Journal of Information Systems and Informatics
Vol. 6, No. 4, December 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v6i4.889 Published By DRPM-UBD

Enhancing Real-time Herbal Plant Detection in Agricultural Environments with YOLOv8

Ranty Deviana Siahaan[1], Herimanto[2], Iustisia Natalia Simbolon[3], Ivanston Simbolon[4], Dian Jorgy Gultom[5]

[1, 2,3,4,5] Informatics, Faculty of Informatics and Electronic Engineering, Del Institute of Technology
Email: [1]ranty.siahaan@del.ac.id, [2]pardzheri@gmail.com, [3]iustisia.simbolon@del.ac.id, [4]ivanstonsimbolon99@gmail.com, [5]dianjorgy02@gmail.com

Abstract

The detection of herbal plants plays a crucial role in the utilization of traditional medicine, particularly in the Toba region of Indonesia. This study aims to develop an Android application capable of real-time detection of herbal plants using the YOLOv8 algorithm. The five types of herbal plants targeted in this study are tempuyung, rimbang, papaya leaves, turmeric leaves, and aloe vera. The research methodology includes the collection of a dataset of herbal plant images, which were then labeled using the Roboflow platform. The YOLOv8 model was trained with this dataset to detect herbal plant objects. After training, the model was exported to TensorFlow Lite and integrated into an Android application. Testing was conducted to evaluate the accuracy and real-time detection performance of the application. The results show that the YOLOv8 model achieved a mean Average Precision (mAP) of 92.4%, with optimal real-time detection capabilities on Android devices. The developed application can quickly and accurately detect and identify herbal plants, providing a practical solution for users to recognize herbal plants. This study indicates that the YOLOv8 algorithm is effective for herbal plant recognition applications in a mobile context, opening up opportunities for further development in the integration of AI technology into everyday applications.

Keywords: Herbal plant detection, YOLOv8, object detection, Android Application, computer vision, Real – Time Identification.

1. INTRODUCTION

Herbal plants have become a significant focus in research due to their extensive applications in health, food, and cosmetics. Despite their potential, the accurate identification of plant species demands in-depth botanical knowledge and often necessitates expert assistance, making the process time-consuming and inaccessible for general populations[1], [2]. In response to these challenges, advanced algorithms have emerged as promising solutions in detecting and identifying objects, including herbal plants. YOLOv8, the latest in these series of algorithms, promises to enhance the efficiency and speed of object detection in images, crucial for real-time applications [2][3].

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

This study aims to harness the YOLOv8 algorithm in an Android-based application to identify five specific herbal plants in the Toba region, Indonesia. The application intends to provide a real-time identification tool through the camera of an Android device, addressing the need for rapid and accurate plant identification [4][5]. Previous studies have shown that while the VGG16 model achieved a detection accuracy of 92%, it is hampered by slow processing times due to its two-stage detection approach [6]. In contrast, the YOLOv5 algorithm, although fast, only achieved an mAP of 83% in real-time applications [7]. YOLOv8, with its advanced processing capabilities and improved algorithmic structure, offers a solution that could potentially bridge the gap between traditional botanical knowledge and modern technology, thus supporting sustainable health and environmental practices[2][3][7].

The objectives of this research are twofold. First, it seeks to evaluate the accuracy of the YOLOv8 algorithm in visually detecting specific herbal plant objects in a real-time setting. Second, it aims to develop an application that can be used seamlessly on Android devices to identify these plants accurately and swiftly, contributing positively to the fields of traditional medicine and agricultural technology[1][4][8]. By addressing these aims, the study not only seeks to advance the technology of object detection but also to enhance the accessibility and efficiency of herbal plant identification, facilitating broader utilization of these plants in various beneficial applications.

2. METHODS

The main stages carried out in real-time object detection with YOLOv8 can be seen in Figure 1. The methodology involves a detailed configuration of the YOLOv8 model, where parameters such as learning rate, batch size, and epochs are optimized for agricultural imagery. The model utilizes advanced data augmentation techniques including rotation, scaling, and brightness adjustment to improve robustness and performance across varied environmental conditions. These methods are meticulously documented to ensure reproducibility and clarity in application, providing a comprehensive blueprint for implementing this technology in real-world scenarios.

The dataset used in this study is herbal plant image data. The images were collected from the Kaggle Platform, Mendeley Dataset, Scraping image from Bing and also direct image capture. The data generated uses the .jpg, .jpeg, and .png formats. The result of obtaining herbal plant image data are 694 images divided into 5 classes. The variations collected are images of Turmeric Leaves, Papaya Leaves, Aloe Vera, Rimbang, Tempuyung. The following is an explanation of the research stage.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

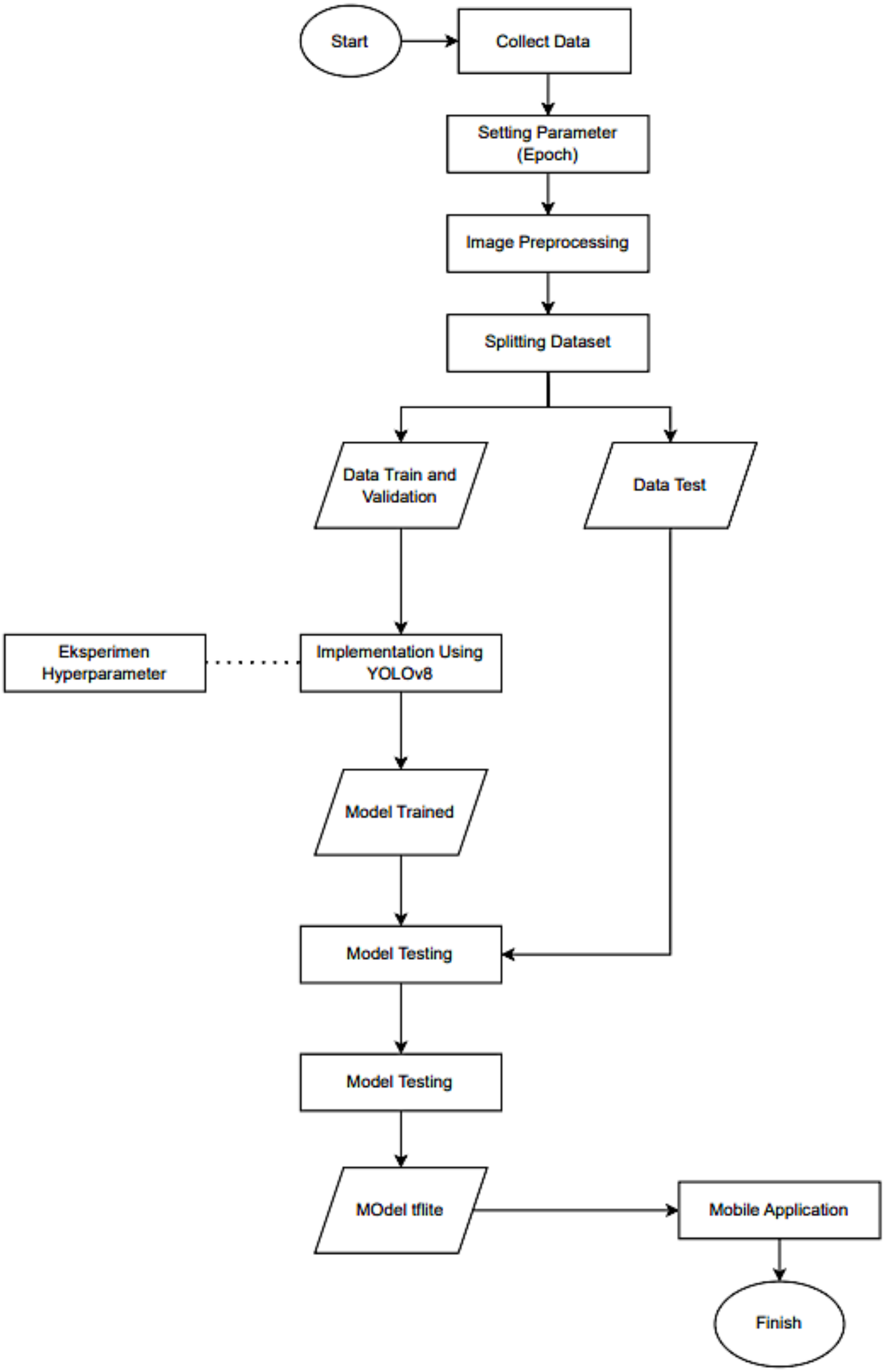

Figure 1. Research Methods

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.1. Image Preprocessing

Image Preprocessing is a data processing technique that consists of steps to transform image so that it can be easily processed by machines[9]. Auto-orient ensures consistent image orientation before training a machine learning model, preventing inaccurate object detection and improving model performance. Auto-orientation adjusts the image according to EXIF metadata, ensuring that the data provided to the model is a correct visual representation[10]. Object annotation is a stage carried out by providing information on each existing data to be used in the training and testing process[11]. Annotation is done using the Roboflow tool. The result of object is a file in .txt formats as shown in Figure 2.

<Object_id> <x_center> <y_center> <width> <height>

Figure 2. Object Annotation

Information related to object annotation includes object class identification (usually an integer number starting from 0), x and y coordinates of the center of the bounding box, and width and heigh of the bounding box. The dataset is divided into training data, validation data, and testing data to avoid overfitting and evaluate the model. The augmented train data is used for model training with the YOLOv8 algorithm. At this stage, the model is trained with the train data, which enables it to learn the patterns and features of the objects in the images.

The process begins with augmentation which is performed to increase the variety of image data. The train data that has gone through the preprocessing stage is augmented using the flip technique, 90° rotation, and brightness adjustments, increasing the diversity of the training data and improving the model's ability to generalize to new, unseen data. The augmented train data is then used for model training with the YOLOv8 algorithm, enabling the model to learn the patterns and features of the objects in the images. The model's performance is evaluated using object detection metrics such as precision, recall, and mAP (mean Average Precision), providing a comprehensive understanding of the model's performance. To ensure compatibility with Android devices, the model is converted from PyTorch format to TFLITE format using the TensorFlow Lite Converter. Finally, a mobile application is developed, with a prototype design created to reflect the appearance and functionality of the application, which is then implemented into a ready-to-use application, providing a user-friendly interface for object detection.

2.2. YOLOv8

Deep learning has demonstrated superior capabilities in the field of computer vision [12], and the introduction of YOLOv8 in 2023 by Ultralytics has brought significant improvements with the use of a backbone with the C2f module. This

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

model is capable of handling various computer vision tasks such as object detection, segmentation, pose estimation, tracking, and classification, with better performance than previous versions. In object detection with YOLOv8, the image is divided into a grid measuring SxS[13], where each grid predicts the bounding box, coordinates x, y, width, height, and confidence score, which describes the confidence that the bounding box contains the object to be detected. Bounding boxes with confidence scores below the threshold are ignored, and only those with confidence scores higher than the threshold are retained. The Intersection over Union (IoU) is calculated to measure the overlap between the prediction bounding box and ground truth[14], and Non-Maximal Suppression (NMS) is applied to overcome overlapping bounding boxes by selecting the bounding box with the highest confidence score and removing the others. The final result is a bounding box on the input object detection along with a label as the object's identification [15]. The YOLOv8 algorithm architecture consists of three main parts, namely: Backbone, Neck, and Head, as illustrated in Figure 3.

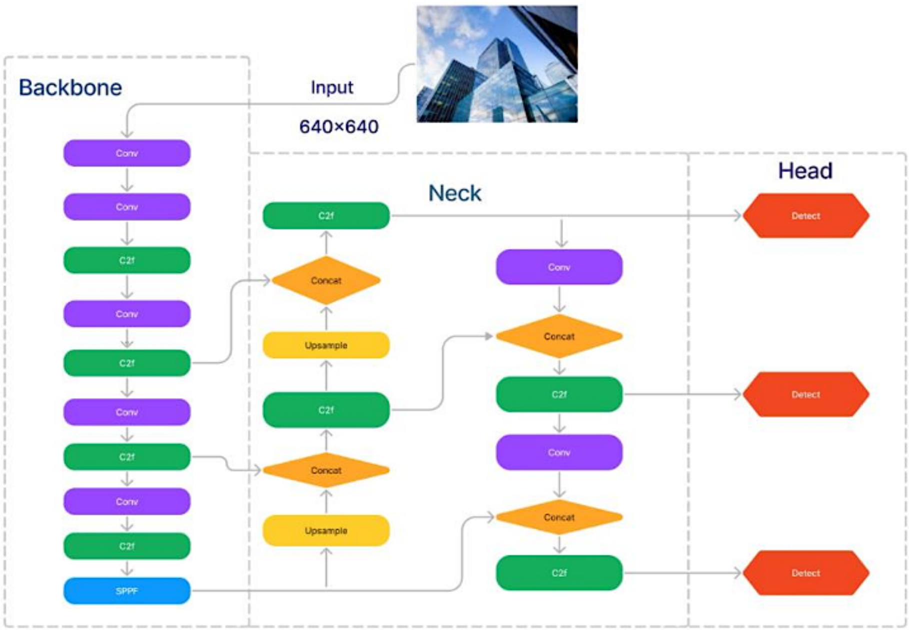

Figure 3. YOLOv8 Architecture [15]

The YOLOv8 backbone maintains a similar structure to YOLOv5, with the C2f module, based on the Cross Stage Partial (CSP) concept and the idea of ELAN in YOLOv7, providing a richer gradient flow and improving the model's ability to capture important image features. The Spatial Pyramid Pooling Fast (SPPF) module at the end of the backbone ensures high-accuracy object detection at various scales. The combination of C2f and SPPF modules enables YOLOv8 to maintain a lightweight structure, suitable for real-time applications that require fast

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

inference without sacrificing accuracy[16]. In the Neck section, the feature merging method used is still PAN-FPN (Path Aggregation Network – Feature Pyramid Network), which strengthens the fusion and utilization of information from feature layers at various scales, and the upsampling process increases feature resolution to retain more details. The C2f module ensures that important information is not lost during the fusion process, and the separate head structure at the end of the neck module, first introduced in YOLOx, separates the classification and regression tasks to improve model accuracy. The Head module separates the classification branch from the localization branch to reduce the inherent conflict between classification and regression tasks, allowing YOLOv8 to handle classification and localization more effectively and independently, and providing more consistent and accurate results [16].

2.3. Roboflow

Roboflow is computer vision platform that enables users to build computer vision models faster and more accurately by providing better data collection, data pre-processing where pre-processing is a technique consisting of steps to change data so that it can be easily processed by machines[17], and model training technique. One of the features of this platform is the creation of datasets for object detection, which is commonly used in the YOLO algorithm. Roboflow was founded in January 2020.

2.4. Confusion Matrix

In using the YOLO algorithm for object detection, the Confusion Matrix consisting of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) is generated from the Intersection over Union (IoU) values determined, with and IoU threshold set at 0.5 for this study. This matrix is important for calculating precision and recall, which are then used in calculating Average Precision (AP) and mean Average Precision (mAP). The precision, recall, and mean Average Precision (mAP) are calculated as shown in Equation 1 to 3 [18].

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$mAP = \sum_{i=1}^{N} \sum \frac{AP(i)}{N} \times 100\% \tag{3}$$

Furthermore, this study utilizes Tensorflow Lite (TFLite), an open source framework developed by Google to run machine learning models on mobile and edge devices, which demonstrates its ability to support AI-based applications in

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

various domains, particularly in devices with limited resources such as embedded systems, where memory and computing power are very limited.

2.5. Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) is a process that describes the stages and activities involved in developing software from concept to deployment and maintenance, providing a structured framework for managing and controlling the software development process[19]. The waterfall method, a classic and well-established SDLC methodology, offers a structured and linear approach to project development, progressing through distinct phases, with each phase building on the completion of the previous phase, including requirements gathering, design, development, and testing [20]. This approach is best suited for projects with well-defined and stable requirements, where changes are expected to be minimal.

3. RESULTS AND DISCUSSION

3.1 Image Preprocessing

Image data was collected from the Kaggle platform, Mendeley Dataset, scraping image from Bing and direct image capture. The data generated uses .jpg, .jpeg, and .png formats. The results of obtaining herbal plant image data are 694 images divided into 5 classes can be seen in Table 1.

Table 1. Data Collection

Herbal Plant	Amount Of Data
Turmeric-leaves	109
PaPaya-leaves	175
Aloe vera	117
Rimbang	166
Tempuyung	127

The data collected above will be used in the preprocessing stage with the Roboflow platform. Each image collected will be used for training and evaluating the model to detect herbal plants. After collecting the image data to be used, the image will then go through the image preprocessing stage. This stage begins by activating the auto-orient feature to ensure that the visual orientation of the image is consistent before the machine learning model training process is carried out. Furthermore, the image data will be resized to match the overall size of the image data. The results of the image resize that has been collected can be seen in Figure 4.

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Figure 4. Image Resizing

The image data is processed using auto-orient to avoid a common problem in computer vision projects where images that are not properly oriented can lead to inaccurate annotations or object detections, affecting the overall model performance. The image data is resized to 640 x 640 pixels to ensure that all images have a uniform size.

3.2 Result of Image Data Annotation

The results of the images that have been standardized to 640 x 640 pixels, the images are then labeled using Roboflow. The total annotated images are 694 image data and in all of the image data there are 880 herbal plant objects that are annotated. This annotation is done by marking herbal plant objects with rectangular bounding boxes. With this annotation, the dataset used in this study is image data in .jpg, .jpeg and .png formats, and the annotation data is .txt format data. The following are the results of the annotation with .txt format as in Figure 5.

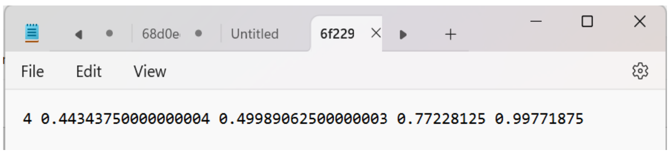

Figure 5. Annotation Image

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The result of the image annotation is a .txt file containing 5 attributes. the first attribute is the class id which indicates the plant class, namely "4", then followed by the midpoint of the bounding box (x, y) namely the x attribute has a value of "0.44343750000000004" and the y attribute has a value of "0.49989062500000003", the next attribute is the width of the bounding box, namely "0.77228125" and the last attribute is the height of the bounding box, namely "0.99771875". The results of this annotation are very important for training object detection models, because they ensure that the model can learn from data that has been correctly labeled.

3.3 Augmentation

Augmentation is done to enrich the dataset with visual variations without requiring additional data. This is useful for increasing the model's robustness to new data and ensuring better performance when applied to real-world scenarios. The augmentation techniques applied are 90-degree rotation, brightness, and flip.
In this study, 90° clockwise, counter-clockwise, and upside-down rotations were performed as in Figure 6.

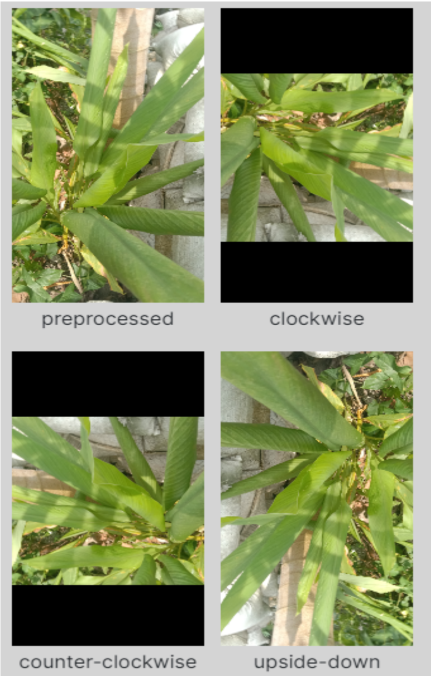

Figure 6. 90° Rotate

Brightness augmentation is done by adjusting the intensity of the image pixels to make them 20% brighter or 20% darker than the original. The purpose of this

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

augmentation is to simulate varying lighting conditions, which is important in improving the robustness of the computer vision model to changes in brightness. This adjustment improves the generalization ability of the model on images with different lighting conditions. It can be seen in Figure 7. Flip augmentation, which includes horizontal flip and vertical flip, is used to enrich the dataset by increasing the variation in image orientation. This allows the computer vision model to learn to recognize objects and features in different orientations. This can be seen in Figure 8.

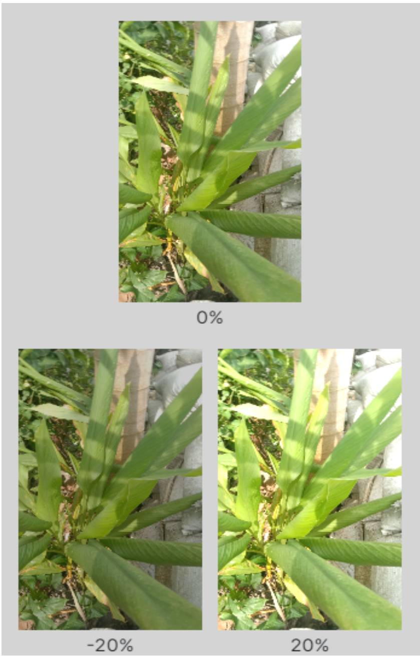

Figure 7. Brightness.

Figure 8. Flip

3.4 Result of Splitting Dataset

The train-test split ratio involves dividing the dataset into three subsets, namely the training set, validation set, and test set From Table 2.

Table 2. Splitting Dataset

Train Set (%)	Validation set (%)	Test set (%)	mAP	Training Time
54	36	10	0.916	0.085 *hours*
63	27	10	0.944	0.095 *hours*
72	18	10	0.946	0.104 *hours*
81	9	10	0.938	0.113 *hours*

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Based on the results of this experiment, dividing the dataset with 72% for the train set, 18% for the validation set, and 10% for the test set produces the best model performance with the highest mAP of 0.946. The 72:18:10 dataset division produces a higher mAP than other divisions for several reasons. First, this division provides an optimal balance between the amount of training data and validation data. With 72% of the data for training, the model has enough data to thoroughly learn important features of the dataset. At the same time, 18% of the validation data provides a large enough dataset for accurate validation without sacrificing too much training data.

3.5 Training Model

Based on the data splitting experiment, the best ratio for train, validation, and testing is 72:18:10. From these results, 72% of the train data will be used to train the model. The following are the results of the experiment in forming a model in detecting herbal plants which were carried out to obtain high accuracy or mAP.

Table 3. Training Model

Epoch	Model Evaluation		
	Precision	**Recall**	**mAP50**
25	0.924	0.875	0.946
50	0.889	0.915	0.938
100	0.854	0.901	0.918
500	0.893	0.872	0.917

The results of Table 3 show that the training time increases with the increase in the number of epochs. At 25 epochs, the training time required is 0.111 hours, at 50 epochs the training time increases to 0.222 hours, at 100 epochs the training time further increases to 0.439 hours, and at 500 epochs the training time required is 2.202 hours. From these results, it can be seen that the increase in the number of epochs is directly proportional to the training time required. The reason why mAP at epoch 25 is higher than the others is because at a lower number of epochs, the model is at an optimal point in the learning process. At this stage, the model has learned enough important features of the dataset without experiencing overfitting.

3.6 Model Evaluation

This section will discuss the model evaluation results obtained from previous training and object detection on test data. During the evaluation process, the model uses the knowledge gained from training to process validation data.

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	125	159	0.924	0.875	0.946	0.686
Daun Kunyit	125	32	0.835	0.688	0.887	0.69
Daun Pepaya	125	39	0.915	0.949	0.983	0.738
Lidah Buaya	125	26	0.987	1	0.995	0.722
Rimbang	125	43	0.89	0.791	0.879	0.582
Tempuyung	125	19	0.992	0.947	0.986	0.699

Figure 9. Model Evaluation

The performance evaluation of the epoch 25 model as the model with the highest mAP value is presented in the figure above which includes several important metrics for each class of herbal objects detected.

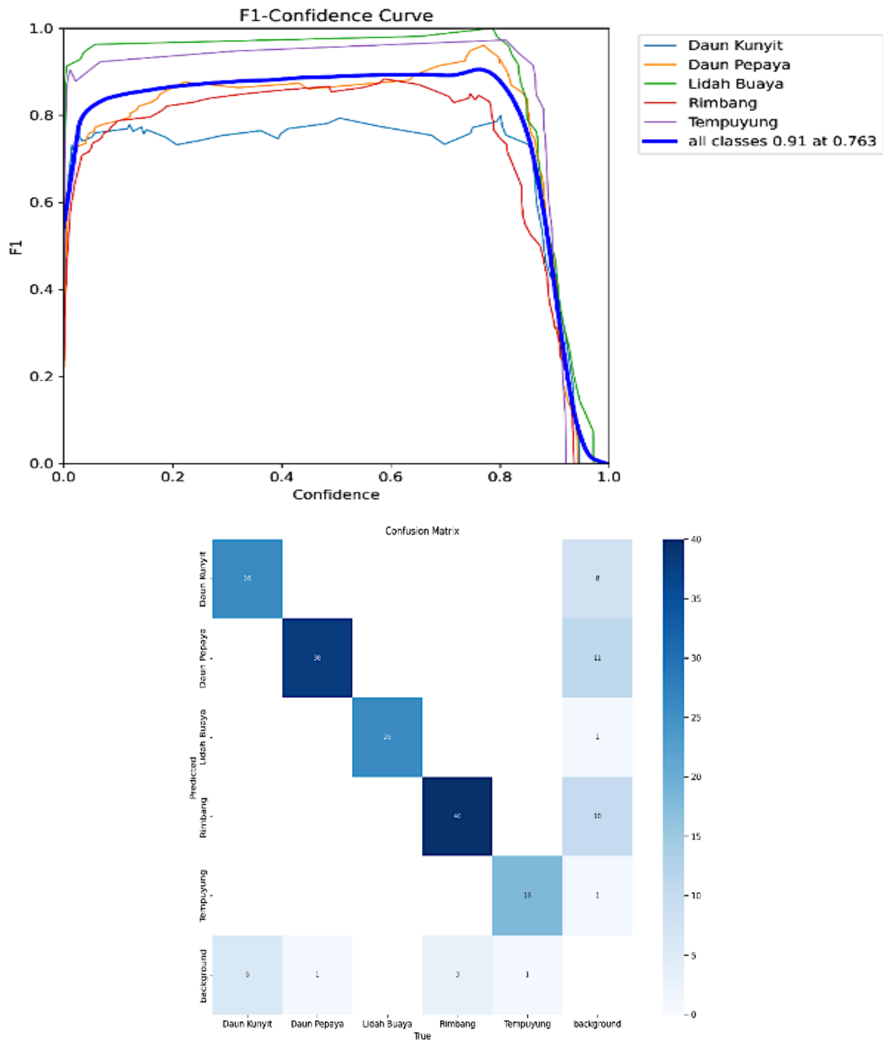

Figure 10. F1 Score and Confusion Matrix

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

In Figure 10 there is a graph that shows a comparison of the F1 score with the confidence value. The F1 score is a measure of the harmonic mean of recall and precision. That is, this score considers the number of objects successfully identified by the model and the accuracy of its predictions. A high F1 value indicates that the model has a good level of precision and recall. The confusion matrix above is a very useful evaluation tool in machine learning. The matrix displays six different classes: Kunyit-leaves, Aloe vera, Papaya-leaves, Rimbang, Tempuyung, and background. Each cell in the matrix represents the number of predictions for each pair of actual and predicted classes, with rows indicating the actual classes and columns indicating the predicted classes.

3.7 Real-Time Herbal Detection Test

After the best herbal plant detection model has been obtained, namely table 3 the best model at epoch 25 with precision 0.924, recall 0.875, and mAP50 0.946. The model is converted into Tensorflow Lite format (.tflite). Then the model is integrated into the Android application to perform object detection in real time.

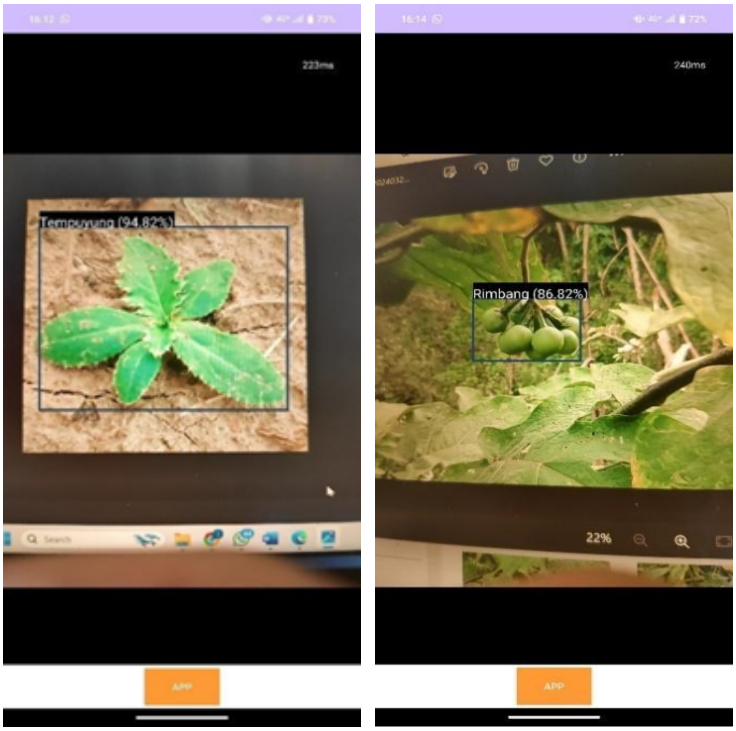

Figure 11. Real-Time Plant Detection Results

The following shows the results of real-time detection of herbal plants using a camera on an Android application. This application allows users to point their mobile phone camera at objects containing herbal plants and the model will immediately detect and recognize the plants. The detection results will be displayed in the form of annotations on the camera screen.

3.8 Discussion

The research on detecting herbal plants using machine learning techniques yielded valuable insights into the preprocessing, training, and evaluation processes. The discussion highlights key aspects of the methodology and their implications for improving herbal plant detection models. The preprocessing stage ensured consistent image orientation and uniform size, which are critical for training robust machine learning models. The use of the Roboflow platform facilitated efficient image standardization to 640 x 640 pixels, minimizing orientation-related errors and ensuring consistency across the dataset. Annotation of 880 herbal plant objects using bounding boxes was essential for creating an accurate and meaningful dataset, as this step enables models to learn spatial and class-specific features effectively. The careful preparation of this annotated dataset provided a strong foundation for subsequent augmentation and model training, demonstrating that high-quality preprocessing directly influences the performance of computer vision models (e.g., precision and recall metrics).

Data augmentation played a pivotal role in enhancing the model's robustness. Techniques such as rotation, brightness adjustment, and flipping enriched the dataset by simulating various real-world scenarios. For instance, brightness adjustments addressed variations in lighting, while flips and rotations introduced geometric transformations to diversify image orientations. These augmentations ensured that the model could generalize to unseen data, thereby reducing overfitting. The inclusion of these augmentations highlights the importance of synthetic data diversity in achieving better performance when deploying models in real-world conditions.

The experimental results of the dataset splitting strategies revealed that a 72:18:10 ratio for training, validation, and testing yielded the highest mAP of 0.946. This ratio provided a balanced trade-off between sufficient training data for learning key features and adequate validation data for model evaluation. The findings support the assertion that optimal data distribution is critical to model performance. The analysis further underscores that excessive allocation of data to training at the expense of validation can lead to suboptimal results due to insufficient model validation during training.

The training results demonstrated that a lower number of epochs (25 epochs) yielded better mAP compared to higher epochs, with 25 epochs achieving the highest mAP of 0.946. This outcome indicates that the model reached an optimal

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

learning point before overfitting began to occur. Training time, as expected, increased with more epochs, reinforcing the trade-off between computational cost and model accuracy. These results highlight the need to balance model complexity and computational efficiency while avoiding overfitting through careful monitoring of validation metrics.

Evaluation metrics, including the F1 score and confusion matrix, provided a comprehensive understanding of the model's precision and recall for different classes. The F1 score highlighted the harmonic balance between precision and recall, validating the model's capacity to accurately identify and predict plant objects. The confusion matrix further elucidated the model's ability to handle class imbalances and identify specific herbal plant types, emphasizing its real-world applicability. The successful deployment of the model in a real-time detection system integrated into an Android application demonstrated its practical potential. The model's ability to detect and annotate herbal plants in real time, with high precision and recall, is a significant milestone, suggesting its feasibility for applications in agriculture, medicine, and botanical research.

The study demonstrated the efficacy of combining robust preprocessing, augmentation, and model optimization techniques to achieve high-performance object detection. However, certain challenges remain, such as addressing class imbalances and further improving detection accuracy for underrepresented classes. Future research could explore the integration of more advanced augmentation techniques, transfer learning with pre-trained models, and fine-tuning for specific plant species to enhance model performance further. The findings underscore the critical role of comprehensive data preparation, augmentation, and evaluation in developing reliable object detection systems. This research lays the groundwork for future advancements in automated herbal plant detection, with potential applications in environmental monitoring, agricultural productivity, and herbal medicine quality assurance.

4. CONCLUSION

This study successfully implements the YOLOv8 algorithm for real-time herbal plant detection in the Toba area, achieving highly satisfactory results with a precision of 0.924, recall of 0.875, and mAP50 of 0.946. The integration of the detection model into an Android application using TensorFlow Lite was accomplished effectively, enabling users to identify herbal plants in real time with visible class labels on their devices. The structured Waterfall approach was utilized for its efficiency in linearly planning, implementing, and evaluating each development stage.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The success of this project opens several avenues for practical applications in real-world agricultural settings, where rapid and accurate plant identification can significantly aid in sustainable farming practices and biodiversity conservation. Future research could extend this model to detect a broader range of plant species, enhancing its utility in diverse ecological zones. Additionally, optimizing the model for resource-constrained mobile devices could broaden its accessibility, making this technology beneficial in regions with limited technological infrastructure. Such advancements could further integrate traditional botanical knowledge with modern technology, supporting health, environmental sustainability, and the global economy.

REFERENCES

[1] D. B. Valdez, C. J. G. Aliac, and L. S. Feliscuzo, "Detecting Medicinal Plants Using YOLOv5: A Mobile Vision Approach," *6th Int. Conf. Inven. Comput. Technol. ICICT 2023 - Proc.*, no. Icict, pp. 533–538, 2023, doi: 10.1109/ICICT57646.2023.10134246.

[2] Haryono, Khairul Anam, and Azmi Saleh, "Autentikasi Daun Herbal Menggunakan Convolutional Neural Network dan Raspberry Pi," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 3, pp. 278–286, 2020, doi: 10.22146/.v9i3.302.

[3] H. Xinru, C. Limin, X. Qing, L. Chongyuan, W. Yinchai, and F. Peijun, "Plant disease detection based on improved YOLOv5," *2023 2nd Int. Conf. Robot. Artif. Intell. Intell. Control. RAIIC 2023*, pp. 162–165, 2023, doi: 10.1109/RAIIC59453.2023.10280962.

[4] A. Kumar and D. B. Kumar, "Automatic Recognition of Medicinal Plants using Machine Learning Techniques," *Gedrag Organ. Rev.*, vol. 33, no. 01, pp. 166–175, 2020, doi: 10.37896/gor33.01/012.

[5] F. Liantoni and H. Nugroho, "Klasifikasi Daun Herbal Menggunakan Metode Naïve Bayes Classifier Dan Knearest Neighbor," *J. Simantec*, vol. 5, no. 1, pp. 9–16, 2015.

[6] A. E. Sukmayadi, S. A. Sumiwi, M. I. Barliana, and A. D. Aryanti, "The Immunomodulatory Activity of Ethanol Extract of Tempuyung Leaves (Sonchus arvensis Linn.)," *Indones. J. Pharm. Sci. Technol.*, vol. 1, no. 2, pp. 65–72, 2014, doi: 10.15416/ijpst.v1i2.7515.

[7] Z. Chen *et al.*, "Plant Disease Recognition Model Based on Improved YOLOv5," *Agronomy*, vol. 12, no. 2, 2022, doi: 10.3390/agronomy12020365.

[8] C. Y. Pelokang, R. Koneri, and D. Katili, "Pemanfaatan Tumbuhan Obat Tradisional oleh Etnis Sangihe di Kepulauan Sangihe Bagian Selatan, Sulawesi Utara (The Usage of Traditional Medicinal Plants by Sangihe Ethnic in the Southern Sangihe Islands, North Sulawesi)," *J. Bios Logos*, vol. 8, no. 2, p. 45, 2018, doi: 10.35799/jbl.8.2.2018.21446.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

[9] H. Herimanto, "Perbandingan Matriks Loss Pada Model Deep Learning Resnet50 dan Xception dalam Deteksi Objek," *J. Media Inform. Budidarma*, vol. 7, no. 4, pp. 1994–2002, 2023, doi: 10.30865/mib.v7i4.6849.
[10] B. Dwyer, "When Should I Auto-Orient My Images."
[11] M. Geiß, R. Wagner, M. Baresch, J. Steiner, and M. Zwick, "Automatic Bounding Box Annotation with Small Training Datasets for Industrial Manufacturing," *Micromachines*, vol. 14, no. 2, pp. 1–20, 2023, doi: 10.3390/mi14020442.
[12] L. Marifatul Azizah, S. Fadillah Umayah, and F. Fajar, "Deteksi Kecacatan Permukaan Buah Manggis Menggunakan Metode Deep Learning dengan Konvolusi Multilayer," *Semesta Tek.*, vol. 21, no. 2, pp. 230–236, 2018, doi: 10.18196/st.212229.
[13] C. Y. Pelokang, R. Koneri, and D. Katili, "Pemanfaatan tumbuhan obat tradisional oleh Etnis Sangihe di Kepulauan Sangihe bagian selatan, Sulawesi Utara," *J. Bioslogos*, vol. 8, no. 2, pp. 45–51, 2018.
[14] K. Khairunnas, E. M. Yuniarno, and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *J. Tek. ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373539.v10i1.61622.
[15] C. Geraldy and C. Lubis, "Pendeteksian Dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once Dan Convolutional Neural Network," *J. Ilmu Komput. dan Sist. Inf.*, vol. 8, no. 2, p. 197, 2020, doi: 10.24912/jiksi.v8i2.11495.
[16] A. K. *et al.*, "Malayalam Sign Language Identification using Finetuned YOLOv8 and Computer Vision Techniques," 2024, [Online]. Available: http://arxiv.org/abs/2405.06702
[17] Herimanto, A. S. Dharma, H. Manurung, and A. B. Nababan, "Comparison of Support Vector Regression and Artificial Neural Network Algorithm in Predicting the Number of Prospective Student Registrans," *Proc. 2023 IEEE Int. Conf. Data Softw. Eng. ICoDSE 2023*, pp. 43–48, 2023, doi: 10.1109/ICoDSE59534.2023.10291384.
[18] B. Wang, Y. Yan, Y. Lan, M. Wang, and Z. Bian, "Accurate Detection and Precision Spraying of Corn and Weeds Using the Improved YOLOv5 Model," *IEEE Access*, vol. 11, no. January, pp. 29868–29882, 2023, doi: 10.1109/ACCESS.2023.3258439.
[19] U. Islam and N. Sumatera, "ALACRITY : Journal Of Education," vol. 1, no. 2, pp. 1–12, 2021.
[20] M. I. H. -, "Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management," *Int. J. Multidiscip. Res.*, vol. 5, no. 5, 2023, doi: 10.36948/ijfmr.2023.v05i05.6223.