International Journal on Advanced Science Engineering Information Technology

Vol.15 (2025) No. 2
ISSN: 2088-5334

Comparative Evaluation of Machine Learning Models for Rice Seed Quality Identification on an Android Platform

Feny Rahmasari [a], Sidiq Syamsul Hidayat [a,*], Kurnianingsih [a]

[a] *Department of Electrical Engineering, Politeknik Negeri Semarang, Jl Prof Soedarto, Semarang, Indonesia*
Corresponding author: [*]sidiqsh@polines.ac.id

***Abstract*— The challenge of significantly increasing rice production in Indonesia necessitates a granular focus on the farm level, with particular emphasis on the utilization of high-quality seeds. Unfortunately, the accurate assessment of seed quality is often only feasible post-planting. While traditional methods persist, they are notably time-consuming and prone to inaccuracies. This research introduces a novel solution: an Android-based machine learning model for rice seed quality identification based on morphological seed structure. By enabling early detection of low-quality seeds, the system aims to mitigate the associated risks and contribute to increased rice yield. Two cutting-edge machine learning models were trained using datasets sourced from the Roboflow platform and subsequently integrated into an Android application. A comparative analysis of these models was conducted to determine their efficacy in discerning rice seed quality. The evaluation results unequivocally demonstrated the superior performance of the Roboflow Train 3.0 Object Detection (Fast) model, achieving an impressive mean average precision (mAP) of 97.1%, precision of 96.2%, and recall of 93.4%. Given its exceptional accuracy and speed, the Roboflow Train 3.0 Object Detection (Fast) model has been proven to be the ideal option for integration into Android applications. Future research could concentrate on optimizing the model for edge inference, thereby enhancing identification efficiency and accuracy. This advancement holds the potential to revolutionize rice cultivation practices in Indonesia by empowering farmers to make informed decisions based on precise seed quality assessment, ultimately leading to substantial improvements in rice production and food security.**

***Keywords*— Rice seed quality; machine learning; Roboflow train 3.0; object detection; YOLO NAS; Android application.**

Manuscript received 7 Aug. 2024; revised 19 Oct. 2024; accepted 24 Nov. 2024. Date of publication 30 Apr. 2025.

I. INTRODUCTION

Rice serves as the primary food source for most Indonesian population [1], [2], [3]. It is a crucial commodity with significant implications for social, economic, political, and national security aspects, while also meeting food needs and achieving sustainable self-sufficiency goals [4], [5]. The total rice production for food consumption in 2023 reached 31.10 million tons, marking a decrease of 439.24 thousand tons or 1.39 percent compared to the 31.54 million tons produced in 2022 [6], [7]. The enhancement of national rice production must be accompanied by an increase in the production of high-quality rice seeds, which play a vital role in boosting productivity [8], [9], [10], [11]. However, assessing the quality of rice seeds before planting is challenging due to limited human observation and potential seed damage [12]. Traditional human evaluation methods are slow, costly, inconsistent, and subjective [13]. Therefore, there is a need for more efficient and reliable methods for determining rice seed quality [14], [15].

Since the inception of the Making Indonesia 4.0 initiative, there has been a pressing need for advanced technologies within the agricultural sector to enhance productivity, quality, and sustainability [16]. The utilization of machine learning in the field of agriculture, with specific reference to rice production, has enabled considerable progress in the areas of seed selection, germination, and plant growth [17]. A variety of digital tools and online platforms have been developed to provide support for these endeavors. This has resulted in contributions to the field of artificial intelligence, with the creation of intelligent algorithms designed to assist farmers in assessing the quality of rice seeds [18], [19].

A recent study conducted by Sidiq [12] developed a superior/non superior rice seed classification. This system uses Deep-CNN, facilitating a rapid, accurate, and cost-efficient method for farmers to assess rice seeds, with 93% precision and 95% accuracy. However, this research has not been supported by a platform as a practical form of the system

created. Research by Zhao [20] developed an artificial neural network (YOLO-r) to detect the germination status of rice seeds and automatically evaluate the total number of germinations. However, this research did not seek to make comparison with other algorithms, nor did it establish a practical platform. A significant number of researchers have presented digital image processing techniques for non-destructive manipulation of rice seeds [21]. Most of these studies have focused on two key features: color and morphology. This research addressed these features in turn.

This research project aimed to develop a system for assessing the quality of rice seeds by employing the most recent algorithms. Roboflow platform was used for the development of the model, and an Android application was created to run the inference model developed on Roboflow. The Android application served as a practical system for evaluating the model's performance. The testing parameters included an assessment of the computational speed of the inference model, employing the profiling tool on the Android Studio integrated development environment (IDE). Furthermore, end-to-end testing was used to evaluate the model's accuracy in identifying rice seed quality.

II. Materials and Method

This research used the following method and design system for the development of a rice seed quality identification system with an Android platform:

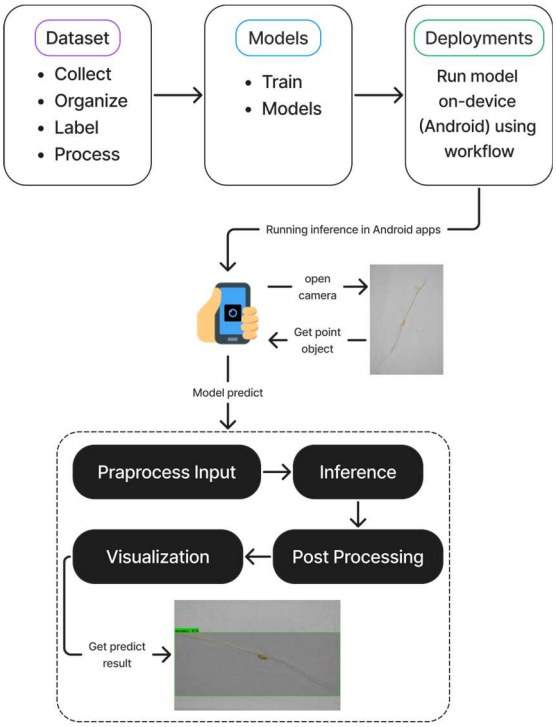

Fig. 1 Method and Design System

The methods involved in the development of the Android-based rice seed quality identification system was dataset creation with balanced dataset health criteria, model creation, model deployment, model integration with Android applications, and running inference models.

A. Deep Learning Model Development Method

The top level of Figure 1 shows a diagram of the deep learning model development process, which consists of three main stages: dataset creation, model training, and model deployment.

As shown in Figure 1, the first stage in the dataset creation was the collecting image stage, where seed image data was gathered to form the dataset, which was a pivotal step in dataset development. This stage, as highlighted in [22], [23], directly shaped the performance and outcomes of the machine learning algorithm, underlining the direct influence of the work on the overall results.

The image data used in this study were rice seeds in the germination period with the following quality categories:

TABLE I
GERMINATION CRITERIA FOR EACH CLASSIFICATION

Classification	Criteria	Image
Normal (N)	a. Primary roots grow strongly with secondary roots. The secondary seminal roots grow strongly, 2-3 roots. Sometimes the primary seminal root does not grow, but at least 2 secondary seminal roots must grow strongly. b. The primary leaf grows along the coleoptile and has emerged from it. In this case, the leaf should look healthy. The plumula can also be curved as long as it is not rotten. c. Good hypocotyl development is complete without any damage to the tissues. d. Have one cotyledon for monocotyledon sprouts and two for dicots. e. Plant seeds with epigeal germination type are considered normal if the root length is 4 × the length of the seed and have a normal structural development. f. Sprouts that are rotten due to infection by other sprouts are still considered normal if the important parts of the sprouts are all present.	
Abnormal (AN)	a. No primary or secondary roots grow. If they do, they are weak and short. b. No first leaves and colorless coleoptiles. Sometimes the plumula grows white or is decayed. c. Damaged sprouts, without cotyledons, embryos, ruptures, and short primary roots. d. Sprouts with deformed shape, weak, or unbalanced development of important parts.	
Dead (BM)	Seeds that spoil before germination or do not sprout after the prescribed testing period, but are not dormant.	
Fresh Not Growing (BSTT)	Seeds that do not grow to the end of the test, but still have the ability to grow normal. This type of seed is capable of absorbing water during the testing process but is inhibited for further development.	

This category was obtained from direct observation of rice seed in a previous study by Sidiq [12] and a recent research project by Li [24].

The next stage was organizing image. Image data was organized based on seed quality. Out of the initial 7,500 image collected, only 637 images were used in the final model training. This was because of an imbalance in the dataset with some images not meeting the size and clarity requirements. Additionally, certain categories had significantly more samples than others, which resulted in a skewed distribution of training data. Unfortunately, an incorrect application of oversampling before cross-validation created significant bias, resulting in lower accuracy and recall for the minority classes. The bias was evident through misclassification in the confusion matrix [25], [26]. The uneven amount of data between quality groups ultimately affected the overall health of the dataset, leading to poor results from the model [27]. This issue was addressed in the processing image stage.

After the organize stage, the next stage was the labeling image stage. The image was annotated according to the seed quality group (ground truth) by creating a bounding box around the seed in the image. The color of the bounding box was determined by the seed quality group it belonged to. For instance, a green bounding box might indicate a high-quality seed, while a red one might indicate a low-quality seed.

TABLE II
COLOR INDICATION FOR RICE SEED QUALITY GROUP

Color	Class Name
	Abnormal
	Dead
	Fresh not growing
	Normal

After all image data had been annotated according to the seed quality group, the data was then divided into train/valid/test with a ratio of 88%: 8%: 4%. This resulted in a data composition of 1338;128;63 images.

The following stage was the processing image, which consisted of 2 steps:

a. Pre-processing, which involved standardization of image data (all images in train, valid, and test) [28].
b. Augmentation, which was applied only to the training set. Data augmentation (rotation, inversion, scaling) increased minority class samples to address imbalance [29]. Additionally, oversampling and under sampling techniques were used during training to improve class distribution.

TABLE III
PRE-PROCESSING AND AUGMENTATION

Pre-processing	Augmentation
• Auto-oriented : Applied • Resize : Stretch to 640*640 • Grayscale : Applied	• Flip : Horizontal • 90^0 Rotate : Clockwise, Counter-Clockwise • Rotation : Between -15^0 and +15^0 • Grayscale: Apply to 15% of images • Saturation: Between -25% and +25% • Brightness: Between -15% and +15% • Blur: Up tp 2.5px • Noise: Up to 0.1% of pixels

The following are the results of the dataset created after applying data augmentation techniques:

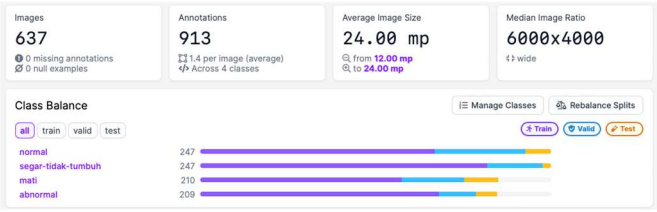

Fig. 2 Dataset Health

Additionally, both oversampling for minority classes and under sampling for majority classes were employed during training to further balance the distribution. These measures helped reduce bias and improve model performance, as reflected in a more evenly distributed error in the confusion matrix. However, it is important to note that while balancing the dataset improved results for common classes, it must be applied carefully, as it can sometimes reduce performance for rare ones [30].

The next step in the process involved train the model using the previously created dataset. The model-building phase began with training the dataset using Roboflow AutoML to generate a Roboflow 3.0 model optimized for speed. This model was selected for its ease of training and deployment on the Android application. The study also trained the YOLO NAS [31], [32] model for comparison in terms of modernity. The training process with Roboflow AutoML typically takes around 1.5 hours with 300 epochs. Following training, the model was evaluated using three key metrics: mean average precision (mAP), precision, and recall. F1 score was calculated using the formula [33]:

$$F1\ Score = 2x(Precision\ x\ Recall)/(Precision + Recall) \quad (1)$$

The model was then deployed using Roboflow's hosting server for online inference. To run the inference model with Roboflow, a workflow scheme was first created on the Roboflow website to improve the accessibility and efficiency of image segmentation tasks, paving the way for future advancements in cloud-based machine learning applications [34]. The workflow for running the Roboflow Train 3.0 Object Detection (Fast) and YOLO NAS inference models is as follows:

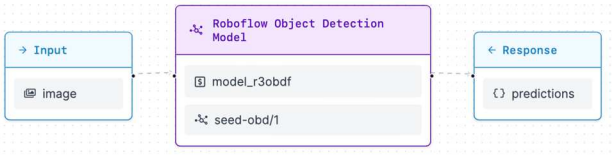

Fig. 3 Workflow Model

B. Mobile Application Development Methods

The workflow used the Roboflow API to create an Android app in Kotlin functioning primarily as a camera for capturing rice seed images. Feature identification began with input pre-processing, followed by activating the inference model and post-processing results. The app displayed a colored bounding box to assess seed quality. To ensure that the user interface (UI) was effective in agricultural settings, we prioritized simplicity and accessibility. The layout supported intuitive

navigation, allowing users to quickly capture images and view results without technical expertise. In addition, user feedback enhanced usability. Figure 4 shows the UI design of the system application.

Fig. 4 Mobile application general design

Figure 5 illustrates the inference process on the Android application using the trained model. The input image was encoded to Base64, which was then used to run the inference API. If encoding failed, a toast notification provided feedback on the error. Once successfully encoded, the API call was made. If it returned a 200 status, the application displayed a bounding box around the rice seed image along with the detection percentage. For any response other than 200, a toast notification alerted the user to the error and suggested actions such as re-capturing the image.

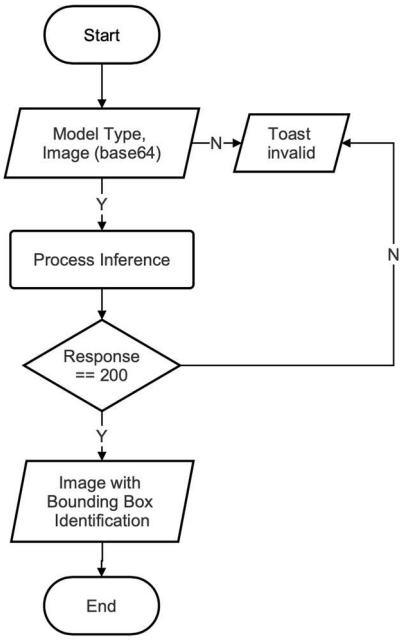

Fig. 5 Flowchart of the Android App System

Computational testing of the application was carried out using the App Inspector tool in the Android Studio IDE. Parameters taken included the Network Inspector, employed to read the application in the movement of transferred and received data [35].

III. Results and Discussion

A. The Model and Model Evaluations

The Roboflow Train 3.0 Object Detection (Fast) model demonstrated robust performance in object detection, as evidenced by a stable mean average precision (mAP) value with mAP@50:95.

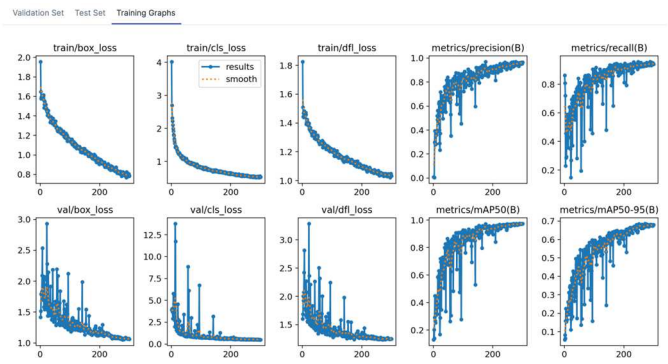

Fig. 6 Training Graph Roboflow 3.0 Object Detection (Fast)

Based on Figure 6, the decline in all loss functions suggests that the model developed the ability to accurately detect and classify objects. However, it should be noted that the model may be overfitting, as evidenced by the initial stabilization of its performance. It is possible that overfitting may cause some bugs, and indeed this model was tested and found to exhibit bugs. Bugs arising from this overfitting Roboflow model included:

a. *High Performance on Training Data*: The model demonstrated excellent performance on the training data, reflecting its ability to effectively learn the nuances and complexities of the data, including both relevant details and irrelevant noise.
b. *Poor Performance on Test Data:* The model struggled to generalize to unseen data. It performed well on training data but struggled with random images, often providing inaccurate feedback.
c. *Unstable Predictions:* The model's predictions were unreliable when applied to new data, leading to inconsistent results even with identical images.

To address this issue, solutions such as data augmentation and regularization should be explored further, along with techniques like dropout and early stopping to prevent overfitting.

Both precision and recall demonstrated improvement, indicating enhanced positive predictions and object identification. The upward trajectory of the mAP 50% and mAP 50-95% curves indicate an aptitude for extrapolation to novel data sets.

Meanwhile, as shown in the training graphics of the YOLO NAS model in Figure 7, the mean average precision (mAP) fluctuated but generally showed an upward trend, indicating improvement in the model's ability to detect objects.

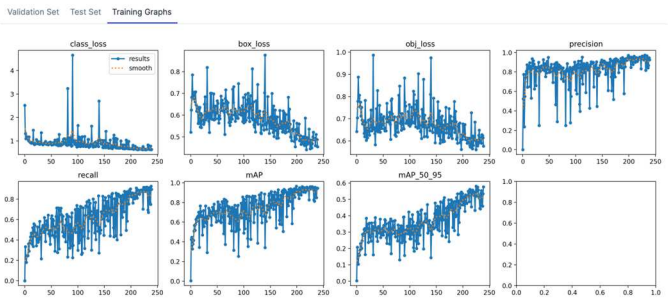

Fig. 7 Training Graph YOLO NAS

The MAP for detections with Intersection over Union (IoU) scores of above 0.5 showed an increasing trend, indicating the

model's improving ability to find objects with higher precision. The training graph demonstrated that the YOLO NAS model learnt and evolved, but it also highlighted the significant room for further optimization. This challenge presents an opportunity to actively contribute to the model's improvement, enhancing its performance and achieving better object detection results.

The evaluation of the Roboflow Train 3.0 Object Detection (Fast) and YOLO NAS models involved comparing key metrics, including mean average precision (mAP), precision, recall, F1 score, and computational speed. Figure 8 illustrates the differences in precision, recall, and F1 score between the two models.

Fig. 8 Comparative Diagram of Results from Model Evaluation

a. Precision: Roboflow achieved 96.2%, while YOLO NAS slightly outperformed with 97.1%.
b. Recall: Roboflow had a higher recall of 93.4% compared to YOLO NAS's 90.1%, indicating Roboflow's better ability to detect more objects.
c. F1 Score: Roboflow's F1 score was 0.947, slightly superior to YOLO NAS's 0.933, making it more balanced in both precision and recall.

This shows that the Roboflow Train 3.0 Object Detection (Fast) model can better detect objects present in the image and minimize missed objects.

The results of the calculation of F1 score with formula (1) for both models are as follows:

- Roboflow Train 3.0 Object Detection Fast
  F1 Score = 2 × (P×R) ÷ (P+R)
  = 2 × (0.962 × 0.934) ÷ (0.962+0.934)
  = 2 × (0.898) ÷ (1.896)
  = 0.947
- YOLO NAS
  F1 Score = 2 × (P×R) ÷ (P+R)
  = 2 × (0.971 × 0.901) ÷ (0.971+0.901)
  = 2 × (0.874) ÷ (1.872)
  = 0.933

Roboflow Train 3.0 Object Detection (Fast) achieved a slightly superior F1 score of 0.947, closely rivaling the YOLO NAS's score of 0.933.

B. *Integrating Inference Model to Android Application*

Furthermore, the inference of both models was tested when running on the Android application. This system application, called Seed-Snap, was developed as simple as possible and focused on a user-friendly concept. A camera feature was available to shoot seed objects, select photos from the gallery, and take photos of existing rice seeds. Furthermore, it was equipped with a spinner to select the model to be tested. In this study, only two models were tested. Table 4 shows the results of the android application that can run inference.

TABLE IV
SYSTEM APPLICATION FOR IDENTIFYING RICE SEED QUALITY

Splash Screen	Home
	
Model Picker	**Image Picker**
	
UI Identification Result	
	

C. *End to End Testing of the Android Application*

At this stage, the application underwent end-to-end tests to ensure its functionality operated as intended. We prepared several test scenarios for this Seed-Snap application.

TABLE V
END TO END TESTING OF THE SEED-SNAP ANDROID APPLICATIONS

Test scenario	Result	Evidence
Users can select a model.	Success	
Users can select an image resource.	Success	
Users can take pictures with the device's camera and the resulting images can be displayed for viewing.	Success	

Test scenario	Result	Evidence
Users can take a picture through the device's gallery, and the resulting images can be displayed for viewing.	Success	 
Users can run inference 1. When a picture of a dead seed category is uploaded, a red bounding box will appear, an indication of the dead seed and its corresponding percentage. 2. When an image of a normal seed category is uploaded, a green bounding box will appear, an indication of a normal seed and its corresponding percentage. 3. When an image of an abnormal seed category is uploaded, a yellow Bounding Box will appear, an indication of an abnormal seed and its corresponding percentage. 4. When a picture of a fresh non-sprouting seed category is uploaded, an orange Bounding Box will appear, an indication of a fresh non-sprouting seed and its corresponding percentage.	Success	   
The application can recognize rice seed objects	Success	 
The application can distinguish the rice seed objects from other objects	Failed	
The application can read multiple rice seed objects in one image	Success	

Based on the results of the end-to-end testing, all features worked as expected. However, in one scenario, the model incorrectly classified a non-seed object as a seed from a specific category. This highlights a limitation in the model's ability to differentiate between seed and non-seed objects. To address this issue, future improvements should focus on enhancing the model's object detection capabilities through additional training, data augmentation, and more advanced classification techniques to ensure more accurate predictions.

D. Analysis of Computational Speed of The Android Application

The parameter evaluated was the computational speed, which involved the application's attempt to capture the image until the bounding box was accurately visualized. Figure 9 shows the results of the computational speed of the model as seen from the Roboflow API invocation process for the Roboflow Train 3.0 Object Detection (Fast) model.

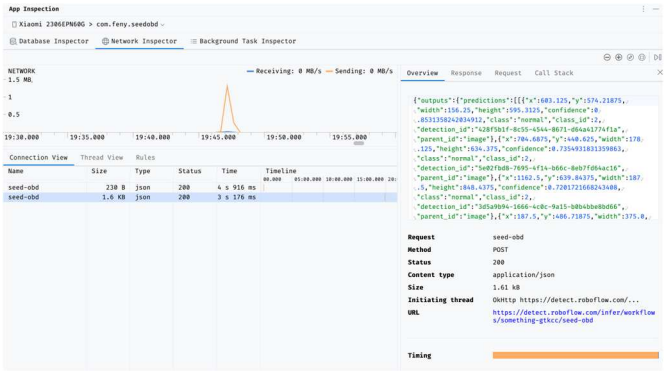

Fig. 9 The Inference Model for Data Network Inspector API of Roboflow Train 3.0 Object Detection (Fast).

Based on the results of the network inspector carried out by testing a single seed image, the data transfer speed recorded was 4s 916 ms with a size response of 230 B. Meanwhile, when multiple seeds were analyzed in one photo, it took 3s 176 ms with a size response of 1.6KB.

Figure 10 shows the results with the YOLO NAS model.

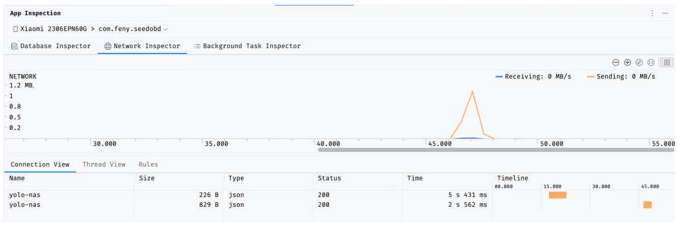

Fig. 10 The Inference Model for Data Network Inspector API of YOLO NAS

The network inspector results from YOLO NAS for single seed images indicated that the data transfer process took 5s 431ms. For multiple seeds within a single photo, the transfer process took 2s 562ms. However, the YOLO NAS model could only form fewer bounding boxes than the Roboflow model.

The differences in computational speed between the two models had significant implications for real-time applications. The YOLO NAS model was faster when handling multiple seeds, but its slower performance on single images and fewer bounding boxes limited its accuracy and effectiveness in field use. The Roboflow model was slightly slower with multiple objects but provided more bounding boxes, enhancing precision in real-world scenarios. The choice of model may

impact the user experience, especially in time-sensitive applications where speed and accuracy are both critical.

IV. Conclusion

This study compared two object detection models, Roboflow Train 3.0 Object Detection (Fast) and YOLO NAS, using an Android application. Both models showed good results in terms of object detection quality. Roboflow Train 3.0 Object Detection (Fast) had a slightly higher mAP percentage, recorded at 97.1%, compared to that of YOLO NAS with 96.6%. It also had a higher overall accuracy, with a precision of 96.2%, recall of 93.4%, and F1 score of 0.947. In terms of computational speed, Roboflow Train 3.0 Object Detection (Fast) performed better on the Android application, with a data transfer time of 4s 916ms for a single seed image and 3s 176ms for multiple seeds in a single photo, compared to YOLO NAS with 5s 431ms and 2s 562ms, respectively. However, YOLO NAS produced fewer bounding boxes. Overall, it is recommended to use Roboflow Train 3.0 Object Detection (Fast) for Android applications due to its better accuracy and speed. Future research could focus on optimizing these models for edge inference using Docker, such as model pruning, quantization, or utilizing lightweight architectures to improve efficiency and speed in mobile environments. Additionally, expanding the testing dataset to include diverse types of rice seeds would enhance model robustness and generalization, ensuring more reliable performance across various seed characteristics.

Acknowledgment

This research was made possible by the generous support from professors in State Polytechnic of Semarang, as well as funding from the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) for the 2024 Master's Thesis Research category (PTM).

References

[1] Sutardi et al., "The transformation of rice crop technology in Indonesia: Innovation and sustainable food security," *Agronomy*, vol. 13, no. 1, p. 1, Dec. 2022, doi: 10.3390/agronomy13010001.

[2] P. Luna and E. A. Suryana, "Rice fortification and biofortification as a potential strategy to alleviate stunting and bolster food security in Indonesia," in *BIO Web Conf.*, 2024, doi: 10.1051/bioconf/202411905004.

[3] A. Suryana et al., "Stability of rice availability and prices in Indonesia during the COVID-19 pandemic and Russia-Ukraine war," in *BIO Web Conf.*, 2024, doi: 10.1051/bioconf/202411902013.

[4] J. Lagga, A. A. Ambar, and A. Abdullah, "Strategi pengembangan penangkaran benih melalui kegiatan desa mandiri benih," *Ascarya: J. Islam. Sci., Cult., Soc. Stud.*, vol. 2, no. 1, pp. 13-31, Jan. 2022, doi: 10.53754/iscs.v2i1.94.

[5] M. Arsyad, A. Amiruddin, and Y. Kawamura, "Food security and political ecology for sustainable agriculture: Some crucial notes from the ICEFS 2020," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 681, no. 1, 2021, doi: 10.1088/1755-1315/681/1/012051.

[6] Badan Pusat Statistik, "Luas panen dan produksi padi di Indonesia 2023 (angka tetap)," Jakarta, Indonesia, 2024. [Online]. Available: https://www.bps.go.id/id/pressrelease/2024/03/01/2375

[7] Badan Pusat Statistik, "Berita resmi statistik," Jakarta, Indonesia, 2024. [Online]. Available: https://www.bps.go.id/id/pressrelease/2024/03/01/2375

[8] R. Simanjuntak, T. Supriana, and A. Rauf, "Analysis of production efficiency of rice seed producers in Serdang Bedagai Regency, North Sumatera, Indonesia," *Int. J. Res. Rev.*, vol. 10, no. 7, pp. 350-365, Jul. 2023, doi: 10.52403/ijrr.20230747.

[9] M. K. Misra, A. Harries, and M. Dadlani, "Role of seed certification in quality assurance," in *Seed Science and Technology: Biology, Production, Quality*, Springer Nature, 2023, pp. 267-297, doi: 10.1007/978-981-19-5888-5_12.

[10] A. Roy et al., "Seed quality enhancement through seed biopriming to increase productivity: A review," *Agric. Res. Commun. Cent.*, Jun. 2022, doi: 10.18805/ag.r-2477.

[11] K. U. Ghodasaini and H. Ghimire, "Role of quality seeds in food security and food self-sufficiency in Nepal," *INWASCON Technol. Mag.*, vol. 4, pp. 52-55, 2022, doi: 10.26480/itechmag.04.2022.52.55.

[12] S. S. Hidayat et al., "Determining the rice seeds quality using convolutional neural network," *Int. J. Inform. Vis.*, vol. 7, no. 2, 2023, doi: 10.30630/joiv.7.2.1175.

[13] T. T. H. Phan, Q. T. Vo, and H. D. Nguyen, "A novel method for identifying rice seed purity using hybrid machine learning algorithms," *Heliyon*, vol. 10, no. 14, Jul. 2024, doi: 10.1016/j.heliyon.2024.e33941.

[14] R. Rajalakshmi et al., "RiceSeedNet: Rice seed variety identification using deep neural network," *J. Agric. Food Res.*, vol. 16, p. 101062, 2024, doi: 10.1016/j.jafr.2024.101062.

[15] R. Jindal and S. K. Mittal, "Artificial intelligence and machine vision-based assessment of rice seed quality," in *Applied Data Science and Smart Systems*, J. Singh et al., Eds., 2024, pp. 603-609, doi: 10.1201/9781003471059.

[16] D. Johan et al., "Agricultural digitalization in Indonesia: Challenges and opportunities for sustainable development," *Educ. Admin.: Theory Pract.*, 2024, doi: 10.53555/kuey.v30i7.6599.

[17] C. J. Sharma and M. Kumar, "Advances in deep learning-based technologies in rice crop management," in *Computer Vision and Machine Learning in Agriculture*, vol. 3, M. S. Bansal and M. Uddin, Eds., Singapore: Springer, 2023, pp. 79-89, doi: 10.1007/978-981-99-3754-7_6.

[18] R. Srinivasaiah et al., "Analysis and prediction of seed quality using machine learning," *Int. J. Elect. Comput. Eng.*, vol. 13, no. 5, pp. 5770-5781, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5770-5781.

[19] Y. Basol and S. Toklu, "A deep learning-based seed classification with mobile application," *Turk. J. Math. Comput. Sci.*, May 2021, doi: 10.47000/tjmcs.897631.

[20] J. Zhao et al., "Deep-learning-based automatic evaluation of rice seed germination rate," *J. Sci. Food Agric.*, vol. 103, no. 4, pp. 1912-1924, 2023, doi: 10.1002/jsfa.12318.

[21] M. M. Shivamurthaiah and H. K. K. Shetra, "Non-destructive machine vision system based rice classification using ensemble machine learning algorithms," *Recent Adv. Elect. Electron. Eng.*, vol. 17, no. 5, pp. 486-497, Jun. 2024, doi: 10.2174/2352096516666230710144614.

[22] G. Vidhya, D. Nirmala, and T. Manju, "Quality challenges in deep learning data collection in perspective of artificial intelligence," *J. Inf. Technol. Comput.*, vol. 4, no. 1, pp. 46-58, Jun. 2023, doi: 10.48185/jitc.v4i1.725.

[23] F. Nobells, "Data collection," May 2022, doi: 10.31219/osf.io/h9an8.

[24] D. Li et al., "Rice seedling row detection based on morphological anchor points of rice stems," *Biosyst. Eng.*, vol. 226, pp. 71-85, 2023, doi: 10.1016/j.biosystemseng.2022.12.012.

[25] A. Demircioğlu, "Applying oversampling before cross-validation will lead to high bias in radiomics," *Sci. Rep.*, vol. 14, no. 1, p. 11563, 2024, doi: 10.1038/s41598-024-62585-z.

[26] D. A. Dablain and N. V. Chawla, "The hidden influence of latent feature magnitude when learning with imbalanced data," in *Pattern Recognit. Artif. Intell.*, 2025, pp. 76-91, doi: 10.1007/978-981-97-8702-9_6.

[27] Y. Gong et al., "A survey on dataset quality in machine learning," *Inf. Softw. Technol.*, vol. 162, p. 107268, 2023, doi: 10.1016/j.infsof.2023.107268.

[28] A. Amato and V. Di Lecce, "Data preprocessing impact on machine learning algorithm performance," *Open Comput. Sci.*, vol. 13, no. 1, Jan. 2023, doi: 10.1515/comp-2022-0278.

[29] D. A. Dablain et al., "Efficient augmentation for imbalanced deep learning," in *Proc. IEEE 39th Int. Conf. Data Eng. (ICDE)*, 2023, pp. 1433-1446, doi: 10.1109/ICDE55515.2023.00114.

[30] R. C. Moore et al., "Dataset balancing can hurt model performance," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10095255.

[31] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, Dec. 2023, doi: 10.3390/make5040083.

[32] A. Team, "YOLOv8 vs. YOLO-NAS showdown: Exploring advanced object detection," Deci, 2024. [Online]. Available: https://deci.ai/blog/yolov8-vs-yolo-nas-showdown-exploring-advanced-object-detection/

[33] K. Wilkinghoff and K. Imoto, "F1-EV score: Measuring the likelihood of estimating a good decision threshold for semi-supervised anomaly detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2024, pp. 256-260, doi: 10.1109/ICASSP48485.2024.10446011.

[34] J. M. Paule et al., "Integration of AWS and Roboflow Mask R-CNN model for a fully cloud-based image segmentation platform," in *Proc. IEEE 15th Int. Conf. HNICEM*, 2023, pp. 1-6, doi: 10.1109/HNICEM60674.2023.10589105.

[35] J. D. Rathod and R. Bhatti, "Vulnerability detection in the Android application based on dynamic analysis," in *Proc. 3rd Int. Conf. Intell. Comput., Inf. Control Syst.*, R. Pandian et al., Eds., Singapore: Springer, 2022, pp. 287-302, doi: 10.1007/978-981-16-7330-6_22.