/obj_loss, train/box_loss, and train/cls_loss) continued to show a consistent decline, contributing to overall performance improvements as the training progressed to 200 epochs. This indicates that while the model showed some instability during validation, the training data maintained a steady trajectory toward better accuracy and loss reduction.

The validation data metrics, specifically valid/box_loss and valid/cls_loss, exhibited minimal downward movement and reached a stable state at around 105 epochs. In terms of the precision matrix, the curve demonstrated significant instability from the 9th training batch to the 94th epoch, which impacted the model's performance during this period. However, starting from the 95th training batch up until the 200th epoch, the precision curve showed only slight fluctuations, leading to improved model performance.

Similarly, the recall matrix also experienced performance instability from the 2nd training batch through the 133rd epoch. From the 134th batch onward until the 200th epoch, the recall curve remained stable with minimal decline, contributing to better recall performance. For the mAP 0.5 metric, instability was observed between the 1st and 76th epochs. However, after the 77th training batch, the mAP 0.5 curve showed consistent stability, with no further decline up to the 200th epoch.

Regarding the mAP 0.5:0.9 matrix, the model experienced minor instability throughout the training process from the 1st to the 200th epoch. Despite this, the overall accuracy remained strong, indicating that the model achieved good accuracy for further development and refinement, even when exposed to more stringent evaluation criteria using the mAP 0.5:0.9 metric. Based on the confusion matrix derived from the test results shown in Figure 8, it can be concluded that the model correctly identified all plant species, with a prediction accuracy of 1.0 for each species, indicating no errors in the predictions. This demonstrates the high precision of the proposed model.

The performance of the YOLOv5 model for both the training and testing phases was evaluated using a private dataset and compared with other plant identification models in terms of accuracy. The dataset comprises 1,220 images, covering nine plant species, including *Alocasia Macrorrhizos*, *Cactus*, *Costus Spicatus*, *Euphorbia tirucalli*, *Excoecaria Cochinchinensi*, *Frangipani*, *Kalanchoe*, *Sansevieria*, and *Syzygium paniculatum*. Out of the total dataset, 1,220 images were used for training, and 500 images were allocated for testing. All images were selected from these nine species, and the resolution for each image was standardized to 640 × 640 pixels.

The YOLOv5-based plant species identification model was evaluated by training and testing the model on multiple splits of the dataset to obtain a reliable estimate of its performance. The model achieved an average training accuracy of 96.45%, indicating that it effectively learned from the training data. The average testing accuracy was also 96.45%, showing that the model maintained its accuracy on unseen data. This result demonstrates that the YOLOv5 model outperformed other machine learning models for plant species identification, such as Faster R-CNN, which achieved 74.96% accuracy [38], and InceptionV3, which achieved 82.50% accuracy [39]. This indicates that the proposed YOLOv5-based model provides a robust and accurate solution for plant species identification, outperforming established benchmarks in the field.

IV. Conclusion

This study developed and evaluated a plant species identification model based on the YOLOv5 architecture, addressing the limitations of manual identification methods. The model demonstrated high efficiency and accuracy, achieving a precision of 98.27%, recall of

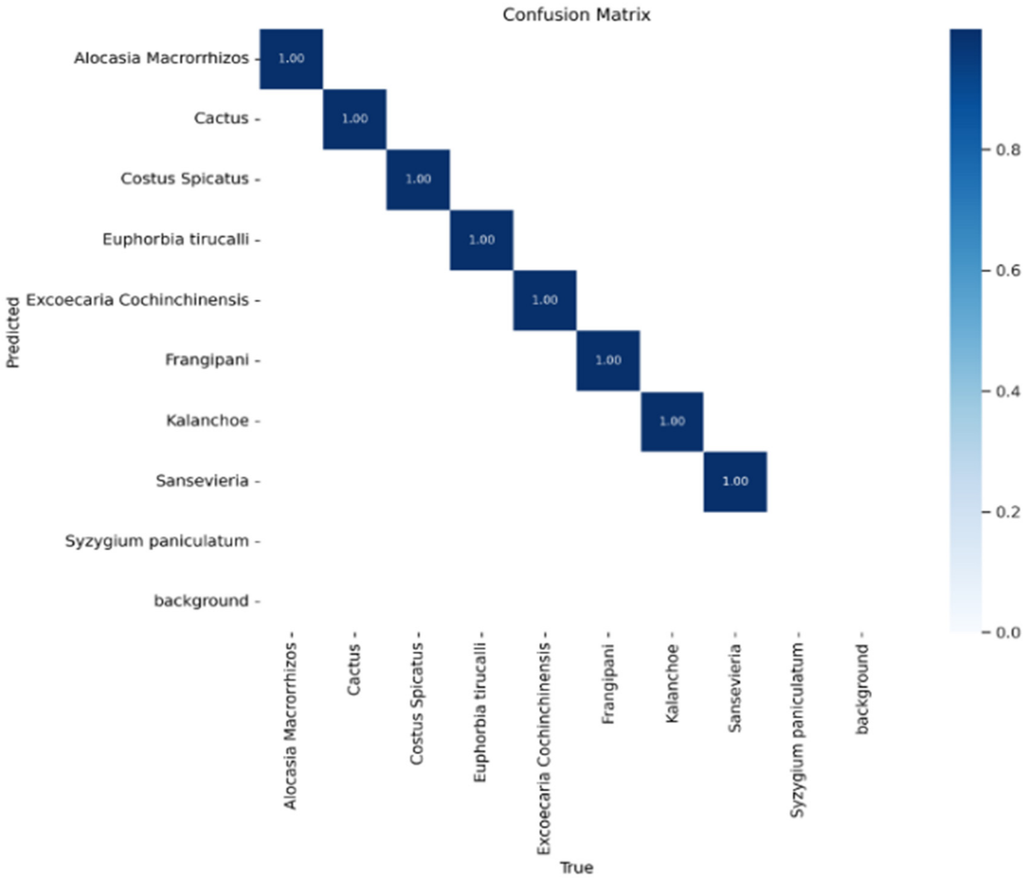

Figure 8. Confusion matrix.

94.36%, and mAP of 85.73%. Both the training and testing phases yielded an accuracy of 96.45%, significantly outperforming established models like Faster R-CNN (74.96%) and InceptionV3 (82.50%). The model successfully identified nine plant species, with a prediction accuracy of 1.0 for each species, as validated by the confusion matrix. Its robust performance across varying environmental conditions makes it suitable for real-time applications in agriculture and biodiversity conservation. The YOLOv5-based approach presents a reliable and scalable solution for plant species identification, offering significant improvements in detection accuracy and speed. This research provides a strong foundation for future advancements in automated plant recognition systems.

DECLARATIONS

Conflict of Interest

In completing this paper, the authors affirm that no competing interests exist.

CRediT Authorship Contribution

Billi Clinton: Conceptualization, Investigation, Methodology, Data curation, Validation, Visualization, Formal analysis, Writing-Original draft; Amperawan: Conceptualization, Supervision, Resources, Methodology, Validation, and Formal analysis; Tresna Dewi: Resources, Formal analysis, Writing-Reviewing and Editing.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Acknowledgment

The authors would like to thank the Electrical Engineering Department and Politeknik Negeri Sriwijaya for providing the academic and research environment that supported this study.

REFERENCES

[1] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "PANet: few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9196–9205. doi: 10.1109/ICCV.2019.00929.

[2] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Comput. Electron. Agric.*, vol. 157, pp. 417–426, 2019, doi: 10.1016/J.COMPAG.2019.01.012.

[3] I. H. Sarker, M. H. Furhad, and R. Nowrozy, "AI-driven cybersecurity: An overview, security intelligence modeling and research directions," *SN Comput. Sci.*, vol. 2, Art. no. 173, Mar. 2021, doi: 10.1007/S42979-021-00557-0.

[4] I. Sarker, M. M. Hoque, M. K. Uddin, and T. Alsanoosy, "Mobile data science and intelligent apps: Concepts, AI-based modeling and research directions," *Mob. Networks Appl.*, vol. 26, pp. 285–303, 2021, doi: 10.1007/s11036-020-01650-z.

[5] P. Akhtar, A. M. Ghouri, H. U. R. Khan, M. A. ul Haq, U. Awan, N. Zahoor, Z. Khan, and A. Ashraf, "Detecting fake news and disinformation using artificial intelligence and machine learning to avoid supply chain disruptions," *Ann. Oper. Res.*, vol. 327, pp. 633–657, 2023, doi: 10.1007/s10479-022-05015-5.

[6] R. Kaur, D. Gabrijelčič, and T. Klobučar, "Artificial intelligence for cybersecurity: Literature review and future research

directions," *Inf. Fusion*, vol. 97, 2023, Art. no. 101804 doi: https://doi.org/10.1016/j.inffus.2023.101804.

[7] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2022, doi: https://doi.org/10.1016/j.procs.2022.01.135.

[8] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *2017 IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690.

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

[10] A. M. Roy, J. Bhaduri, T. Kumar, and K. Raj, "WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection," *Ecol. Inform.*, vol. 75, 2023, Art. no. 101919, doi: https://doi.org/10.1016/j.eco inf.2022.101919.

[11] J. Wu, J. Dong, W. Nie, and Z. Ye, "A lightweight YOLOv5 optimization of coordinate attention," *Appl. Sci.,* vol. 13, no. 3, 2023, Art. no. 1746, doi: 10.3390/APP13031746.

[12] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Comput. Electron. Agric.*, vol. 178, 2020, Art. no. 105742, doi: 10.1016/j.compag.2020.105742.

[13] M. I. Mardiyah, "Implementasi Deep learning untuk image classification menggunakan algoritma convolutional neural network (CNN) pada citra kebun dan sawah," B.S. thesis, Univ. Islam Indonesia, 2020. [Online]. Available: https://dspace.uii.ac.id/123456789/28083

[14] S. M. Ali, "Comparative analysis of YOLOv3, YOLOv4 and YOLOv5 for sign language detection," *Int. J. Adv. Res. Innov. Ideas Educ.*, vol. 7, no. 4, pp. 2393–2398, 2021, Accessed: Dec. 16, 2024. [Online]. Available: http://ijariie.com/FormDetails.aspx?MenuScriptId=214132

[15] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," *J. Big Data*, vol. 8, 2021, Art. no. 66, doi: 10.1186/s40537-021-00434-w.

[16] M. Afonso, A. Mencarelli, G. Polder, R. Wehrens, D. Lensink, and N. Faber, "Detection of tomato flowers from greenhouse images using colorspace transformations," in *Proc. 19th EPIA Conf. Artif. Intell. Lect. Notes Comput. Sci.,* vol. 11804, 2019, pp. 146–155. doi: 10.1007/978-3-030-30241-2_13.

[17] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," Jun. 2016, Accessed: May 08, 2024. [Online]. Available: http://arxiv.org/abs /1606.02147

[18] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and Matti Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, pp. 261–318, 2020, doi: 10.1007/s11263-019-01247-4.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, 1998, pp. 2278–2324, doi: 10.1109/5.726791.

[20] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia Comput. Sci.*, vol. 132, pp. 377–384, 2018, doi: https://doi.org/10.1016 /j.procs.2018.05.198.

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732. doi: 10.1109/CVPR.2014.223.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[23] H. Yuan, Z. Lu, R. Zhang, J. Li, S. Wang, and J. Fan, "An effective graph embedded YOLOv5 model for forest fire detection," *Comput. Intell.*, vol. 40, no. 2, Apr. 2024, Art. no. e12640, doi: https://doi.org/10.1111/coin.12640.

[24] ShengjieCheng, P. Zhou, YuLiu, HongjiMa, A. Aysa, and K. Ubul, "Research on knowledge distillation algorithm based on Yolov5 attention mechanism," *Expert Syst. Appl.*, vol. 240, 2024, Art. no. 122553, doi: 10.1016/j.es wa.2023.122553.

[25] T. Carneiro, R. V. M. Da Nobrega, T. Nepomuceno, G.-B. Bian, V. H. C. Albuquerque, and P. P. R. Filho, "Performance analysis of google colaboratory as a tool for accelerating deep learning applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018, doi: 10.1109/ACCESS.2018.2874767.

[26] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, 2021, Art. no. 217, doi: 10.3390/f12020217.

[27] M. Horvat, L. Jelečević, and G. Gledec. (Sep. 2022). A comparative study of YOLOv5 models performance for image localization and classification. Presented at 33rd Central European Conf. Information and Intelligent Systems. [Online]. Available: http://archive.ceciis.foi.hr/public/conferences/2022/Proceedings/IS/IS4.pdf

[28] M. H. Hamzenejadi and H. Mohseni, "Fine-tuned YOLOv5 for real-time vehicle detection in UAV imagery: Architectural improvements and performance boost," *Expert Syst. Appl.*, vol. 231, 2023, Art. no. 120845, doi: https: //doi.org/10.1016/j.eswa.2023.120845.

[29] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, 2022, Art. no. 464, doi: 10.3390/S22020464.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[31] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, Tong Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network" in *Proc. 2019 IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8439–8448, doi: 10.1109/ICCV.2019.00853.

[32] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," Apr. 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[33] M. N. Bramasta, M. Anshar, and I. Nurtanio, "Prototipe sistem pengenalan pelat kendaraan otomatis berbasis YOLO pada mekanisme pintu masuk departemen elektro UNHAS," in *Proc. Seminar Nasional Elektroteknik dan Teknologi Informasi,* 2022. [Online]. Available: https://www.researchgate.net/publication/360756309_Prototipe_Sistem_Pengenalan_Pelat_Kendaraan_Otomatis_Berbasis_YOLO_pada_Mekanisme_Pintu_Masuk_Departemen_Elektro_UNHAS

[34] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Univ. Washington, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[35] Y. Song, Q.-K. Pan, L. Gao, and B. Zhang, "Improved non-maximum suppression for object detection using harmony search algorithm," *Appl. Soft Comput.*, vol. 81, 2019, Art. no. 105478, doi: https://doi.org/10.1016/j.asoc.2019.05.005.

[36] C. Guo, M. Cai, N. Ying, H. Chen, J. Zhang, and D. Zhou, "ANMS: attention-based non-maximum suppression," *Multimed. Tools Appl.*, vol. 81, pp. 11205–11219, 2022, doi: 10.1007/s11042-022-12142-5.

[37] B. B. Dursa and K. K. Tune, "Developing traffic congestion detection model using deep learning approach : A case study of Addis Ababa city road," 2020, doi: 10.21203/rs.3.rs-113234/v1.

[38] A. Tiendrebeogo, "Identification of plants from the convolutional neural network," *Multimed. Tools Appl.*, vol. 83, pp. 63121–63131, 2024, doi: 10.1007/s11042-023-18030-w.

[39] S. Patil and M. Sasikala, "Accurate plant species analysis for plant classification using convolutional neural network architecture," *Int. J. Reconfigurable Embed. Syst.*, vol. 13, no. 1, pp.160–170, 2024, doi: 10.11591/IJRES.V13.I1.PP160-170.