Vol. 14, No. 4 (2025): 1128 - 1139 https://doi.org/10.23960/jtepl.v14i4.1128-1139

JURNAL TEKNIK PERTANIAN LAMPUNG

ISSN 2302-559X (print) / 2549-0818 (online)
Journal homepage : https://jurnal.fp.unila.ac.id/index.php/JTP

Deep Learning-Based Detection for Early Germination Stages of Chili Pepper (*Capsicum annuum* L) Seedling in Greenhouse

Jasmine Tasmara[1], Supriyanto[1,✉], Mohamad Solahudin[1]

[1] IPB University, Bogor, INDONESIA.

Article History:

Received : 20 January 2025
Revised : 08 April 2025
Accepted : 20 April 2025

Keywords:

Chili seedling,
Deep learning,
Detection system,
Germination.

Corresponding Author:
✉ debasupriyanto@apps.ipb.ac.id
(Supriyanto)

ABSTRACT

Nursery plays an important role on starting chili cultivation, determining the crop health, fertility from disease, and growth performance. Early-stage germination detection is necessary to minimize nursery failure and improve plant health, but manual detection is challenging for large scale nursery in the greenhouse. The aim of this research was to develop an automatic detection model integrated with a You Only Look Once (YOLO) based deep learning algorithm using RGB camera to monitor the chili germination stages. Method to detect germination was YOLO with several steps, included: (1) early stages chili germination images acquisition, (2) datasets preparations, (3) dataset annotation and labeling, (4) model development using deep learning YOLO algorithms, and (5) model testing and validation. The training of 11,423 images was conducted utilizing the YOLOv5 and YOLOv8 algorithms, which categorized into, three classes (germinated, not germinated, and cotyledon appearance). The model was evaluated using mean Average Precision (mAP), precision, accuracy, and recall with the respective values of 0.697, 73%, 75%, and 73% for YOLOv8, and 0.664, 70%, 73%, and 70% for YOLOv5. Both model achieved high accuracy, but YOLOv8 was better to detect and classify chili seedling growth stages than YOLOv5. This study also demonstrated that model can be implemented in real applications integrated with automatic monitoring system included in the model.

1. INTRODUCTION

Chili peppers (*Capsicum annuum* L.) are an important product in Indonesia. In 2023, the consumption of chili peppers is 2,42 kg per capita per year (increase 7.11% from previous), and productions 1.55 million tons (Portal Satu Data Pertanian, 2023). Chili cultivations are starting with the nursery process to produce high quality seed (Muslimin *et al*., 2021; do Rêgo *et al*., 2016). The seedling stages consist of media preparations, and transplanting sprouts into seed trays. There are several stages on nursery are emergence (3 to 7 days after seedling), cotyledon expansion (7 to 14 days after seedling), first true leaves (14-21 days after seedling), and preparation to planting in the filed or greenhouse (21 to 30 days after seedling) (Lutfiana *et al*., 2019; Wahyudi & Topan, 2011). Optimal management and monitoring in these stages are crucial process to achieve the productivity and plant health (Jean & Sihombing, 2024). The present monitoring of chili nurseries still relies on direct observation by eye. This method is labour-intensive and time-consuming, and frequently results in a incomplete survey of the entire nursery area. One potential solution to this issue is the integration of deep learning technology in the monitoring process.

Currently, nursery process are conduct inside the controlled environment such as greenhouse or plant factory with artificial light to achieve the high quality product (Anam *et al*., 2020; Niam *et al*., 2019). Modern nursery greenhouses are using IoT based environment control of several parameters such as light intensity, temperature, and nutrient. This integration enables automated responses, such as adjusting irrigation or lighting, based on real-time environmental data and plant conditions (Feng & Hu, 2021; Nugrahapsari *et al*., 2020). The seedling process is using the tray with

automatic seeder to maximize the speed and minimize the human errors (Sun *et al*., 2023). Then, seedling trays move the seedling rack continue to the grow stages. In a commercial greenhouse with an area of 760 m² (40 x 19 m), it is possible to accommodate 20 rack of seedling, 920 trays, and 172,040 seed per batch (30 days). Assuming 10 batches of seedling per year, the greenhouse capacity is approximately 1.720.400 seeds, sufficient for 10 ha cultivation area.

Monitoring seedling inside the greenhouse should use the sensing technology to monitor seedling stages, health, and growth of chili seedling (Singh *et al*., 2021). Recently, research on monitoring the growth of seedlings has received special attention from researchers. Cui *et al.* (2023) developed a real-time method for detecting and counting missing rice seedlings using an improved YOLOv5s and ByteTrack, achieving 93.2% accuracy and five times faster counting than manual methods, making it practical and efficient for real-time application in paddy fields. YOLO receives input images divided into $S \; x \; S$ boxes sent by the Neural Network for Bounding box generation and the system predicts using the dimensional division as the anchor box (Redmon *et al*., 2016). In another study, an improved mask R-CNN model was developed for the detection and segmentation of lettuce seedlings from seedling trays. The method enabled early-stage monitoring of seedlings and efficient size estimation for better management of environmental stresses (Islam *et al*., 2024). Perugachi-Diaz, Tomczak & Bhulai, (2021) conducted a study to classify the seedbed stages in white cabbage seedlings using Convolutional Neural Networks (CNN). Another study used deep learning with the CNN algorithm to classify 12 different types of seedlings in the seedbed process, this study used a dataset of 4,234 images and produced an accuracy of more than 90% (Alimboyong *et al*., 2019). The datasets division is based on the size of the model, the largest size has higher accuracy, and the detection time for a single image will increase (Li *et al*., 2022).

However, the limitation of deep learning was difficult to implement in other locations without updated the datasets with the specific condition. These studies highlight the critical importance of early detection and the integration of advanced technologies in agricultural practices. In this research we proposed a deep learning algorithm to detect early stages of nursery process inside the commercial greenhouse in Indonesia. The aim of this research was to develop a deep learning algorithm using YOLOv5 and YOLOv8, along with camera technology, to detect the early germination stages of chili pepper seedlings in a nursery greenhouse. By leveraging these advanced algorithms, the study seeks to enhance the accuracy and efficiency of monitoring seedling development, focusing on critical stages such as germination, not germination, and the appearance of cotyledon leaves. This approach will facilitate timely interventions during the nursery process, ultimately contributing to improved seedling health and productivity. Through this research, the goal is to advance sustainable agricultural practices in Indonesia by integrating innovative technology into the management of chili pepper cultivation.

2. MATERIAL AND METHODS

2.1. Materials and Tools

The materials used in this study included seeds of chili pepper (*Capsicum annuum* L.), sourced from a reputable local seed supplier. The seedlings were cultivated in rockwool, a growing medium known for its excellent water retention and aeration properties, which supports optimal seedling growth. The seedlings were placed in plastic seed trays. Clean water, free from contaminants, was used for irrigation. The camera utilized a red-green-blue (RGB) configuration to ensure detailed imagery. The images undergo a preprocessing step before annotation, which involves selecting high-quality images and removing unsuitable images to improve the datasets. The selected images were then annotated and labeled using RoboFlow, a software tool designed for dataset preparation. After annotation, dataset augmentation was performed to ensure the robustness of the training data. Augmentation techniques included auto-orientation, applying blur up to 4.5px, horizontal and vertical flipping, static cropping, and the addition of noise. The training of the deep learning models was conducted using Google Colaboratory, which provided the necessary computational resources. Additionally, initial data processing and development were performed using a Lenovo Ideapad 330s laptop equipped with an Intel Core i7 processor, 4GB RAM, and 1TB HDD storage. The deep learning algorithms implemented in this study were YOLOv5 and YOLOv8, developed in a Python environment using TensorFlow and PyTorch libraries. The training was conducted using specific hyperparameters to optimize model performance, including an input image size of 640 pixels, a batch size of 64, and 200 training epochs.

2.2. Methods

This research was carried out at the Laboratory of Systems Engineering and Agricultural Informatics (Bioinformatics Engineering), IPB University, Bogor Regency, West Java. This research was conducted from January to August 2024. This study was conducted in several stages such as dataset collections, dataset labeling, training model, and testing (Figure 1).

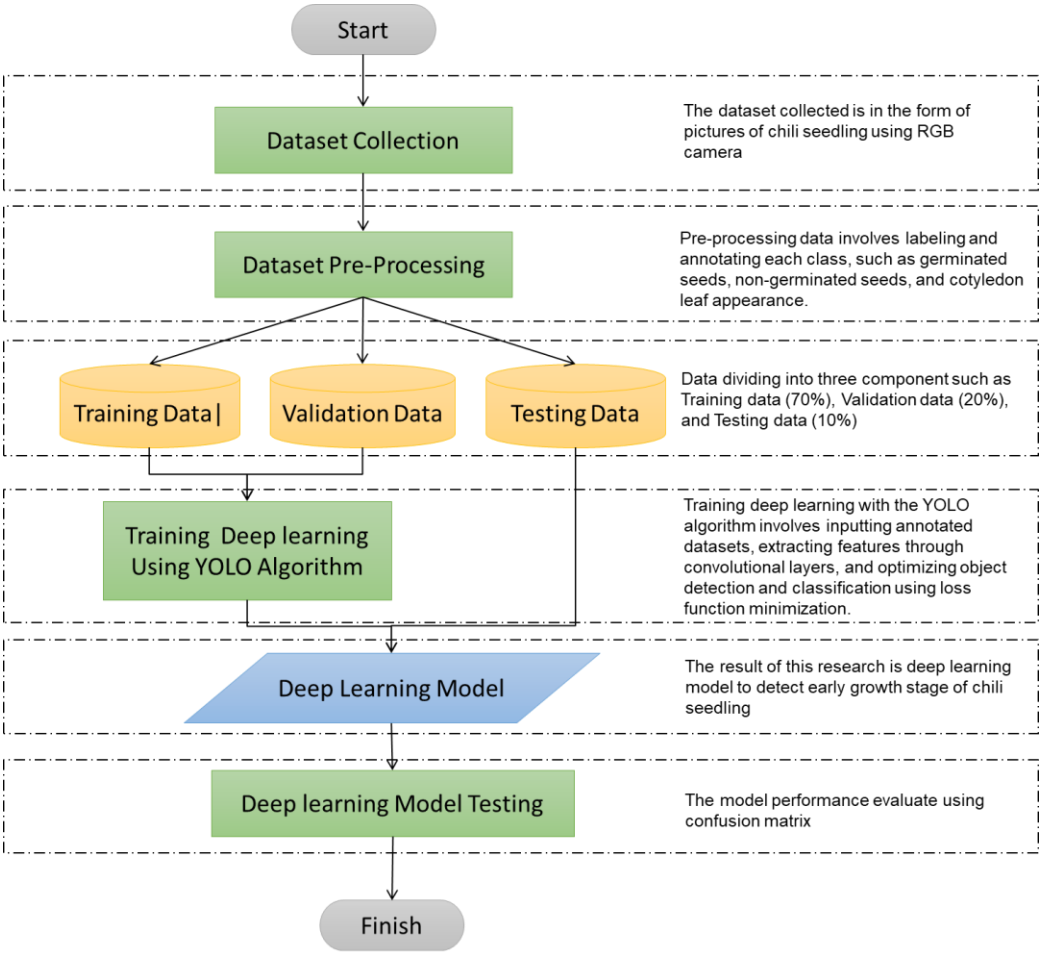

Figure 1. Stage of the research that consist of data collections, data labelling, training model, and testing

2.2.1. Dataset Collections

Dataset was collected in early stages of nursery, starting from ungerminated seeds to seedlings ready for transplanting. The dataset was collected using the red-green-blue camera to capture high resolution images. Two types of lighting conditions were applied during image capture: natural light, using sunlight during the day, and artificial light, using a controlled light source in indoor settings. This process was using the automatic capture during the morning time (08.00 am), afternoon (13.00 pm), and evening (15.00 pm). The use of both natural and artificial lighting introduced variability in visual conditions, which affected image consistency and, consequently, model performance. Natural light resulted in fluctuations in brightness, color saturation, and shadow intensity depending on the time of day, while artificial light provided more consistent illumination but lacked the subtle gradients of natural lighting. These variations occasionally reduced detection accuracy due to reflections, overexposure, or reduced contrast.

Nonetheless, the inclusion of both lighting types was intentional to enrich the dataset and improve the model's robustness in diverse, real-world environments. To mitigate the impact of lighting inconsistencies, preprocessing was

conducted to select only high-quality images. Furthermore, data augmentation techniques such as, auto-orientation, blur (up to 4.5 px), noise addition, flipping, and cropping were applied to improve model generalization. The camera was capture at 30 and 40 cm above the seedling trays to capture optimum view without images distortions or damages. Figure 2 shows the configuration of camera was used to capture the images.

The dataset, collected 20 days after planting, contains over 13,000 images of individual chili seeds categorized into germinated seeds, non-germinated seeds, and cotyledon appearance. All images were resized to 640 × 640 pixels for model compatibility and split into training (70%), validation (20%), and testing (10%) sets. Figure 3 presents sample images of chili seedlings at early stages.

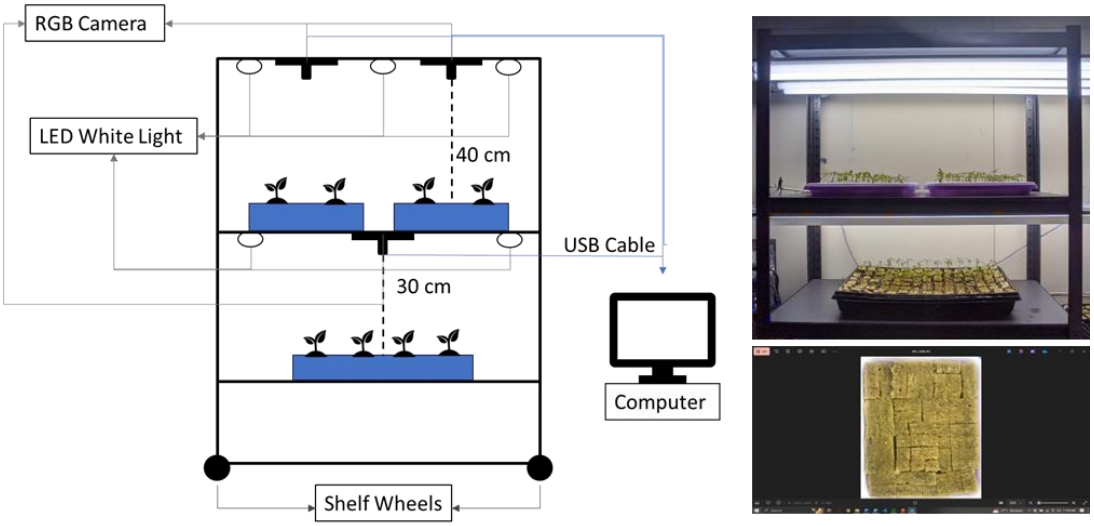

Figure 2. Schematic diagram of camera capture

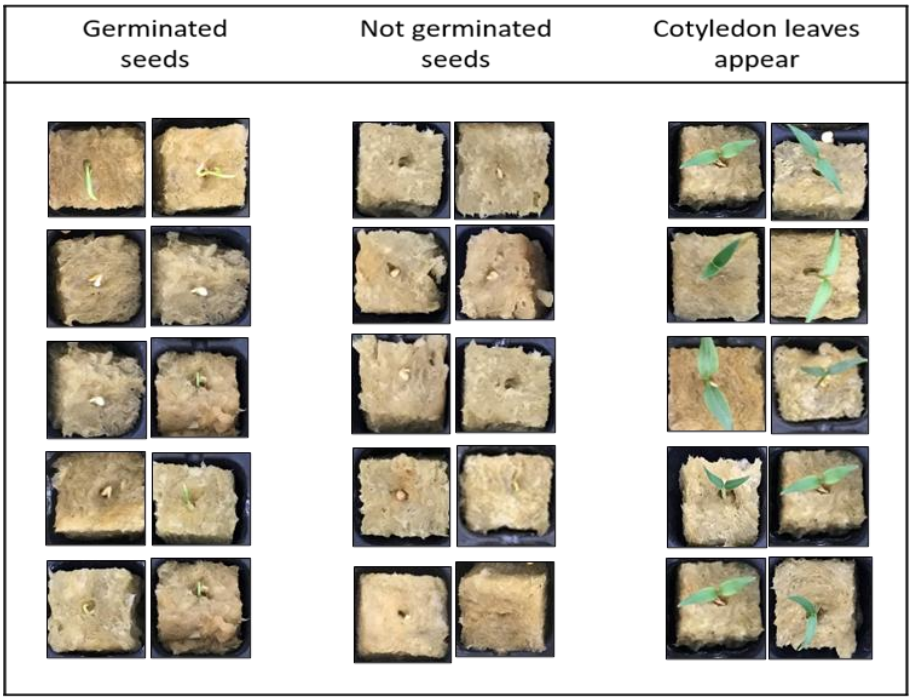

Figure 3. Sample of dataset of chili seedling in early stages

2.2.2. Data Pre-Processing

In this step, data was pre-processed using starting by selecting a good image and remove unwanted images for training process. Then, the dataset was labelled with three classes such as germinated seeds, non-germinated seeds, and the appearance of cotyledon leaves using RoboFlow software (Figure 4). Then the output of these stages was the annotated images included the XML based data to store the annotation information. This format incorporates essential parameters, including the class ID, which serves to uniquely identify each object category, the central coordinates of the bounding box (x and y), which represent the middle position of the object, and the width and height of the bounding box, which define the size of the rectangular boundary.

These parameters enable precise classification and detection of objects in the images. The bounding box itself acts as a spatial reference, ensuring the object of interest is accurately located and segmented for further analysis. Finally, to ensure the quality of training data was pre-processed with several options such as auto-orientation, blur up to 4.5px, flip horizontal and vertical, static crop, and noise. Figure 5 shows several examples of data pre-processing techniques applied to the images in the datasets.

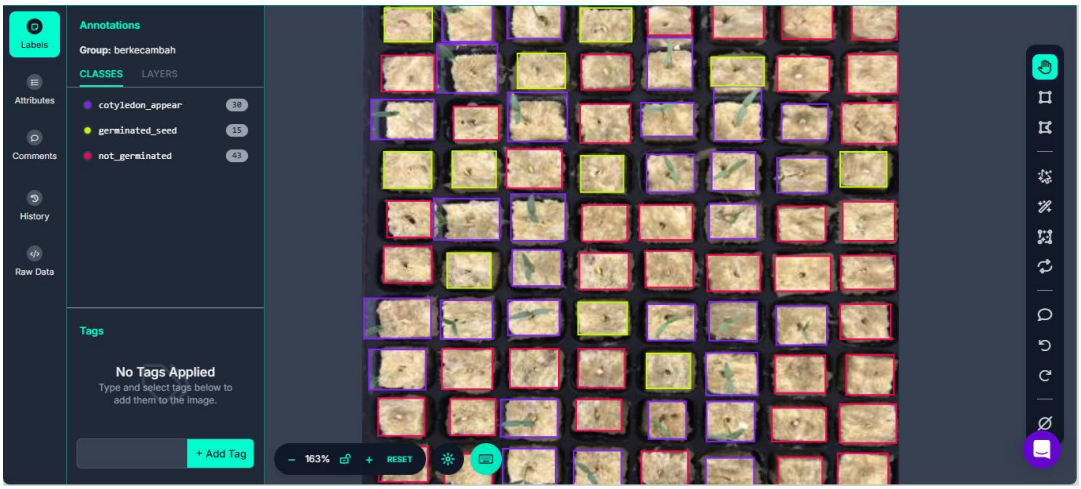

Figure 4. Datasets labelling using RoboFlow software with three classes

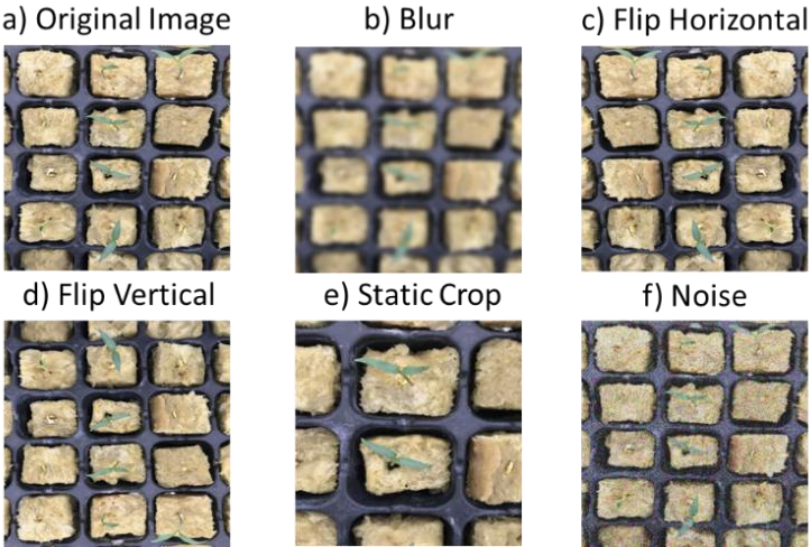

Figure 5. Sample of data pre-processing. a) original image, b) blur, c) flip horizontal, d) flip vertical, e) static crop, f) noise

2.2.3. Training Deep Learning Model

In this step, annotated datasets were trained, validated, and tested using separate datasets with the composition 70% allocated for training, 20% for validation, and 10% for testing. The proportions of training, validation, and testing dataset ensure the quality of model. The training process uses *.xml files generated during the object labeling phase, which contain structured annotation data specifying the locations and classes of objects in the images. The training process used two deep learning algorithms (YOLOv5 and YOLOv8), 15,000 datasets and 200 epoch iterations to process the dataset. The YOLO (You Only Look Once) model architecture is composed of convolutional layers responsible for extracting features, identifying patterns and detecting objects in input images, followed by fully connected layers to convert the two-dimensional feature maps into a one-dimensional matrix for efficient classification and bounding box prediction (Vilar-Andreu *et al*., 2024). YOLOv5 uses a CSPNet-based backbone for feature extraction, while YOLOv8 introduces several architectural improvements such as an anchor-free detection head, a decoupled head for classification and localisation, and an improved backbone and neck structure, including the use of C2f modules for better gradient flow and efficiency (Bochkovskiy *et al*., 2020; Yaseen, 2024). Figure 6 shows the number of instances of each class, bounding boxes dimension, the *y* center against the *x* center of each label, and the height of each label against its width. In all subfigures, the axes are normalized to the image dimensions, with values ranging from 0 to 1. In subfigure (a), the y-axis indicates the total count of objects per class. Subfigures (b) and (c) show the relative positions of object centers across the images, while subfigure (d) displays the distribution of bounding box sizes, with both height and width also expressed in normalized units. This normalization ensures consistency and robustness of the model across varying image sizes.

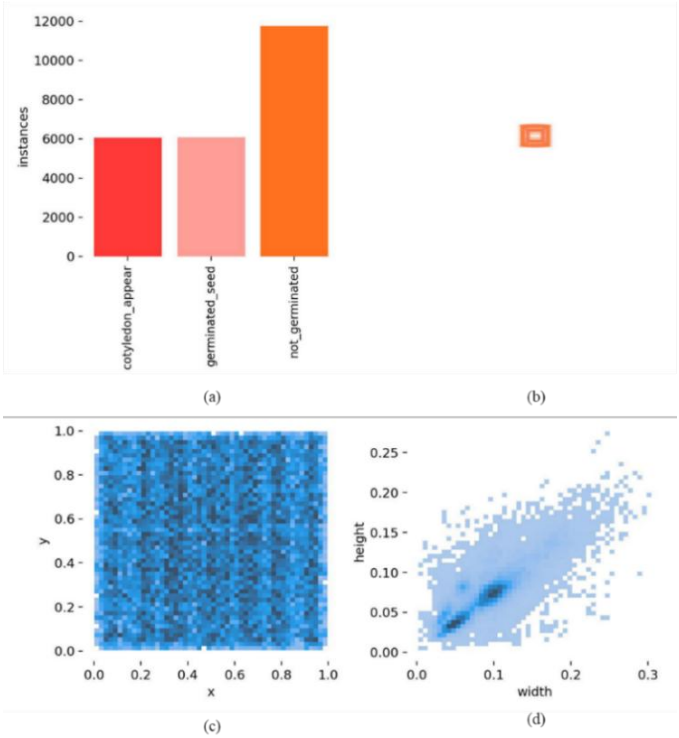

Figure 6. Bounding box analysis: (a) The number of instances of each class, (b) Every bounding box, (c) The y center against the x center of each label, and (d) The height of each label against its width.

2.2.4. Performance Evaluation

The evaluation metrics used in this research consisted of accuracy, recall, precision, and mean average precision (mAP). These metrics are used to assess both the detection accuracy and real-time efficiency of the proposed method. In our study, we selected these metrics to evaluate our trial results. Precision was the proportion of true sample that

was correctly determined to the total of actual instance. Then, recall was the percentages of amount rightly true divided to all true datasets. The mAP is a metric of choice when it comes to evaluating the performance of object detection models. It measures the area under the precision-recall curve, providing a quantitative assessment of the model's accuracy in identifying objects. The mAP calculation typically considers a confidence threshold, which determines how confident the model must be in its predictions to be counted as correct. This metric can be calculated at specific thresholds or across a range of values. For instance, $M_{0.5}$ uses a confidence threshold of $\tau = 0.5$, meaning predictions with at least 50% confidence are evaluated. On the other hand, $M_{0.95}$ provides a more comprehensive evaluation by averaging the precision across a range of thresholds from $\tau = 0.5$ to $\tau = 0.95$, incremented by 0.05.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$mAP = \frac{1}{m}\sum_{i=1}^{m} AP_i \quad (3)$$

where TP is true positive, when the model predicts a positive outcome and the actual result is positive; FP is false positive, when the model predicted a positive outcome, but the actual value is negative; and FN is false negative, when a prediction by the model of a negative outcome when the actual result is positive.

This ensures that the evaluation considers a variety of confidence levels, offering a better understanding of the model's performance across different detection scenarios. The mAP metric is particularly useful because it captures the balance between precision and recall. A higher mAP indicates that the model is both precise and comprehensive in its detections (Terven *et al*., 2023). By evaluating the performance at different confidence thresholds, $M_{0.5}$ and $M_{0.95}$ provide insights into how well the model handles varying levels of prediction certainty, making it a standard benchmark in the field of object detection.

3. RESULT AND DISCUSSION

Figure 7 shows loss values during training bounding box, training classification, and training distribution focal for YOLOv5 and YOLOv8 algorithms. The models were trained using deep learning YOLOv5 and YOLOv8 algortihms on the same dataset for 200 epochs. The training process was monitored through various loss metrics, including box loss,

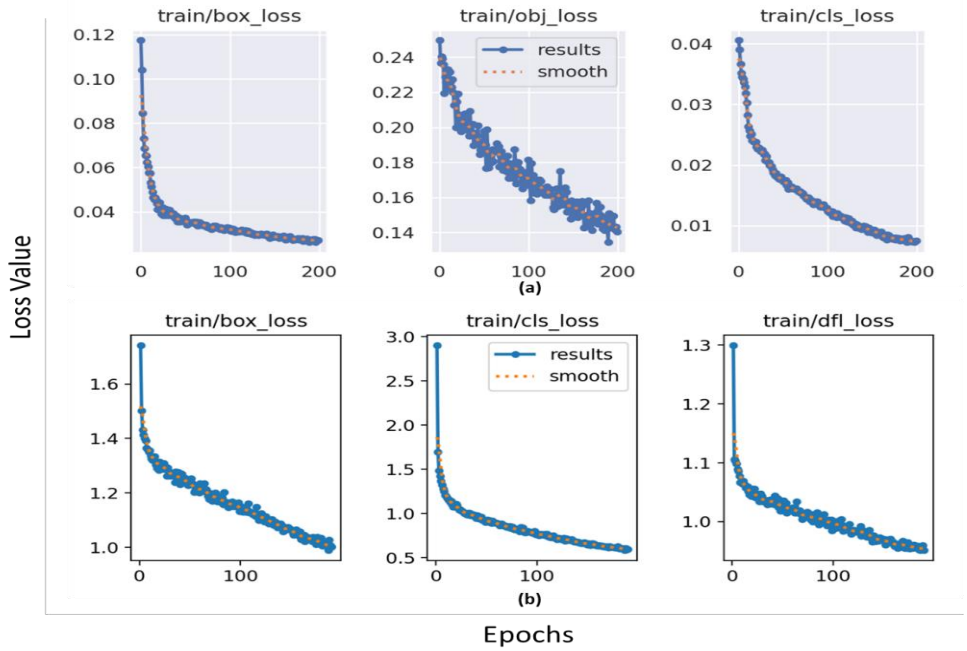

Figure 7. Loss values at training (bounding box, classification, and distribution focal): (a) YOLOv5, and (b) YOLOv8 algorithm

object loss, and classification loss. Additionally, validation metrics such as precision, recall, and mAP were recorded to evaluate model performance. YOLOv5 exhibited lower training loss values across all metrics, with box loss decreasing from 0.12 to 0.03, object loss from 0.24 to 0.14, and classification loss from 0.04 to 0.01. This indicates effective learning and convergence. YOLOv8, on the other hand, started with higher training loss values, with box loss ranging from 1.6 to 1.0, classification loss from 3.0 to 0.5, and the introduction of distribution-focused loss (dfl_loss), which showed variability (Figure 7).

In the validation phase, YOLOv5 maintained lower loss values, with box loss stabilizing around 0.04, object loss around 0.25, and classification loss around 0.02. YOLOv8 showed higher initial validation losses, with box loss fluctuating between 1.55 and 1.40, and classification loss decreasing from 2.5 to 1.0, indicating potential instability during training (Figure 8).

The evaluation of precision and recall metrics revealed that YOLOv5 achieved a precision of approximately 0.7 and a recall of 0.8. In comparison, YOLOv8 demonstrated a similar recall rate but exhibited higher precision, stabilizing around 0.7. This indicates that while both models are effective in detecting objects, YOLOv8 may have a slight advantage in precision (Figure 9).

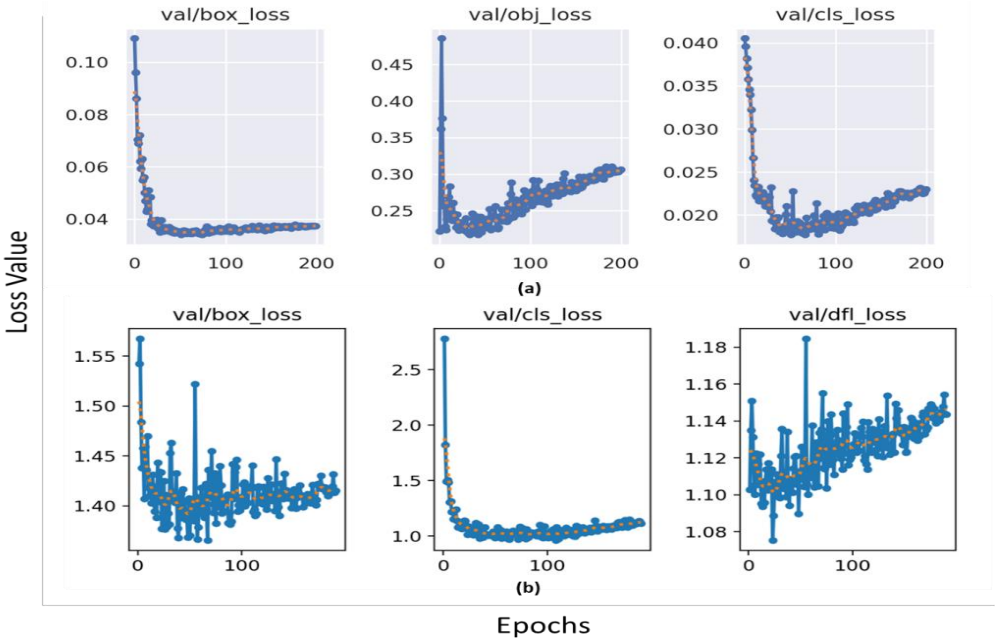

Figure 8. Validation of bounding box loss, classification loss, and distribution focal loss for (a) YOLOv5, (b) YOLOv8 algorithm

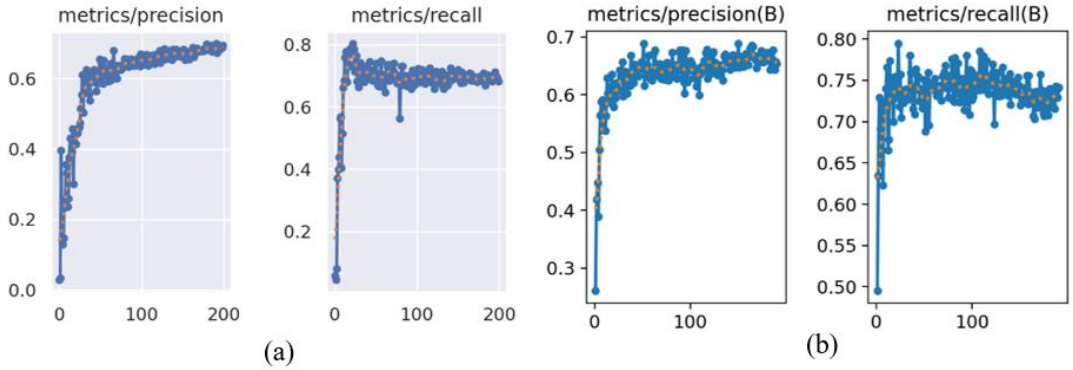

Figure 9. Precision and recall for (a) YOLOv5 algorithm, (b) YOLOv8 algorithm

In terms of mean Average Precision (mAP), YOLOv5 recorded an mAP of 0.6 and an mAP0.5–0.95 of 0.4. Conversely, YOLOv8 achieved an mAP of 0.7 and an mAP0.5–0.95 of 0.4. These results suggest that YOLOv8 performs better at the 0.5 Intersection over Union (IoU) threshold, indicating its greater effectiveness in accurately identifying objects compared to YOLOv5.

As shown in Table 1, the detection performance of YOLOv5 and YOLOv8 was compared based on their mean Average Precision (mAP) at two different Intersection over Union (IoU) thresholds: 0.5 and 0.5–0.95. The results revealed that YOLOv8 outperforms YOLOv5 in terms of mAP@0.5, with a mean value of 0.697 compared to YOLOv5's 0.664. This difference was found to be statistically significant, as confirmed by a paired t-test ($t = 98.65$, $p < 0.001$). Although YOLOv5 and YOLOv8 showed a smaller difference in their mAP0.5–0.95 values, with YOLOv8 achieving a mean of 0.397 compared to YOLOv5's 0.382, this difference was also statistically significant ($p = 0.0021$).

Table 1 Comparison of detection performance between YOLOv5 and YOLOv8

Metric	YOLOv5 (Mean ± SD)	YOLOv8 (Mean ± SD)	*p*-value	Significance
mAP	0.664 ± 0.015	0.697 ± 0.016	< 0.001	Significant
mAP0.5:0.95	0.382 ± 0.010	0.397 ± 0.012	0.0021	Significant

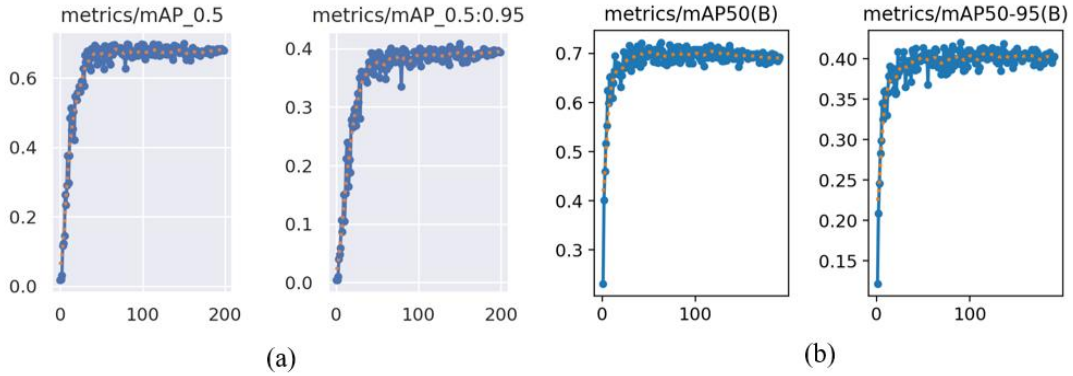

Figure 10. mAP0.5, and mAP0.95 for : (a) YOLOv5 algorithm, (b) YOLOv8 algorithm

Figure 10 provides a visual comparison of the mAP values at both IoU thresholds for each model, highlighting the relative performance between YOLOv5 and YOLOv8. The confusion matrix for the YOLOv5 and YOLOv8 algorithm provides a detailed overview of its classification performance across four categories such as cotyledon_appear, germinated_seed, not_germinated, and background (Figure 11). The YOLOv5 model demonstrates strong accuracy in identifying the cotyledon appear class, achieving a True Positive Rate (TPR) of 83%, which indicates high reliability in detecting this early growth stage. However, for the germinated seed class, the performance is moderate, with a TPR of 60%. This class also experiences a notable False Negative Rate (FNR) of 14%, where actual germinated seeds are misclassified as other categories, and a False Positive Rate (FPR) of 30%, primarily due to not germinated instances being incorrectly predicted as germinated seeds. Similarly, the not germinated class reaches a TPR of 68%, but 27% of its instances are misclassified as germinated seed, contributing to the observed confusion between early germination stages. The background class shows the lowest detection performance, with a TPR only 17%, suggesting a tendency of the model to misclassify background regions as plant-related classes, particularly cotyledon appear and not germinated.

In comparison, the YOLOv8 model yields a higher TPR of 89% for the cotyledon appear class, indicating even greater detection accuracy than YOLOv5. The germinated seed class under YOLOv8 achieves a TPR of 56%, slightly lower than YOLOv5, but it benefits from a lower FNR of 13%, demonstrating improved sensitivity. The not germinated class performs better under YOLOv8, with a TPR of 72%. However, it still suffers from a considerable FPR of 32%, reflecting persistent confusion with the germinated seed class. The background class again shows limited performance, with a TPR of only 14%, similar to YOLOv5, indicating ongoing difficulty in distinguishing background from seed objects.

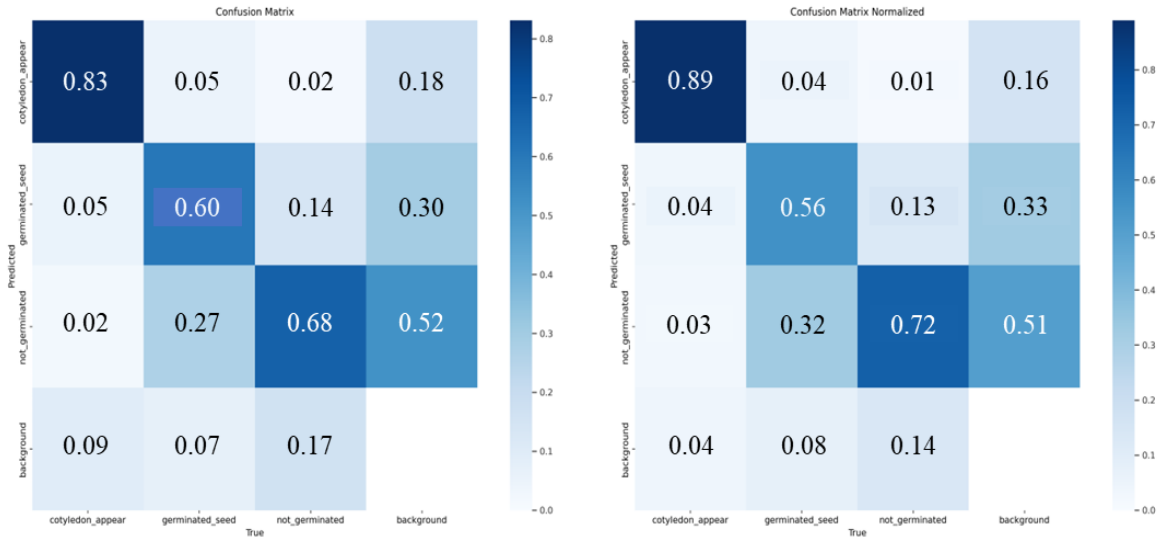

Figure 11 Confusion Matrix for (a) YOLOv5 algorithm, (b) YOLOv8 algorithm

Overall, a comparison of the confusion matrices showed that although both models performed well in identifying the emerged cotyledon class, YOLOv5 showed better accuracy in classifying the non-germinated class, while YOLOv8 achieved higher precision in detecting the emerged cotyledon class. Although YOLOv8 experienced a slight improvement in detection accuracy, the speed of inference remains an important factor for real-time applications such as greenhouse monitoring. Based on the evaluation log, YOLOv5 achieved an inference speed of about 49 Frames Per Second (FPS), processing 56 images in 1.15 seconds. In contrast, YOLOv8 processes the same number of images in 4.00 seconds, resulting in a lower effective speed of 14 FPS under the current setup. Thus, while the YOLOv8 offers better detection for certain classes, the YOLOv5 may be preferable for real-time systems that require higher frame rates and lower latency.

The model developed in this study was specifically designed for chili (*Capsicum annuum* L.), and its application to other crop species or environmental conditions requires further validation. Different species have different growth characteristics, which may affect performance, and transfer learning can be used to adapt the model to other crops. In addition, the model was trained under controlled greenhouse conditions, and its robustness to environmental variability (e.g., light, temperature, humidity) should be tested through the addition of real-world data and evaluation. Performance may also vary with different imaging systems, so retraining or refinement with varying camera data is recommended. However, based on our knowledge this models has acceptable compared to other deep learning models on agriculture application.

4. CONCLUSION

A deep learning model for detecting the early germination stages of chili pepper has been successfully developed using the YOLOv5 and YOLOv8 algorithms. The results indicate that the model is stable and accurate in detecting objects, as evidenced by the higher number of correct detections compared to incorrect ones. The detection errors are likely due to unclear images, which hinder the model's ability to distinguish between objects effectively. Future research should focus on expanding the dataset and utilizing other cameras, such as multispectral or UV cameras, which would provide richer image data and improve the detection of early germination stages. Additionally, the computational limitations encountered during training deep learning models have posed challenges in handling larger datasets. Optimizing computational resources would enable the inclusion of more data and enhance the model's performance and scalability. These models show significant potential for supporting monitoring systems and detecting the early germination stages of chili peppers in nursery greenhouses.

ACKNOWLEDGEMENT

Authors would like to express my sincere gratitude to the Department of Mechanical Engineering and Biosystems, and the Agribusiness and Technology Park, Directorate of Community Development, IPB University, Indonesia, for their valuable support and facilitation of this research.

REFERENCES

Alimboyong, C.R., Hernandez, A.A., & Medina, R.P. (2019). Classification of plant seedling images using deep learning. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 1839–1844. https://doi.org/10.1109/TENCON.2018.8650178

Anam, B.K., Sasongko, L.A., & Subantoro, R. (2020). Komparasi kelayakan usaha pembibitan cabai merah keriting (*Capisum annum* L.) dengan cabai rawit (*Capisum frutescens* L.) di Desa Banyukuning Kecamatan Bandungan Kabupaten Semarang. *Jurnal Ilmu-Ilmu Pertanian*, ***16***(1), 1–11.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y.M. (2020). YOLOv4: Optimal speed and accuracy of object detection. http://dx.doi.org/10.48550/arXiv.2004.10934

Cui, J., Zheng, H., Zeng, Z., Yang, Y., Ma, R., Tian, Y., Tan, J., Feng, X., & Qi, L. (2023). Real-time missing seedling counting in paddy fields based on lightweight network and tracking-by-detection algorithm. *Computers and Electronics in Agriculture*, ***212***(July), 108045. https://doi.org/10.1016/j.compag.2023.108045

Feng, J., & Hu, X. (2021). An IoT-based hierarchical control method for greenhouse seedling production. *Procedia Computer Science*, ***192***, 1954–1963. https://doi.org/10.1016/j.procs.2021.08.201

Islam, S., Reza, M.N., Chowdhury, M., Ahmed, S., Lee, K.-H., Ali, M., Cho, Y.J., Noh, D.H., & Chung, S.-O. (2024). Detection and segmentation of lettuce seedlings from seedling-growing tray imagery using an improved mask R-CNN method. *Smart Agricultural Technology*, ***8***(April), 100455. https://doi.org/10.1016/j.atech.2024.100455

Sihombing, D.J.C. (2024). User needs analysis for developing plant monitoring information system: enhancing agricultural efficiency and productivity. *Jurnal Info Sains: Informatika dan Sains*, ***14***(01), 846–854.

Li, Z., Wang, Y., Zhang, N., Zhang, Y., Zhao, Z., Xu, D., Ben, G., & Gao, Y. (2022). Deep learning-based object detection techniques for remote sensing images: A survey. *Remote Sensing*, ***14***(10), 1–41. https://doi.org/10.3390/rs14102385

Lutfiana, Z., Fangohoi, L., & Saikhu, M. (2019). Pengaruh intensitas penyiraman terhadap persemaian cabai rawit (*Capsicum frutescens* L.) dengan media semai pelepah batang pisang di Kelompok Tani Morgo Utomo Kelurahan Bence Kecamatan Garum, Kabupaten Blitar, Provinsi Jawa Timur. *Agrovigor: Jurnal Agroekoteknologi*, ***12***(2), 82–86. https://doi.org/10.21107/agrovigor.v12i2.5605

Muslimin., Taufik, M., Thamrin, M., & Suddin, A.F. (2021). Financial analysis of red chili farming business with green cultivation technology in South Sulawesi. *IOP Conference Series: Earth and Environmental Science*, ***911***, 12079. https://doi.org/10.1088/1755-1315/911/1/012079

Niam, A.G., Muharam, T.R., Widodo, S., Solahudin, M., & Sucahyo, L. (2019). CFD simulation approach in determining air conditioners position in the mini plant factory for shallot seed production. *AIP Conference Proceedings*, ***2062***(October). https://doi.org/10.1063/1.5086564

Nugrahapsari, R.A., Sayekti, A.L., Yufdy, M.P., & Arsanti, I.W. (2020). Faktor-faktor yang memengaruhi keputusan petani dalam mengadopsi teknologi persemaian bibit cabai di Provinsi Jawa Barat. *Jurnal Agro Ekonomi*, ***38***(2), 143–153.

Portal Satu Data Pertanian. (2023). *Outlook Komoditas Pertanian Subsektor Hortikultura Cabai*. Pusat Data dan Sistem Informasi Pertanian, Sekretariat Jenderal Kementerian Pertanian, Jakarta.

Perugachi-Diaz, Y., Tomczak, J.M., & Bhulai, S. (2021). Deep learning for white cabbage seedling prediction. *Computers and Electronics in Agriculture*, ***184***(December 2020), 106059. https://doi.org/10.1016/j.compag.2021.106059

do Rêgo, E.R., do Rêgo, M.M., & Finger, F.L. (2016). *Production and breeding of chilli peppers (Capsicum spp.)*. Springer. https://doi.org/10.1007/978-3-319-06532-8

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, 779–788. https://doi.org/10.1109/CVPR.2016.91

Singh, A., Jones, S., Ganapathysubramanian, B., Sarkar, S., Mueller, D., Sandhu, K., & Nagasubramanian, K. (2021). Challenges and opportunities in machine-augmented plant stress phenotyping. *Trends in Plant Science*, ***26***(1), 53–69. https://doi.org/10.1016/j.tplants.2020.07.010

Sun, E., Xiao, Z., & Tan, Y. (2023). A fast path planning method of seedling tray replanting based on improved particle swarm optimization. *Agronomy*, ***13***(3). https://doi.org/10.3390/agronomy13030853

Terven, J., Córdova-Esparza, D.M., & Romero-González, J.A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, ***5***(4), 1680–1716. https://doi.org/10.3390/make5040083

Vilar-Andreu, M., Garcia, L., Garcia-Sanchez, A.J., Asorey-Cacheda, R., & Garcia-Haro, J. (2024). Enhancing precision agriculture pest control: A generalized deep learning approach with YOLOv8-based insect detection. *IEEE Access*, ***12***(June), 84420–84434. https://doi.org/10.1109/ACCESS.2024.3413979

Wahyudi, W., & Topan, M. (2011). *Panen Cabai di Pekarangan Rumah*. Agromedia Pustaka, Jakarta.

Yaseen, M. (2024). What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. *Computer Vision and Pattern Recognition*. http://arxiv.org/abs/2408.15857