Identification types of plant using convolutional neural network

Radityo Hendratmojo Jati Notonegoro, Hustinawaty
Department of Information System Management, Universitas Gunadarma, Depok, Indonesia

Article Info

Article history:

Received Sep 25, 2024
Revised Jul 14, 2025
Accepted Sep 14, 2025

Keywords:

Convolutional neural network
Identification
Plant
Real time
You only look once

ABSTRACT

Artificial intelligence can be implemented in fields that related to environmental education by providing knowledge for taxonomy which recognize and identify plant species based on its features. The variety of plant species that inhabit in a certain area allows many plant species to be found that look similar so that difficult to distinguish and recognize a particular plant. Convolutional neural network (CNN) often used in object detection, you only look once (YOLO), one of CNN's object detections, could identify object in real time and obtained good performance and accuracy in several researched. However, no studies have ever identified a plant from its flowers, leaves, and fruits. Therefore, the main object of this paper is identified types of plant with CNN (YOLOv8). The YOLOv8 model with 0.01 learning rate, 32 batch size, stochastic gradient descent (SGD) optimizer obtained highest precision of 69.62% and F1 score of 61.22%, recall of 54.73%, mAP50 and mAP50 – 90 on the training data of 57.61% and 42.49%.

Corresponding Author:

Radityo Hendratmojo Jati Notonegoro
Department of Information System Management, Universitas Gunadarma
Kartika Jaya G 44, Cipayung, Jakarta, Indonesia
Email: raditjati06@gmail.com

1. INTRODUCTION

Artificial intelligence (AI) can significantly reduce human errors and save time and energy by handling simple tasks efficiently. In the field of environmental education, AI can also play a key role—particularly in taxonomy—by helping to identify and classify plant species based on their characteristics. Given the wide variety of plant species in a specific region, many of which appear quite similar, AI can assist in accurately distinguishing between them. This capability supports the conservation of endangered species and enhances our understanding of unique, endemic plants that exist only in certain areas. Additionally, AI can help identify edible and non-edible plants, as well as medicinal plants and economically valuable forest species. This knowledge can contribute to sustainable resource management and improve the commercial value of these plants. In agriculture, AI is useful for monitoring crop growth at every stage, enabling more sustainable and optimized harvests.

Object detection is an image classification-based task that requires bounding boxes as markers and identifies input images into appropriate categories [1]. The use of object detection to identify certain plants in real time can make it easier to recognize a type of plant. Convolutional neural network (CNN) often used to object detection, there are two types of detection algorithm, two stages and one stage. Two stages has dedicated module for generating region proposals, module firstly identifies a variable number of object proposals within an image, then the module classifying and localizing those proposals on second stage, like regions with CNN (RCNN), while one stage, like you only look once (YOLO), provides object classification and bounding boxes directly using a single feed forward fully convolutional neural network and key points of different scales and aspect ratios to identify objects. This design offers advantages over two-stage detectors in terms of real-time performance and simplicity [2]–[4].

YOLO models could implemented in many things: real time detection for ginger shoot and seeds [5], detection robots for picking mangos fruits in real time [6] and tomatoes for detecting ripe (mature), unriped (immature), semi mature, and diseased fruits [7], [8], disease detection on tomato leaf [9], [10], jute (*Corchorus olitorius* or *Corchorus capsularis*) disease from leaves and stem and pests (*Jute Hairy Caterpillar* and *Comophila sabulifers*) [11], tea leaf disease (blight) and pest (*Apolygus lucorum*) [12], determine which plant is healthy and has disease on soybean (*Glycine max*), okra (*Abelmoschus esculentus*), and maize (*Zea mays*) leaves [13], white grape fruit real time counting and bunch detection for grape yield decrease time estimation [14], Counting leaves of Arabidopsis plant (*Arabidopsis thaliana*) [15]. Based on research by Khan *et al.* [16] work on real time weeds detection in potato (*Solanum tuberosum*) crops using YOLOv4-tiny, the adopted model get 49.4% accuracy on very limited dataset. Abozar *et al.* [17] detect the damaged of the sugar beet (*Beta vulgaris*) roots by mechanical stress during harvesting using YOLOv4, the method be able to detect the damage with recall 92%, precision 94%, and F1 score 93% (better performance). Research from Yao *et al.* [18] detect the defect in kiwifruit using YOLOv5, the model reached 94.7% of mAP50. Another disease detector for tomato fruit with comparing method using obtained mAP Faster RCNN (80.8%), SSD (76.7%), and YOLO version 4, 5, 7, 8 (88.4%, 91.2%, 91.6%, 91.9%) [19], the rice and cotton disease using Fast RCNN, YOLO v7, YOLOv8 get mAP values (49.33%, 61.80%, 66.47%) for rice and (76.88%, 78.36%, 79.56%) for cotton [20].

YOLO version 8 (YOLOv8) was chosen because it has advantages: not using anchor boxes, reducing the number of prediction boxes, and accelerating non maximum impression [21]. This version of the YOLO model is considered more effective because it has an updated feature map and convolutional network [22], uses a task aligned assigner that computes a task alignment task matric using regression coordinates and the classification scores, combine with the value of intersection over union (IoU), allows localization and classification optimization simultaneously while suppressing prediction boxes which have low quality [23]. Many research in identifying plant using YOLO model implemented for detect and identify the diseases and crop damages in real time, but implementation YOLO model with version 8 for identifying plant based on the flowers, leaves, and fruits has not been carried out in the recognition of an object. The objective of this study is identifying types of plant with YOLOv8 model based on its features.

2. METHOD

In this research, the process of creating the model for plant species identification involves several key steps, as outlined in the main framework shown in Figure 1. The first step is to collect a diverse set of image data that accurately represents the plant species under study. Once the data is gathered, the next stage involves preprocessing, which includes resizing the images to a uniform size, annotating them to label the plant species, and applying data augmentation techniques to increase the dataset's variability. After preprocessing, the data is split into three distinct sets: a training set, a validation set, and a test set. The training and validation sets are used to train and fine-tune the model, while the test set is used to evaluate its performance. Roboflow is employed to manage the data and assist in model development, and the training and evaluation of the identification model are carried out using Google Collab, providing a flexible environment for deep learning tasks.

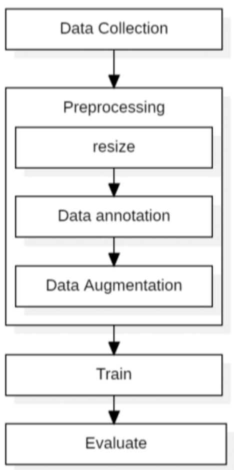


Figure 1. Main workflow

2.1. Data collection

The image data used as training, validation, and validation was collected from Kaggle (*https://www.kaggle.com*), Global Biodiversity Information Facility (GBIF) (*https://www.gbif.org*), and Roboflow universe (*https://universe.roboflow.com*) websites as many as 2,526 images with specifications of 640×640 pixels in JPEG and JPG formats. Data collected in the form of pictures of flowers, fruits, or leaves from 25 different plant species, such as: adam hawa ungu (*Tradescantia pallida*), anggrek (*Orchidaceae* spp.), heliconia (*Heliconia latispatha*), jengger ayam (*Celosia argentea*), kembang sepatu (*Hibiscus rosa sinensi*), kencana ungu (*Ruellia simplex*), marigold (*Tagetes* spp.), bunga matahari (*Helianthus* spp.), kana (*Canna* spp.), thunbergia (*Thunbergia laurifolia*), bunga telang (*Clitoria ternatea*), bunga patrakomala (*Caesalpinia pulcherrima*), cabai (*Capsium annum*), daun bawang (*Allium fistulosum*), kembang bokor (*Hydrangea macrophylla*), kemuning (*Murraya paniculata*), lidah mertua (*Sansevieria trifasciata*), miana (*Coleus scutellarioides),* pacar air (*Impatiens balsamina*), pucuk merah (*Syzygium myrtifolium*), puring (*Codiaeum variegatum*), sawi (*Brassica* spp.), selada (*Lactuca sativa*), sri rezeki (*Aglaonema* spp.), tomat (*Solanum lycopersicum*). One of sample picture representing each type of plant collected in Figure 2.

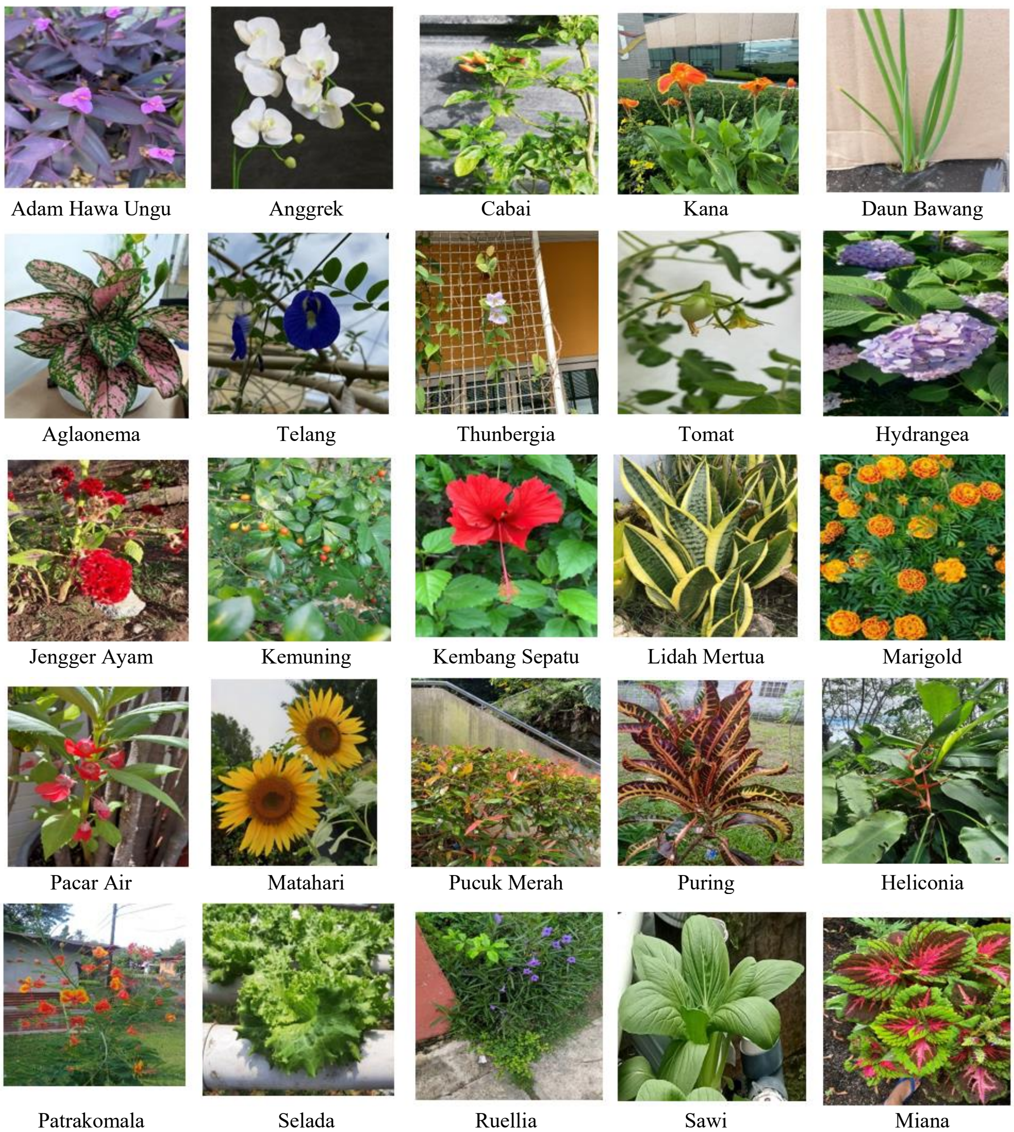
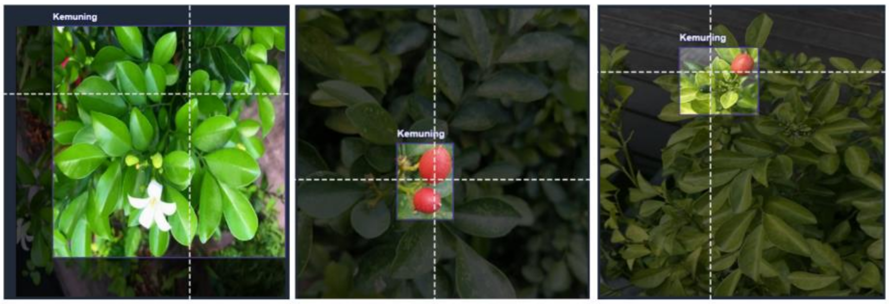


Figure 2. Types of plant sample

2.2. Preprocessing

The collected image data, which initially varies in pixel sizes, is resized to a standard resolution of 640×640 pixels to ensure consistency across the dataset. After resizing, the images are labeled and annotated with bound box around the object according to specific plant features such as the flower, leaf, and fruit to facilitate accurate classification, as shown in Figure 3. To enhance the robustness of the model and improve its generalization ability, data augmentation techniques are applied. These techniques include horizontal and vertical flipping, 90º rotations in both clockwise and counter-clockwise directions, as well as upside-down rotations. Additional augmentations such as varying the saturation between -50% and +50%, adjusting the brightness between -20% and +20%, and modifying exposure between -15% and +15% help simulate different environmental conditions. Furthermore, a blur effect of up to 1.5 px is applied to random images in the training set. These augmentation methods significantly increase the dataset size by generating up to 4,206 images, as shown in Figures 4(a) to 4(m), reducing the risk of overfitting by ensuring the model is exposed to a diverse range of image variations [24]. This also helps address class imbalances by artificially increasing the representation of underrepresented classes in the dataset, improving the model's ability to generalize across different plant types [25].



Leaf and Flower Fruits Leaf and Fruits

Figure 3. Annotated and labeled the kemuning (*Murraya paniculata*) with bounding box

(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l) (m)

Figure 4. Augmented image, (a) original, (b) flip (vertical), (c) flip (horizontal), (d) 90º rotations (clockwise), (e) 90º rotations (counter clockwise), (f) 90º rotations (upside down), (g) saturation (50%), (h) saturation (+50%), (i) brightness (darken: -20%), (j) brightness (brighten: +20%), (k) exposure (-15%), (l) exposure (+15%), and (m) blur 1.5 px

2.3. Train

The dataset contains 25 classes: adam hawa ungu (101 images), anggrek (100 images), heliconia (100 images), jengger ayam (102 images), kembang sepatu (100 images), kencana ungu (98 images), marigold (98 images), bunga matahari (100 images), kana (105 images), thunbergia (101 images), bunga telang (114 images), bunga patrakomala (102 images), cabai (102 images), daun bawang (109 images), kembang bokor (100 images), kemuning (101 images), lidah mertua (99 images), miana (97 images), pacar air (102 images), pucuk merah (88 images), puring (100 images), sawi (103 images), selada (104 images), sri rezeki (100 images), tomat (100 images), and allocated as shown in Table 1 split into: 80% of the data is for training the model, 10% is for validating the model and the remaining 10% is for testing the model. To enhance the robustness and generalization capability of the model, data augmentation techniques were applied prior to the training phase. These augmented and pre-processed images were then used to train the YOLOv8 model under various experimental settings, using combinations of the stochastic gradient descent (SGD) and Adam optimizers, batch sizes of 16 and 32, learning rates of 0.01 and 0.001, and training durations of 25, 50, 75, and 100 epochs. These variations were designed to explore and identify the most optimal set of parameters for accurate and efficient plant classification. The training process was conducted on Google Colab, utilizing a cloud-based environment equipped with an NVIDIA Tesla T4 GPU, running on CUDA version 12.0, and supported by 16 GB of memory. This hardware setup ensured sufficient computational power for handling high-resolution image data and large model architectures, while also accelerating training times across all experimental configurations. The details of the dataset allocation for training, validation, and testing purposes are comprehensively presented in Table 1, which serves as a reference for understanding the data distribution used throughout the model development process.

Table 1. Dataset allocation

Dataset	Train	Validation	Testing
Types of plant	3660	423	423

2.4. Evaluate

The results of the model training get precision, recall, as well as the accuracy of the mean average precision (mAP)50 and mAP50-90. The precision value measures the YOLOv8 model's prediction of positive instances correctly, recall measures the identification of the YOLOv8 model against positive instances, the F1 score is the average balance between precision and sensitivity, the F1 score is useful when the method has low sensitivity but high precision or high sensitivity but low precision [26], measure the classifier performance comprehensively [27], and a high F1 score indicates the model more robust [28], mAP calculates average precision against sensitivity values in the range 0-1 [29], compares performance between detectors [3], assesses the models of object detection performance across multiple categories [30], and provides model summary [31]. mAP50 expresses average precision at the IoU threshold of 50% and mAP50-90 expresses the average precision at the IoU threshold of 50% to 90% [10]. IoU calculate the quantification similarity of predicted bounding box (k_p) and ground truth bounding box (k_g) [32], IoU values that exceed a certain threshold, can be considered to produce true positive detection results [33], and objects that exceed IoU value of 50% can be classified as detected [34]. Average prediction (AP) value is needed to compute mAP [35]. The equations of precision [36], recall [37], F1 score [38] and mAP are formulated in (1) through (6).

$$precision\ (P) = \frac{TP}{TP+FP} \tag{1}$$

$$recall\ (R) = \frac{TP}{TP+FN} \tag{2}$$

$$F1\ score = \frac{2 \times Recall \times Precision}{recall+precision} \tag{3}$$

$$IoU = \frac{Area\ (k_p \cap k_g)}{Area\ (k_p \cup k_g)} \tag{4}$$

$$AP = \int_0^1 P(R)dR \tag{5}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AP}_i \tag{6}$$

In the evaluation of the YOLOv8 model's performance, several key metrics were used to interpret the accuracy and reliability of object detection and classification. True positive (TP) refers to instances where the model correctly identifies and classifies a specific plant object, and the predicted bounding box overlaps significantly with the ground truth. False positive (FP) occurs when the model correctly classifies an object but the predicted bounding box does not correspond to any actual object, leading to a mismatch. Conversely, False negative (FN) indicates that the model fails to detect or correctly classify a plant object that is present in the image. N denotes the number of object categories being detected, while AP_i represents the average precision (AP) for the *i-th* category, reflecting how well the model performs per class.

The performance outcomes are further visualized using a series of evaluation graphs. The box loss metric evaluates how accurately the predicted bounding boxes align with the true locations of the objects, serving as an indicator of localization performance. Classification loss (CLS loss) reflects how well the model distinguishes between different plant categories, highlighting its classification accuracy. The distribution focal loss (DFL loss) is particularly useful in scenarios involving class imbalance, as it helps refine predictions for categories that are underrepresented in the dataset. Additional metrics include accuracy, precision, and recall, which collectively describe the model's overall correctness and completeness in detection. The mean average precision (mAP) is reported at both mAP50) and mAP50-90, providing a more nuanced understanding of the model's robustness.

3. RESULTS AND DISCUSSION

The YOLOv8 model was trained through eight different experiments, each using a unique combination of hyperparameters such as learning rate, batch size, and optimizer settings, all conducted over 100 epochs. The outcomes of these experiments were evaluated using key performance metrics including precision, recall, F1 score, mean average precision at IoU thresholds of 0.50 (mAP50), and 0.50 to 0.95 (mAP50-90), with the summarized results presented in Table 2. In addition to the tabulated metrics, Table 2 also includes evaluation graphs illustrating the trends of box loss, classification loss (CLS loss), and distribution focal loss (DFL loss), as well as curves depicting the evolution of accuracy, precision, recall, and both mAP50 and mAP50-90 over the course of the training epochs. Furthermore, the table provides information on the total time required for each experiment, enabling a comprehensive comparison of training efficiency and model performance under different parameter settings.

Table 2. Precision, recall, F1 score, mAP50, and mAP50-90 values

Learning rate	Batch Size	Optimizer	Precision (%)	Recall (%)	F1 (%)	mAP50 (%)	mAP50-90 (%)	Time
0.01	16	SGD	63.11	53.44	57.88	56.35	41.14	125m 23s
0.01	16	Adam	65.17	51.53	57.48	53.44	37.22	125m 31s
0.01	32	SGD	**69.62**	54.73	**61.22**	57.61	42.49	123m 40s
0.01	32	Adam	56.23	52.76	54.30	52.86	37.20	125m 34s
0.001	16	SGD	55.61	57.17	56.44	55.49	39.71	113m 8s
0.001	16	Adam	60.07	58.59	59.28	**58.97**	**43.55**	114m 59s
0.001	32	SGD	62.24	**58.99**	60.64	55.94	39.85	**110m 44s**
0.001	32	Adam	63.89	53.85	58.52	57.53	42.05	111m 12s

The result in Figure 5 shows the train box loss decreasing trend over the training epochs, indicating that the model is learning to predict more accurate bounding boxes as training progresses. It seems to plateau towards the end, suggesting convergence. Also, the train class loss exhibits a decreasing trend, implying that the model is improving its ability to classify objects correctly during training. It also appears to be converging. Follows a similar decreasing pattern, the train DFL loss suggesting that the model is becoming better at predicting the precise distribution of bounding box coordinates. Generally, increases over training, the train precision indicating that the model is making fewer false positive bounding box predictions as it learns. It fluctuates, which is common during training. Shows an increasing trend initially on train recall, meaning the model is learning to detect more of the actual objects present. It seems to plateau or slightly decrease towards the end, which could be a sign of overfitting if the validation recall doesn't follow the same trend. The validation box loss decreases initially on but then seems to stabilize and might even slightly increase or fluctuate in the later epochs. This suggests that the model's ability to generalize its bounding box predictions on unseen data might have plateaued or started to slightly degrade. Decreases in the early stages but then plateaus and shows some fluctuations on validation class loss. This indicates that the classification performance on unseen data is no longer significantly improving. Similar to the box loss, the validation DFL loss shows an initial decrease followed by stabilization and some fluctuations. Increases significantly in the beginning and then starts to plateau. This suggests that the overall detection performance at a 50% IoU threshold on the validation set has reached a certain level and is no longer improving much. Shows a similar trend to mAP50 but with generally lower values, as expected due to the stricter IoU thresholds. The plateauing indicates that the model's ability to precisely localize objects on unseen data is also not improving significantly. Overall, the model on training phase losses is generally decreasing, indicating that the model is learning on the training data. However, the validation phase losses and mAP metrics have plateaued, suggesting that the model might have reached its optimal performance on the unseen data or is starting to overfit to the training data. Figure 6 shows the confusion matrix of 0.01 learning rate, 32 batch size, and SGD optimizer.

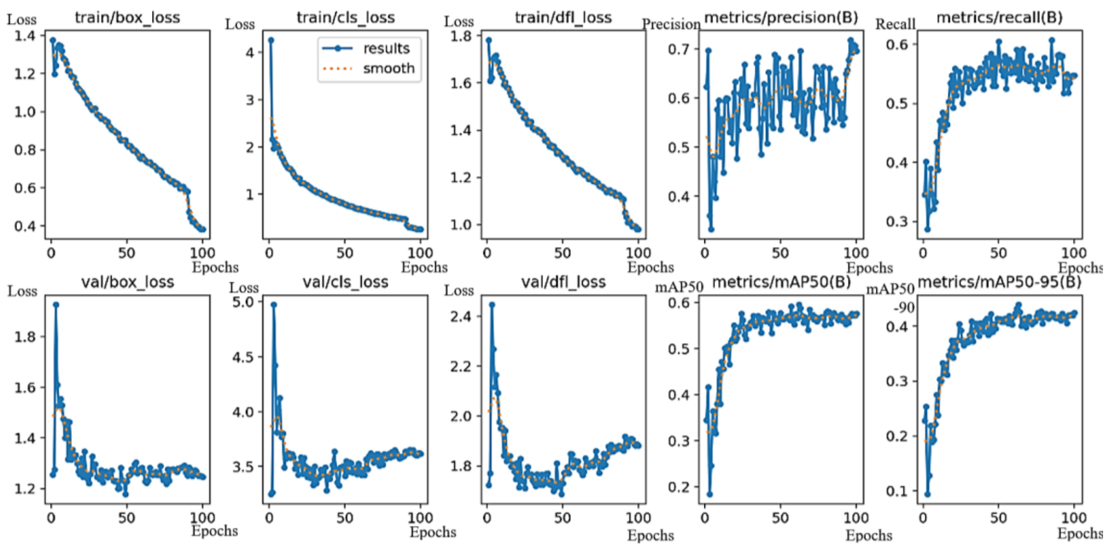


Figure 5. Model result of 0.01 learning rate, 32 batch size, and SGD optimizer

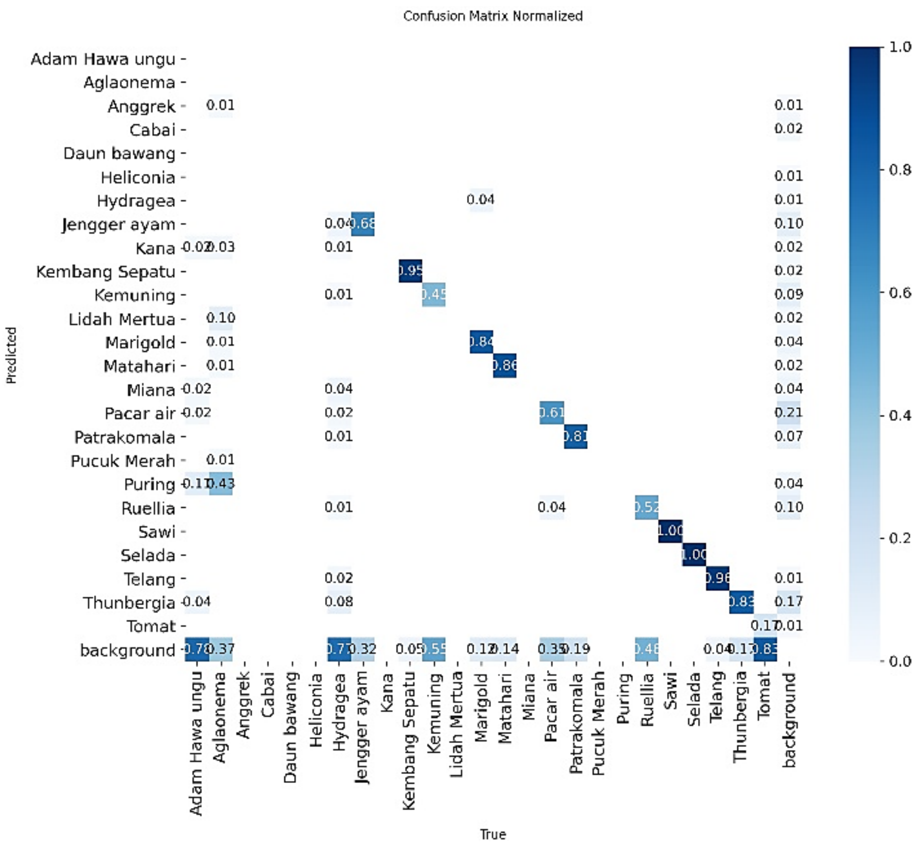


Figure 6. The confusion matrix of 0.01 learning rate, 32 batch size, and SGD optimizer

4. CONCLUSION

The YOLOv8 model with 0.01 learning rate, 32 batch size, SGD optimizer obtained highest precision of 69.62% and F1 score of 61.22%, recall of 54.73%, mAP50 and mAP50 – 90 on the training data of 57.61% and 42.49% for 123m 40s. The confusion matrix indicates the proportion of predictions for each actual class. The numbers outside the main diagonal clearly indicate classification errors. The main errors of the model lie in the difficulty of distinguishing between plants and the background, as well as confusion between certain plant classes, this because most of images collected in dataset have many objects complexity at background. the high valued classes, Sawi and Selada, those images do not have much object at background so the model could easily recognize the object. Besides, the label also using other parts like, flower, leaves, and fruits of the plants and not all of the type in dataset have those parts. In future, the collected data should reduce the image background to avoid data complexity, gather more image data and augmentation to avoid overfitting, and adjusting model hyperparameter such as learning rate and optimizer to prevent underfitting and instability in training process.

ACKNOWLEDGEMENTS

The author would like to express sincere appreciation to the second author, Hustinawaty, for her valuable guidance throughout this research. The author also gratefully acknowledges the Faculty of Information Systems and Management, Universitas Gunadarma, Indonesia, for their support in facilitating this study.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration. Radityo Hendratmojo Jati Notonegoro contributed to methodology, software development, validation, formal analysis, investigation, resources, data curation, writing – original draft preparation, writing – review and editing, visualization, and project administration. Hustinawaty contributed to conceptualization, writing – review and editing, supervision, and funding acquisition.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Radityo Hendratmojo Jati Notonegoro		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Hustinawaty	✓									✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available from multiple public repositories, including Kaggle, GBIF, and Roboflow Universe. Specific datasets were obtained from these sources and combined for the purpose of this research. The datasets can be accessed via the following platforms: Kaggle (*https://www.kaggle.com*) GBIF (*https://www.gbif.org*) Roboflow Universe (*https://universe.roboflow.com*).

REFERENCES

[1] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[2] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one stage and two stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, p. 89, Dec. 2020, doi: 10.3390/rs13010089.

[3] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing: A Review Journal*, vol. 126, 2022, doi: 10.1016/j.dsp.2022.103514.

[4] X. Yue, H. Li, M. Shimizu, S. Kawamura, and L. Meng, "YOLO-GD: a deep learning based object detection algorithm for empty dish recyclings robots," *Machines*, vol. 10, no. 5, p. 294, Apr. 2022, doi: 10.3390/machines10050294.
[5] L. Fang *et al.*, "Using channel and network layer pruning based on deep learning real time detection of ginger images," *Agriculture*, vol. 11, no. 12, p. 1190, Nov. 2021, doi: 10.3390/agriculture11121190.
[6] Z. Cao and R. Yuan, "Real-time detection of mango based on improved YOLOv4," *Electronics*, vol. 11, no. 23, p. 3853, Nov. 2022, doi: 10.3390/electronics11233853.
[7] G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, "A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention," *Agronomy*, vol. 13, no. 7, p. 1824, Jul. 2023, doi: 10.3390/agronomy13071824.
[8] P. Li, J. Zheng, P. Li, H. Long, M. Li, and L. Gao, "Tomato maturity detection and counting model based on MHSA-YOLOv8," *Sensors*, vol. 23, no. 15, p. 6701, Jul. 2023, doi: 10.3390/s23156701.
[9] A. M. Roy, R. Bose, and J. Bhaduri, "A fast accurate fine-grain object detection model based on YOLOv4 deep neural network," *Neural Computing and Applications*, vol. 34, no. 5, pp. 3895–3921, Mar. 2022, doi: 10.1007/s00521-021-06651-x.
[10] S. Zhao, Y. Peng, J. Liu, and S. Wu, "Tomato leaf disease diagnosis based on improved convolutional neural network by attention module," *Agriculture*, vol. 11, no. 7, p. 651, Jul. 2021, doi: 10.3390/agriculture11070651.
[11] D. Li, F. Ahmed, N. Wu, and A. I. Sethi, "YOLO-JD: a deep learning network for jute diseases and pest detection from images," *Plants*, vol. 11, no. 7, 2022, doi: 10.3390/plants11070937.
[12] Z. Xue, R. Xu, D. Bai, and H. Lin, "YOLO-Tea: A tea disease detection model improved by YOLOv5," *Forests*, vol. 14, no. 2, 2023, doi: 10.3390/f14020415.
[13] M. Khalid, M. Sarfraz, U. Iqbal, M. Aftab, G. Niedbała, and H. Rauf, "Realtime plant health detection using deep convolutional neural network," *Agriculture*, vol. 13, no. 2, p. 510, Feb. 2023, doi: 10.3390/agriculture13020510.
[14] M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, and F. Marinello, "Automatic bunch detection in white grapes varieties using YOLOv3, YOLOv4, and YOLOv5 deep learnings algorithms," *Agronomy*, vol. 12, no. 2, 2022, doi: 10.3390/agronomy12020319.
[15] M. Buzzy, V. Thesma, M. Davoodi, and J. M. Velni, "Real-time plant leaf counting using deep object detection networks," *Sensors (Switzerland)*, vol. 20, no. 23, pp. 1–14, 2020, doi: 10.3390/s20236896.
[16] F. Khan, N. Zafar, M. N. Tahir, M. Aqib, S. Saleem, and Z. Haroon, "Deep learning based approach for weed detection in potato crops," in *The 1st International Precision Agriculture Pakistan Conference 2022 (PAPC 2022)—Change the Culture of Agriculture*, Nov. 2022, p. 6, doi: 10.3390/environsciproc2022023006.
[17] A. Nasirahmadi, U. Wilczek, and O. Hensel, "Sugar beet damage detection during harvesting using different convolutional neural network models," *Agriculture*, vol. 11, no. 11, p. 1111, Nov. 2021, doi: 10.3390/agriculture11111111.
[18] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A realtime detection algorithm for kiwifruit defects based on YOLOv5," *Electronics*, vol. 10, no. 14, p. 1711, Jul. 2021, doi: 10.3390/electronics10141711.
[19] G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, "A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention," *Agronomy*, vol. 13, no. 7, 2023, doi: 10.3390/agronomy13071824.
[20] Y. Wang, C. Yi, T. Huang, and J. Liu, "Research on intelligent recognition for plant pest and disease based on improved YOLOv8 model," *Applied Sciences (Switzerland)*, vol. 14, no. 12, 2024, doi: 10.3390/app14125353.
[21] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: from YOLOv1 and beyond," *Computer Vision and Pattern Recognition*, pp. 1–34, 2023, [Online]. Available: http://arxiv.org/abs/2304.00501.
[22] A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, "Realtime multiclass helmet violation detection using few shot data sampling technique and YOLOv8," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2023-June, pp. 5350–5358, 2023, doi: 10.1109/CVPRW59228.2023.00564.
[23] Y. Egi, M. Hajyzadeh, and E. Eyceyurt, "Drone computer communication based tomato generative organ counting model using YOLO v5 and deep sort," *Agriculture*, vol. 12, no. 9, p. 1290, Aug. 2022, doi: 10.3390/agriculture12091290.
[24] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, Nov. 2021, doi: 10.3390/rs13224712.
[25] T. M. Alam *et al.*, "An efficient deep learning based skin cancer classifier for an imbalanced dataset," *Diagnostics*, vol. 12, no. 9, p. 2115, Aug. 2022, doi: 10.3390/diagnostics12092115.
[26] M. A. As'ari and N. I. A. Manap, "Covid-19 detection from chest x-ray images: comparison of well-established convolutional neural networks models," *International Journal of Advances in Intelligent Informatics*, vol. 8, no. 2, p. 224, Jul. 2022, doi: 10.26555/ijain.v8i2.807.
[27] T. Jiang, T. Liu, W. Dong, Y. Liu, and Q. Zhang, "Security risk level prediction of carbofuran pesticide residues in chinese vegetables based on deep learning," *Foods*, vol. 11, no. 7, p. 1061, Apr. 2022, doi: 10.3390/foods11071061.
[28] J.-W. Chen, W.-J. Lin, H.-J. Cheng, C.-L. Hung, C.-Y. Lin, and S.-P. Chen, "A smartphone based application for scale pest detection using multiple object detection methods," *Electronics*, vol. 10, no. 4, p. 372, Feb. 2021, doi: 10.3390/electronics10040372.
[29] V. Ponnusamy, A. Coumaran, A. S. Shunmugam, K. Rajaram, and S. Senthilvelavan, "Smart glass: Realtime leaf desease detection using YOLO transfer learning," in *Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing, ICCSP 2020*, 2020, pp. 1150–1154, doi: 10.1109/ICCSP48568.2020.9182146.
[30] H. Chen, G. Zhou, and H. Jiang, "Student behavior detection in the classroom based on improved YOLOv8," *Sensors*, vol. 23, no. 20, p. 8385, Oct. 2023, doi: 10.3390/s23208385.
[31] G. Oh and S. Lim, "One stage break light status detection based on YOLOv8," *Sensors*, vol. 23, no. 17, p. 7436, Aug. 2023, doi: 10.3390/s23177436.
[32] M. Hussain, "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, pp. 677–689, 2023.
[33] C. Jiang *et al.*, "Object detection from UAV thermal infrared images and videos using YOLO models," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, 2022, doi: 10.1016/j.jag.2022.102912.
[34] X. Wang, H. Gao, Z. Jia, and Z. Li, "BL-YOLOv8: an improved road defect detection model based on YOLOv8," *Sensors (Basel, Switzerland)*, vol. 23, no. 20, 2023, doi: 10.3390/s23208361.
[35] D. G. Stuparu, R. I. Ciobanu, and C. Dobre, "Vehicle detection in overhead satellite images using a one-stage object detection model," *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–18, 2020, doi: 10.3390/s20226485.
[36] W. M. Elmessery *et al.*, "YOLO based model for automatic detection of broiler pathological phenomena through visual and thermal images in internsive poultry houses," *Agriculture*, vol. 13, no. 8, p. 1527, Jul. 2023, doi: 10.3390/agriculture13081527.
[37] F. Zhou, H. Deng, Q. Xu, and X. Lan, "CNTR-YOLO: improved YOLOv5 based on convnext and transformer for aircraft detection in remote sensing images," *Electronics*, vol. 12, no. 12, p. 2671, Jun. 2023, doi: 10.3390/electronics12122671.
[38] D. Kolosov, V. Kelefouras, P. Kourtessis, and I. Mporas, "Anatomy of deep learning image classification and object detection on commercial edge devices: a case study on face mask detection," *IEEE Access*, vol. 10, pp. 109167–109186, 2022, doi: 10.1109/ACCESS.2022.3214214.

BIOGRAPHIES OF AUTHORS

Radityo Hendratmojo Jati Notonegoro holds a bachelor's degree in computer science (S.Kom.) with a focus on information systems and is currently pursuing a master's degree in information systems management (MMSI), specializing in software information systems. In addition to his academic background, he has obtained various certifications and professional skills. He is currently employed as a staff member at Gunadarma University. His research interests include artificial intelligence, image processing, and data science. He can be reached via email: radithjn@staff.gunadarma.ac.id.

Hustinawaty earned her Doctorate in information technology from Gunadarma University, Indonesia, in 2012, after completing her MMSI degree at the same university in 1994. Currently, she holds the position of Head of the Master of Information Systems Management Program at Gunadarma University. Her research focuses on information systems management and technology. She can be reached via email at hustina@staff.gunadarma.ac.id.